



(10) **DE 11 2017 006 701 T5** 2019.09.19

(12)

Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2018/125989**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2017 006 701.4**
(86) PCT-Aktenzeichen: **PCT/US2017/068683**
(86) PCT-Anmeldetag: **28.12.2017**
(87) PCT-Veröffentlichungstag: **05.07.2018**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **19.09.2019**

(51) Int Cl.: **H04W 4/70 (2018.01)**

H04W 4/08 (2009.01)
H04L 29/08 (2006.01)
H04L 29/12 (2006.01)
H04L 9/32 (2006.01)
H04W 84/18 (2009.01)

(30) Unionspriorität:
62/441,070 30.12.2016 US

(71) Anmelder:
Intel Corporation, Santa Clara, Calif., US

(74) Vertreter:
**Maiwald Patentanwalts- und
Rechtsanwaltsgesellschaft mbH, 80335 München,
DE**

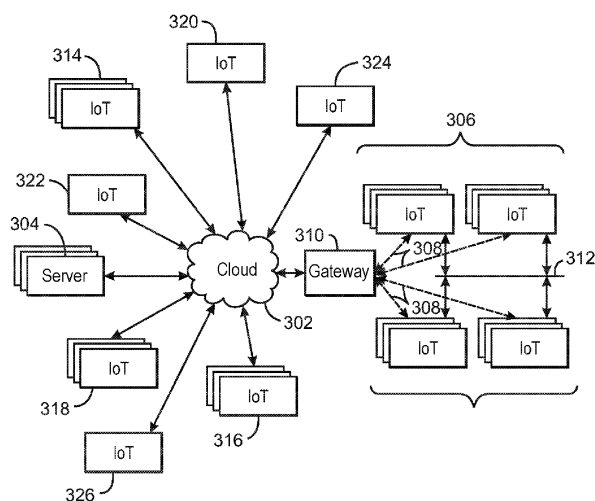
(72) Erfinder:
**Smith, Ned M., Beaverton, Oreg., US; Nolan,
Keith, Mullingar, IE; Kelly, Mark, Leixlip KE Co, IE;
Burns, Gregory, Seattle, WA, US; Nolan, Michael,**

**Maynooth KE, IE; Brady, John, Celbridge, KE, IE;
Ni Scanail, Clíodhna, Broadford, KE, IE; Cahill,
Niall, Dublin, IE; Macieira, Thiago J., Portland,
OR, US; Zhang, Zheng, Portland, Oreg., US;
Anderson, Glen J., Beaverton, Oreg., US; Muttik,
Igor, Aylesbury, Buckinghamshire, GB; Carboni,
Davide, London, GB; Ryan, Eugene, Dublin, IE;
Davies, Richard, Dublin, IE; Kohlenberg, Toby M.,
Portland, OR, US; Koning, Maarten, Bloomfield,
Ontario, CA; Wenus, Jakub, Maynooth, KE, IE;
Poornachandran, Rajesh, Portland, Oreg., US;
Deleeuw, William C., Beaverton, OR, US; Chukka,
Ravikiran, Portland, OR, US**

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Internet der Dinge**

(57) Zusammenfassung: Das Internet kann dafür ausgelegt werden, einer großen Anzahl von Internet-der-Dinge (IoT)-Vorrichtungen Kommunikationen bereitzustellen. Vorrichtungen können dafür ausgelegt werden, den Bedarf an Netzschichten von zentralen Servern über Gateways bis hinab zu Edge-Vorrichtungen zu adressieren, ungehindert zu wachsen, angeschlossene Ressourcen zu entdecken und zugänglich zu machen und die Fähigkeit zu unterstützen, angeschlossene Ressourcen zu verbergen und aufzuteilen. Netzprotokolle können Teil des Gewebes sein, das von Menschen zugängliche Dienste unterstützt, die unabhängig vom Ort, von der Zeit oder vom Raum arbeiten. Innovationen können die Dienstauflieferung und die zugeordnete Infrastruktur in der Art von Hardware und Software umfassen. Dienste können entsprechend spezifizierten Dienstqualitäts(QoS)-Bedingungen bereitgestellt werden. Die Verwendung von IoT-Vorrichtungen und Netzen kann in ein heterogenes Verbindbarkeitsnetz, einschließlich festverdrahteter und drahtloser Technologien, aufgenommen werden.



Beschreibung

Querverweis auf verwandte Anmeldung

[0001] Die vorliegende Anmeldung beansprucht den Vorteil des Einreichungsdatums der am 30. Dezember 2016 eingereichten vorläufigen US-Patentanmeldung mit der laufenden Nummer 62/441 070 von Ned M. Smith u. a. mit dem Titel „THE INTERNET OF THINGS“, welche hier durch Verweis aufgenommen ist.

Technisches Gebiet

[0002] Die vorliegenden Techniken betreffen allgemein Internet-der-Dinge(IoT)-Vorrichtungen. Insbesondere betreffen die vorliegenden Techniken Vorrichtungen, die Fernerfassungs- und -betätigungsfunktionen ausführen können.

Hintergrund

[0003] Eine aktuelle Sicht des Internets ist die Verbindung von Clients in der Art von Personalcomputern, Tablets, Smartphones, Servern, digitaler Bilderrahmen und vieler anderer Vorrichtungstypen mit öffentlich zugänglichen Datenzentren, die in Server-Farmen gehostet sind. Diese Sicht repräsentiert jedoch einen kleinen Teil der Gesamtverwendung des global verbundenen Netzes. Gegenwärtig existiert eine sehr große Anzahl verbundener Ressourcen, die jedoch nicht öffentlich zugänglich sind. Beispiele umfassen Firmennetze, private Organisationssteuernetze und den Globus umspannende Überwachungsnetze, häufig unter Verwendung von Peer-to-Peer-Relais für die Anonymität.

[0004] Es wurde geschätzt, dass das Internet der Dinge bzw. Internet of Things (IoT) bis zum Jahre 2020 mehr als 15 Milliarden Vorrichtungen Internetverbindbarkeit bringen kann. Für Organisationen können IoT-Vorrichtungen Gelegenheiten für die Überwachung, Verfolgung oder Steuerung anderer Vorrichtungen und Elemente, einschließlich weiterer IoT-Vorrichtungen, anderer Heim- und industrieller Vorrichtungen, Elemente in Herstellungs- und Lebensmittelerzeugungsketten und dergleichen, bereitstellen. Die Entstehung von IoT-Netzen hat als Katalysator für eine tiefgreifende Änderung in der Entwicklung des Internets gedient. In der Zukunft wird sich das Internet wahrscheinlich von einem primär personenorientierten Werkzeug zu einer Infrastruktur entwickeln, wobei Personen schließlich Minderheitsakteure in einer Welt miteinander verbundener Vorrichtungen sein können.

[0005] In dieser Hinsicht wird das Internet zu einem Kommunikationssystem für Vorrichtungen und Netze von Vorrichtungen, um nicht nur mit Datenzentren, sondern auch miteinander zu kommunizieren. Die Vorrichtungen können funktionelle Netze oder virtuelle Vorrichtungen bilden, um Funktionen auszuführen, die sich auflösen können, sobald die Funktion ausgeführt wurde. Es existieren Herausforderungen bei der Ermöglichung zuverlässiger, sicherer und identifizierbarer Vorrichtungen, die Netze bilden können, wie es erforderlich ist, um Aufgaben zu erreichen.

Figurenliste

Fig. 1 zeigt Zwischenverbindungen, die im Internet gemäß einigen Ausführungsformen vorhanden sein können.

Fig. 2 zeigt eine Netztopologie, die für eine Anzahl von Internet-der-Dinge(IoT)-Netzen, die durch Backbone-Strecken mit Gateways gekoppelt sind, verwendet werden kann, gemäß einigen Ausführungsformen.

Fig. 3 ist eine Zeichnung eines Cloud-Rechnernetzes oder einer Cloud in Kommunikation mit einer Anzahl von IoT-Vorrichtungen gemäß einigen Ausführungsformen.

Fig. 4 ist eine Zeichnung eines Cloud-Rechnernetzes oder einer Cloud in Kommunikation mit einem Mesh-Netz von IoT-Vorrichtungen, das als Fog-Vorrichtung bezeichnet werden kann, das am Rand der Cloud arbeitet, gemäß einigen Ausführungsformen.

Fig. 5 ist eine Schemazeichnung, welche die Bildung eines zusammengesetzten Objekts anhand einer Anzahl atomarer Objekte zeigt, gemäß einigen Ausführungsformen.

Fig. 6 ist eine Schemazeichnung der Bildung eines Gruppenobjekts aus einer Sammlung atomarer Objekte und zusammengesetzter Objekte gemäß einigen Ausführungsformen.

- Fig. 7** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Gruppenerzeugung unter Verwendung einer Sammlung von Objekten gemäß einigen Ausführungsformen.
- Fig. 8** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Abladen von Daten vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 9** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Gruppenobjekte zu bilden, gemäß einigen Ausführungsformen.
- Fig. 10** ist eine Schemazeichnung, welche die Verwendung von Enhanced Privacy Identification (EPID) für die Objekttypidentität zeigt, gemäß einigen Ausführungsformen.
- Fig. 11** ist ein Leiterdiagramm eines beispielhaften Verfahrens zur dynamischen Erzeugung eines Objekttyps gemäß einigen Ausführungsformen.
- Fig. 12** ist ein Leiterdiagramm eines beispielhaften Verfahrens zur Typintrospektion unter Verwendung einer Rekursion gemäß einigen Ausführungsformen.
- Fig. 13** ist ein Leiterdiagramm eines beispielhaften Verfahrens zur rekursiven Typattestierung gemäß einigen Ausführungsformen.
- Fig. 14** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Zuweisung von Typen zu zusammengesetzten Objekten, wenn sie gebildet werden, vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 15** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Gruppenobjekte zu bilden, gemäß einigen Ausführungsformen.
- Fig. 16** ist eine Schemazeichnung der Bildung einer Koalitionsgruppe gemäß einigen Ausführungsformen.
- Fig. 17** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Einschreiben von Mitgliedern in eine Koalitionsgruppe gemäß einigen Ausführungsformen.
- Fig. 18** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 19** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Koalitionsgruppen zu erzeugen, gemäß einigen Ausführungsformen.
- Fig. 20** ist ein schematisches Diagramm einer semi-permissionierten verteilten Ledger-Transaktion gemäß einigen Ausführungsformen.
- Fig. 21** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Ausführung semi-permissionierter Transaktionen gemäß einigen Ausführungsformen.
- Fig. 22** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 23** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, sicher in Gruppen zu kommunizieren, gemäß einigen Ausführungsformen.
- Fig. 24** ist ein schematisches Diagramm der Verwendung einer vertrauenswürdigen Ausführungsumgebung bzw. Trusted Execution Environment (TEE) zum sicheren Booten einer Vorrichtung in einer IoT-Umgebung gemäß einigen Ausführungsformen.
- Fig. 25** ist ein Blockdiagramm eines Blockchain-Blocks, der Boot-Integritätstransaktionen enthält, gemäß einigen Ausführungsformen.
- Fig. 26** ist ein schematisches Diagramm der Verwendung einer Whitelist-Bildsammlung mit einer Blockchain gemäß einigen Ausführungsformen.
- Fig. 27** ist eine Zeichnung eines Blockchain-Blocks mit Integritätstransaktionen für Whitelist-Bilder gemäß einigen Ausführungsformen.
- Fig. 28** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für einen sicheren Boot-Prozessfluss unter Verwendung von Blockchain-Vertrauens-Roots gemäß einigen Ausführungsformen.
- Fig. 29** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 30 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, sicher in Gruppen zu kommunizieren, gemäß einigen Ausführungsformen.

Fig. 31 ist eine Schemazeichnung, worin eine Zusammenarbeitsfähigkeit zwischen öffentlichen Domains, privaten Domains und öffentlich-privaten Domains dargestellt ist, gemäß einigen Ausführungsformen.

Fig. 32 ist eine Schemazeichnung der Zusammenarbeitsfähigkeit zwischen einem heterogenen Netz aus festverdrahteten Netzen und Drahtlosnetzen gemäß einigen Ausführungsformen.

Fig. 33 ist eine Schemazeichnung eines Inline-Routingsystems, das zwei verschiedene Fog- oder Cloud-Entitäten in der Art einer Cloud A mit einer Cloud B verbindet, gemäß einigen Ausführungsformen.

Fig. 34 ist eine Schemazeichnung eines Inline-Routings, worin ein implizites Durchlass-Routing durch eine IoT-Vorrichtung dargestellt ist, gemäß einigen Ausführungsformen.

Fig. 35 ist eine Schemazeichnung eines expliziten permissionierten Routens durch eine IoT-Vorrichtung gemäß einigen Ausführungsformen.

Fig. 36 ist eine Schemazeichnung einer Dienstbarkeitsschicht für das Inline-Routing, welche für die Durchlassrichtlinienkontrolle verwendet wird, gemäß einigen Ausführungsformen.

Fig. 37 ist ein Leiterdiagramm eines beispielhaften Verfahrens zum expliziten Durchlass-Routing auf der Grundlage von Erlaubnissen gemäß einigen Ausführungsformen.

Fig. 38 ist ein Leiterdiagramm eines beispielhaften Verfahrens eines zeitlich begrenzten Mietansatzes für ein explizites Durchlassen gemäß einigen Ausführungsformen.

Fig. 39 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 40 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, Kommunikationen zwischen Vorrichtungen durch Dienstbarkeiten zu übertragen, gemäß einigen Ausführungsformen.

Fig. 41 ist eine Schemazeichnung der Verwendung einer Frame-Struktur zur Übertragung von Herkunftsnachweis- bzw. Proof-of-Provenance(PoP)-Informationen durch Vorrichtungen in einem Netz gemäß einigen Ausführungsformen.

Fig. 42 ist ein schematisches Diagramm einer Prozedur, die zur Erzeugung eines PoP-Durchleitungs-codes oder -Schlüssels verwendet werden können, gemäß einigen Ausführungsformen.

Fig. 43 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Erzeugung eines PoP-Schlüssels gemäß einigen Ausführungsformen.

Fig. 44 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Überprüfung der PoP-Schlüssel in einem Paket gemäß einigen Ausführungsformen.

Fig. 45 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Verfolgung eines Herkunftsnachweises in Paketen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 46 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, Kommunikationen zwischen Vorrichtungen durch Dienstbarkeiten zu übertragen, gemäß einigen Ausführungsformen.

Fig. 47 ist eine Schemazeichnung eines Beispiels eines Pakets, das Mikrozahlungsinformationen in einem Token-Bucket aufweist, gemäß einigen Ausführungsformen.

Fig. 48 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung eines Token-Buckets zur Übertragung von Mikrozahlungen zu übertragenden Systemen gemäß einigen Ausführungsformen.

Fig. 49 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Verwendung von Token-Buckets zum Erleichtern von Zahlungen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 50 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code zum Anweisen eines Prozessors aufweist, Kommunikationen zwischen Vorrichtungen auf der Grundlage von Zahlungen von einem Token-Bucket zu übertragen, gemäß einigen Ausführungsformen.

- Fig. 51** ist eine Zeichnung einer heterogenen Netzinfrastruktur (hetnet-Infrastruktur), welche an mehreren Stufen IP-Domains mit Nicht-IP-Domains verbindet, gemäß einigen Ausführungsformen.
- Fig. 52** ist eine Schemazeichnung einer Protokollverpackung, die für die Verpackung von Frames von einem Protokoll in ein anderes Protokoll verwendet wird, gemäß einigen Ausführungsformen.
- Fig. 53** ist eine Schemazeichnung einer Protokollverpackung, die zur Verpackung eines Low-Power-Wide-Area-Network (LPWAN)-Protokoll-Frames in der Art eines LoRaWAN-Frames in einem IEEE-802.11 (oder Wi-Fi®)-MAC-Frame verwendet wird, gemäß einigen Ausführungsformen.
- Fig. 54** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Protokollverpackung für die Übertragung eines Frames gemäß einigen Ausführungsformen.
- Fig. 55** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Verpackung von Frames in einem ersten Protokoll in Frames eines anderen Protokolls vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 56** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Frames in einem ersten Protokoll in Frames eines anderen Protokolls zu verpacken, gemäß einigen Ausführungsformen.
- Fig. 57** ist eine Zeichnung einer Frame-Struktur, die als Nutzinhalt in einem Niederleistungs-Weitbereichs- bzw. Low-Power-Wide-Area(LPWA)-Frame in der Art eines LoRaWAN-Frames verwendet werden kann, gemäß einigen Ausführungsformen.
- Fig. 58** ist eine Schemazeichnung von Übertragungsdaten-Nutzinhalt, der vor dem Senden in eine Anzahl von Sub-Blöcken fragmentiert wird, gemäß einigen Ausführungsformen.
- Fig. 59** ist eine Schemazeichnung **5900** einer Network-Division-Multiplexing-NDM-Seriell-zu-Parallel-Übertragung gemäß einigen Ausführungsformen.
- Fig. 60** ist eine Schemazeichnung des Empfangs der Sub-Blöcke gemäß einigen Ausführungsformen.
- Fig. 61** ist eine Schemazeichnung der Rekombination der Sub-Blöcke zur Bildung des empfangenen Datennutzeinhalts gemäß einigen Ausführungsformen.
- Fig. 62** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Fragmentieren und Absenden eines Nutzinhalts über mehrere parallele Kommunikationskanäle gemäß einigen Ausführungsformen.
- Fig. 63** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Empfangen und Rekombinieren unter Verwendung einer NDM-Technik gesendeter Pakete gemäß einigen Ausführungsformen.
- Fig. 64** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Fragmentieren von Nutzgehalten für die Übertragung entlang mehreren parallelen Wegen vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 65** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Nutzhalte zu fragmentieren und entlang mehreren parallelen Wegen zu übertragen, gemäß einigen Ausführungsformen.
- Fig. 66** ist eine Schemazeichnung eines Überlagerungsbakensystems, bei dem ein Bakenknoten einer nahe gelegenen IoT-Vorrichtung eine Ortsnachricht bereitstellt, gemäß einigen Ausführungsformen.
- Fig. 67** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Erzeugung eines Orts-Nutzeinhalts gemäß einigen Ausführungsformen.
- Fig. 68** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Parsen eines Frames, der einen Orts-Nutzeinhalt aufweist, gemäß einigen Ausführungsformen.
- Fig. 69** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem Bakenknoten vorhanden sein können, um ein Bakenknotensystem zum Teilen von Ortsdaten einzurichten, gemäß einigen Ausführungsformen.
- Fig. 70** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Orts-Nutzeinhalt zu senden und zu empfangen, gemäß einigen Ausführungsformen.
- Fig. 71** ist eine Schemazeichnung eines verteilten Inhaltsverteilungssystems für heterogene Netze gemäß einigen Ausführungsformen.

Fig. 72 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für eine dispergierte Inhaltsverteilung gemäß einigen Ausführungsformen.

Fig. 73 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Implementation eines verteilten Inhaltsverteilungssystems vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 74 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, ein verteiltes Inhaltsverteilungssystem zu implementieren, gemäß einigen Ausführungsformen.

Fig. 75 ist eine Schemazeichnung eines Drahtlosspeichersystems gemäß einigen Ausführungsformen.

Fig. 76 ist eine andere Schemazeichnung des Drahtlosspeichersystems gemäß verschiedenen Ausführungsformen.

Fig. 77 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Fragmentieren und Speichern von Daten in einer Übertragungsschleife zwischen Vorrichtungen gemäß einigen Ausführungsformen.

Fig. 78 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Datenspeicherung und zum Zugriff unter Verwendung eines Kommunikationskanals zur Speicherung gemäß einigen Ausführungsformen.

Fig. 79 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Speichern von Daten in Übertragungskanälen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 80 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Nutzinhalte zu fragmentieren und entlang mehreren parallelen Wegen zu übertragen, gemäß einigen Ausführungsformen.

Fig. 81 ist eine Zeichnung einer Struktur, die zur dynamischen Signalisierung verwendet werden kann, gemäß einigen Ausführungsformen.

Fig. 82 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Übertragung von Daten unter Verwendung einer Zadoff-Chu(ZC)-Präambel-Struktur gemäß einigen Ausführungsformen.

Fig. 83 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Empfangen von Daten über mehrere Kanäle unter Verwendung der ZCverschobenen Sequenz gemäß einigen Ausführungsformen.

Fig. 84 ist eine Reihe von Auftragungen, welche den in der vorstehenden Gleichung für jede der durch K gegebenen Sequenzen detaillierten Korrelationsprozess zeigen, gemäß einigen Ausführungsformen.

Fig. 85 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Verwendung von ZC-Sequenzen für das Senden von Daten in mehreren gleichzeitigen Kanälen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 86 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, über unter Verwendung von ZC-Sequenzen modulierte Kanäle zu kommunizieren, gemäß einigen Ausführungsformen.

Fig. 87 ist eine Schemazeichnung eines Mehrfunk-Koexistenzsystems in einer IoT-Vorrichtung gemäß einigen Ausführungsformen.

Fig. 88 ist ein Leiterdiagramm eines beispielhaften Verfahrens zur Steuerung und Behandlung des Betriebs und der Koexistenz mehrerer Funkeinrichtungen gemäß einigen Ausführungsformen.

Fig. 89 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Verwendung mehrerer koexistierender Funkeinrichtungen für die Kommunikation mit anderen Knoten vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 90 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, über unter Verwendung von ZC-Sequenzen modulierte Kanäle zu kommunizieren, gemäß einigen Ausführungsformen.

Fig. 91 ist ein schematisches Diagramm einer Netzdienste-Overlay-Funktion über ein heterogenes Netz gemäß einigen Ausführungsformen.

Fig. 92 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Behandlung neuer Anforderungen für einen Dienst gemäß einigen Ausführungsformen.

Fig. 93 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Registrieren einer Endpunkt- oder Dienstkomponente bei einer Netzdomain-Steuereinrichtung (NDC) oder einem anderen Dienstkoordinator gemäß einigen Ausführungsformen.

Fig. 94 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Koordinieren oder Erfüllen von Dienstanforderungen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 95 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor oder Prozessoren anzuweisen, Dienstanforderungen zu koordinieren oder zu erfüllen, gemäß einigen Ausführungsformen.

Fig. 96 ist ein schematisches Diagramm der Ad-hoc-Bildung eines umgekehrten Verteilte-Hash-Tabelle bzw. Distributed-Hash-Table(DHT)-Netzes für IoT-Dienste gemäß einigen Ausführungsformen.

Fig. 97 ist ein schematisches Diagramm eines Prozesses zur Verfolgung, welche Knoten für das Speichern oder Übertragen von Dateidaten verwendet werden können, gemäß einigen Ausführungsformen.

Fig. 98 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Vorgabe von Speicher- oder Sendeknoten gemäß einigen Ausführungsformen.

Fig. 99 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Speichern oder Übertragen von Daten unter Verwendung einer verteilten Hash-Tabelle bzw. Distributed Hash Table (DHT) gemäß einigen Ausführungsformen.

Fig. 100 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Koordinieren oder Erfüllen von Dienstanforderungen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 101 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor oder Prozessoren anzuweisen, Dienstanforderungen zu koordinieren oder zu erfüllen, gemäß einigen Ausführungsformen.

Fig. 102 ist ein schematisches Diagramm eines Mehrweg-Kommunikationssystems, worin drei beispielhafte Wege zwischen zwei Endpunkten, die für eine mögliche Verwendung verfügbar sein können, dargestellt sind, gemäß einigen Ausführungsformen.

Fig. 103 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Auswahl eines Kommunikationswegs gemäß einigen Ausführungsformen.

Fig. 104 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Senden von Daten über mehrere Kommunikationskanäle vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 105 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Daten über mehrere Kommunikationskanäle zu senden, gemäß einigen Ausführungsformen.

Fig. 106 ist eine Schemazeichnung eines IoT-Gateways für sichere Kommunikationen und Übersetzungen zwischen Domains gemäß einigen Ausführungsformen.

Fig. 107 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Übersetzung von Arbeitslasten in einem sicheren IoT-Gateway gemäß einigen Ausführungsformen.

Fig. 108 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem IoT-Gateway zur Übersetzung von Arbeitslasten zwischen Domains vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 109 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, eine Arbeitslast zwischen einem Eingangsnetz und einem Ausgangsnetz zu übersetzen, gemäß einigen Ausführungsformen.

Fig. 110 ist ein schematisches Diagramm von Vorrichtungen, die durch verschiedene Domains eingegliedert werden, welche von einer geteilten Domain aufgenommen werden, die erzeugt wurde, um es den Vorrichtungen zu ermöglichen, als Komponenten einer neuen Domain teilzunehmen, gemäß einigen Ausführungsformen.

Fig. 111 ist ein schematisches Diagramm der Erzeugung einer geteilten Ressource, um es einer Vorrichtung zu ermöglichen, über Domains teilzunehmen, gemäß einigen Ausführungsformen.

- Fig. 112** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Einrichtung einer kombinierten IoT-Domain mit geteilten Ressourcen gemäß einigen Ausführungsformen.
- Fig. 113** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung geteilter Ressourcen vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 114** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, geteilte Ressourcen über Domains einzurichten, gemäß einigen Ausführungsformen.
- Fig. 115** ist ein Leiterdiagramm, das Stufen in einem Produktlebenszyklus für die Implementation eines Produktverfolgungssystems zeigt, gemäß einigen Ausführungsformen.
- Fig. 116** ist eine Schemazeichnung der Verwendung privater Datenspeicher, wobei ein Datensatzschlüssel verwendet werden kann, um auf die Verfolgbarkeitsdatensätze für jede Stufe zuzugreifen, gemäß einigen Ausführungsformen.
- Fig. 117** ist eine Schemazeichnung der Verwendung eines öffentlichen oder gemeinsamen Datenspeichers gemäß einigen Ausführungsformen.
- Fig. 118** ist ein schematisches Diagramm eines Prozesses zur Implementation eines Verfolgbarkeitssystems gemäß einigen Ausführungsformen.
- Fig. 119** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung vorhanden sein können, um Verfolgbarkeitsdatensätze für ein Produkt bereitzustellen, gemäß einigen Ausführungsformen.
- Fig. 120** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **27500**, das Code aufweist, um einen Prozessor **902** anzuweisen, Operationen von Pre-first-Respondervorrichtungen zu behandeln, gemäß einigen Ausführungsformen.
- Fig. 121(A)** ist eine Schemazeichnung des in vielen heutigen Computernetzen verwendeten hierarchischen Richtlinien-Behandlungssystems, gemäß einigen Ausführungsformen.
- Fig. 121(B)** ist eine Schemazeichnung der Richtlinienbehandlung in einem Peer-to-Peer(P2P)-Netz in der Art eines IoT-Mesh-Netzes gemäß einigen Ausführungsformen.
- Fig. 122** ist ein schematisches Diagramm von Systemen in Knoten zur Implementation eines verteilten Richtlinien-Behandlungssystems gemäß einigen Ausführungsformen.
- Fig. 123(A)** ist ein Leiterdiagramm eines beispielhaften Verfahrens eines neuen nicht konfigurierten Knotens, der versucht, Richtlinien auf einem Netz zu entdecken, beispielsweise von einem Peer-Knoten, gemäß einigen Ausführungsformen.
- Fig. 123(B)** ist ein Leiterdiagramm eines beispielhaften Verfahrens eines neuen nicht konfigurierten Knotens, der Richtlinien von einem konfigurierten Knoten entdeckt, gemäß einigen Ausführungsformen.
- Fig. 124** ist ein Leiterdiagramm eines beispielhaften Verfahrens eines konfigurierten Knotens, der mit einem Knoten kommuniziert, der eine aktualisierte Richtlinie zur Aktualisierung der Richtlinien des konfigurierten Knotens aufweist, gemäß einigen Ausführungsformen.
- Fig. 125** ist ein Leiterdiagramm eines beispielhaften Verfahrens zur Verkettung von verschiedenen Knoten durch den konfigurierten Knoten erhaltener Richtlinien gemäß einigen Ausführungsformen.
- Fig. 126** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur verteilten Behandlung von Richtlinien vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 127** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Richtlinien in einem IoT-Netz in Zusammenarbeit mit anderen IoT-Vorrichtungen zu behandeln, gemäß einigen Ausführungsformen.
- Fig. 128** ist eine Zeichnung einer Netzsteckervorrichtung, die verwendet werden kann, um die Verfügbarkeit einer IoT-Vorrichtung zu verbessern, gemäß einigen Ausführungsformen.
- Fig. 129** ist eine Auftragung eines globalen Zustandsübergangs auf der Grundlage einer Selbstanpassung für die Netzsteckervorrichtung gemäß einigen Ausführungsformen.
- Fig. 130** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung einer Netzsteckervorrichtung zur Erhöhung der Zuverlässigkeit einer IoT-Vorrichtung gemäß einigen Ausführungsformen.

Fig. 131 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer Netzsteckervorrichtung zur Erhöhung der Verfügbarkeit einer IoT-Vorrichtung vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 132 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, die Verfügbarkeit einer IoT-Vorrichtung zu erhöhen, gemäß einigen Ausführungsformen.

Fig. 133 ist ein schematisches Diagramm eines Ausfallsicherungsmechanismus für eine ausgefallene Vorrichtung gemäß einigen Ausführungsformen.

Fig. 134 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine bzw. Trusted Reliability Engine (TRE) gemäß einigen Ausführungsformen.

Fig. 135 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 136 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, einen Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine zu implementieren, gemäß einigen Ausführungsformen.

Fig. 137 ist ein schematisches Diagramm der Konstruktion eines Schlüssels unter Verwendung fraktioneller Schlüssel, die zwischen Knoten in einem IoT-Netz ausgetauscht werden, gemäß einigen Ausführungsformen.

Fig. 138 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Zusammenstellung eines vollständigen Schlüssels aus in einzelnen Knoten in einem IoT-Netz gespeicherten fraktionellen Schlüsseln gemäß einigen Ausführungsformen.

Fig. 139 ist ein schematisches Diagramm der Zusammenstellung eines vollständigen Schlüssels aus von fünf Knoten A - E bereitgestellten fraktionellen Schlüsseln gemäß einigen Ausführungsformen.

Fig. 140 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 141 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, fraktionelle Schlüssel zu empfangen, die fraktionellen Schlüssel zu einem endgültigen Schlüssel zusammenzustellen und den endgültigen Schlüssel zu verwenden, gemäß einigen Ausführungsformen.

Fig. 142 ist ein schematisches Diagramm einer Prozedur zur Erzeugung von Schlüsseln nach Bedarf für Vorrichtungen auf verlustbehafteten Netzen gemäß einigen Ausführungsformen.

Fig. 143 ist ein schematisches Diagramm eines Schlüsselerzeugungsverfahrens, dass im vorstehend beschriebenen bedarfsweisen Prozess für die Schlüsselerzeugung sowie für die Erzeugung von Schlüsseln in anderen Zusammenhängen verwendet werden kann, gemäß einigen Ausführungsformen.

Fig. 144 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Erzeugung von Schlüsseln gemäß einigen Ausführungsformen.

Fig. 145 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Erzeugung von Schlüsseln nach Bedarf vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 146 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code aufweist, um einen Prozessor anzuweisen, Schlüssel nach Bedarf zu erzeugen, gemäß einigen Ausführungsformen.

Fig. 147 ist ein schematisches Diagramm eines Entropiemultiplexierprozesses zur Erzeugung einer Anzahl von Seeds, die für die Erzeugung neuer Schlüssel verwendet werden können, gemäß einigen Ausführungsformen.

Fig. 148 ist ein schematisches Diagramm, das einen Prozess zur Erzeugung eines Orts-Seed-Baums zeigt, gemäß einigen Ausführungsformen.

Fig. 149 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Erzeugung von Seeds unter Verwendung einer Entropiemultiplexierung und zur Verwendung dieser Seeds, um Schlüssel für verschlüsselte Kommunikationen zu erzeugen, gemäß einigen Ausführungsformen.

Fig. 150 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 151 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, eine Entropiemultiplexierung zur Erzeugung eines gemeinsamen Geheimnisses zwischen Vorrichtungen zu verwenden, gemäß einigen Ausführungsformen.

Fig. 152 ist ein Leiterdiagramm eines beispielhaften Verfahrens zur vereinheitlichten Schlüsselbehandlung in einer IoT-Netzumgebung gemäß einigen Ausführungsformen.

Fig. 153 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Behandlung von Schlüsseln in einem Netz von IoT-Mesh-Vorrichtungen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 154 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen.

Fig. 155 ist ein schematisches Diagramm eines Prozesses zum Bootstrappen und zur Entdeckung einer Vorrichtung gemäß einigen Ausführungsformen.

Fig. 156 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Bootstrappen und zur Entdeckung von Vorrichtungen gemäß einigen Ausführungsformen.

Fig. 157 ist ein schematisches Diagramm eines Prozesses zum Bootstrappen, zur Entdeckung und für den Lebenszyklus von Vorrichtungen unter Verwendung von Smart-Contract-Funktionen gemäß einigen Ausführungsformen.

Fig. 158 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Bootstrappen, zur Entdeckung und für den Lebenszyklus von Vorrichtungen unter Verwendung eines Smart Contracts gemäß einigen Ausführungsformen.

Fig. 159 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Bootstrappen, zur Entdeckung und zur Lebenszyklusbehandlung vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 160 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen.

Fig. 161 ist ein schematisches Diagramm eines Prozesses, bei dem Bloomfilter-Hops zur Entdeckung von Ressourcen verwendet werden, gemäß einigen Ausführungsformen.

Fig. 162 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Ressourcenentdeckung unter Verwendung von DHT gemäß einigen Ausführungsformen.

Fig. 163 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 164 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, ein Bloomfilter-Hop-Verfahren zur Ressourcenentdeckung zu verwenden, gemäß einigen Ausführungsformen.

Fig. 165 ist ein schematisches Diagramm eines beispielhaften Verfahrens zur Aufgabendefinition und -kommissionierung gemäß einigen Ausführungsformen.

Fig. 166 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Protokollumwandlungsvermittlung durch einen Protokollumwandlungsvermittler gemäß einigen Ausführungsformen.

Fig. 167 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung vorhanden sein können, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen.

Fig. 168 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen.

Fig. 169 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Behandlung eines schwebenden Dienstes und Werts in einem digitalen Wallet gemäß einigen Ausführungsformen.

Fig. 170 ist ein schematisches Diagramm einer beispielhaften Schwebender-Dienst-Datenstruktur zur Behandlung eines schwebenden Dienstes und der Optionen, Konditionen und Bedingungen gemäß einigen Ausführungsformen.

Fig. 171 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Behandlung schwebender Dienste gemäß einigen Ausführungsformen.

Fig. 172 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Behandlung schwebender Dienste vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 173 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um schwebende Dienste zu behandeln, gemäß einigen Ausführungsformen.

Fig. 174 ist ein schematisches Diagramm eines beispielhaften Erlaubnisanleitungs-Verhandlungsprozesses gemäß einigen Ausführungsformen.

Fig. 175 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Erlaubnisanleitungsverhandlung gemäß einigen Ausführungsformen.

Fig. 176 ist ein schematisches Diagramm einer beispielhaften Datenstruktur zur Beurteilung des Werts einer Dateneinheit und zum Zuweisen des Werts zu der Dateneinheit gemäß einigen Ausführungsformen.

Fig. 177 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für die Verhandlung mit einem Wert versehener Dateneinheiten vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 178 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen.

Fig. 179 ist ein schematisches Diagramm einer beispielhaften Organisation für das dezentralisierte Netzzugangs-Proxy zur Verwendung von Funktionen gemäß einigen Ausführungsformen.

Fig. 180 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für ein dezentralisiertes Netzzugangs-Proxy zur Verwendung von Funktionen gemäß einigen Ausführungsformen.

Fig. 181 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für die Verhandlung mit einem Wert versehener Dateneinheiten vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 182 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen.

Fig. 183 ist ein schematisches Diagramm einer beispielhaften Organisation für eine dezentralisierte Version des Bereitstellens einer Authentifizierung, Autorisierung und Abrechnung mit einer Erlaubnisanleitung gemäß einigen Ausführungsformen.

Fig. 184 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für eine dezentralisierte Version zur Bereitstellung von Authentifizierung, Autorisierung und Abrechnung mit einer Erlaubnisanleitung gemäß einigen Ausführungsformen.

Fig. 185 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 186 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das einen Code zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung aufweist, gemäß einigen Ausführungsformen.

Fig. 187 ist ein schematisches Diagramm einer Technik zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung unter Verwendung eines Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)-Protokolls oder eines DIAMETER-Protokolls gemäß einigen Ausführungsformen.

Fig. 188 ist ein Leiterdiagramm eines beispielhaften Verfahrens für die Komponenten aus **Fig. 187**, um durch eine dezentralisierte API **18706** zur Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung zu wirken, gemäß einigen Ausführungsformen.

Fig. 189 ist ein Leiterdiagramm eines beispielhaften Verfahrens für die Komponenten aus **Fig. 187**, um durch eine dezentralisierte API **18706** zur Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung zu wirken, gemäß einigen Ausführungsformen.

Fig. 190 ist ein schematisches Aktionsdiagramm zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung gemäß einigen Ausführungsformen.

Fig. 191 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 192 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorzunehmen, gemäß einigen Ausführungsformen.

Fig. 193 ist ein schematisches Diagramm eines Prozesses zur Konfiguration und zum Betrieb eines Konsensnetzes unter Verwendung einer nativen dezentralisierten Datenbank gemäß einigen Ausführungsformen.

Fig. 194 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für den Beitritt und den Betrieb innerhalb eines Konsensnetzes unter Verwendung einer nativen dezentralisierten Datenbank gemäß einigen Ausführungsformen.

Fig. 195 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für den Beitritt zu einer dezentralisierten Datenbank und zum Betreiben von dieser vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 196 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, einer dezentralisierten Datenbank beizutreten und diese zu betreiben, gemäß einigen Ausführungsformen.

Fig. 197 ist ein schematisches Diagramm einer logischen Unterteilung für die Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen.

Fig. 198 ist ein schematisches Diagramm logischer Unterteilungen zwischen einem Aufruferberechtigungs-nachweis und einer Anforderung zur Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen.

Fig. 199 ist ein schematisches Diagramm logischer Unterteilungen zwischen einer Objektfähigkeit zur Zugriffssteuerung unter Verwendung von Schichten in einem IoT-Objekt gemäß einigen Ausführungsformen.

Fig. 200 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen.

Fig. 201 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Zugriffssteuerung in einem IoT-Objekt vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 202 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **19600**, das Code aufweist, um einen Prozessor für die Zugriffssteuerung in einem IoT-Objekt anzuweisen, gemäß einigen Ausführungsformen.

Fig. 203 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Abbildung von Ressourcen und Anforderungen selbstbeschreibender Hardware.

Fig. 204 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung vorhanden sein können, um Ressourcen und Anforderungen selbstbeschreibender Hardware abzubilden, gemäß einigen Ausführungsformen.

Fig. 205 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Ressourcen und Anforderungen selbstbeschreibender Hardware abzubilden, gemäß einigen Ausführungsformen.

Fig. 206 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Abbildung von Ressourcen und Anforderungen selbstbeschreibender Hardware gemäß einigen Ausführungsformen.

Fig. 207 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung für ein Berechnungswerkzeug für selbstbeschreibende Hardware vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 208 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Ressourcen und Anforderungen selbstbeschreibender Hardware abzubilden, gemäß einigen Ausführungsformen.

Fig. 209 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Konfiguration einer Signalaufbereitungs-Schaltungsanordnung gemäß einigen Ausführungsformen.

Fig. 210 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Konfiguration einer Signalaufbereitungs-Schaltungsanordnung vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 211 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Signalaufbereitungs-Schaltungsanordnung zu konfigurieren, gemäß einigen Ausführungsformen.

Fig. 212 ist ein schematisches Diagramm einer hierarchischen Vorrichtungs- und Netz-Funktionstüchtigkeitsberichterstattung gemäß einigen Ausführungsformen.

Fig. 213 ist ein schematisches Diagramm einer Vorrichtungsniveau-Bloomfilter- und einer Shadow-Filter-Funktionstüchtigkeitsberichterstattung gemäß einigen Ausführungsformen.

Fig. 214 ist ein schematisches Diagramm eines Netzebenen-Bloomfilters, wodurch über einen historischen intermittierenden Verlust der Watchdog-Berichterstattung berichtet wird, gemäß einigen Ausführungsformen.

Fig. 215 zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Berichterstattung über die Funktionstüchtigkeit unter Verwendung von Shadow- und Bloomfiltern gemäß einigen Ausführungsformen.

Fig. 216 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Berichten über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 217 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigen Ausführungsformen.

Fig. 218 ist ein schematisches Diagramm eines drahtlosen Weitbereichsnetzes bzw. Wireless Wide Area Network (WWAN), wobei ein Steuerkanal über jede Verbindung verwendet werden kann, gemäß einigen Ausführungsformen.

Fig. 219 ist ein schematisches Diagramm einer Karte eines in Zonen unterteilten physikalischen Bereichs gemäß einigen Ausführungsformen.

Fig. 220 zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Mitteilung des geographischen Orts unter Verwendung der Ankunftszeitdifferenz gemäß einigen Ausführungsformen.

Fig. 221 ist ein schematisches Diagramm eines Netzes zur Bestimmung einer Zeitdifferenz auf der Grundlage von Ankunftszeitinformationen in einem heterogenen Netz teilweise unter Verwendung einer Zonenkennung gemäß einigen Ausführungsformen.

Fig. 222 ist ein schematisches Diagramm einer beispielhaften Steuerkanal-Frame-Struktur, die in einen beispielhaften Frame eines Weitbereichsnetzes mit geringer Leistungsaufnahme (LPWAN) gepackt ist, gemäß einigen Ausführungsformen.

Fig. 223 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Entdeckung von Ressourcen und für die Sektoridentifikation geografischer Orte vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 224 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigen Ausführungsformen.

- Fig. 225** ist ein schematisches Diagramm eines Konzeptmodells der Datenanalytik gemäß einigen Ausführungsformen.
- Fig. 226** zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung zur Bereitstellung einer Datenanalytik von IoT-Systemen gemäß einigen Ausführungsformen.
- Fig. 227** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Bereitstellen einer Datenanalytik von IoT-Systemen vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 228** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten.
- Fig. 229** zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung durch eine IoT-Vorrichtung bei einer verteilten Zuordnung eines neuronalen Netzes und der Ressourcenverwaltung gemäß einigen Ausführungsformen.
- Fig. 230** ist ein schematisches Diagramm für eine verteilte Zuordnung eines neuronalen Netzes zur Ressourcenverwaltung gemäß einigen Ausführungsformen.
- Fig. 231** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur verteilten Zuordnung eines neuronalen Netzes und zur Ressourcenverwaltung vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 232** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigen Ausführungsformen.
- Fig. 233** ist ein schematisches Diagramm einer Hierarchie von Blockchains in Zusammenhang mit Ebenen in einer Netzhierarchie gemäß einigen Ausführungsformen.
- Fig. 234** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Bildung einer Blockchain-Hierarchie gemäß einigen Ausführungsformen.
- Fig. 235** ist eine erweiterte Ansicht der mit Bezug auf **Fig. 233** beschriebenen Merkle-Bäume gemäß einigen Ausführungsformen.
- Fig. 236** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zum Durchsuchen einer Blockchain-Hierarchie unter Verwendung von Merkle-Baumindizes gemäß einigen Ausführungsformen.
- Fig. 237** ist ein schematisches Diagramm eines in einem Cloud-Server gespeicherten gecachten Merkle-Baums gemäß einigen Ausführungsformen.
- Fig. 238** zeigt ein schematisches Diagramm eines verteilten Merkle-Baum-Caches auf der IoT-Netzebene H1, wie mit Bezug auf **Fig. 233** beschrieben, gemäß einigen Ausführungsformen.
- Fig. 239** ist ein schematisches Diagramm einer Technik zum Unterhalten eines verteilten Caches mit Kohärenz gemäß einigen Ausführungsformen.
- Fig. 240** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Bildung eines kohärenten Caches für eine Hierarchie von Blockchains gemäß einigen Ausführungsformen.
- Fig. 241** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Aufrechterhaltung eines kohärenten Caches für eine Hierarchie von Blockchains gemäß einigen Ausführungsformen.
- Fig. 242** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Implementation hierarchischer Blockchains mit zugeordneten Indizes vorhanden sein können, gemäß einigen Ausführungsformen.
- Fig. 243** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen.
- Fig. 244** ist ein schematisches Diagramm der Verwendung der Pub-Sub-Weiterleitung auf der Grundlage von Bloomfiltern gemäß einigen Ausführungsformen.
- Fig. 245** ist ein schematisches Diagramm der Verwendung eines Whitelist-Bloomfilters zum Ermöglichen der Verteilung von Inhalt gemäß einigen Ausführungsformen.
- Fig. 246** ist ein schematisches Diagramm der Verwendung eines Blacklist-Bloomfilters zum Verhindern der Verteilung von Inhalt gemäß einigen Ausführungsformen.

Fig. 247 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Implementation von Pub-Sub mit Blacklist- oder Whitelist-Bloomfiltern zur Inhaltssteuerung gemäß einigen Ausführungsformen.

Fig. 248 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zur Implementation eines Pub-Sub-Inhaltsverteilungssystems unter Verwendung von Bloomfiltern vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 249 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, ein Pub-Sub-System unter Verwendung von Bloomfiltern zur Inhaltsverteilung zu behandeln, gemäß einigen Ausführungsformen.

Fig. 250 ist ein schematisches Diagramm einer Themenbenachrichtigung mit verschlüsseltem Inhalt gemäß einigen Ausführungsformen.

Fig. 251(A) ist ein schematisches Diagramm einer Gruppe von Routern, die Benachrichtigungen über ein Thema empfangen, das verschlüsselten Inhalt aufweist, gemäß einigen Ausführungsformen.

Fig. 251(B) ist ein schematisches Diagramm einer Gruppe von Routern, welche ihre Caches in Erwartung eines Subskribenten, der ein verschlüsseltes Thema anfordert, vorwarnen, gemäß einigen Ausführungsformen.

Fig. 252 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Verwendung einer Schlüsselbehandlungsbenachrichtigung und zum Vorwärmen des Schlüssel-Cachings gemäß einigen Ausführungsformen.

Fig. 253 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Verwalten der Themenbenachrichtigung mit verschlüsseltem Inhalt vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 254 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, eine Themenbenachrichtigung mit verschlüsseltem Inhalt zu behandeln, gemäß einigen Ausführungsformen.

Fig. 255 ist ein schematisches Diagramm eines Subskribenten, der einen Themengruppenschlüssel erhält, gemäß einigen Ausführungsformen.

Fig. 256 ist ein schematisches Diagramm eines Veröffentlichers, der ein Subskriptions-Bloomfilter zur Benachrichtigung von Subskribenten über verfügbare Themen erzeugt, gemäß einigen Ausführungsformen.

Fig. 257 ist ein Leiterdiagramm eines beispielhaften Verfahrens zur Themenverschlüsselung gemäß einigen Ausführungsformen.

Fig. 258 ist ein schematisches Diagramm der Verwendung von Mehrniveau-Sicherheitsetiketten in einer Veröffentlichen-Subskribieren-Umgebung gemäß einigen Ausführungsformen.

Fig. 259 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Implementation von Bloomfiltern für die Anwendung von Mehrniveau-Sicherheitsrichtlinien auf Benachrichtigungsnachrichten gemäß einigen Ausführungsformen.

Fig. 260 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung zum Verwalten der Themenbenachrichtigung mit verschlüsseltem Inhalt vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 261 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, das Code aufweist, um einen Prozessor anzuweisen, eine Themenbenachrichtigung mit verschlüsseltem Inhalt zu behandeln, gemäß einigen Ausführungsformen.

Fig. 262 ist eine Zeichnung eines Beispiels eines scheuen Roboters gemäß einigen Ausführungsformen.

Fig. 263 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem scheuen Roboter vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 264 ist ein schematisches Diagramm des Betriebs der Kontextmaschine gemäß einigen Ausführungsformen.

Fig. 265 ist ein schematisches Diagramm des Betriebs eines Schwarms scheuer Roboter gemäß einigen Ausführungsformen.

Fig. 266 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens für den Betrieb eines scheuen Roboters in einem Schwarm gemäß einigen Ausführungsformen.

Fig. 267 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **27500**, dass Code aufweist, um einen Prozessor **902** anzuweisen, Operationen von Pre-first-Respondervorrichtungen zu behandeln, gemäß einigen Ausführungsformen.

Fig. 268 ist ein schematisches Diagramm eines Verwendungsfalls, worin Drohnen als Pre-First-Responder-Vorrichtungen an einer Szene innerhalb eines Zuständigkeitsbereichs dargestellt sind, gemäß einigen Ausführungsformen.

Fig. 269 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur Ausführung eines Beitritts- und Registrierungsprozesses in Zusammenhang mit der einzelnen und den mehreren Zuständigkeitsbereichs-Kontrollzonen in **Fig. 268** gemäß einigen Ausführungsformen.

Fig. 270 ist ein schematisches Diagramm einer Fahrtplanung für einen Notfall-Responder (ER) oder eine andere Entität, um einen Weg zu einem Ziel zu bestimmen, gemäß einigen Ausführungsformen.

Fig. 271 ist ein schematisches Diagramm der Verwendung von Emergency-Management(EM)-Teilbäumen an jedem Wegpunkt gemäß einigen Ausführungsformen.

Fig. 272 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens zur dynamischen Konfiguration eines Pre-First-Responder-Dispatch(PFRD)-Netzes an einer Szene gemäß einigen Ausführungsformen.

Fig. 273 ist ein Leiterdiagramm eines beispielhaften Verfahrens zum Versehen von Szeneninformationen mit Baken durch eine PFRD gemäß einigen Ausführungsformen.

Fig. 274 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer PFRD vorhanden sein können, gemäß einigen Ausführungsformen.

Fig. 275 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums, dass Code aufweist, um einen Prozessor anzuweisen, Operationen von Pre-first-Respondervorrichtungen zu behandeln, gemäß einigen Ausführungsformen.

[0006] Überall in dieser Offenbarung und in den Figuren werden die gleichen Zahlen verwendet, um auf gleiche Komponenten und Merkmale Bezug zu nehmen. Zahlen in der 100-Serie beziehen sich auf Merkmale, die ursprünglich in **Fig. 1** angetroffen wurden, Zahlen in der 200-Serie beziehen sich auf Merkmale, die ursprünglich in **Fig. 2** angetroffen wurden, usw.

Beschreibung der Ausführungsformen

[0007] Das Internet der Dinge (IoT) ist ein System, bei dem eine große Anzahl von Rechenvorrichtungen miteinander und mit einem Kommunikationsnetz (beispielsweise dem Internet) verbunden ist, um eine Funktionalität in der Art einer Datenerfassung und Betätigung auf sehr niedrigen Ebenen in Netzen bereitzustellen. Niedrige Ebenen geben Vorrichtungen an, die sich an oder in der Nähe der Ränder des Netzes befinden können, wie die letzten Vorrichtungen, bevor die Netze enden. Hier kann eine IoT-Vorrichtung eine Vorrichtung einschließen, die eine Funktion in der Art unter anderem einer Messung oder Steuerung in Kommunikation mit anderen IoT-Vorrichtungen und einem Kommunikationsnetz ausführt. Die IoT-Vorrichtung kann eine autonome Vorrichtung oder eine halbautonome Vorrichtung, die dafür ausgelegt ist, eine oder mehrere Funktionen auszuführen, einschließen. Häufig können IoT-Vorrichtungen in Bezug auf den Speicher, die Größe oder die Funktionalität beschränkt sein, wodurch größere Anzahlen zu ähnlichen Kosten wie eine kleinere Anzahl größerer Vorrichtungen eingesetzt werden können. Eine IoT-Vorrichtung kann jedoch auch ein Smartphone, ein Laptop, ein Tablet, ein PC und/oder eine andere größere Vorrichtung sein. Ferner kann eine IoT-Vorrichtung eine virtuelle Vorrichtung in der Art einer Anwendung auf einem Smartphone oder einer anderen Rechenvorrichtung sein. IoT-Vorrichtungen können IoT-Gateways aufweisen, die zur Kopplung von IoT-Vorrichtungen mit anderen IoT-Vorrichtungen und mit Cloud-Anwendungen für die Datenspeicherung, Prozesssteuerung und dergleichen verwendet werden.

[0008] Netze von IoT-Vorrichtungen können kommerzielle und Heimvorrichtungen in der Art von Wasserverteilungssystemen, Systemen zur Verteilung elektrischer Energie, Pipelinesteuersystemen, Anlagensteuersystemen, Lichtschaltern, Thermostaten, Schlössern, Kameras, Alarmen, Bewegungssensoren und dergleichen aufweisen. Die IoT-Vorrichtungen können durch eine Steuereinrichtung in der Art von Computern, Servern und anderen Systemen, beispielsweise zum Steuern von Systemen oder zum Zugriff auf Daten, zugänglich sein. Die Steuereinrichtung und die IoT-Vorrichtungen können sich fern voneinander befinden.

[0009] Das Internet kann dafür ausgelegt werden, Kommunikationen mit einer großen Anzahl von IoT-Vorrichtungen bereitzustellen. Dementsprechend wird, wie hier beschrieben, eine Anzahl von Innovationen für das

künftige Internet entwickelt, um den Bedarf an Netzschichten von zentralen Servern über Gateways bis hinab zu Edge-Vorrichtungen zu adressieren, um ungehindert zu wachsen, angeschlossene Ressourcen zu entdecken und zugänglich zu machen und die Fähigkeit zu unterstützen, angeschlossene Ressourcen zu verbergen und aufzuteilen. Es kann eine beliebige Anzahl von Netzprotokollen und Kommunikationsstandards verwendet werden, wobei jedes Protokoll und jeder Standard dafür ausgelegt ist, spezifische Ziele zu adressieren. Ferner sind die Protokolle Teil des Gewebes, wodurch von Menschen zugängliche Dienste unterstützt werden, die unabhängig vom Ort, von der Zeit oder vom Raum arbeiten. Die Innovationen umfassen die Dienstausslieferung und die zugeordnete Infrastruktur in der Art von Hardware und Software. Die Dienste können entsprechend den auf der Dienstebene spezifizierten Dienstqualität(QoS)-Bedingungen und Dienstausslieferungsvereinbarungen bereitgestellt werden. Die Verwendung von IoT-Vorrichtungen und -Netzen führt zu einer Anzahl neuer Herausforderungen in einem heterogenen Verbindbarkeitsnetz, einschließlich einer Kombination drahtgebundener und drahtloser Technologien, wie in den **Fig. 1** und **Fig. 2** dargestellt ist.

[0010] **Fig. 1** zeigt Zwischenverbindungen, die zwischen dem Internet **100** und IoT-Netzen gemäß einigen Ausführungsformen vorhanden sein können. Die Zwischenverbindungen können kleinere Netze **102** bis hinab zur einzelnen IoT-Vorrichtung **104** mit dem Backbone **106** des Internets **100** koppeln. Zur Vereinfachung der Darstellung ist nicht jede Vorrichtung **104** oder ein anderes Objekt bezeichnet.

[0011] In **Fig. 1** sind Top-Level-Provider, die als Tier-1 („T1“-)Provider **108** bezeichnet werden können, durch den Backbone **106** des Internets mit anderen Providern in der Art sekundärer oder Tier-2(„T2“-)Providern **110** gekoppelt. Bei einigen Aspekten kann der Backbone **106** optische Faserstrecken aufweisen. Bei einem Beispiel kann ein T2-Provider **110** beispielsweise durch weitere Verbindungsstrecken, durch Mikrowellenkommunikationen **114** oder durch andere Kommunikationstechnologien, mit einem Mast **112** eines LTE-Mobilfunknetzes koppeln. Der Mast **112** kann durch eine LTE-Kommunikationsstrecke **116**, beispielsweise durch einen zentralen Knoten **118**, mit einem Mesh-Netz koppeln, das IoT-Vorrichtungen **104** aufweist. Die Kommunikationen zwischen den einzelnen IoT-Vorrichtungen **104** können auch auf LTE-Kommunikationsstrecken **116** beruhen.

[0012] Bei einem anderen Beispiel kann ein schneller Uplink **119** einen T2-Provider **110** mit einem Gateway **120** koppeln. Eine Anzahl von IoT-Vorrichtungen **104** kann mit dem Gateway **120** und miteinander durch den Gateway **120**, beispielsweise über Bluetooth-Low-Energy(BLE)-Strecken **122**, kommunizieren.

[0013] Der Backbone **106** kann niedrigere Ebenen von Diensteanbietern mit dem Internet verbinden, beispielsweise Tier-3(„T3“-)Provider **124**. Ein T3-Provider **124** kann als ein allgemeiner Internetdiensteanbieter (ISP) angesehen werden, der beispielsweise vom T2-Provider **110** einen Zugriff auf den Backbone **106** kauft und einem Firmen-Gateway **126** und anderen Kunden Zugang bereitstellt.

[0014] Vom Firmen-Gateway **126** kann ein drahtloses lokales Netz (WLAN) zur Kommunikation mit IoT-Vorrichtungen **104** durch WiFi®-Strecken **128** verwendet werden. Eine WiFi-Strecke **128** kann auch zur Kopplung mit einem Weitbereichs-Gateway mit geringer Leistungsaufnahme (LPWA-Gateway) **130** verwendet werden, der über LPWA-Strecken **132** mit IoT-Vorrichtungen **104** kommunizieren kann, welche beispielsweise mit der von der LoRa-Allianz verbreiteten LoRaWan-Spezifikation kompatibel sind.

[0015] Der T3-Provider **124** kann durch eine Koordinatorvorrichtung **136**, die unter Verwendung einer Anzahl von Kommunikationsstrecken in der Art einer LTE-Mobilfunkstrecke, einer LPWA-Strecke oder einer Strecke **138** auf der Grundlage des IEEE 802.15.4-Standards in der Art von Zigbee® mit dem T3-Provider **124** kommuniziert, auch Zugang zu einem Mesh-Netz **134** bereitstellen. Andere Koordinatorvorrichtungen **136** können eine Kette von Verbindungsstrecken bereitstellen, die einen oder mehrere Cluster-Bäume verbundener Vorrichtungen bilden.

[0016] Bei einigen Aspekten weisen eine oder mehrere IoT-Vorrichtungen **104** den geeigneten Sendeempfänger für die Kommunikationen mit anderen Vorrichtungen auf. Ferner können eine oder mehrere IoT-Vorrichtungen **104** andere Funk-, optische oder akustische Sendeempfänger sowie drahtgestützte Netzschnittstellen für Kommunikationen unter Verwendung zusätzlicher Protokolle und Frequenzen aufweisen. Bei einigen Aspekten weisen eine oder mehrere IoT-Vorrichtungen **104** mit Bezug auf **Fig. 8** beschriebene Komponenten auf.

[0017] Die Technologien und Netze können das Wachstum von Vorrichtungen und Netzen ermöglichen. Wenn die Technologien wachsen, kann das Netz für die Selbstverwaltung, funktionelle Entwicklung und/oder Zusammenarbeit entwickelt werden, ohne dass ein direkter menschlicher Eingriff erforderlich wäre. Demgemäß können die Technologien ermöglichen, dass Netze ohne zentralisierte gesteuerte Systeme funktionieren. Die hier beschriebenen Technologien können die Netzverwaltungs- und Betriebsfunktionen über aktuelle Fähigkeiten

hinaus automatisieren. Ferner können die Ansätze die Flexibilität, eine zentralisierte Steuerung, die ohne menschlichen Eingriff funktioniert, eine zentralisierte Steuerung, die automatisiert ist, oder Kombinationen davon zu haben, bereitstellen.

[0018] Fig. 2 zeigt eine Netztopologie **200**, die für eine Anzahl von Internet-der-Dinge(IoT)-Netzen, die durch Backbone-Strecken **202** mit Gateways **204** gekoppelt sind, verwendet werden kann, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 1 beschrieben wurden. Ferner ist zur Vereinfachung der Darstellung nicht jede Vorrichtung **104** oder Kommunikationsstrecke **116**, **122**, **128** oder **132** bezeichnet. Die Backbone-Strecken **202** können eine beliebige Anzahl drahtgebundener oder drahtloser Technologien aufweisen und Teil eines lokalen Netzes (LAN), eines Weitbereichsnetzes (WAN) oder des Internets sein.

[0019] Wenngleich die Topologien in Fig. 2 Hub-and-Spoke-Topologien sind und die Topologien in Fig. 1 Peer-to-Peer-Topologien sind, kann beobachtet werden, dass diese nicht in Konflikt sind, sondern dass sich Peer-to-Peer-Knoten durch Gateways als Hub-and-Spoke-Knoten verhalten können. Es kann in Fig. 2 auch beobachtet werden, dass eine Subnetztopologie mehrere Gateways aufweisen kann, so dass sie zu einer hybriden Topologie an Stelle einer reinen Hub-and-Spoke-Topologie an Stelle einer konsequenten Hub-and-Spoke-Topologie wird.

[0020] Die Netztopologie **200** kann eine beliebige Anzahl von IoT-Netztypen in der Art eines Mesh-Netzes **206** unter Verwendung von Bluetooth-Low-Energy(BLE)-Strecken **122** aufweisen. Andere IoT-Netze, die vorhanden sein können, umfassen ein WLAN-Netz **208**, ein Mobilfunknetz **210** und ein LPWA-Netz **212**. Jedes dieser IoT-Netze kann Gelegenheiten für neue Entwicklungen bereitstellen, wie hier beschrieben wird.

[0021] Beispielsweise können Kommunikationen zwischen IoT-Vorrichtungen **104** beispielsweise über die Backbone-Strecken **202** durch ein zentralisiertes System zur Authentifizierung, Autorisierung und Abrechnung (AAA) geschützt werden. In einem dezentralisierten AAA-System können verteilte Zahlungs-, Kredit-, Audit-, Autorisierungs-, Handels-, Arbitrierungs- und Authentifizierungssysteme über verbundene heterogene Infrastruktur implementiert werden. Dies ermöglicht es, dass sich Systeme und Netze zu autonomen Operationen bewegen.

[0022] Bei diesen Typen autonomer Operationen können Maschinen menschliche Ressourcen kontraktieren und Partnerschaften mit anderen Maschinennetzen verhandeln. Dies kann es ermöglichen, wechselseitige Ziele und eine ausgeglichene Dienstausslieferung anhand dargelegter, geplanter Dienstniveauvereinbarungen zu erreichen und Lösungen zu erreichen, die eine Dosierung, Messungen und Verfolgbarkeit und Nachvollziehbarkeit bereitstellen. Die Erzeugung neuer Versorgungskettenstrukturen und -Verfahren kann es ermöglichen, eine Vielzahl von Diensten zu erzeugen, sie auf Wert hin zu durchsuchen und sie ohne menschlichen Eingriff zu kollabieren.

[0023] Die IoT-Netze können ferner durch die Integration von Messtechnologien in der Art von Schall-, Licht-, elektronischen Verkehrs-, Gesichts- und Mustererkennungs-, Geruchs- und Vibrationsmesstechnologien in die autonomen Organisationen erweitert werden. Die Integration sensorischer Systeme kann eine systematische und autonome Kommunikation und Koordination der Dienstausslieferung anhand vertraglicher Dienstziele, eine Orchestrierung und eine Dienstqualitäts(QoS)-basierte Verbreitung und Fusion von Ressourcen ermöglichen.

[0024] Das Mesh-Netz **206** kann durch Systeme erweitert werden, die Inline-Daten-zu-Informationen-Transformationen ausführen. Beispielsweise können sich selbst bildende Ketten von Verarbeitungsressourcen, die ein Mehrverbindungsnetz umfassen, die Transformation von Rohdaten in Informationen wirksam verteilen. Dies kann eine solche Funktionalität als eine erste Stufe ermöglichen, die eine erste numerische Operation ausführt, bevor das Ergebnis an eine andere Stufe übergeben wird, wobei die nächste Stufe dann eine andere numerische Operation ausführt und das Ergebnis an die andere Stufe übergibt. Das System kann die Fähigkeit, zwischen Vermögensgegenständen und Ressourcen zu unterscheiden, und die zugeordnete Verwaltung dieser jeweiligen bereitstellen. Ferner können die geeigneten Infrastruktur- und Ressourcenkomponenten auf der Grundlage von Vertrauens- und Dienstindizes eingefügt werden, um die Datenintegrität zu verbessern, die Qualität zu gewährleisten und eine Datenvertrauensmetrik zu liefern.

[0025] Wie hier beschrieben, kann das WLAN-Netz **208** Systeme verwenden, welche eine Wandlung zwischen Standards ausführen, um eine Konnektivität zwischen mehreren Standards bereitzustellen, wodurch es IoT-Vorrichtungen **104** ermöglicht wird, unter Verwendung verschiedener Protokolle zu kommunizieren. Weitere

Systeme können eine nahtlose Verbindbarkeit über eine mehrere Standards aufweisende Infrastruktur bereitstellen, welche sichtbare Internet-Ressourcen und verborgene Internet-Ressourcen umfasst.

[0026] Kommunikationen im Mobilfunknetz **210** können durch Systeme erweitert werden, die Daten abladen, Kommunikationen zu fernen Vorrichtungen erweitern oder beides ausführen. Das LPWA-Netz **212** kann Systeme aufweisen, welche Nicht-Internetprotokoll(IP)-zu-IP-Verbindungen, eine entsprechende Adressierung und eine entsprechende Weiterleitung ausführen.

[0027] **Fig. 3** ist eine Zeichnung **300** eines Cloud-Rechnernetzes oder einer Cloud **302** in Kommunikation mit einer Anzahl von Internet-der-Dinge(IoT)-Vorrichtungen gemäß einigen Ausführungsformen. Die Cloud **302** kann das Internet repräsentieren oder ein lokales Netz (LAN) oder ein Weitbereichsnetz (WAN) in der Art eines proprietären Netzes für eine Firma sein. Die IoT-Vorrichtungen können eine beliebige Anzahl verschiedener Vorrichtungstypen, die in verschiedenen Kombinationen gruppiert sind, aufweisen. Beispielsweise kann eine Verkehrssteuergruppe **306** IoT-Vorrichtungen entlang Straßen in einer Stadt aufweisen. Diese IoT-Vorrichtungen können Ampelanlagen, Verkehrsfluss-Überwachungseinrichtungen, Kameras, Wettersensoren und dergleichen einschließen. Die Verkehrssteuergruppe **306** oder andere Untergruppen können durch drahtlose Strecken **308** in der Art von LPWA-Strecken und dergleichen in Kommunikation mit der Cloud **302** stehen. Ferner kann ein drahtgestütztes oder drahtloses Subnetz **312** den IoT-Vorrichtungen ermöglichen, miteinander zu kommunizieren, beispielsweise durch ein lokales Netz, ein drahtloses lokales Netz und dergleichen. Die IoT-Vorrichtungen können eine andere Vorrichtung in der Art eines Gateways **310** zur Kommunikation mit der Cloud **302** verwenden.

[0028] Andere Gruppen von IoT-Vorrichtungen können ferne Wetterstationen **314**, lokale Informationsendgeräte **316**, Alarmsysteme **318**, Geldautomaten **320**, Alarmtafeln **322** oder sich bewegende Fahrzeuge in der Art von Einsatzfahrzeugen **324** oder anderen Fahrzeugen **326** sowie viele andere einschließen. Jede dieser IoT-Vorrichtungen kann in Kommunikation mit anderen IoT-Vorrichtungen, mit Servern **304** oder beiden stehen.

[0029] Wie anhand **Fig. 3** ersichtlich ist, kann eine große Anzahl von IoT-Vorrichtungen über die Cloud **302** kommunizieren. Dies kann es verschiedenen IoT-Vorrichtungen ermöglichen, von anderen Vorrichtungen autonom Informationen anzufordern oder diesen autonom Informationen bereitzustellen. Beispielsweise kann die Verkehrssteuergruppe **306** eine aktuelle Wettervorhersage von einer Gruppe ferner Wetterstationen **314** anfordern, welche die Vorhersage ohne menschlichen Eingriff bereitstellen können. Ferner kann ein Einsatzfahrzeug **324** durch einen Geldautomaten **320** gewarnt werden, dass ein Einbruch abläuft. Während sich das Einsatzfahrzeug **324** dem Geldautomaten **320** nähert, kann es auf die Verkehrssteuergruppe **306** zugreifen, um einen freien Zugang zum Ort anzufordern, beispielsweise durch Ampeln, die auf Rot schalten, um Querverkehr an einer Kreuzung genügend lange zu blockieren, damit das Einsatzfahrzeug **324** ungehinderten Zugang zur Kreuzung hat.

[0030] Cluster von IoT-Vorrichtungen in der Art ferner Wetterstationen **314** oder der Verkehrssteuergruppe **306** können dafür ausgerüstet sein, mit anderen IoT-Vorrichtungen sowie mit der Cloud **302** zu kommunizieren. Dies kann es den IoT-Vorrichtungen ermöglichen, ein Ad-hoc-Netz zwischen den Vorrichtungen zu bilden, so dass sie als eine einzige Vorrichtung wirken können, die als Fog-Vorrichtung bezeichnet werden kann. Die Fog-Vorrichtung wird weiter mit Bezug auf **Fig. 4** erörtert.

[0031] **Fig. 4** ist eine Zeichnung **400** eines Cloud-Rechnernetzes oder einer Cloud **302** in Kommunikation mit einem Mesh-Netz von IoT-Vorrichtungen, das als Fog-Vorrichtung **402** bezeichnet werden kann, das am Rand der Cloud **302** arbeitet, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 3** beschrieben wurden. Hier ist eine Fog-Vorrichtung **402** ein Cluster von Vorrichtungen, die gruppiert werden können, um eine spezifische Funktion in der Art einer Verkehrssteuerung, einer Wettersteuerung, einer Anlagensteuerung und dergleichen auszuführen.

[0032] Bei diesem Beispiel weist die Fog-Vorrichtung **402** eine Gruppe von IoT-Vorrichtungen an einer Verkehrskreuzung auf. Die Fog-Vorrichtung **402** kann unter anderem gemäß vom OpenFog Consortium (OFC) herausgegebenen Spezifikationen eingerichtet sein. Diese Spezifikationen ermöglichen die Bildung einer Hierarchie von Rechenelementen zwischen den Gateways, welche die Fog-Vorrichtung **402** mit der Cloud **302** und mit Endpunktvorrichtungen in der Art von Ampeln **404** und Daten-Aggregatoren **406** bei diesem Beispiel koppeln. Die Fog-Vorrichtung **402** kann die kombinierten Verarbeitungs- und Netzressourcen ausnutzen, die das Kollektiv der IoT-Vorrichtungen bereitstellt. Dementsprechend kann eine Fog-Vorrichtung **402** für eine beliebige Anzahl von Anwendungen verwendet werden, einschließlich beispielsweise der finanziellen Modellierung, der Wettervorhersage, von Verkehrsanalysen und dergleichen.

[0033] Beispielsweise kann der Verkehrsfluss durch die Kreuzung durch mehrere Ampeln **404** (beispielsweise drei Ampeln **404**) gesteuert werden. Analysen des Verkehrsflusses und von Steuerschemata können durch Aggregatoren **406** implementiert werden, die durch ein Mesh-Netz in Kommunikation mit den Ampeln **404** und miteinander stehen. Daten können durch Gateways **310**, die durch das Mesh-Netz in Kommunikation mit den Ampeln **404** und den Aggregatoren **406** stehen, zur Cloud **302** hochgeladen werden, und Befehle können dadurch von der Cloud **302** empfangen werden.

[0034] Eine beliebige Anzahl von Kommunikationsstrecken kann in der Fog-Vorrichtung **402** verwendet werden. Kommunikationsstrecken **408** kürzerer Reichweite, die beispielsweise mit IEEE 802.15.4 kompatibel sind, können lokale Kommunikationen zwischen IoT-Vorrichtungen bereitstellen, die sich in der Nähe der Kreuzung befinden. Kommunikationsstrecken **410** längerer Reichweite, die beispielsweise mit LPWA-Standards kompatibel sind, können Kommunikationen zwischen den IoT-Vorrichtungen und den Gateways **310** bereitstellen. Zur Vereinfachung des Diagramms ist nicht jede Kommunikationsstrecke **408** oder **410** mit einer Bezugszahl bezeichnet.

[0035] Die Fog-Vorrichtung **402** kann als ein massiv zwischenverbundenes Netz angesehen werden, wobei eine Anzahl von IoT-Vorrichtungen in Kommunikation miteinander steht, beispielsweise durch die Kommunikationsstrecken **408** und **410**. Das Netz kann unter Verwendung der am 23. Dezember 2015 von der Open Connectivity Foundation™ (OCF) herausgegebenen Open-Interconnect-Consortium(OIC)-Standardspezifikation 1.0 eingerichtet werden. Dieser Standard ermöglicht es Vorrichtungen, einander zu entdecken und Kommunikationen für Zwischenverbindungen einzurichten. Andere Zwischenverbindungsprotokolle können auch verwendet werden, einschließlich beispielsweise des AllJoyn-Protokolls von der AllSeen-Allianz, des Optimized-Link-State-Routing(OLSR)-Protokolls oder des Better-Approach-To-Mobile-Ad-hoc-Networking (B.A.T.M.A.N.) sowie anderer.

[0036] Bei einigen Aspekten können Kommunikationen von einer IoT-Vorrichtung über den zweckmäßigsten Weg geleitet werden, um die Gateways **310** zu erreichen, beispielsweise den Weg mit der geringsten Anzahl von Zwischen-Hops oder mit der höchsten Bandbreite etc. Bei diesen Netzen stellt die Anzahl der Zwischenverbindungen eine erhebliche Redundanz bereit, wodurch es ermöglicht wird, Kommunikationen selbst beim Verlust einer Anzahl von IoT-Vorrichtungen aufrechtzuerhalten.

[0037] Bei einigen Aspekten kann die Fog-Vorrichtung **402** temporäre IoT-Vorrichtungen aufweisen. Mit anderen Worten können nicht alle IoT-Vorrichtungen permanente Mitglieder der Fog-Vorrichtung **402** sein. Beispielsweise haben sich beim beispielhaften System **400** drei transiente IoT-Vorrichtungen der Fog-Vorrichtung **402** angeschlossen, nämlich ein erstes Fahrzeug **412**, ein zweites Fahrzeug **414** und ein Fußgänger **416**. In diesen Fällen kann die IoT-Vorrichtung in die Fahrzeuge **412** und **414** eingebaut sein oder eine App auf einem vom Fußgänger **416** getragenen Smartphone sein. Es können auch andere IoT-Vorrichtungen vorhanden sein, wie IoT-Vorrichtungen in Fahrradcomputern, Motorradcomputern, Drohnen und dergleichen.

[0038] Die aus den ein IoT-Vorrichtungen gebildete Fog-Vorrichtung **402** kann Clients in der Cloud **302** in der Art des Servers **304** als einzelne Vorrichtung, die sich am Rand der Cloud **302** befindet, präsentiert werden. Bei diesem Beispiel können die Steuerkommunikationen zu spezifischen Ressourcen in der Fog-Vorrichtung **402** geschehen, ohne eine spezifische IoT-Vorrichtung innerhalb der Fog-Vorrichtung **402** zu identifizieren. Falls eine IoT-Vorrichtung innerhalb der Fog-Vorrichtung **402** ausfällt, können demgemäß andere IoT-Vorrichtungen in der Fog-Vorrichtung **402** in der Lage sein, eine Ressource in der Art eines Aktuators oder einer anderen an eine IoT-Vorrichtung angeschlossenen Vorrichtung zu entdecken und zu steuern. Beispielsweise können die Ampeln **404** so verdrahtet sein, dass es jeder der Ampeln **404** ermöglicht wird, Lichter für die anderen Ampeln **404** zu steuern. Die Aggregatoren **406** können auch eine Redundanz bei der Steuerung der Ampeln **404** und andere Funktionen der Fog-Vorrichtung **402** bereitstellen.

[0039] Bei einigen Beispielen können die IoT-Vorrichtungen unter Verwendung eines imperativen Programmierstils konfiguriert werden, wobei beispielsweise jede IoT-Vorrichtung eine spezifische Funktion und spezifische Kommunikationspartner aufweist. Die IoT-Vorrichtungen, welche die Fog-Vorrichtung **402** bilden, können jedoch in einem deklarativen Programmierstil konfiguriert werden, wodurch es den IoT-Vorrichtungen ermöglicht wird, ihre Operationen und Kommunikationen zu rekonfigurieren, beispielsweise um erforderliche Ressourcen ansprechend auf Bedingungen, Abfragen und Vorrichtungsfehler zu bestimmen. Dies kann ausgeführt werden, wenn sich transiente IoT-Vorrichtungen in der Art des Fußgängers **416** der Fog-Vorrichtung **402** anschließen.

[0040] Weil es wahrscheinlich ist, dass sich der Fußgänger **416** langsamer bewegt als die Fahrzeuge **412** und **414**, kann sich die Fog-Vorrichtung **402** rekonfigurieren, um zu gewährleisten, dass der Fußgänger **116** genügend Zeit hat, um es über die Kreuzung zu schaffen. Dies kann durch die Bildung einer temporären Gruppe der Fahrzeuge **412** und **414** und des Fußgängers **416** zur Steuerung der Ampeln **404** geschehen. Falls eines oder beide der Fahrzeuge **412** oder **414** autonom ist, kann die temporäre Gruppe die Fahrzeuge anweisen, vor den Ampeln **404** zu verlangsamen. Ferner kann, falls alle Fahrzeuge an der Kreuzung autonom sind, die Notwendigkeit von Verkehrssignalen verringert werden, weil Kollisionsvermeidungssysteme autonomer Fahrzeuge stark verschachtelte Verkehrsmuster ermöglichen können, die zu komplex sind, um durch Ampeln behandelt zu werden. Die Ampel **404** kann jedoch für den Fußgänger **416**, für Radfahrer oder für nicht autonome Fahrzeuge weiter wichtig sein.

[0041] Wenn die transienten Vorrichtungen **412**, **414** und **416** den näheren Bereich der Kreuzung der Fog-Vorrichtung **402** verlassen, kann sich die Fog-Vorrichtung **402** rekonfigurieren, um diese IoT-Vorrichtungen aus dem Netz zu beseitigen. Wenn sich andere transiente IoT-Vorrichtungen der Kreuzung nähern, kann sich die Fog-Vorrichtung **402** rekonfigurieren, um diese Vorrichtungen aufzunehmen.

[0042] Die Fog-Vorrichtung **402** kann die Ampeln **404** für eine Anzahl von Kreuzungen, beispielsweise entlang einer Straße, zusammen mit allen transienten IoT-Vorrichtungen entlang der Straße aufweisen. Die Fog-Vorrichtung **402** kann sich in funktionelle Einheiten in der Art der Ampeln **404** und anderer IoT-Vorrichtungen in der Nähe einer einzigen Kreuzung unterteilen. Dieser Kombinationstyp kann die Bildung größerer IoT-Konstrukte, beispielsweise Gruppen von IoT-Vorrichtungen, die eine bestimmte Funktion ausführen, in der Fog-Vorrichtung **402** ermöglichen.

[0043] Falls sich beispielsweise ein Einsatzfahrzeug der Fog-Vorrichtung **402** anschließt, kann ein Notkonstrukt oder eine virtuelle Vorrichtung erzeugt werden, welche alle Ampeln **404** für die Straße aufweist, wodurch die Verkehrsflussmuster für die gesamte Straße gesteuert werden können. Das Notkonstrukt kann die Ampeln **404** entlang der Straße anweisen, für Gegenverkehr rot zu bleiben und für das Einsatzfahrzeug grün zu bleiben, wodurch die Durchfahrt des Einsatzfahrzeugs beschleunigt wird.

[0044] Wie für die Fog-Vorrichtung **402** erläutert, ist die organische Entwicklung von IoT-Netzen für die Verbesserung oder Maximierung der Nutzbarkeit, Verfügbarkeit und Widerstandsfähigkeit von IoT-Implementationen zentral. Ferner gibt das Beispiel die Nützlichkeit von Strategien zur Verbesserung des Vertrauens und daher der Sicherheit an. Die lokale Identifikation von Vorrichtungen kann bei Implementationen wichtig sein, weil die Dezentralisierung der Identität gewährleistet, dass eine zentrale Autorität nicht ausgenutzt werden kann, um eine Nachahmung von Objekten zu ermöglichen, die innerhalb der IoT-Netze existieren können. Ferner verringert eine lokale Identifikation den Overhead und die Latenz bei der Kommunikation.

[0045] Blockchains können verwendet werden, um die Identifikation zu dezentralisieren, weil sie eine Vereinbarung zwischen Vorrichtungen in Bezug auf Namen und Identitäten, die gegenwärtig verwendet werden, bereitstellen können. So wie sie hier verwendet wird, ist eine Blockchain eine verteilte Datenbank von Identitätsdatensätzen, die aus Datenstrukturblocken besteht. Ferner kann der Begriff Blockchain, wie er hier verwendet wird, eines oder mehrere von anderen verteilten Ledger-Systemen einschließen. Andere verteilte Ledger-Ansätze umfassen Ripple, Hyperledger, Multichain, Keyless Signature Infrastructure und dergleichen. Jeder Datenstrukturblock beruht auf einer Transaktion, wobei das Erteilen eines neuen Namens an eine Vorrichtung, eine zusammengesetzte Vorrichtung oder eine virtuelle Vorrichtung ein Beispiel einer Transaktion ist.

[0046] Bei Verwendung von Blockchains zur Identifikation kann eine Nachahmung durch Beobachten der Wiedererteilung von Namen und Identitäten ohne einen entsprechenden Abschluss erkannt werden. Öffentliche Blockchains können am nützlichsten sein, weil sie es einer diversifizierten Gemeinschaft von Beobachtern ermöglichen können, eine fehlerhafte Namensgebung, eine bösartige Namensgebung oder einen Fehler einer Namensgebungsinfrastruktur zu erkennen. Demgemäß kann eine vertrauenswürdige Identitätsinfrastruktur für das Vertrauen in IoT-Netze zentral sein.

[0047] Fig. 5 ist eine Schemazeichnung **500**, welche die Bildung eines zusammengesetzten Objekts **502** anhand einer Anzahl atomarer Objekte **504**, **506** und **508** zeigt, gemäß einigen Ausführungsformen. Ein Objekt weist eine Datenmodellrepräsentation der Funktionalität, des Zustands und der Schnittstellensemantik, wodurch ein Knoten eines verteilten Systems gebildet wird, auf. Wie hier verwendet, kann ein Objekt oder IoT-Objekt eine physische Vorrichtung aus IoT-Vorrichtungen, eine virtuelle Vorrichtung aus einer Gruppe physischer oder virtueller Vorrichtungen oder eine Anzahl anderer Konfigurationen sein.

[0048] Objekte können interagieren, um eine größere Funktion, ein größeres Ziel oder einen größeren Arbeitsablauf zu erreichen. Objekte können in Bezug auf ihren Typ, beispielsweise die ausgeführte Funktion und die Instanz, beispielsweise das Vorhandensein, identifiziert werden. Mehrere Objektinstanzen können die gleiche Typidentität aufweisen, jedoch eindeutige Instanzidentitäten aufweisen. Ferner können mehrere Objektinstanzen zu Gruppen organisiert werden, wobei eine Instanz der Gruppierung eine Identität aufweisen kann. Eine Gruppe von Objekten, die in einer bestimmten Weise interagieren, können, wenn ihr Typ gegeben ist, beispielsweise ihre Funktions-, Zustands- und Schnittstellensemantik, ein zusammengesetztes Objekt repräsentieren. Die Zusammensetzung selbst kann eine Typ- und Instanzabstraktion aufweisen. Daher folgen zusammengesetzte Objekte den gleichen Identitätsregeln wie atomare Objekte. Eine Zusammensetzung mit Typ- und Instanzeigenschaften ermöglicht eine Objekterweiterbarkeit durch Zusammensetzung.

[0049] Das Objekt kann so lange halten wie eine einzige Vorrichtung in der Art eines Kühlschranks oder nur bis eine aktuelle Funktion abgeschlossen ist. Beispielsweise kann ein Kühlschrank als ein zusammengesetztes Objekt **502** angesehen werden, das aus mehreren anderen Objekten besteht, in der Art einer Lampe, eines Kompressors, eines Temperatursensors, eines Thermostats, eines Wasserspenders, eines Eisbereiters und dergleichen. Die anderen Objekte können jeweils atomare Objekte **504**, **506** und **508** sein oder selbst zusammengesetzte Objekte **502** sein. Der Eisbereiter kann ein zusammengesetztes Objekt **502** sein, das aus atomaren Objekten **504**, **506** und **508** gebildet ist, beispielsweise einem Temperatursensor, einem Thermostat, einem magnetbetätigten Wasserventil, einem Zeitgeber, einem Eiseinsatz und dergleichen. Ein Beispiel eines virtuellen zusammengesetzten Objekts **502**, das aus einer Anzahl physischer Vorrichtungen besteht, ist der Kreuzungs- und Notfall-Cluster, wie mit Bezug auf **Fig. 4** beschrieben wurde.

[0050] Dementsprechend kann die Objektidentität in Zusammenhang mit drei Abstraktionen verstanden werden, nämlich Objektinstanz, Objekttyp und Metaidentität. Eine Objektinstanz ist ein Rechelement, das begrenzte Ressourcen in der Art von Speicher, CPU, Bandbreite, Status und dergleichen belegt. Eine Objektinstanziierung hat einen Lebenszyklus, der die Erzeugung, Mutation und das Löschen einschließt. Ein Objekttyp ist ein logisches Konstrukt, das erwartetes oder mögliches Verhalten, erwartete oder mögliche Zustände und die erwartete oder mögliche Zusammensetzung deklariert. Der Objekttyp kann Randbedingungen auferlegen, wie sich Objekte verhalten und interagieren, wenn sie instanziiert wurden. Der Objekttyp kann auch den Typ von Anforderungen angeben, worauf das Objekt antworten kann, beispielsweise die Schnittstelle.

[0051] Die Metaidentität ist ein Weg zur Definition eines Metadatenzusammenhangs, in dem das Objekt existieren kann. Ein Objekt kann sich einer einkapselnden Metaidentität nicht bewusst sein. Objektinstanzen können durch Definieren einer Gruppe mit einem gewünschten Metadatenzusammenhang und anschließendes Einschreiben des Objekts in die Gruppe dynamisch Stereotypinformationen anwenden.

[0052] Die Authentifizierung und Identität sind kollationierte Gegebenheiten. Einer Objektidentität kann nicht vertraut werden, falls sie nicht authentifiziert wurde. Eine Authentifizierung ohne eine Identität hat jedoch eine begrenzte Nützlichkeit. Eine asymmetrische Schlüsselsignierung in der Art von ECDSA (Elliptic Curve Digital Signature Algorithm), RSA oder dergleichen ist für die Authentifizierung unter der Erwartung nützlich, dass die Fähigkeit zum Replizieren und Verteilen des privaten Schlüssels beschränkt ist. Die Verwendung des Schlüssels beweist, dass ein Vollmachtgeber oder Agent einen wenn auch beschränkten Zugriff auf den Schlüssel hat. Daher muss der Vollmachtgeber oder Agent authentisch sein.

[0053] Die Semantik der Authentifizierung, wenn sie auf Objektidentitäten angewendet wird, folgt auch den drei Abstraktionen der Objektinstanz, des Objekttyps und der Metaidentität. Für eine Objektinstanz legt die Authentifizierungs-Aufforderung-Antwort fest, dass die aktuelle Interaktion nur mit einer bestimmten Instanziierung des Objekts erfolgen kann. Für einen Objekttyp attestiert die Authentifizierungs-Aufforderung-Antwort, dass die aktuelle Interaktion durch die Semantik der Typidentifikation beschränkt ist. Für die Metaidentität kategorisiert die Authentifizierungs-Aufforderung-Antwort die aktuelle Interaktion entsprechend dem definierten Zusammenhang.

[0054] **Fig. 6** ist eine Schemazeichnung **600** der Bildung eines Gruppenobjekts **602** aus einer Sammlung atomarer Objekte **604** und zusammengesetzter Objekte **606**. Das Gruppenobjekt **602** gehört zu einer Objektklasse, die eine Untermenge des Objekttyps ist. Eine Objektklasse könnte beispielsweise ein Wärmetauscher sein, während ein Objekttyp der Klasse Wärmetauscher eine spezifischere Vorrichtung in der Art eines Kühlschranks, einer Wärmepumpe, einer Klimaanlage, eines Verdampfungskühlers und dergleichen sein kann.

[0055] Die Authentifizierung einer Objektklasse kann durch die Verwendung von EPID (Enhanced Privacy ID) erleichtert werden, wobei es sich um ein asymmetrisches Verschlüsselungssystem handelt, das einen einzigen

öffentlichen Schlüssel, der zu mehreren privaten Schlüsseln passt, aufweist. Eine durch einen der privaten Schlüssel erzeugte Signatur kann mit dem einzigen öffentlichen Schlüssel verifiziert werden. Demgemäß kann das Gruppenobjekt **602** einen einzigen öffentlichen Schlüssel aufweisen, während allen der atomaren Objekte **604** und der zusammengesetzten Objekte **606** jeweils eine eindeutige private Kennung erteilt wird. Das System ist nicht auf die Verwendung von EPID beschränkt, sondern kann andere Identifikationstechniken verwenden, in der Art geteilter Zugangssignaturen.

[0056] Falls einer Objektklasse eine einer EPID-Gruppenkennung (gid) entsprechende Zahl zugeordnet ist und Objektinstanzen desselben Typs der EPID-Gruppe entsprechende private Schlüssel erteilt sind, können Objektinstanzen ihre Klasse für einen Prüfer authentifizieren. Die Objektklassenauthentifizierung ist eine Attestierungsform, die es anderen ermöglicht, auf der Grundlage typisierter Regeln mit den Objekten zu interagieren. Dies ist in der Industrie auch als Typerzwangung bekannt. Die Konstruktion einer zusammengesetzten Objektklassenkennung unter Verwendung der Typkennungen ihrer Komponentenobjekte ist ein Objekttyp-Erweiterbarkeitsverfahren. Beispielsweise erzeugt eine Funktion $f()$, die als Argumente $C = (c_1, c_2, c_3, \dots, c_n)$ akzeptiert, wobei c_X die Objekttypen für die jeweiligen Komponentenobjekte sind, einen EPID-gid-Wert $C2\_id$, der die Typkennung des zusammengesetzten Objekts repräsentiert. Die Implementation von $f()$ kann die Verwendung eines kryptographischen Hashes von jedem c_x in C aufweisen. Bei einem anderen Beispiel kann $f()$ eine OID(Objektkennung)-Namensgebungshierarchie verwenden, wobei jedes c_x ein OID-Unterbaum einer übergeordneten OID für C ist. Es kann auch andere Verfahren zur Berechnung von $f()$ geben.

[0057] Erweiterbare zusammengesetzte Objektklassenkennungen ermöglichen es, dass Systeme von IoT-Objekten zu jeder beliebigen Zeit während der Lebenszeit des Vorrichtungseigentümers **608**, der die Objekte hostet, kombiniert werden. Eine Blockchain **610** kann die Entwicklung zusammengesetzter Objekte verfolgen, so dass Autorisierungswerkzeuge durch bereits existierende Zusammensetzungen informiert werden können. Eine verteilte Schemabibliothek kann unter Verwendung der Blockchain **610** gebildet werden, indem eine Transaktion **612**, welche die Objekttypkennung, beispielsweise gid, registriert, mit der zusammengesetzten Objektdefinition, beispielsweise C , versorgt wird. Aktuelle zentralisierte Objektablageschemata hängen häufig von einer einzigen logischen Vorrichtung ab, die autoritativ Klassendefinitionen auf zentralen Servern unterhält. Eine Modifikation an den zentralen Servern könnte jedoch zu nicht autorisierten Schemaänderungen führen. Im Vergleich kann die Verwendung einer Blockchain **610** gewährleisten, dass ein Schwellenkonsens über eine Anzahl von IoT-Vorrichtungen, beispielsweise in einer Fog, existiert, bevor eine existierende Objektklassendefinition geändert werden kann.

[0058] Die Blockchain **610** erleichtert die Identifikation isomorpher Objektklassifikationen. Wenn eine neue Objektklasse vorgeschlagen wird, beispielsweise in einer Nachricht **614**, kann die Blockchain **610** durchsucht werden, um herauszufinden, ob C bereits existiert.

[0059] Die Zusammensetzung eines Gruppenobjekts **602** aus Sub-Objekten, wodurch zusammengesetzte Objekte gebildet werden, ist ein Erweiterbarkeitsmechanismus für ein IoT-Objektmodell. Zusammengesetzte Objekte können unter Verwendung einer Funktion benannt werden, die sich auf die Sub-Objekte bezieht, wie „Kreuzung XYZ“. Die Sammlung von Objektinstanzen kann das Gruppenobjekt **602** bilden, wenn jedes vorgeschlagene Element der Gruppe eine Nachricht **616** sendet, um eine Nachricht **218** zu erhalten, welche den Berechtigungsnachweis aufweist, der die Sammlung identifiziert. Wenn EPID als Berechtigungsnachweismechanismus verwendet wird, kann jedes Objekt in der Sammlung mit jedem anderen Objekt oder mit anderen IoT-Vorrichtungen als Agent der Sammlung interagieren.

[0060] Die Blockchain **610** wird vom System verwendet, um die Notwendigkeit des Vertrauens vom Nameserver **620** zu entfernen. Falls ein Gruppenname wiederverwendet wird, während eine Gruppe des gleichen Namens gegenwärtig verwendet wird, kann die Blockchain **610** den Missbrauch des Nameservers **620** unterbinden. Die Wiederverwendung eines Gruppennamens kann von den IoT-Vorrichtungen festgestellt werden, welche die Blockchain **610** speichern und überwachen. Diese Feststellung kann durch die Identifikation geschehen, dass eine aktuelle Namensanforderung einen vorhergehenden Block überlappt, der aktiv ist und den Gruppennamen aufweist.

[0061] Bei einigen Aspekten ist das primäre Sammlungsgruppenmitglied (PCGM) oder Gruppenobjekt **602** dafür ausgelegt, den Gruppennamen auf der Grundlage der jeweiligen Konfiguration der Sammlung zu bestimmen. Das PCGM kommuniziert mit den Sammlungsmitgliedern, beispielsweise den zusammengesetzten Objekten **606** und den atomaren Objekten **604** oder anderen Sammlungsmitgliedern und führt die gleichen Operationen aus wie das PCGM, um zum gleichen Gruppennamen zu gelangen. Eine Funktion $F()$ kann den Sammlungsgruppennamen $C2\_id$ unter Verwendung einer festgelegten Mitglied-

schaftslogik berechnen, um Differenzen in Introspektionsreihenfolge-Nichtdeterminismus zu vermeiden, wenn verschiedene Mitglieder einen Gruppennamen getrennt berechnen.

[0062] Beispielsweise kann die EPID-Gruppenkennung (gid) einen 32-Bit- oder 128-Bit-Wert annehmen. Wenn ein 32-Bit-Wert verwendet wird, kann die Funktion $F()$ die höherwertigen 12 Bytes abschneiden. Der Nameserver **620** kann ungeachtet der gid-Länge verifizieren, ob die gid wiederverwendet wird. Kürzere gid-Längen können in beschränkten Umgebungen, beispielsweise solchen, die begrenztere IoT-Vorrichtungen verwenden, nützlich sein. Auch wenn Namenskollisionen von $F()$ selten sein können, kann eine Kollisionsauflösung durch rekursives Aufrufen von $F()$ erreicht werden, wodurch die Gruppenmitgliedschaftswerte erneut zugeführt werden (beispielsweise $F' = F(m_1, m_2, \dots, m_n, F(m_1, m_2, \dots, m_n))$).

[0063] **Fig. 7** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **700** zur Gruppenerzeugung unter Verwendung einer Sammlung von Objekten gemäß einigen Ausführungsformen. Das Verfahren **700** kann unter Verwendung des mit Bezug auf **Fig. 8** beschriebenen Systems **802** ausgeführt werden. Block **702** repräsentiert beispielsweise den Fall, in dem ein neues Gruppenobjekt erwünscht ist. Dies kann geschehen, wenn sich ein transientes Objekt in der Nähe eines aktuellen Gruppenobjekts bewegt, wie mit Bezug auf den Notfall-Cluster aus **Fig. 4** beschrieben wurde, der gebildet werden kann, wenn sich ein Einsatzfahrzeug einer Straße nähert. Bei einem anderen Beispiel kann das Einschalten einer Vorrichtung in der Art des mit Bezug auf die **Fig. 5** und **Fig. 6** beschriebenen Kühlschranks die Erzeugung eines Gruppenobjekts einleiten.

[0064] In Block **704** wird ein zusammengesetztes Objekt durch Aufrechterhalten einer Referenz zu der Kennung jedes der atomaren (A) oder zusammengesetzten (C) Sub-Objekte, welche das Gruppenobjekt bilden, in einer Liste der primären Sammlungsgruppenmitglieder (PCGM) des zusammengesetzten Objekts gebildet. Die Objekte, die das zusammengesetzte Objekt ausmachen, können durch die Objekte bestimmt werden, die zur Ausführung der Funktion erforderlich sind, wie durch einen Konsens der Objekte, durch ein vorhergehendes Programm bei einem Vorrichtungseigentümer oder durch eine Anzahl anderer Techniken, wie die Bildung eines Objekts mit einer Anzahl von IoT-Vorrichtungen, bestimmt wird.

[0065] In Block **706** wird eine Sammlungsgruppenkennung gebildet. Dies kann durch Anwenden einer Funktion auf die Liste von Objektkennungen im PCGM geschehen, wodurch das Gruppenobjekt gebildet ist. Die Funktion kann einen Hash-Code der Objektkennungen kombinieren und bilden, beispielsweise $C2\_ID = \text{SHA2}(C1, C2, C3, \dots, A1, A2, A3, \dots, A_n)$.

[0066] In Block **708** kommuniziert eines oder mehrere der Sub-Objekte (beispielsweise alle Sub-Objekte) mit einem Nameserver, beispielsweise beim Vorrichtungseigentümer, um einen Gruppenschlüssel zu erhalten. Dies kann durch die Verwendung eines EPID-Beitrittsprotokolls geschehen. Beim Beitrittsprotokoll sendet das Sub-Objekt eine Beitrittsnachricht zum Nameserver und empfängt als Antwort einen EPID-Berechtigungsnachweis beispielsweise für das $C2\_ID$ -Gruppenobjekt.

[0067] In Block **710** akzeptiert der Gruppennameserver den für die Gruppe berechneten Namen anhand der Liste im PCGM. Der Nameserver kann den Namen dann an eine Blockchain übergeben. In Block **712** erhält der Nameserver den Namen, beispielsweise $C2\_ID$, anhand der Blockchain. Hier ist die Blockchain eine verteilte Datenbank an einer Anzahl einzelner IoT-Vorrichtungen gespeicherter Transaktionen. Die Bestätigung der Gültigkeit der Transaktion kann von jeder der IoT-Vorrichtungen ausgeführt werden, wodurch mehrere Bestätigungen der Authentizität und Identität bereitgestellt werden.

[0068] In Block **714** wird festgestellt, ob der Name bereits verwendet wird, beispielsweise in einem früheren Transaktionsblock vorhanden ist, ohne dass der Name für das Objekt entsprechend verstrichen ist. Falls dies der Fall ist, kann in Block **716** ein neuer Name durch rekursives Aufrufen von $F()$ bestimmt werden, wodurch wiederum die Gruppenmitgliedschaftswerte $F' = F(m_1, m_2, \dots, m_n, F(m_1, m_2, \dots, m_n))$ zugeführt werden.

[0069] Falls der Name gegenwärtig nicht verwendet wird, wird in Block **718** festgestellt, ob die Gruppenmitgliedschaft privatheitssensitiv ist. Dies kann geschehen, falls das Vorhandensein einer IoT-Vorrichtung an einem Ort nicht öffentlich bekannt sein sollte, beispielsweise dass ein Fahrzeug an einer Reihe von Kreuzungen vorhanden ist. Falls dies der Fall ist, wirkt das PCGM in Block **720** als Proxy zur Verhandlung von Beitrittsprotokollanforderungen von Sub-Objekten. Falls dies nicht der Fall ist, findet der Nameserver in Block **722** den Sub-Objekt-Mitgliedsnamen anhand der Blockchain.

[0070] In Block **724** wird festgestellt, ob ein Antragsteller ein autorisiertes Gruppenmitglied ist. Falls dies der Fall ist, wird in Block **726** eine Beitrittsanforderung ausgeführt. In Block **728** übergibt der Nameserver den Gruppennamen, beispielsweise C2_ID, an eine Blockchain.

[0071] In Block **730** wird festgestellt, ob ein anderes Sub-Objekt existiert und demgemäß einen Gruppenberechtigungs-nachweis benötigt. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **712** für die Berechtigungs-nachweisvergabe für das Sub-Objekt zurück. Falls dies nicht der Fall ist oder falls festgestellt wird, dass ein Antragsteller kein autorisiertes Gruppenmitglied war, endet der Prozess in Block **732**.

[0072] **Fig. 8** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 800 zum Abladen von Daten vorhanden sein können. Die IoT-Vorrichtung 800 kann beliebige Kombinationen der im Beispiel dargestellten Komponenten aufweisen. Die Komponenten können als IC, Teile davon, diskrete elektronische Vorrichtungen oder andere Module, Logik, Hardware, Software, Firmware oder eine Kombination davon, die in der IoT-Vorrichtung 800 angepasst sind, oder als Komponenten, die auf andere Weise in ein Chassis eines größeren Systems aufgenommen sind, implementiert sein. Das Blockdiagramm aus **Fig. 8** soll eine Ansicht höherer Ebene der Komponenten der IoT-Vorrichtung 800 zeigen. Bei anderen Implementationen können jedoch einige der gezeigten Komponenten fortgelassen sein, es können dabei zusätzliche Komponenten vorhanden sein, und es kann eine andere Anordnung der gezeigten Komponenten auftreten.

[0073] Die IoT-Vorrichtung 800 kann einen Prozessor **802** aufweisen, der ein Mikroprozessor, ein Mehrkernprozessor, ein Multi-Thread-Prozessor, ein Prozessor mit sehr niedriger Spannung, ein eingebetteter Prozessor oder ein anderes bekanntes Verarbeitungselement sein kann. Der Prozessor **802** kann Teil eines Systems-auf-einem-Chip (SoC) sein, wobei der Prozessor **802** und andere Komponenten zu einer einzigen integrierten Schaltung oder einer einzigen Baugruppe ausgebildet sind, wie bei den Edison™- oder Galileo™-SoC-Platinen von Intel. Beispielsweise kann der Prozessor **802** einen Intel®-Architecture-Core™-basierten Prozessor in der Art eines Quark™-, eines Atom™-, eines i3-, eines i5-, eines i7- oder eines MCU-Class-Prozessors oder einen anderen derartigen von Intel® Corporation, Santa Clara, CA, erhältlichen Prozessor einschließen. Es kann jedoch eine Anzahl anderer Prozessoren, wie sie von Advanced Micro Devices, Inc. (AMD) aus Sunnyvale, CA verfügbar sind, ein MIPS-basierter Entwurf von MIPS Technologies, Inc. aus Sunnyvale, CA, ein ARM-basierter Entwurf, der von ARM Holdings, Ltd. oder einem Kunden davon oder ihren Lizenznehmern oder Verwendern lizenziert wird, verwendet werden. Die Prozessoren können Einheiten in der Art eines A5-A9-Prozessors von Apple® Inc., eines Snapdragon™-Prozessors von Qualcomm® Technologies, Inc. oder eines OMAP™-Prozessors von Texas Instruments, Inc. einschließen.

[0074] Der Prozessor **802** kann mit einem Systemspeicher **804** über einen Bus **806** kommunizieren. Eine beliebige Anzahl von Speichervorrichtungen kann verwendet werden, um einen gegebenen Umfang an Systemspeicher bereitzustellen. Der Speicher kann beispielsweise ein Direktzugriffsspeicher (RAM) gemäß einem Joint-Electron-Devices-Engineering-Council(JEDEC)-Low-Power-Double-Data-Rate(LPDDR)-basierten Entwurf in der Art des aktuellen LPDDR2-Standards gemäß JEDEC JESD 209-2E (veröffentlicht im April 2009) oder eines LPDDR-Standards der nächsten Generation in der Art von LPDDR3 oder LPDDR4 sein, wodurch Erweiterungen von LPDDR2 geboten werden, um die Bandbreite zu vergrößern. Bei verschiedenen Implementationen können die einzelnen Speichervorrichtungen beliebige von einer Anzahl verschiedener Baugruppentypen in der Art von Einzel-Die-Baugruppe (SDP), Dual-Die-Baugruppe (DDP) oder Quad-Die-Baugruppe (Q17P) sein. Diese Vorrichtungen können gemäß einigen Ausführungsformen direkt auf die Hauptplatine gelötet werden, um eine Lösung mit einem niedrigeren Profil bereitzustellen, während die Vorrichtungen gemäß anderen Ausführungsformen als ein oder mehrere Speichermodule ausgelegt werden, die wiederum durch einen gegebenen Verbinder mit der Hauptplatine koppeln. Es kann eine beliebige Anzahl anderer Speicherimplementationen verwendet werden, wie andere Typen von Speichermodulen, beispielsweise Doppelreihen-Speichermodule (DIMM) verschiedener Spielarten, einschließlich microDIMM oder MiniDIMM, jedoch ohne Einschränkung darauf. Beispielsweise kann ein Speicher eine Größe zwischen 2 GB und 16 GB aufweisen und als ein DDR3LM-Baugruppen- oder ein LPDDR2- oder LPDDR3-Speicher ausgelegt sein, der über eine Kugelgitteranordnung (BGA) auf eine Hauptplatine gelötet wird.

[0075] Um eine dauerhafte Speicherung von Informationen in der Art von Daten, Anwendungen, Betriebssystemen usw. bereitzustellen, kann auch ein Massenspeicher **808** über den Bus **806** mit der Prozessor **802** gekoppelt werden. Um einen dünneren und leichteren Systementwurf zu ermöglichen, kann der Massenspeicher **808** durch ein Halbleiterlaufwerk (SSD) implementiert werden. Andere Vorrichtungen, die für den Massenspeicher **808** verwendet werden können, umfassen Flash-Speicherkarten in der Art von SD-Karten, microSD-Karten, xD-Bildkarten und dergleichen sowie USB-Flash-Laufwerke.

[0076] Bei Implementationen mit geringer Leistungsaufnahme kann der Massenspeicher **808** ein On-Die-Speicher oder Register in Zusammenhang mit dem Prozessor **802** sein. Bei einigen Beispielen kann der Massenspeicher **808** jedoch unter Verwendung einer Mikrofestplatte (HDD) implementiert werden. Ferner kann eine beliebige Anzahl neuer Technologien zusätzlich oder an Stelle der beschriebenen Technologien für den Massenspeicher **808** verwendet werden, wie Widerstandsänderungsspeicher, Phasenänderungsspeicher, holographische Speicher oder chemische Speicher sowie andere. Beispielsweise kann die IoT-Vorrichtung **800** die 3D-XPOINT-Speicher von Intel® und Micron® aufweisen.

[0077] Die Komponenten können über den Bus **806** kommunizieren. Der Bus **806** kann eine Anzahl von Technologien aufweisen, einschließlich Industry Standard Architecture (ISA), extended ISA (EISA), Peripheral Component Interconnect (PCI), Peripheral Component Interconnect extended (PCIx), PCI Express (PCIe) oder einer Anzahl anderer Technologien. Der Bus **806** kann ein proprietärer Bus sein, der beispielsweise in einem SoC-basierten System verwendet wird. Andere Bussysteme können enthalten sein, wie eine I²C-Schnittstelle, eine I³C-Schnittstelle, eine SPI-Schnittstelle, Point-to-Point-Schnittstellen und ein Stromversorgungsbus, einschließlich anderer.

[0078] Der Bus **806** kann den Prozessor **802** mit einem Mesh-Sendeempfänger **810** zur Kommunikation mit anderen Mesh-Vorrichtungen **812** koppeln. Der Mesh-Sendeempfänger **810** kann eine beliebige Anzahl von Frequenzen und Protokollen verwenden, wie 2,4-Gigahertz(GHZ)-Übertragungen nach dem IEEE-802.15.4-Standard unter Verwendung des Bluetooth®-Low-Energy(BLE)-Standards, wie von der Bluetooth® Special Interest Group definiert, oder des ZigBee®-Standards usw. Eine beliebige Anzahl von Funkeinrichtungen, die für ein bestimmtes Drahtloskommunikationsprotokoll ausgelegt sind, kann für die Verbindungen mit den Mesh-Vorrichtungen **812** verwendet werden. Beispielsweise kann eine WLAN-Einheit zur Implementation von Wi-Fi™-Kommunikationen gemäß dem Institute-of-Electrical-and-Electronics-Engineers(IEEE)-802.11 -Standard verwendet werden. Zusätzlich können drahtlose Weitbereichskommunikationen, beispielsweise gemäß einem Mobilfunk- oder anderen drahtlosen Weitbereichsprotokoll, über eine WWAN-Einheit geschehen.

[0079] Der Mesh-Sendeempfänger **810** kann unter Verwendung mehrerer Standards oder Funkeinrichtungen für Kommunikationen mit verschiedenen Reichweiten kommunizieren. Beispielsweise kann die IoT-Vorrichtung **800** mit geographisch nahen Vorrichtungen, beispielsweise innerhalb von etwa 10 Metern, unter Verwendung eines lokalen Sendeempfängers auf der Grundlage von BLE oder einer anderen Funkeinrichtung mit geringem Stromverbrauch kommunizieren, um Strom zu sparen. Fernere Mesh-Vorrichtungen **812**, beispielsweise innerhalb von etwa 50 Metern, können über ZigBee oder andere Funkübertragungen mittlerer Leistung erreicht werden. Beide Kommunikationstechniken können über eine einzige Funkeinrichtung bei unterschiedlichen Leistungspegeln stattfinden oder über getrennte Sendeempfänger stattfinden, beispielsweise einen lokalen Sendeempfänger unter Verwendung von BLE und einen getrennten Mesh-Sendeempfänger unter Verwendung von ZigBee. Der Mesh-Sendeempfänger **810** kann als eine direkt durch den Chip in der Art der von Intel erhältlichen Curie®-Einheiten zugängliche Adresse in eine MCU aufgenommen werden.

[0080] Ein Uplink-Sendeempfänger **814** kann aufgenommen werden, um mit Vorrichtungen in der Cloud **302** zu kommunizieren. Der Uplink-Sendeempfänger **814** kann eine LPWA-Sendeempfänger sein, der unter anderem dem IEEE-802.15.4-, IEEE-802.15.4g-, IEEE-802.15.4e-, IEEE-802.15.4k- oder dem NB-IoT-Standard folgt. Die IoT-Vorrichtung **800** kann unter Verwendung von LoRaWAN™ (Long Range Wide Area Network), entwickelt von Semtech und von der LoRa-Allianz, über einen weiten Bereich kommunizieren. Die hier beschriebenen Techniken sind nicht auf diese Technologien beschränkt, sondern können mit einer beliebigen Anzahl anderer Cloud-Sendeempfänger verwendet werden, die Kommunikationen über eine große Reichweite und mit einer geringen Bandbreite implementieren, wie Sigfox und andere Technologien. Ferner können andere Kommunikationstechniken, wie Zeitschlitz-Kanalsprung, wie in der IEEE-802.15.4e-Spezifikation beschrieben ist, verwendet werden.

[0081] Eine beliebige Anzahl anderer Funkkommunikationen und Protokolle kann zusätzlich zu den für den Mesh-Sendeempfänger **810** und den Uplink-Sendeempfänger **814**, wie hier beschrieben, erwähnten Systemen verwendet werden. Beispielsweise können die Funksendeempfänger **810** und **812** einen LTE- oder anderen Mobilfunk-Sendeempfänger einschließen, der Spreizspektrums(SPA/SAS)-Kommunikationen zur Implementation schneller Kommunikationen in der Art von Videoübertragungen verwendet. Ferner kann eine beliebige Anzahl von anderen Protokollen verwendet werden, wie WiFi®-Netze für Kommunikationen mittlerer Geschwindigkeit, wie für stehende Bilder, Sensorablesungen und die Bereitstellung von Netzkommunikationen.

[0082] Die Funksendeempfänger **810** und **812** können Funkeinrichtungen aufweisen, die mit einer Anzahl von 3GPP(Third Generation Partnership Project)-Spezifikationen, unter anderem insbesondere Long Term

Evolution (LTE), Long Term Evolution-Advanced (LTE-A), Long Term Evolution-Advanced Pro (LTE-A Pro) oder Narrow Band IoT (NB-IoT) kompatibel sind. Es sei bemerkt, dass Funkeinrichtungen, die mit einer beliebigen Anzahl anderer fester, mobiler oder Satellitenkommunikationstechnologien und -Standards kompatibel sind, ausgewählt werden können. Diese können beispielsweise eine beliebige Mobilfunk-Weitbereichs-Funkkommunikationstechnologie aufweisen, die beispielsweise ein 5th-Generation(5G)-Kommunikationssystem, eine Global-System-for-Mobile-Communications-(GSM)-Funkkommunikationstechnologie, eine General-Packet-Radio-Service-(GPRS)-Funkkommunikationstechnologie oder eine Enhanced-Data-Rates-for-GSM-Evolution(EDGE)-Funkkommunikationstechnologie einschließen kann. Eine andere Third-Generation-Partnership-Project(3GPP)-Funkkommunikationstechnologie, die verwendet werden kann, umfasst UMTS (Universal Mobile Telecommunications System), FOMA (Freedom of Multimedia Access), 3GPP LTE (Long Term Evolution), 3GPP LTE Advanced (Long Term Evolution Advanced), 3GPP LTE Advanced Pro (Long Term Evolution Advanced Pro)), CDMA2000 (Code Division Multiple Access 2000), CDPD (Cellular Digital Packet Data), Mobitex, 3G (Third Generation), CSD (Circuit Switched Data), HSCSD (High-Speed Circuit-Switched Data), UMTS (3G) (Universal Mobile Telecommunications System (Third Generation)), W-CDMA (UMTS) (Wideband Code Division Multiple Access (Universal Mobile Telecommunications System)), HSPA (High-speed Packet Access), HSDPA (High-Speed Downlink Packet Access), HSUPA (High-Speed Uplink Packet Access), HSPA+ (High-speed Packet Access Plus), UMTS-TDD (Universal Mobile Telecommunications System - Time-Division Duplex), TD-CDMA (Time Division - Code Division Multiple Access), TD-SCDMA (Time Division - Synchronous Code Division Multiple Access), 3GPP Rel. 8 (Pre-4G) (3rd Generation Partnership Project Release 8 (Pre-4th Generation)), 3GPP Rel. 9 (3rd Generation Partnership Project Release 9), 3GPP Rel. 10 (3rd Generation Partnership Project Release 10), 3GPP Rel. 11 (3rd Generation Partnership Project Release 11), 3GPP Rel. 12 (3rd Generation Partnership Project Release 12), 3GPP Rel. 13 (3rd Generation Partnership Project Release 13), 3GPP Rel. 14 (3rd Generation Partnership Project Release 14), 3GPP LTE Extra, LTE Licensed-Assisted Access (LAA), UTRA (UMTS Terrestrial Radio Access), E-UTRA (Evolved UMTS Terrestrial Radio Access), LTE Advanced (4G) (Long Term Evolution Advanced (4th Generation)), cdmaOne (2G), CDMA2000 (3G) (Code Division Multiple Access 2000 (Third generation)), EV-DO (Evolution-Data Optimized or Evolution-Data Only), AMPS (1G) (Advanced Mobile Phone System (1st Generation)), TACS/ETACS (Total Access Communication System/Extended Total Access Communication System), D-AMPS (2G) (Digital AMPS (2nd Generation)), PTT (Push-to-talk), MTS (Mobile Telephone System), IMTS (Improved Mobile Telephone System), AMTS (Advanced Mobile Telephone System), OLT (Norwegisch für Offentlig Landmobil Telefoni, öffentliche Landmobiltelefonie), MTD (schwedische Abkürzung für Mobiltelefonisystem D oder Mobiltelefoniesystem D), Autotel/PALM (Public Automated Land Mobile), ARP (Finnisch für Autoradiopuhelin, „Autoradiotelefon“), NMT (Nordic Mobile Telephony), Hicap (Version hoher Kapazität von NTT (Nippon Telegraph and Telephone)), CDPD (Cellular Digital Packet Data), Mobitex, DataTAC, iDEN (Integrated Digital Enhanced Network), PDC (Personal Digital Cellular), CSD (Circuit Switched Data), PHS (Personal Handy-phone System), WiDEN (Wideband Integrated Digital Enhanced Network), iBurst, Unlicensed Mobile Access (UMA, auch als 3GPP-Generic-Access-Network- oder GAN-Standard bezeichnet), Wireless-Gigabit-Alliance(WiGig)-Standard, mmWave-Standards allgemein (drahtlose) Systeme, die bei 10 - 90 GHz und darüber arbeiten, wie WiGig, IEEE 802.11ad, IEEE 802.11ay und dergleichen. Zusätzlich zu den vorstehend angeführten Standards kann eine beliebige Anzahl von Satelliten-Uplink-Technologien für den Uplink-Sendeempfänger **814** verwendet werden, einschließlich unter anderem beispielsweise Funkeinrichtungen, die mit Standards kompatibel sind, die von der ITU (International Telecommunication Union) oder der ETSI (European Telecommunications Standards Institute) herausgegeben wurden. Die hier bereitgestellten Beispiele sind demgemäß als auf verschiedene andere Kommunikationstechnologien anwendbar zu verstehen, und zwar sowohl existierende als auch noch nicht formulierte.

[0083] Eine Netzschnittstellen-Steuereinrichtung (NIC) **816** kann aufgenommen werden, um eine drahtgebundene Kommunikation zur Cloud **302** oder zu anderen Vorrichtungen in der Art der Mesh-Vorrichtungen **812** bereitzustellen. Die drahtgebundene Kommunikation kann eine Ethernet-Verbindung bereitstellen oder auf anderen Netztypen in der Art von Controller Area Network (CAN), Local Interconnect Network (LIN), DeviceNet, ControlNet, Data Highway+, PROFIBUS oder PROFINET usw. beruhen. Eine zusätzliche NIC **816** kann aufgenommen werden, um die Verbindung mit einem zweiten Netz zu ermöglichen, beispielsweise eine NIC **816**, die Kommunikationen mit der Cloud über das Ethernet bereitstellt, und eine zweite NIC **816**, die Kommunikationen mit anderen Vorrichtungen über einen anderen Netztyp bereitstellt.

[0084] Der Bus **806** kann den Prozessor **802** mit einer Schnittstelle **818** koppeln, die für den Anschluss externer Vorrichtungen verwendet wird. Die externen Vorrichtungen können Sensoren **820** in der Art von Beschleunigungsmessern, Höhensensoren, Durchflusssensoren, Temperatursensoren, Drucksensoren, barometrischen Drucksensoren und dergleichen aufweisen. Die Schnittstelle **818** kann zur Verbindung der IoT-Vorrichtung **800** mit Betätigungselementen **822** in der Art von Leistungsschaltern, Ventilbetätigungselementen, eines Generators für hörbare Töne, einer visuellen Warnvorrichtung und dergleichen verwendet werden.

[0085] Wenngleich dies nicht dargestellt ist, können innerhalb der IoT-Vorrichtung 800 verschiedene Ein-/Ausgabe(E/A)-Vorrichtungen vorhanden sein oder damit verbunden sein. Beispielsweise kann eine Anzeige aufgenommen sein, um Informationen in der Art von Sensormesswerten und Betätigungselementpositionen zu zeigen. Eine Eingabevorrichtung in der Art eines Touchscreens oder eines Tastenfelds kann für die Entgegennahme von Eingaben aufgenommen sein.

[0086] Eine Batterie 824 kann die IoT-Vorrichtung 800 mit Strom versorgen, wenngleich sie bei Beispielen, bei denen die IoT-Vorrichtung 800 an einem festen Ort montiert ist, eine mit dem Stromnetz gekoppelte Stromversorgung aufweisen kann. Die Batterie 824 kann eine Lithiumionenbatterie, eine Metall-Luft-Batterie in der Art einer Zink-Luft-Batterie, einer Aluminium-Luft-Batterie, einer Lithium-Luft-Batterie, ein Hybrid-Superkondensator und dergleichen sein.

[0087] Ein Batterieüberwachungs-/Ladegerät 826 kann in die IoT-Vorrichtung 800 aufgenommen sein, um den Ladezustand (SoCh) der Batterie 820 zu verfolgen. Das Batterieüberwachungs-/Ladegerät 826 kann zur Überwachung anderer Parameter der Batterie 824 verwendet werden, um Fehlervorhersagen bereitzustellen, wie den Gesundheitszustand (SoH) und den Funktionstüchtigkeitszustand (SoF) der Batterie 824. Das Batterieüberwachungs-/Ladegerät 826 kann einen Batterieüberwachungs-IC in der Art eines LTC4020 oder eines LTC2990 von Linear Technologies, eines ADT7488A von ON Semiconductor aus Phoenix, Arizona oder eines IC aus der UCD90xxx-Familie von Texas Instruments aus Dallas, TX, aufweisen. Das Batterieüberwachungs-/Ladegerät 826 kann die Informationen über die Batterie 824 über den Bus 806 zum Prozessor 802 übermitteln. Das Batterieüberwachungs-/Ladegerät 826 kann auch einen Analog-Digital-Wandler (ADC) aufweisen, der es dem Prozessor 802 ermöglicht, die Spannung der Batterie 826 oder den Stromfluss von der Batterie 824 direkt zu überwachen. Die Batterieparameter können zur Bestimmung von Aktionen verwendet werden, welche die IoT-Vorrichtung 800 ausführen können, in der Art der Sendefrequenz, des Mesh-Netz-Betriebs, der Messfrequenz und dergleichen.

[0088] Ein Stromversorgungsblock 828 oder eine andere mit einem Versorgungsnetz gekoppelte Stromversorgung kann mit dem Batterieüberwachungs-/Ladegerät 826 gekoppelt werden, um die Batterie 824 zu laden. Bei einigen Beispielen kann der Stromversorgungsblock 828 durch einen drahtlosen Stromempfänger ersetzt werden, um den Strom drahtlos zu erhalten, beispielsweise durch eine Schleifenantenne in der IoT-Vorrichtung 800. Eine drahtlose Batterieladeschaltung in der Art eines LTC4020-Chips von Linear Technologies aus Milpitas, CA kann unter anderem in das Batterieüberwachungs-/Ladegerät 826 aufgenommen werden. Die spezifischen gewählten Ladeschaltungen hängen von der Größe der Batterie 824 und damit vom benötigten Strom ab. Das Laden kann unter anderem unter Verwendung des von der Airfuel Alliance vertretenen Airfuel-Standards, des vom Wireless Power Consortium vertretenen Qi-Drahtlosladestandards oder des von der Alliance for Wireless Power vertretenen Rezence-Ladestandards ausgeführt werden. Bei einigen Beispielen kann der Stromversorgungsblock 828 durch Solarzellen, einen Windgenerator, einen Wassergenerator oder andere natürliche Stromversorgungssysteme erweitert oder ersetzt werden.

[0089] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher 808 kann eine Sub-Objekt-Liste 830 atomarer Objekte und zusammengesetzter Objekte aufweisen, die zur Bildung eines Gruppenobjekts verwendet werden können. Eine Sammlungsgruppenkennung 832 kann die Sub-Objekt-Liste 830 zur Erzeugung einer Gruppenkennung verwenden, beispielsweise unter Verwendung einer auf die Sub-Objekt-Liste 830 angewendeten Hash-Formel.

[0090] Ein Nameserver 834 kann aufgenommen werden, um eine Namensunterstützung bereitzustellen und Namen an die Blockchain 836 zu übergeben. Der Nameserver 834 kann bestätigen, dass der ausgewählte Name gegenwärtig nicht verwendet wird, und Berechtigungsnachweise an Sub-Objekte ausgeben, um im Auftrag des Gruppenobjekts zu wirken.

[0091] Die Blockchain 836 weist eine Transaktionsdatenbank auf, welche Datenblöcke enthält, die Namen von Gruppenobjekten, den Sub-Objekten, die das Gruppenobjekt bilden, und dem aktuellen Zustand der Gruppenobjekte, wie gebildet, entwickelt oder aufgelöst, entsprechende Transaktionen aufweisen. Zusätzlich zu Identifikationsinformationen kann die Blockchain 836 Autorisierungsinformationen in der Art öffentlicher Verschlüsselungsschlüssel für Gruppenobjekte und Sub-Objekte aufweisen. Eine Kopie der Blockchain 836 kann auf einem Teil oder allen IoT-Vorrichtungen in einem Mesh-Netz gehalten werden. Dies ermöglicht es anderen IoT-Vorrichtungen, Änderungen in der Blockchain 836 zu bestätigen und jegliche Versuche, die Blockchain 836

ohne geeignete Autorisierung zu ändern, anzuzeigen. Wenngleich sie bei diesem Beispiel für Gruppenidentifikationstransaktionen verwendet wird, kann die Blockchain **836** für eine Anzahl anderer Transaktionen in Bezug auf Sicherheit, Bezahlungen, Transaktionen und dergleichen verwendet werden, wie hier beschrieben wird.

[0092] Ein Proxy-Broker **838** kann Sub-Objekten für ein Gruppenobjekt Berechtigungsnachweise von der Blockchain **836** bereitstellen, falls die Zusammensetzung der Gruppe als privat angesehen wird. Dies kann beispielsweise zur Erhöhung der Sicherheit von IoT-Netzen, die sich an öffentlichen Orten in der Art von Kreuzungen und Straßen befinden, verwendet werden.

[0093] Ein EPID-Server **840** kann aufgenommen werden, um Verschlüsselungsdienste in der Art des Verschlüsseln und Entschlüsseln von Daten unter Verwendung eines öffentlichen oder privaten Schlüssels bereitzustellen. Ferner kann der EPID-Server **840** öffentliche Schlüssel oder andere Berechtigungsnachweise bereitstellen, die zur Autorisierung von Sub-Objekten, im Auftrag eines Gruppenobjekts sowie als ein Schlüsselverifikationsserver zu wirken, verwendet werden können. Der EPID-Server **840** kann auch bei anderen Anwendungen zur Bildung und Ausgabe von Schlüsseln oder zur Erzeugung von Typidentitäten verwendet werden, wie mit Bezug auf die **Fig. 10** bis **Fig. 15** erörtert.

[0094] **Fig. 9** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **900**, das Code aufweist, um einen Prozessor **902** anzuweisen, Gruppenobjekte zu bilden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **900** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf den Prozessor **802** und den Bus **806** aus **Fig. 8** beschrieben ausgewählt werden. Das nichtflüchtige maschinenlesbare Medium **900** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0095] Das nichtflüchtige maschinenlesbare Medium **900** kann Code **906** aufweisen, um den Prozessor **902** anzuweisen, einen Gruppennamen aus einer Liste von Sub-Objekten zu berechnen, wie beispielsweise mit Bezug auf die **Fig. 6** und **Fig. 7** beschrieben wurde. Der Code **908** kann aufgenommen werden, um den Prozessor **902** anzuweisen, auf eine Blockchain **910** zuzugreifen, beispielsweise um festzustellen, ob ein Gruppenobjektname in der Blockchain **910** ist, und, falls dies der Fall ist, den Status des Gruppenobjekts festzustellen. Der Code **908** kann auch den Prozessor **902** anweisen, Transaktionen an die Blockchain **910** zu übergeben, sobald der Name bestätigt wurde. Der Code **908** kann auch den Prozessor **902** anweisen, Änderungen an der Blockchain **910** zu anderen Einheiten in einem IoT-Netz zu migrieren.

[0096] Das maschinenlesbare Medium **900** kann einen Code **912** aufweisen, um den Prozessor **902** anzuweisen, die Identitäten von Sub-Objekten für das Gruppenobjekt in einer Liste zu speichern. Der Code **912** kann auch den Prozessor anweisen, festzustellen, ob Anforderungen, sich der Gruppe anzuschließen, von autorisierten Sub-Objekten kommen. Falls dies der Fall ist, kann der Code **912** auch den Prozessor anweisen, den anfordernden Sub-Objekten Berechtigungsnachweise zu erteilen. Das maschinenlesbare Medium **900** kann einen Code **914** aufweisen, um den Prozessor anzuweisen, als ein Proxy-Server zu wirken, um Sub-Objekten Berechtigungsnachweise für ein privatheitsgeschütztes Gruppenobjekt bereitzustellen.

[0097] Das maschinenlesbare Medium **900** kann einen Code **916** aufweisen, um den Prozessor **902** anzuweisen, als ein Nameserver für ein Gruppenobjekt zu wirken. Das maschinenlesbare Medium **900** kann einen Code **918** aufweisen, um den Prozessor **902** anzuweisen, Berechtigungsnachweise anzufordern, um einer Gruppe beispielsweise als ein Sub-Objekt beizutreten.

[0098] **Fig. 10** ist eine Schemazeichnung **1000**, welche die Verwendung von EPID für die Objekttypidentität zeigt, gemäß einigen Ausführungsformen. Ein Vorrichtungseigentümer **1002** weist einen Typnameserver **1004** auf, der neue Typen auf der Grundlage von Einschreibanforderungen **1006** oder **1008** von zusammengesetzten Objekten **1010** oder **1012** einschreibt. Hier bedeutet das Einschreiben eines Typs oder Objekts das Registrieren des Typs oder Objekts in einer Datenbank oder Liste von Typen oder Objekten. Beispielsweise kann das Einschreiben das Senden einer Transaktion zu einer Blockchain zur Speicherung des Typs umfassen. Ein neuer Objekttyp T_{n+1} kann durch Zusammensetzung einer zusammengesetzten Gruppe **1010** oder **1012** aus Sub-Objekten **1014** bis **1018** abgeleitet werden. Die Typnamen für die Sub-Objekte **1014** bis **1018** können zur Bildung eines neuen Typnamens für die zusammengesetzten Objekte **1010** oder **1012** verwendet werden.

[0099] Das zusammengesetzte Objekt **1010** oder **1012** kann einen Typnamen durch Inspizieren der Sub-Objekte **1014** bis **1018**, mit denen es interagiert, dynamisch bestimmen. Zwei Verfahren können zur Inspektion der Konfiguration eines Sub-Objekts **1014** bis **1018** verwendet werden. Ein erstes Verfahren, das mit Bezug

auf das Leiterdiagramm aus **Fig. 12** beschrieben wird, verwendet eine Introspektion. Bei der Introspektion kann das IoT-Objekt oder eine andere Vorrichtung mit einer Ressourcenmodelldefinition sich selbst beschreiben, wenn dies gefordert wird. Beispielsweise kann das IoT-Objekt ausgelegt werden, um die Strukturen, Schnittstellen und Semantiken, die durch das Objekt implementiert wurden, bereitzustellen, wenn dies gefordert wird. Bei einigen Aspekten kann das IoT-Objekt die Strukturen, Schnittstellen und Semantiken unter Verwendung einer Datenmodellsprache (DML) in der Art eines XML-Schemas, eines JSON-Schemas und/oder YANG beschreiben. Die tatsächliche Implementation kann das Datenmodell nicht direkt interpretieren, weil dies mit einer langsamen Ausführung verbunden sein könnte. Es kann jedoch ein Testen verwendet werden, um zu zeigen, dass das durch Introspektion erzeugte DM mit dem implementierten Verhalten übereinstimmt.

[0100] Ein zweites Verfahren, das mit Bezug auf das Leiterdiagramm aus **Fig. 13** beschrieben wird, kann eine Attestierung verwenden, um die Integrität, Berechtigungsnachweise oder die Identität der Vorrichtung zu validieren. Wie hier verwendet, ist die Attestierung ein sicheres Verfahren zum Offenbaren der Vertrauenseigenschaften einer Vorrichtung oder Plattform, wobei die Vorrichtung oder Plattform selbst die Vertrauenseigenschaften mitteilt. Die Vertrauenseigenschaften können den Plattformhersteller, durch den Verkäufer erreichte Zertifizierungen, welche die Sicherheitshärtung widerspiegeln, wie FIPS140-2 für eine Kryptomodulimplementation, aufweisen. Ferner können ISO9000 sowie Verkäuferprozesse, denen gefolgt wird, um die Qualität zu sichern, relevant sein. Die Attestierung offenbart typischerweise die Hardware-, Firmware- und Softwareversionen und Patch-Niveaus. Sie kann Informationen über Schlüssel, die durch die gehärtete Umgebung in der Art einer Trusted Execute Environment (TEE) geschützt werden, und Informationen über Schlüsseltypen offenbaren. Beispielsweise kann eine gehärtete Umgebung ein Trusted Platform Module (TPM) verwenden, um Schlüsseltypen zu definieren, wobei ein Schlüsseltyp nicht aus dem TPM zu einem weniger gehärteten kryptographischen Modul migriert werden kann.

[0101] Sowohl die Introspektion als auch die Attestierung können einen Satz erzeugen, der eine Objektmitgliedschaft auf der Grundlage einer Sub-Objekt-Typisierung enthält. Ferner können die beiden Verfahren zusammen verwendet werden, beispielsweise unter Verwendung des Attestierungsschlüssels, um zu bestätigen, dass die Introspektionsidentität von einer bestimmten Einheit gekommen ist. Alle IoT-Objekte sind typisiert, auch wenn nicht alle einen Berechtigungsnachweis, der den Typ authentifiziert, aufweisen können.

[0102] Ein Verfahren zur Ableitung neuer Objekttypnamen, wie beispielsweise mit Bezug auf das Leiterdiagramm aus **Fig. 12** beschrieben, unterstützt die automatisierte Erzeugung von Objekttypen. Die automatische Erzeugung ermöglicht nützliche Sammlungen von Objekten zur Bildung eines Musters zur Objektreplikation. Eine nützliche Sammlung kann dann leichter anderswo im Netz unter Verwendung des Typnamens und des Musters als Eingabeparameter instanziiert werden.

[0103] Der Typnameserver **1004** kann EPID verwenden, um die Objekttypkennung durch Einlassen jeder Instanz des gleichen Objekttyps in eine EPID-Gruppe unter Verwendung des Typnamens als Gruppenkennung zu authentifizieren. Eine Blockchain **1022** kann verwendet werden, um die Erzeugung dynamisch abgeleiteter Typen, beispielsweise durch Übergeben einer Transaktion **1024** vom Typnameserver **1004**, aufzuzeichnen, so dass Objekte des gleichen Typs ohne isomorphe Redundanz zuverlässig instanziiert werden können.

[0104] **Fig. 11** ist ein Leiterdiagramm eines beispielhaften Verfahrens **1100** zur dynamischen Erzeugung eines Objekttyps gemäß einigen Ausführungsformen. Das Verfahren **1100** aus **Fig. 11** kann durch die mit Bezug auf **Fig. 14** beschriebene IoT-Vorrichtung 1400 implementiert werden. In Schritt **1104** kann ein zusammengesetztes Objekt **1102** eine Anforderung **1104** zur Erzeugung einer Typgruppe, beispielsweise T1, zu einem Typnameserver **1106** senden. Ein Typnameserver **1106** kann in einen Vorrichtungseigentümer **1002** aufgenommen werden, wie mit Bezug auf **Fig. 10** beschrieben, oder er kann sich in einer zentralen oder getrennten Vorrichtung in der Art eines Aggregators **406** befinden, wie mit Bezug auf **Fig. 4** beschrieben.

[0105] In Schritt **1108** antwortet der Typnameserver **1106** auf das zusammengesetzte Objekt **1102** mit einer Anforderung einer Typintrospektion. In Schritt **1110** löst die Anforderung ein rekursives Senden von Anforderungen an Sub-Objekte **1112**, Introspektions- oder Attestierungsinformationen bereitzustellen, aus. Die Sub-Objekte **1112** antworten **1114** mit den angeforderten Informationen. Sobald die Rekursion abgeschlossen ist, kann das zusammengesetzte Objekt **1102** einen Typnamen **1116** anhand der Typen aller Sub-Objekte **1112**, beispielsweise als $T1 = F(t1, t2, t3, \dots, tn)$, berechnen. Das zusammengesetzte Objekt **1102** kann dann den Typ unter Verwendung des Objektinstanzschlüssels in einer EPID-Beitrittsanforderung **1118** an den Typnameserver **1106** attestieren.

[0106] Der Typnameserver **1106** sendet eine Anforderung **1120** einer vorhergehenden Instanziierung des Erzeugungsereignisses an den Administrator der Blockchain **1122**. Eine Nachricht **1124**, dass der Typ bereits existiert, kann vom Administrator der Blockchain **1122** empfangen werden. Dies kann auch durch eine Feststellung des Typnameservers **1106** geschehen, dass ein vorhergehender Typ bereits mit diesem Namen erzeugt wurde und in der Blockchain **1122** vorhanden ist.

[0107] Falls der Typ nicht erzeugt wurde, erteilt der Typnameserver **1126** eine Anforderung **1128** zur Erzeugung des Typs in der Blockchain **1122**. Dies kann durch Übergeben der Erzeugungstransaktion an die Instanziierung der Blockchain **1122** geschehen, welche sich in der IoT-Vorrichtung befindet, die den Typnameserver **1106** hostet.

[0108] Bei einigen Beispielen kann es anderen IoT-Vorrichtungen, welche die Blockchain **1122** speichern, nicht gelingen, die neue Erzeugung zu validieren, beispielsweise eine andere Instanziierung des Typs in der Blockchain **1122**, die bei ihnen gespeichert ist, zu lokalisieren. Falls bei einer Mehrzahl die Validierung der Erzeugung fehlschlägt, wird sie abgelehnt, und die Blockchain **1122** kehrt zur vorhergehenden Chain zurück. Der Typnameserver **1106** kann dann den Typ umbenennen **1126** und die Erzeugung erneut versuchen **1128**.

[0109] Falls die Erzeugung erfolgreich ist, wie beispielsweise durch eine vom Administrator der Blockchain **1122** empfangene Nachricht **1130** oder durch eine Bestätigung der neuen Blockchain **1122** durch eine Mehrzahl der IoT-Vorrichtungen angegeben, kann der Typnameserver **1106** dann eine EPID-Beitrittsanforderung **1132** an das zusammengesetzte Objekt **1102** ausgeben. Die EPID-Beitrittsanforderung **1132** weist die EPID-Berechtigungsnachweise für den Typ auf. Diese können direkt durch das zusammengesetzte Objekt **1102** mit den Sub-Objekten **1112** geteilt werden, oder die Sub-Objekte **1112** können eine Beitrittsanforderung mit dem neuen Typnamen senden, damit der Typnameserver **1106** die Berechtigungsnachweise bereitstellt.

[0110] Fig. 12 ist ein Leiterdiagramm eines beispielhaften Verfahrens **1200** zur Typintrospektion unter Verwendung einer Rekursion gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 11 erörtert wurden. Das Verfahren **1200** aus Fig. 12 kann durch die mit Bezug auf Fig. 14 beschriebene IoT-Vorrichtung **1400** implementiert werden. Die Introspektion stellt eine Verbindungsgraphabstammung von einem zusammengesetzten Objekt zu Blatt-Objekten bereit. Blatt-Objekte sind ansonsten als atomare Objekte bekannt, weil sie keine Sub-Objekte aufweisen.

[0111] In Schritt **1202** kann das zusammengesetzte Objekt **1102** ein Befehl **1202** zu einem Sub-Objekt **1204** senden, um das Sub-Objekt **1204** anzuweisen, eine Introspektion auszuführen. Falls das Sub-Objekt **1204** ein atomares Objekt ist, gibt es eine Signatur als Identifikation des Typs zurück. Falls das Sub-Objekt **1204** selbst ein zusammengesetztes Objekt ist, sendet es einen Befehl **1206** zu jedem das Sub-Objekt **1204** bildenden Sub-Sub-Objekt **1208**, eine Introspektion auszuführen. Dies geschieht rekursiv vom zusammengesetzten Objekt **1102** zu jedem Sub-Objekt **1204** und von jedem Sub-Objekt **1204** zu jedem Sub-Sub-Objekt **1208**, wie durch einen Befehl **1210**, der zu einer niedrigeren Schicht gesendet wird, und einen Typgraphen **1212** oder **1214**, der von der niedrigeren Schicht zurückgegeben wird, angegeben wird.

[0112] Die Introspektion verwendet die Rekursion als Verfahren zum Durchlaufen eines Sub-Objekt-Graphen. Die Rekursion hält, wenn eine von zwei möglichen Bedingungen gegeben ist, erstens, falls ein atomares Objekt angetroffen wird, und zweitens, falls ein bereits angetroffenes Objekt wieder angetroffen wird. Das rekursive Durchlaufen des Sub-Objekt-Graphen erzeugt einen Baum (gerichteter azyklischer Graph), der aus wenigstens und höchstens einem Weg zum Erreichen jedes Knotens im Graphen besteht. Der Typgraph kann das Format $G = (gn), [G]K_{n\_Instanz}$, aufweisen, wobei gn die Gruppennummer ist, $[G]$ der Gruppenname ist und $K_{n\_Instanz}$ der Schlüssel für die spezifische Gruppe ist. Nach der Rückkehr vom rekursiven Durchlaufen des Baums belegt der aktuelle Knoten einen Eintrag in einem Manifest, wodurch eine Baumstruktur gebildet ist, welche die Typinformationen des Objekts enthält. Falls das Objekt einen Instanzschlüssel besitzt, werden die Typinformationen signiert. Demgemäß kann das Sub-Sub-Objekt **1208** einen Typgraphen des Formats $G' = (gn+1 | gn), [G']K_{n\_Instanz}$ an das Sub-Objekt **1204** zurückgeben. Sobald alle Sub-Sub-Objekte **1208** ihre Typen oder Typgraphen an das Sub-Objekt **1204** zurückgegeben haben, kann es seinen eigenen Typgraphen **1216** an das zusammengesetzte Objekt **1102** zurückgeben, beispielsweise $G'' = (gn+2 | gn+1 | gn), [G'']K_{n\_Instanz}$.

[0113] Das resultierende Manifest wird vom zusammengesetzten Objekt **1102** als Root-Objekt zur Erzeugung **1218** seines eigenen Typnamens verwendet. Dies kann auch einen Eigenschaftsnamen mit einer lokalen Reichweite als Eingabe in die zur Erzeugung eines Typnamens verwendete Funktion $F()$ einschließen. Das Manifest kann einem Typnameserver **1106** zugeführt werden, wie mit Bezug auf Fig. 11 erörtert, welcher die Signaturen und den Aufbau des Typnamens verifizieren kann. Der Typnameserver **1106** kann auf eine vorher-

gehende Typnamensreservierung in einer Blockchain prüfen. Falls ein ursprünglicher Typname gefunden wird und ein Berechtigungsnachweis erteilt wird, kann eine Blockchain aktualisiert werden, wodurch eine unabhängige Überprüfung des Typnamens-Reservierungsstatus ermöglicht wird.

[0114] Fig. 13 ist ein Leiterdiagramm eines beispielhaften Verfahrens 1300 zur rekursiven Typattestierung gemäß einigen Ausführungsformen. Das Verfahren 1300 aus Fig. 13 kann durch die mit Bezug auf Fig. 14 beschriebene IoT-Vorrichtung 1400 implementiert werden. Eine rekursive Objektattestierung ähnelt der rekursiven Objektintrospektion mit dem Unterschied, dass Typinformationen unter Verwendung eines Typnamens-Berechtigungsnachweises signiert werden können, der beispielsweise in die Vorrichtung programmiert ist oder anhand eines in die Vorrichtung programmierten Berechtigungsnachweises gebildet wird. Wenn der Objekttyp-Berechtigungsnachweis verwendet wird, kann der Typname einen zuvor eingetragenen Typ identifizieren, weshalb eine Blockchain eine historische Aufzeichnung seiner Typhierarchie enthalten kann. Dementsprechend kann die Verwendung eines Typberechtigungsachweises die Rekursion anhalten. Gemäß einer Ausführungsform der rekursiven Objektattestierung kann die authentifizierte Typbeendigung als Verfahren zur Verifizierung einer Typhierarchie ignoriert werden.

[0115] In Schritt 1302 sendet das zusammengesetzte Objekt 1102 einen Befehl 1302 zu einem Sub-Objekt 1204, um das Sub-Objekt 1204 anzuweisen, einen Attestierungsberechtigungsnachweis zu senden. Falls das Sub-Objekt 1204 ein atomares Objekt ist, gibt es einen Objektberechtigungsachweis als Typidentifikation zurück. Falls das Sub-Objekt 1204 selbst ein zusammengesetztes Objekt ist, sendet es einen Befehl 1304 zu jedem das Sub-Objekt 1204 bildenden Sub-Sub-Objekt 1208, einen Attestierungsberechtigungsnachweis zu senden. Dies geschieht rekursiv vom zusammengesetzten Objekt 1102 zu jedem Sub-Objekt 1204 und von jedem Sub-Objekt 1204 zu jedem Sub-Sub-Objekt 1208, wie durch einen Befehl 1306, der zu einer niedrigeren Schicht gesendet wird, und einen Typgraphen 1308, der von der niedrigeren Schicht zurückgegeben wird, angegeben wird. Ein ähnlicher Typgraph 1310 wird von jedem Sub-Sub-Objekt 1208 zum Sub-Objekt 1204 zurückgegeben. Ein Typgraph 1312 kann dann von jedem Sub-Objekt 1204 zum zusammengesetzten Objekt 1102 zurückgegeben werden. Was die Introspektion betrifft, kann das zusammengesetzte Objekt 1102 dann jede Signatur verifizieren 1314.

[0116] Fig. 14 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 1400 zur Zuweisung von Typen zu zusammengesetzten Objekten, wenn sie gebildet werden, vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass verschiedene Komponenten ausgewählt und für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und die mit Bezug auf Fig. 14 erörterte IoT-Vorrichtung 1400 verwendet werden können.

[0117] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Typerzeugungsfunktionen aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher 808 kann einen Typnameserver 1402 aufweisen, der atomare Objekttypen und zusammengesetzte Objekttypen, die zur Bildung eines Gruppenobjekts verwendet werden können, auflistet. Der Typnameserver 1402 kann einen Befehl, die Typen von Sub-Objekten und ein zusammengesetztes Objekt bildender Sub-Sub-Objekte zu bestimmen, an einen Typinspektor 1404 ausgeben. Der Typinspektor 1404 kann eine rekursive Inspektion von Mesh-Vorrichtungen 812 unter Verwendung einer Introspektion oder Attestierung ausführen. Ein Typgraphgenerator 1406 kann einen Typgraphen unter Verwendung der Antworten von den Sub-Objekten und Sub-Sub-Objekten, einschließlich von Objekten niedrigerer Ebene erzeugter Typgraphen, erzeugen. Ein Typnamensberechner 1408 kann zur Erzeugung eines Typnamens anhand des erzeugten Typgraphen verwendet werden, beispielsweise durch Berechnen einer Hash-Funktion der Einträge im Typgraphen. Typberechtigungsachweise 1410 können aufgenommen werden, um den Typ der IoT-Vorrichtung 1400 zu identifizieren. Die Typberechtigungsachweise 1410 können vom Hersteller beispielsweise zur Attestierung in die Vorrichtung programmierte Berechtigungsnachweise oder beispielsweise zur Attestierung von einer anderen Vorrichtung der IoT-Vorrichtung 1400 bereitgestellte Berechtigungsnachweise aufweisen. Ein kombinierter Typberechtigungsachweis kann unter Verwendung eines in die Vorrichtung eingebauten Berechtigungsnachweises zur Validierung oder Verschlüsselung eines der Vorrichtung bereitgestellten Berechtigungsnachweises erzeugt werden.

[0118] Eine Blockchain 836 kann in die IoT-Vorrichtung 1400 aufgenommen sein, um Typnamenstransaktionen zusätzlich zu anderen Informationen in der Art von Gruppennamenstransaktionen aufzuzeichnen. Wie hier

beschrieben, können die Transaktionen der Blockchain **836** durch ein Mehrheitsvotum der Mesh-Vorrichtungen **812**, die auch Kopien der Blockchain **836** speichern, validiert werden.

[0119] **Fig. 15** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **1500**, das Code aufweist, um einen Prozessor **902** anzuweisen, Gruppenobjekte zu bilden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **1500** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf den Prozessor **802** und den Bus **806** aus **Fig. 8** beschrieben ausgewählt werden. Das nichtflüchtige maschinenlesbare Medium **1500** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0120] Das nichtflüchtige maschinenlesbare Medium **1500** kann Code **1502** aufweisen, um den Prozessor **902** anzuweisen, eine rekursive Typintrospektion zur Bestimmung der Typen von Vorrichtungen in einem zusammengesetzten Objekt auszuführen. Ein Code **1504** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine sequenzielle Attestierung zur Bestimmung der Typen von Vorrichtungen in einem zusammengesetzten Objekt auszuführen. Ein Code **1506** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Typgraphen mit von Sub-Objekten und Sub-Sub-Objekten zurückgegebenen Informationen zu bilden. Ein Code **1508** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Typnamen für den Typ zu berechnen, beispielsweise durch Berechnen einer Hash-Funktion anhand des Typgraphen. Ein Code **1510** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob der Typname bereits in der Blockchain ist, und falls dies nicht der Fall ist, den Typnamen an die Blockchain zu übergeben. Der Code **910** kann den Prozessor **902** auch anweisen, Änderungen zu der in anderen Vorrichtungen im Mesh-Netz gespeicherten Blockchain zu migrieren. Ein Code **1512** kann aufgenommen sein, um den Prozessor **902** anzuweisen, EPID-Beitrittsanforderungen zu die Typgruppe erzeugenden Sub-Objekten auszusenden. Falls der Name nicht erzeugt wird, beispielsweise infolge einer Redundanz oder eines anderen Fehlers in den Blockchain-Datensätzen, kann ein Code **1514** aufgenommen werden, um den Prozessor **902** anzuweisen, den Typnamen zu regenerieren und den Übergabeprozess zu wiederholen.

[0121] **Fig. 16** ist eine Schemazeichnung der Bildung einer Koalitionsgruppe **1600** gemäß einigen Ausführungsformen. IoT-Netze können eine lose Koalition von Objekten, die nicht regelmäßig interagieren können, welche als Koalitionsgruppe **1600** bezeichnet wird, bilden. Eine Bezeichnung der Objekte als Teil einer Gruppenabstraktion kann jedoch einen semantischen Wert bereitstellen. Koalitionsgruppen **1600** können durch administrative Entscheidungen gebildet werden, beispielsweise zur Angabe eines Gebiets, eines Orts oder für allgemeine Zwecke, beispielsweise für Vorrichtungen, die sich in einem Stockwerk oder in einer Wohnung in einem Einzelgebäude befinden. Eine Verwaltungsautorität in der Art eines Vorrichtungseigentümers **1602** kann die Gruppenkennung, welche die gruppierten Vorrichtungen verwenden, beispielsweise durch einen Koalitionsgruppen-Nameserver **1604** wählen. Koalitionsgruppenmitglieder **1606** können durch Aussenden einer Beitrittsanforderung **1608** zum Vorrichtungseigentümer **1602** einer Koalitionsgruppe **1600** beitreten. Anhand des Koalitionsgruppen-Nameservers **1604** können den Gruppenmitgliedern Berechtigungsnachweise **1610** bereitgestellt werden, einschließlich EPID-Berechtigungsnachweise. Die Berechtigungsnachweise **1610** können ferner von den Koalitionsgruppenmitgliedern **1606**, beispielsweise durch Intra-Objekt-Schnittstellen **1614**, Sub-Objekten **1612** bereitgestellt werden. Auf den Koalitionsgruppennamen kann von einer Blockchain **1616** zugegriffen werden, oder er kann nach der Erzeugung an die Blockchain **1616** übergeben werden.

[0122] Ein Berechtigungsnachweis für eine Koalitionsgruppe **1600** ermöglicht es dem Koalitionsgruppenmitglied **1606**, zu authentifizieren, ohne einen Wert zu offenbaren, der zur Verfolgung der Privatheit verwendet werden kann. Daher können Kriterien für die Mitgliedschaft esoterisch sein, wobei die Größe der Gruppe verwendet wird, um den Grad des Privatheitsrisikos in Zusammenhang mit der Verwendung des Berechtigungsnachweises zu bestimmen.

[0123] Die Einschreibung in eine Koalitionsgruppe **1600** durch eine IoT-Vorrichtung oder ein Objekt ermöglicht es dem Objekt, Eigenschaften und Attribute der Koalitionsgruppe **1600** zu erben. Diese Eigenschaften und Attribute für die Koalitionsgruppenmitglieder **1606** können einen Code, einen Zustand oder Schnittstellen, die Gruppeneigenschaften und -attribute verarbeiten, nicht aufweisen. Nichtsdestoweniger können andere Entitäten Eigenschaften und Attribute zum Sortieren, Kategorisieren, Weiterleiten, Verwalten oder Ausführen einer Analyse benennen. In diesem Sinne ist die Koalitionsgruppierung eine Strategie für die dynamische Anwendung von Objektmetadaten.

[0124] **Fig. 17** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **1700** zum Einschreiben von Mitgliedern in eine Koalitionsgruppe gemäß einigen Ausführungsformen. Das Verfahren **1700** aus **Fig. 17** kann

durch die mit Bezug auf **Fig. 18** beschriebene IoT-Vorrichtung **1800** implementiert werden. Block **1702** repräsentiert beispielsweise, wenn eine Gruppe von IoT-Vorrichtungen durch Stromzufuhr eingeschaltet oder auf andere Weise aktiviert wird, beispielsweise wenn eine virtuelle Vorrichtung gestartet wird. In Block **1704** definiert der Netzdomain-Eigentümer Gruppen (G1, G2, ... , Gn). Eine Gruppe kann eine Lokalisierungsbezeichnung in der Art von im oberen Stock, im unteren Stock und dergleichen oder eine funktionelle Bezeichnung in der Art von Verwaltung, Klimaregelung und dergleichen aufweisen und Kombinationen von Lokalität und Funktion aufweisen, wie Evakuierungs-, Eingangswege an einem Stadion und dergleichen. Eine beliebige Anzahl anderer Bezeichnungen kann verwendet werden. Im Allgemeinen wird der Koalitionsgruppenname ausgewählt, um einem System nützliche Metadaten bereitzustellen.

[0125] In Block **1706** wird festgestellt, ob eine Gruppe, beispielsweise G1, entdeckbar ist. Falls dies nicht der Fall ist, wird die Gruppe in Block **1708** für eine Blockchain öffentlich gemacht. In Block **1710** kann eine Anforderung von einem Objekt, beispielsweise O1, der Gruppe **G1** beizutreten, empfangen werden. In Block **1712** können EPID-Beitrittsparameter vom Objekt **O1** empfangen werden. Diese können ansprechend auf eine Anforderung vom Gruppenvorrichtungseigentümer gesendet werden.

[0126] In Block **1714** verifiziert ein Koalitionsgruppen-Nameserver die Beitrittsanforderung von O1. Die Anforderung kann unter Verwendung einer Vielzahl von Berechtigungsnachweisen oder Techniken authentifiziert werden. Beispielsweise kann der Koalitionsgruppen-Nameserver die Instanz-, Autoritäts- oder Typnamens-Berechtigungsnachweise prüfen, um festzustellen, ob die Werte in der Blockchain sind. Bei Anwendungen mit höherer Sicherheit kann gefordert werden, dass alle Berechtigungsnachweise korrekt sind, bevor es erlaubt wird, dass die Vorrichtung der Koalitionsgruppe beitrifft. Ähnlich kann der Koalitionsgruppen-Nameserver bei Anwendungen mit geringerer Sicherheit keine Berechtigungsnachweise für das Einschreiben einer Vorrichtung in eine Koalitionsgruppe fordern. Falls die Anforderung in Block **1716** als gültig bestimmt wird, kann in Block **1718** ein Koalitionsgruppen-Berechtigungsnachweis in der Art von EPID dem Objekt O1 erteilt werden. Falls die Anforderung nicht als gültig bestimmt wird, endet der Prozess in Block **1720** ohne die Erteilung der Berechtigungsnachweise.

[0127] **Fig. 18** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **1800** zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass verschiedene Komponenten ausgewählt und für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und die mit Bezug auf **Fig. 18** erörterte IoT-Vorrichtung **1800** verwendet werden können.

[0128] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der Erzeugung von Koalitionsgruppen, wie hier beschrieben, aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Koalitionsgruppen-Nameserver **1802** aufweisen, der zweckmäßige Gruppierungen für Objekte aufweist. Wie hier erörtert, können die Gruppen auf der Grundlage des Orts, der Funktionalität oder einer Kombination davon gebildet werden. Ein Benutzer kann die für die Gruppierung zu verwendenden Parameter definieren. Der Typnameserver **1802** kann eine Koalitionsgruppen-Mitgliedsliste **1804** bilden und unterhalten, um einen Namen für die Koalitionsgruppe zu erzeugen. Falls die Gruppe nicht entdeckbar ist, kann ein Veröffentlichender **1806** die Eigenschaften der Gruppe, einschließlich Typen, Orten und anderer Metadaten, anderen IoT-Vorrichtungen verfügbar machen, so dass diese IoT-Vorrichtungen feststellen können, ob sie der Gruppe beitreten sollten. Dies kann beispielsweise geschehen, indem der Gruppenname und die Gruppenzusammensetzung einer Blockchain **836** öffentlich gemacht werden.

[0129] Die Blockchain **836** kann in die IoT-Vorrichtung **1800** aufgenommen sein, um Koalitionsgruppen-Namenstransaktionen zusätzlich zu anderen Informationen in der Art von Typnamenstransaktionen und Transaktionen zusammengesetzter Objekte aufzuzeichnen. Wie hier beschrieben, können die Transaktionen der Blockchain **836** durch ein Mehrheitsvotum der Mesh-Vorrichtungen **812**, die auch Kopien der Blockchain **836** speichern, validiert werden.

[0130] Ein Berechtigungsnachweisprüfer **1808** kann aufgenommen sein, um Berechtigungsnachweise von IoT-Vorrichtungen und zusammengesetzten Objekten, die der Koalition beitreten möchten, zu empfangen. Der Berechtigungsnachweisprüfer **1808** kann in Bezug auf Transaktionen in der Blockchain **836** geprüft werden, um festzustellen, ob die Berechtigungsnachweise gültig sind. Falls dies der Fall ist, kann der Berechtigungsnachweisprüfer **1808** Berechtigungsnachweise vom EPID-Server **840** erhalten und sie der IoT-Vorrichtung oder dem zusammengesetzten Objekt, das die Beitrittsanforderung gesendet hat, erteilen. Der Berechtigungs-

nachweisprüfer **1808** kann die Transaktion dann an die Blockchain **836** übergeben, um aufzuzeichnen, dass die IoT-Vorrichtung oder das zusammengesetzte Objekt der Koalitionsgruppe beigetreten ist.

[0131] Fig. 19 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **1900**, das Code aufweist, um einen Prozessor **902** anzuweisen, Koalitionsgruppen zu erzeugen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **1900** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf den Prozessor **802** und den Bus **806** aus Fig. 8 beschrieben ausgewählt werden. Das nichtflüchtige maschinenlesbare Medium **1900** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0132] Das nichtflüchtige maschinenlesbare Medium **1900** kann Code **1902** aufweisen, um den Prozessor **902** anzuweisen, Koalitionsgruppen zu definieren, beispielsweise durch Orte, die Funktion oder beide. Ein Code **1904** kann aufgenommen werden, um den Prozessor **902** anzuweisen, es festzustellen, ob eine Koalitionsgruppe entdeckbar ist, beispielsweise eingerichtet werden, um auf eine Entdeckungsanforderung mit die Koalitionsgruppe identifizierenden Metadaten zu antworten. Ein Code **1906** kann aufgenommen werden, um den Prozessor **902** anzuweisen, die Koalitionsgruppe einer Blockchain oder direkt den umgebenden Vorrichtungen bekannt zu machen. Dies kann das Vorhandensein der Koalitionsgruppe bekannt machen, entdeckbar machen oder beides.

[0133] Ein Code **1908** kann aufgenommen werden, um den Prozessor **902** anzuweisen, eine Beitrittsanforderung für die Koalitionsgruppe von IoT-Vorrichtungen, einschließlich atomarer Objekte, zusammengesetzter Objekte oder beider, zu akzeptieren. Die Beitrittsanforderung kann die Koalitionsgruppe identifizieren und Verifikationsinformationen aufweisen, wie Orts-, Typ- und andere Berechtigungsnachweise oder Metadaten. Ein Code **1910** kann aufgenommen werden, um den Prozessor **902** anzuweisen, die Berechtigungsnachweise zu validieren, beispielsweise festzustellen, ob sie in der Blockchain vorhanden sind. Ein Code **1912** kann aufgenommen werden, um dem Anfordernden Berechtigungsnachweise in der Art eines EPID-Schlüssels zu erteilen.

[0134] Kommunikationen zwischen IoT-Vorrichtungen, beispielsweise in einer Koalitionsgruppe, einem Mesh-Netz, einer Fog-Vorrichtung oder einer anderen Anordnung können gesichert werden müssen, dies kann jedoch für Vorrichtungen problematisch sein, die von begrenzter Funktionalität sind. Ferner können die IoT-Vorrichtungen über verschiedene Netze verteilt sein, wodurch das Sichern der Kommunikationen anspruchsvoller wird. Ein verteiltes Ledger-System kann die Sicherheit bei Kommunikationen durch IoT-Vorrichtungen erhöhen.

[0135] Fig. 20 ist ein schematisches Diagramm der Ermöglichung von Kommunikationen zwischen Vorrichtungen unter Verwendung einer semi-permissionierten verteilten Ledger-Transaktion **2000** gemäß einigen Ausführungsformen. Wie hier verwendet, benutzt ein semi-permissioniertes Verteilter-Ledger-System Enhanced Privacy-Id(EPID)-Schlüssel zur Eintragung von Transaktionsschlüsseln in den Ledger. Eine als Verteilter-Ledger-Spezifikationsstelle bzw. Distributed Ledger Enumeration Authority (DLEA) **2002** bezeichnete Namespacestelle weist einer Instanz eines Ledgers eine eindeutige Zahl zu. Die DLEA **2002** kann durch die Internet Assigned Numbers Authority (IANA), eine öffentliche Behörde, eine private Entität oder eine Entität betrieben werden, die einen Zahlenraum durch Unternehmen von Schritten, die Wiederverwendung in Gebrauch befindlicher Zahlen zu verhindern, verwaltet.

[0136] Es sei bemerkt, dass der von der DLEA **2002** zur Zuweisung von Namen/Zahlen verwendete Algorithmus verteilt werden kann, weil der Zahlenraum in Bezug auf die zugewiesenen in Gebrauch befindlichen Zahlen spärlich ist. Demgemäß ist die Möglichkeit von Kollisionen gering. Daher ist es möglich, dass mehrere Instanzen der DLEA **2002** unabhängig arbeiten könnten. Dementsprechend kann die DLEA **2002** über geographische Grenzen hinweg gehostet werden, wobei keine Notwendigkeit einer zentralen steuernden Stelle in der Art einer Regierung oder der UN oder einer einzigen großen privaten Organisation besteht. Ferner kann die Unabhängigkeit verteilter Blockchains nicht durch eine zentralisierte Namensgebungsstelle beeinträchtigt werden.

[0137] Die Betriebsintegrität der DLEA **2002** kann unter Verwendung eines öffentlichen verteilten Ledger-Systems, das in Gebrauch befindliche DLEA-Zahlen veröffentlicht, überprüft werden. Diesem Ledger DLS-0 **2004** wird der Wert null „0“ zugewiesen und darf von der DLEA **2002** nicht zugewiesen werden. Das geeignete Verhalten der DLEA-Zahlenzuweisung kann durch Implementieren des Zahlenraum-Zuordnungsalgorithmus unter anderem in einer Trusted Execution Environment (TEE) in der Art einer Intel SGX Enclave, einer ARM TrustZone oder eines Hardware Security Module (HSM) verstärkt werden. In diesen Umgebungen kann der Zahlenzuweisungsalgorithmus durch die globale Gemeinschaft von Experten bestätigt werden. Demgemäß

kann der DLEA **2002** in sehr hohem Maße vertraut werden, die einfache Funktion des Vermeidens der Neu-zuweisung einer bereits zugewiesenen Zahl auszuführen.

[0138] Ein Teilnehmer, beispielsweise P1 **2006**, kann eine Anforderung einer identifizierenden Zahl für Kommunikationstransaktionen, beispielsweise DLS-X#, zur DLEA **2002** senden. Die Anforderung kann die Form [Anforderung DLS-X#, K_{TxRoot}] K_{TxRoot} annehmen, wobei die Informationen in den Klammern die Nachricht sind und die Zahl außerhalb der Klammern der öffentliche Schlüssel für P1 **2006**, K_{TxRoot} ist, wodurch eine Signatur der Nachricht angegeben wird.

[0139] Die DLEA **2002** kann einer Instanz eines semi-permissionierten verteilten Ledger-Systems (DLS) eine eindeutige Zahl zuweisen und die durch die DLEA zugeordnete Zahl zusammen mit dem öffentlichen Schlüssel K_{TxRoot} DLS-0 **2004** posten 2010. DLS-0 **2004** ist das öffentliche verteilte Ledger-System (DLS) und nur von der DLEA **2002** beschreibbar, jedoch für alle sichtbar.

[0140] P1 **2006** kann den Ledger DLS-0 **2004** überwachen **2012**, um es festzustellen, wenn die Zuweisung eines neuen Schlüssels X aufgezeichnet wurde. Die zugewiesene Zahl X kann von P1 **2006** als Root- oder Startschlüssel eines neu gebildeten Ledgers DLS-X **2014** verwendet werden. Dies kann durch Erzeugen des Ledgers DLS-X **2014**, durch Übermitteln einer Nachricht **2016** zum neuen Ledger DLS-X **2014**: [K_{TxRoot}] K_{DLS-X} ; [K_{DLS-X} , perm] K_{TxRoot} ; und [K_{TxP2}] K_{TxRoot} geschehen, wobei K_{TxP2} ein neuer Ledger-Transaktionsschlüssel ist.

[0141] Der neue Ledger DLS-X **2014** kann auch zur Implementation des EPID-Beitrittsprotokolls verwendet werden, das private EPID-Schlüssel für jedes neue Mitglied von DLS-X **2014** einrichtet. Jeder nachfolgende Gebrauch privater EPID-Schlüssel kann durch die Verwendung des öffentlichen Schlüssels K_{TxRoot} der ersten Transaktion für den Ledger DLS-X **2014** verifiziert werden. Jegliche der privaten EPID-Schlüssel können durch Signieren des neuen TxK mit dem EPID-Schlüssel Ledger-Transaktionsschlüssel (K_{Tx}) für DLS-X **2014** einleiten.

[0142] Beispielsweise kann ein anderer Teilnehmer P2 **2018** eine Beitrittsanforderung **2020** zum ersten Teilnehmer P1 **2006** senden. Die Beitrittsanforderung **2020** kann die Nachricht: [BeitretenP DLS-X] K_{Mfg2} ; [K_{TxP2}] K_{TxP2} aufweisen. Der zweite Teilnehmer P2 **2018** kann den Transaktionsschlüssel K_{TxP2} durch Zugreifen auf DLS-X **2014** erhalten haben. Der zweite Schlüssel K_{Mfg2} kann der EPID-Schlüssel eines Herstellers in der Art von K_{Mfg2} sein, wobei der Root- K_{Tx} durch einen vom Hersteller zugeführten EPID-Schlüssel des Formats K_{Mfg} attestiert oder signiert ist. Der K_{Mfg} attestiert, dass die Trusted Execution Environment (TEE), welche den Root-TxK enthält, ausreichend vertrauenswürdig ist. Ebenso wird ein K_{Mfg} in der TEE der neuen Teilnehmer-vorrichtung verwendet, um zu attestieren, dass der für das Schützen der Beitrittsanforderung **2020** verwendete temporäre Schlüssel, beispielsweise K_{TxP2} , legitim und vertrauenswürdig ist.

[0143] Falls P1 **2006** die Anforderung authentifiziert, kann er eine Nachricht **2022** zu P2 **2018** zurückführen, um den Beitritt abzuschließen. Die Nachricht **2022** kann [[BeitretenI DLS-X] K_{TxRoot}] K_{Mfg1} aufweisen, wobei K_{Mfg1} die Hersteller-EPID für P1 **2006** ist. Der Root- K_{Tx} , K_{TxRoot} , wird zur Authentifizierung der Beitrittsprotokollantwort verwendet.

[0144] Die Vorrichtungen P1 **2006** und P2 **2018** können auf zwei verschiedenen hierarchischen Ebenen existieren. Demgemäß kann eine Anzahl von Vorrichtungen auf der Ebene von P2 **2018** beispielsweise P1 **2006** als zusammengesetztes Objekt und Sub-Objekte, wie hier beschrieben, beitreten. Ähnlich können andere Vorrichtungen P2 **2018** auf einer niedrigeren Ebene beitreten, beispielsweise Teilnehmer P3 **2024**. Um beizutreten, kann P3 **2024** eine Beitrittsprotokollanforderung **2026** zu P2 **2018** der Form [BeitretenP DLS-X] K_{Mfg3} ; [K_{TxP3}] K_{TxP3} senden. Falls P2 **2018** die Beitrittsprotokollanforderung **2026** authentifiziert, kann er mit einer Nachricht **2028** des Formats: [[BeitretenI DLS-X] K_{TxP2}] K_{Mfg2} ; [TxDaten, K_{TxP3}] K_{TxP2} antworten. P3 **2024** kann die Transaktion an den verteilten Ledger DLS-X **2014** durch Aufzeichnen einer signierten Nachricht **2030** des Formats: [[TxDaten, K_{TxP3}] K_{TxP2}] $K_{DLS-XP3}$ im Ledger DLS-X **2014** übergeben.

[0145] Statt BeitretenP-Transaktionen zu verwenden, können P2 und P3 Peer-Knoten in der Blockchain (X) sein. Dementsprechend können sie die Transaktionsschlüssel (KT) verwenden, um Handel zu treiben. Beispielsweise kann die Nachricht **2028** eine Ware oder eine Dienstleistung kaufen und kann die Nachricht **2026** die Ware oder die Dienstleistung verkaufen. In diesem Fall benötigen sie nur KTx-Schlüssel, und die Technik beschreibt ein Blockchain-Transaktionsschlüsselverhalten.

[0146] Ferner haben Blockchains im Allgemeinen keinen KDLS-Schlüssel. Das bedeutet, dass die Blockchains nicht in der Lage sein können, semi-permissionierte Transaktionen zu erzwingen. Beispielsweise kauft

in Nachricht **2028** P2 eine Ware oder eine Dienstleistung und weiß P3, dass P2 ein Mitglied eines Clubs, beispielsweise einer kommerziellen Einrichtung, einer Online-Auktionsseite, eines Casinos und dergleichen ist. Dementsprechend kann P2 ein verbilligtes Angebot erhalten, falls der Verkäufer P3 auch Teil des Clubs ist oder falls eine clubeigene Währung in der Art von Spielchips gegen verschiedene von den Clubmitgliedern bereitgestellte Waren oder Dienstleistungen getauscht wird.

[0147] Es kann sinnvoll sein, EPID als Transaktionsschlüssel (KTx) zu verwenden, um aus Gründen der Zweckmäßigkeit mehrere Wallets aufrechtzuerhalten. Wie hier verwendet, kann ein Wallet ein kryptographisch geschützter elektronischer Speicher sein, der eine Währung oder eine Verbindung zu einem Guthabenkonto hält. Bei diesem Beispiel können P2 und P3 verschiedene Wallets sein, die jeweils einen Anteil eines verteilten Wallets, beispielsweise die verteilten Wallets des jeweiligen Anderen, halten.

[0148] Ein anderer Fall, in dem die EPID als Transaktionsschlüssel verwendet werden kann, tritt auf, wenn P2 und P3 jeweils Mitglieder einer Gruppe in der Art einer Gruppe von Angestellten in einer Firma oder einer Gruppe von Personen, die eine Kirche oder ein ziviles Unternehmen repräsentieren, sind, wobei die verschiedenen Mitglieder als Agenten des Unternehmens handeln können. Aus der Perspektive einer Blockchain ist es semantisch nicht bedeutsam, ob der Tx-Schlüssel ein EPID-Schlüssel oder ein Schlüssel anderer Typen ist, solange die Signatur die Identitäten verifiziert.

[0149] **Fig. 21** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **2100** zur Ausführung semi-permissionierter Transaktionen gemäß einigen Ausführungsformen. Das Verfahren **2100** aus **Fig. 21** kann durch die mit Bezug auf **Fig. 22** beschriebene IoT-Vorrichtung 2200 implementiert werden. Der Block **2102** repräsentiert beispielsweise den Fall, in dem eine Vorrichtung angewiesen wird, sich anderen Vorrichtungen anzuschließen. In Block **2104** stellt ein erster Teilnehmer fest, dass eine Gemeinschaft von Dingen, unter anderem in der Art der eine Fog-Vorrichtung bildenden IoT-Vorrichtungen, mit hochintegrierten Garantien interagieren kann.

[0150] In Block **2106** reserviert der erste Teilnehmer einen Namen, der die Gemeinschaft repräsentiert. Dies kann beispielsweise durch Senden eines Namens, beispielsweise DLS-X, und eines öffentlichen Schlüssels für den ersten Teilnehmer zu einer DLEA geschehen. Der Name, beispielsweise DLS-X, kann eine Universally Unique Identifier (UUID) oder eine andere Identifikation sein, die eine sehr geringe Wahrscheinlichkeit einer Replikation aufweist. Die Nachricht kann durch einen privaten Schlüssel für den ersten Teilnehmer signiert werden.

[0151] In Block **2108** stellt die DLEA fest, ob der Name aktuell in Gebrauch ist oder bereits zuvor zugewiesen wurde. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **2106** zurück, damit der erste Teilnehmer einen neuen Namen wählt. Falls dies nicht der Fall ist, reserviert die DLEA in Block **2110** den Namen DLS-X durch Übergeben an einen verteilten Ledger DLS-0. Der für die Authentifizierung der anfänglichen Transaktion für die DLEA verwendete Schlüssel kann zusammen mit dem Namen an den Ledger übergeben werden.

[0152] In Block **2112** kann der erste Teilnehmer den DLS-X-Namen verwenden, wenn dieser Name auf DLS-0 erscheint. Dies kann durch den ersten Teilnehmer, der den DLS-0-Ledger überwacht, festgestellt werden. In Block **2114** legt der erste Teilnehmer einen öffentlichen DLS-X-Gruppenschlüssel unter Verwendung der EPID fest und definiert eine Permissionierungsrichtlinie. Der öffentliche Gruppenschlüssel und die Richtlinie werden unter Verwendung des Transaktionsschlüssels des ersten Teilnehmers an den DLS-X-Ledger übergeben. Die Transaktion des ersten Teilnehmers kann unter Verwendung des privaten EPID-Gruppenschlüssels an DLS-X übergeben werden.

[0153] In Block **2116** kann ein zweiter Teilnehmer der DLS-X-Gruppe beitreten, indem er einen privaten DLS-X-Gruppenschlüssel vom ersten Teilnehmer erhält. Der erste Teilnehmer kann als Ausgeber eines EPID-Gruppenschlüssels wirken. Der zweite Teilnehmer kann die Vertrauenswürdigkeit seiner Vorrichtung unter Verwendung eines Herstellerschlüssels, beispielsweise eines EPID-Herstellerschlüssels, attestieren. In Block **2118** wird festgestellt, ob die Attestierung der zweiten Vorrichtung vertrauenswürdig ist. Falls dies nicht der Fall ist, endet das Verfahren **2100** in Block **2120**.

[0154] Falls die Attestierung vertrauenswürdig ist, empfängt der zweite Teilnehmer in Block **2122** eine EPID-Beitrittsprotokollantwort, wodurch es ihm erlaubt wird, einen zweiten privaten Gruppenschlüssel unter dem öffentlichen EPID-Gruppenschlüssel für DLS-X zu erzeugen. In Block **2124** signiert ein zweiter Teilnehmer seinen Transaktionsschlüssel selbst und übergibt ihn an den ersten Teilnehmer. Der erste Teilnehmer signiert den öffentlichen Schlüssel des zweiten Teilnehmers und übergibt die Transaktion an den Ledger DLS-X, wo-

durch der zweite Teilnehmer in DLS-X eingeführt wird. In Block **2126** wird festgestellt, ob es einen anderen Teilnehmer gibt. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **2116** zurück, um die nächste Registrierung wiederaufzunehmen.

[0155] In Block **2128** kann sich ein dritter Teilnehmer selbst einem zweiten Teilnehmer vorstellen. Dies kann dadurch geschehen, dass der dritte Teilnehmer selbst einen Transaktionsschlüssel des dritten Teilnehmers signiert und ihn zum zweiten Teilnehmer sendet. Der zweite Teilnehmer signiert den öffentlichen Transaktionsschlüssel des dritten Teilnehmers und nimmt optional Transaktionsdaten auf und signiert mit seinem Transaktionsschlüssel und dem DLS-X-Gruppenschlüssel.

[0156] In Block **2130** übergibt der dritte Teilnehmer die Transaktion an DLS-X. Dies kann dadurch geschehen, dass der dritte Teilnehmer die Transaktionsdaten des zweiten Teilnehmers unter Verwendung des privaten DLS-X-Gruppenschlüssels des dritten Teilnehmers signiert, bevor er die Transaktion an die DLS-X-Blockchain übergibt. Der zweite Teilnehmer kann die Transaktionsdaten auch unter Verwendung seines privaten DLS-X-Gruppenschlüssels an den DLS-X-Ledger übergeben. Bei diesem Szenario signiert der dritte Teilnehmer auch seinen selbst signierten Tx-Schlüssel mit dem DLS-X-Gruppenschlüssel des dritten Teilnehmers. Das Verfahren **2100** endet dann in Block **2120**.

[0157] Fig. 22 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **2200** zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **2200** andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0158] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Gruppenerzeuger **2202** aufweisen, der feststellt, ob eine Gruppe von Objekten mit sehr vertrauenswürdigen Garantien interagieren kann.

[0159] Wie hier erörtert, können die Garantien auf in die IoT-Vorrichtung **2200** und andere Mesh-Vorrichtungen **812** durch Hersteller einprogrammierten Attestierungsschlüsseln beruhen. Der Gruppenerzeuger **2202** kann einen Namen für die Gruppe erzeugen. Ein DLEA-Zugreifer **2204** kann auf eine DLEA zugreifen, um festzustellen, ob der Name verfügbar ist oder ob die IoT-Vorrichtung **2200** einen anderen Namen erzeugen muss. Falls der Name verfügbar ist, übergibt die DLEA den Namen an einen verteilten Ledger DLS-0. Der DLEA-Zugreifer **2204** kann DLS-0 überwachen, um festzustellen, ob der Name übergeben wurde. Ein Schlüsselerzeuger **2206** kann einen Schlüssel auf der Grundlage eines vom Gruppenerzeuger **2202** erzeugten Namens, beispielsweise unter Verwendung eines EPID-Servers, erzeugen. Der Schlüsselerzeuger **2206** kann den Schlüssel an einen lokalen verteilten Ledger DLS **2008** übergeben. DLS **2208** kann in der IoT-Vorrichtung **2200** oder in einer anderen Mesh-Vorrichtung **812** existieren. Ein Attestierungsprüfer **2210** kann aufgenommen sein, um festzustellen, ob eine Beitrittsanforderung von einer anderen Vorrichtung gültig ist. Falls dies der Fall ist, kann eine Gruppenbeitrittseinrichtung **2212** eine Beitrittsnachricht mit dem Gruppenschlüssel aussenden.

[0160] Fig. 23 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **2300**, das einen Code aufweist, um einen Prozessor **902** anzuweisen, sicher in Gruppen zu kommunizieren, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **2300** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf den Prozessor **802** und den Bus **806** aus Fig. 8 beschrieben ausgewählt werden. Das nichtflüchtige maschinenlesbare Medium **2300** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0161] Das nichtflüchtige maschinenlesbare Medium **2300** kann einen Code **2302** aufweisen, um den Prozessor **902** anzuweisen, festzustellen, dass eine Gruppe mit hoher Integrität kommunizieren kann. Ein Code **2304** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Namen für die Gruppe zu erzeugen und den Namen mit einer Distributed Ledger Enumeration Authority (DLEA) zu reservieren. Ein Code **2306** kann aufgenommen sein, um den Prozessor **902** anzuweisen, andere Schlüssel anhand des registrierten Namens zu erzeugen und die Informationen an einen neuen verteilten Ledger DLS-X **2308** zu übergeben.

[0162] Ein Code **2310** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Beitrittsanforderung für die Gruppe von IoT-Vorrichtungen, zusammengesetzten Objekten oder für beide zu validieren. Die Beitrittsanforderung kann Attestierungsinformationen in der Art eines einer anfordernden Vorrichtung bereitgestellten Herstellerschlüssels aufweisen. Ein Code **2312** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Berechtigungsnachweise an den Antragsteller in der Art einer EPID auszugeben. Ein Code **2314** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Transaktionsdaten unter Verwendung eines privaten Schlüssels, eines öffentlichen Schlüssels oder einer Kombination von beiden an den verteilten Ledger DLS-X zu übergeben.

[0163] Zusätzlich zu sicheren Kommunikationen kann Sicherheit während des Bootens nützlich sein, um das Netz vor einem Eindringen zu schützen. Wenngleich ein sicheres Booten in einem weniger beschränkten System, einschließlich größerer IoT-Vorrichtungen, unter Verwendung eines Trusted Execution Module (TEM) oder einer anderen Hardwarevorrichtung implementiert werden kann, kann dies für stärker ressourcenbeschränkte IoT-Vorrichtungen eine größere Herausforderung sein.

[0164] Fig. 24 ist ein schematisches Diagramm **2400** der Verwendung einer vertrauenswürdigen Ausführungsumgebung bzw. Trusted Execution Environment (TEE) zum sicheren Booten einer Vorrichtung in einer IoT-Umgebung gemäß einigen Ausführungsformen. Eine vertrauenswürdige Berechnung betrifft in erster Linie die Fähigkeit einer Vorrichtung, vertrauenswürdige Attribute einer Rechenvorrichtung zu attestieren. Attribute, die typischerweise das Vertrauen beeinflussen, umfassen ein vertrauenswürdiges oder sicheres Booten.

[0165] Das vertrauenswürdige Booten instrumentiert die Boot-Sequenz mit Messoperationen, die einen Hash des zu ladenden und auszuführenden nächsten Codeblocks berechnen. Messungen werden im sicheren Speicher in der Art eines vertrauenswürdigen Moduls (TEE) **2402** gespeichert. Bei einer IoT-Vorrichtung kann das vertrauenswürdige Modul **2402** eine separate Vorrichtung sein oder ein geschützter Speicherbereich sein, der verschlüsselt ist oder auf andere Weise für den Prozessor oder allgemeinen Operationscode der IoT-Vorrichtung im Allgemeinen nicht zugänglich ist. Ein sicheres Booten ist eine Erweiterung einer vertrauenswürdigen Boot-Umgebung, wobei das Prüfen von Messungen anhand einer Whitelist zugelassener Prozesse hinzugefügt wird. Typischerweise wird die Boot-Sequenz geändert, falls tatsächliche und Whitelist-Messungen nicht übereinstimmen, beispielsweise durch Booten in eine nicht sichere Umgebung und Informieren anderer Vorrichtungen darüber.

[0166] Sobald das vertrauenswürdige Booten abgeschlossen ist, kann es das TEE für die sichere Ausführung bereitstellen. Falls Code in eine gehärtete Ausführungsumgebung in der Art des TEE geladen wird oder statisch daran gebunden ist, können die ausgeführten Operationen einigen Angriffen widerstehen. Eine gehärtete Ausführungsumgebung kann eine Anzahl hardwareerweiterter Sicherheitssysteme in der Art eines Trusted Platform Module (TPM) zur Erzeugung des TEE aufweisen. Die Härtungstechniken können unter anderem Software Guard Extensions (SGX) von Intel®, TrustZone® von ARM®, Hardwaresicherheitsmodule bzw. Hardware Security Modules (HSM) in der Art eines TPM, von Smartcards oder einer Virtualisierung einschließen.

[0167] Das TEE kann auch eine Umgebung für eine sichere Aktualisierung bereitstellen. Das sichere Booten prüft die Codeauthentizität bei der Ladezeit. Eine sichere Aktualisierung verwendet eine Codesignierung, um Integrität und Authentizität zu gewährleisten, beispielsweise mit der Authenticode™-Technologie von Microsoft. Eine Manifeststruktur kann verwendet werden, um die Assoziation von Code-Hash-Werten und Signaturen über Hash-Werten als Teil des Installationsbilds zu behandeln. Technologien zur Installation von Bildpaketen schließen unter anderem das Itsy Package Management System (IPKG), Debian Linux-Installationsdateien (DEB), RPM-Paketmanagerdateien (RPM) und Clear Linux Bundles ein.

[0168] Das TEE kann ein sicheres Speichern sowohl für eine vorübergehende als auch für eine langzeitige Speicherung sicherheitsrelevanter Daten bereitstellen. Datentypen umfassen Schlüssel, Whitelists, Blacklists, Messungen, Auditierungsprotokolle, Passwörter, biometrische Daten, Zertifikate und Richtlinien. Härtungstechniken umfassen Isolation, Manipulationsverhinderung, Verschlüsselung und Verbergung.

[0169] Die Attestierung kann Teil der sicheren Umgebung sein. Die Attestierung, wie hier beschrieben, ist eine Mitteilungsfunktion, die an eine sichere Ausführungs- oder sichere Speicherfunktion gebunden ist, wobei die Vorrichtung oder Plattform ihre Vertrauenseigenschaften selbst mitteilt. Sie detailliert die Härtungstechniken und -garantien, die auf die betreffende sichere Funktion angewendet werden. Die Attestierungsfunktion selbst muss eine sichere Funktion sein, bei der die Härtung und die Garantien das Qualitätsniveau der Funktion, über die sie berichtet, überschreiten.

[0170] Die Herausforderungen für das vertrauenswürdige Berechnen können infolge mehrerer Faktoren in einer IoT-Vorrichtung zunehmen. Beispielsweise können IoT-Vorrichtungen durch Größe, Funktionalität und wirtschaftliche Gesichtspunkte beschränkt sein. Die Sicherheitshärtung erfolgt häufig als Kompromiss mit diesen Kosten. Die Aufnahme vertrauenswürdiger Rechenbaublöcke kann auf kostenbeschränkten Vorrichtungen fehlen oder unvollständig sein.

[0171] Ferner können IoT-Netze die Funktionalität über mehrere Vorrichtungen verteilen, was zu einer größeren Abhängigkeit von Netzbaublöcken führt. Folglich kann das Netzverhalten problematischer sein, wenn das Netz zu einem größeren Bestandteil des Gesamtrechengewebes wird. Unerwünschte Verhaltensweisen können verstärkt werden, wenn die Netzkomplexität und der Netzzumfang zunehmen.

[0172] IoT-Netze können häufig Vorrichtungen und Anwendungen von einer Anzahl von Verkäufern, Mehrwert-Wiederverkäufern, Integratoren, Lieferanten und Analysten aufweisen. Jeder dieser Akteure kann Systeme erzeugen, die zusammenarbeiten müssen, um zu gewährleisten, dass Schnittstellen, Strukturen, Rechenumgebungen und Operationsprozeduren geeignet zueinander passen, ohne ein unerwartetes und unerwünschtes Verhalten herbeizuführen.

[0173] Bei einigen Aspekten können IoT-Netze, um diese Probleme zu adressieren, eine Verteilung des Vertrauens über mehrere Vorrichtungen aufweisen. Die Verteilung ist ein Weg, um eine verringerte Zuverlässigkeit, Verfügbarkeit und Sicherheit zu adressieren, welche die Zentralisierung mit sich bringt. Die Verteilung verstreut auch Entscheidungsprozesse, wenn sich die natürlichen zentralen Steuerungspunkte auflösen.

[0174] Bei einigen Aspekten kann die Attestierung der vertrauenswürdigen Berechnung in IoT-Netzen durch die Verwendung der Blockchain-Technologie verbessert werden. Vertrauenswürdige Rechenkonzepte definieren einen Satz von Vertrauens-Roots, die eine für die Sicherheit grundlegende Funktion ausführen, wobei implizit vertraut wird, dass das geeignete und erwartete Verhalten der Root-Funktionalität wie erwartet funktioniert. Die vertrauenswürdige Rechengruppe bzw. Trusted Computing Group (TCG) kann beispielsweise im vertrauenswürdigen Modul **2402** mehrere Vertrauens-Roots aufweisen.

[0175] Ein Vertrauens-Root für die Messung bzw. Root of Trust for Measurement (RTM) **2404** ist eine Funktion, welche das erste ladbare Objekt in einem System misst und verifizieren kann. Ein Vertrauens-Root für die Mitteilung bzw. Root of Trust for Reporting (RTR) **2406** ist eine Funktion, welche Werte in der Vertrauens-Root für die Speicherung bzw. Root of Trust for Storage (RTS) **2408** und in der Rechenumgebung, welche die RTM **2404**, RTR **2406** und RTS **2408** implementiert, attestiert. Die Attestierungsfunktion kann innerhalb der RTR **2406** rekursiv definiert sein. Die Vertrauens-Root für die Speicherung bzw. Root of Trust for Storage (RTS) **2408** ist die Funktion, die von den RTM **2404** und RTR **2406** erzeugte und verbrauchte Werte speichert.

[0176] Blockchain-Vertrauens-Roots können in IoT-Netzumgebungen verwendet werden, um die Sicherheit durch Verteilen der Sicherheitsfunktionen zu erhöhen. Durch Verteilen des Vertrauens in die Blockchain verwendenden IoT-Netzen können zwei zusätzliche Vertrauens-Roots für die Blockchain hinzugefügt werden. Eine Vertrauens-Root für eine Verkettung bzw. Root of Trust for Chaining (RTC) **2410** ist eine Funktion, die eine Blockchain-Ressource lokalen vertrauenswürdigen Rechen-Roots in der Art von RTR **2406** offenlegt. Die RTC **2410** und die RTR **2406** können zusammenarbeiten, um attestierte Attribute an eine Blockchain zu übergeben, beispielsweise durch Speichern der attestierten Attribute in einer Chain-Geschichte **2412**. Die Vertrauenseigenschaften von Blockchains sind sehr wünschenswert, weil sie die Verteilung als Mechanismus für das Garantieren eines erwarteten Verhaltens unter Verwendung von Schwellenkonsensprotokollen verwenden.

[0177] Eine Vertrauens-Root-für-die-Archivierung-Funktion bzw. Root of Trust for Archival Function (RTA) **2414** fügt eine Verfügbarkeitskomponente zu den anderen Roots hinzu. Eine eingeschränkte IoT-Vorrichtung kann nicht die Ressourcen zum Aufrechterhalten einer Geschichte von Messungen **2416** und von Messprotokollen, die mehrere Reboots überspannen, aufweisen. Ferner kann sie nicht in der Lage sein, kostspielige Whitelists **2418** zu speichern, die frühere oder vorhergesehene Konfigurationen beschreiben. Eine Untersuchung eines vertrauenswürdigen Berechnens kann das Suchen nach einem historischen Zusammenhang erfordern. Die RTA **2414** fügt RTC-Knoten, die nicht die vollständige Blockgeschichte unterhalten können, eine Archivierungsfähigkeit hinzu.

[0178] Das hier beschriebene System kann mit Blockchain-Logik **2420** verwendet werden, die mit Blockchain-Logik **2422** in anderen Vorrichtungen arbeitet, um die Chain-Geschichte **2412** aufrechtzuerhalten. Dies kann beispielsweise das Weiterleiten **2424** der Chain-Geschichte **2412** der Blockchain zu anderen Vorrichtungen einschließen. In anderen Vorrichtungen kann die Chain-Geschichte **2412** mit lokalen Kopien verglichen wer-

den, um sicherzustellen, dass die vorgenommenen Änderungen autorisiert sind. Falls eine Mehrzahl der Vorrichtungen darin übereinstimmt, dass die Änderung nicht autorisiert war, reuert die Blockchain-Logik **2420** die Chain-Geschichte **2412** zur vorhergehenden Geschichte.

[0179] Fig. 25 ist ein Blockdiagramm **2500** eines Blockchain-Blocks **2502**, der Boot-Integritätstransaktionen enthält, gemäß einigen Ausführungsformen. Mit Bezug auch auf Fig. 24 sei bemerkt, dass der Blockchain-Block **2502** einen einzigen Datensatz in der Chain-Geschichte **2412** oder einem anderen verteilten Ledger-System bildet. Die RTC **2410** bildet einen Block, der Messungen **2504** enthält, in Plattformkonfigurationsregistern bzw. Platform Configuration Registers (PCR). Die PCR können Speicherstellen in einem geschützten Bereich, in einer spezifischen Hardwarevorrichtung oder in beiden sein.

[0180] Bei einigen Aspekten kann die Abtastrate für die Messungen, die für den Blockchain-Block **2502** verwendet werden, granulärer sein als die Rate, mit der Messungen in der PCR gespeichert werden, beispielsweise PCR-Erweiterungen. Jede PCR-Erweiterung kann jedoch eine Transaktion auslösen, die zu einem Block hinzugefügt wird. PCR-Werte werden durch einen Attestierungssignierschlüssel **2506** signiert, der sich vom Block-Signaturschlüssel unterscheiden kann. Im Wesentlichen attestiert die RTR **2406** der Blockchain ihren aktuellen Integritätszustand. Die RTC **2410** attestiert, dass die PCR nicht durch unerkannte Systemrücksetzvorgänge überschrieben wurden.

[0181] Das Blockdiagramm **2500** kann auch das Vorhandensein vorhergehender Blockchain-Blöcke **2510** und **2512** angeben. Wenngleich dies in dieser Figur nicht dargestellt ist, können diese Blöcke **2510** und **2512** andere Boot-Integritätstransaktionen enthalten oder Informationen über zusammengesetzte Objekte, Objekttypen, Koalitionsgruppenzusammensetzungen, sichere Transaktionsdaten oder eine Anzahl anderer Elemente zur Unterstützung der Sicherheit eines IoT-Netzes enthalten.

[0182] Fig. 26 ist ein schematisches Diagramm **2600** der Verwendung einer Whitelist-Bildsammlung mit einer Blockchain gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 24 beschrieben wurden. Ein Boot-Prozess findet auf einer ersten IoT-Vorrichtung **2602** statt. Es kann auf einen Bildablagelpeicher **2604** zugegriffen werden, um ein Whitelist-Bild **2606** zu erhalten, beispielsweise unter Verwendung von Kommunikationen **2608**, die mit einem in das System programmierten Herstellerschlüssel **2612** verschlüsselt sind. Bei einigen Beispielen kann an Stelle des Bildablagelpeichers **2604** oder zusätzlich dazu von einer Chain-Geschichte **2412** oder einer Blockchain auf sie zugegriffen werden. Die Bilder im Bildablagelpeicher **2604** können durch andere ähnliche IoT-Vorrichtungen **2610** gespeichert worden sein, so dass ein Referenzzählwert aufrechterhalten werden kann. Weil jede Vorrichtung ihre Blockchain-Transaktion, welche Boot-Integritätsmitteilungen aufzeichnet, signieren kann, kann der Referenzzählwert zwischen der Reboot-Aktivität von derselben Vorrichtung gegenüber der Aktivität von anderen Vorrichtungen unterschieden werden.

[0183] Messungen werden vorgenommen, wenn die IoT-Vorrichtung **2602** bootet, beispielsweise durch Berechnen eines Hash-Codes der nächsten in der Boot-Sequenz auszuführenden Software. Die Messungen können mit Whitelist-Werten, beispielsweise im Whitelist-Bild **2606**, verglichen werden, um die Integrität zu gewährleisten. Ein Bildmanifest **2614** kann verwendet werden, um den Ursprung des Whitelist-Werts zu validieren. Das Manifest **2614** kann Whitelist-Hash-Werte aufweisen, die mit einem dynamisch erhaltenen Hash des Bilds **2606** verglichen werden können.

[0184] Die Bildung von Whitelists in IoT-Netzen stellt wegen der Rate, mit der sich die Population von Bildern ändert, eine Herausforderung dar, wobei beispielsweise, wenn der Bildablagelpeicher **2604** wächst, die Wahrscheinlichkeit größer wird, dass Vorrichtungen in einem Einsatzgebiet vom Ablagespeicher abhängen, um für die Aufnahme in eine Whitelist ausgewählte Referenzbilder zu finden. Falls es keine Datendeduplikationsfunktion und eine vertrauenswürdige Löschfunktion im Netz gibt, steigt die Anzahl der Bilder monoton an, weil es eine IoT-Vorrichtung geben kann, welche das Bild im Ablagespeicher referenziert. Die Blockchain-Geschichte ist ein Weg, den Bildablagelpeicher über die Beliebtheit von Vorrichtungen, welche seine Bilder referenzieren, zu informieren. Vorrichtungen, die nicht mehr im Dienst sind, würden in der Geschichte **2412** nicht auftreten, so dass für sie keine Referenzzählung durch den Bildablagelpeicher vorgenommen werden würde. Der Bildablagelpeicher **2604** kann eine „Wärmekarte“ enthalten, welche die Vorrichtungen offenbart, die eine Boot-Integritätsprüfung ausführen. Eine Strategie, die ältere Vorrichtungen, die nicht mehr im Einsatz sind, obsolet macht, kann darin bestehen, ihr Bild **2606** aus dem Bildablagelpeicher **2604** zu entfernen und die Whitelist-Referenzierung zu blockieren. Dieser Ansatz kann abgestimmt werden, um eine Dekommissionierungsrate auszuwählen, die mit der Wachstumsrate korreliert ist, mit der neue Bilder erzeugt werden.

[0185] Fig. 27 ist eine Zeichnung eines Blockchain-Blocks 2702 mit Integritätstransaktionen für Whitelist-Bilder gemäß einigen Ausführungsformen. Zur Implementation des Blockchain-Blocks können Verkäufer, Hersteller und Codeerzeugungsfirmen Blockchain-Fähigkeiten in ihren Produktionsprozess aufnehmen. Jedes Whitelist-Bild kann unter Verwendung einer Manifeststruktur 2704, die das Manifest 2706 aufweist, signiert werden. Der Entwickler oder die Firma, wodurch das Bild erzeugt wird, kann es unter Verwendung eines Herstellerschlüssels 2708, der ein EPID-Schlüssel sein kann, signieren, um festzulegen, welche Entität das Bild hergestellt hat. Signierte Manifeste 2704 werden zum Blockchain-Block 2702 hinzugefügt und unter Verwendung eines geeigneten Transaktionsschlüssels, wie hier beschrieben, an die Chain-Geschichte 2412 (Fig. 24) der Blockchain übergeben.

[0186] Fig. 28 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 2800 für einen sicheren Boot-Prozessfluss unter Verwendung von Blockchain-Vertrauens-Roots gemäß einigen Ausführungsformen. Das Verfahren 2800 aus Fig. 28 kann durch die mit Bezug auf Fig. 29 beschriebene IoT-Vorrichtung 2900 implementiert werden. Block 2802 repräsentiert beispielsweise den Fall, in dem ein Boot-Integritätsagent ein Objekt misst. Wie hier erörtert, kann dies durch Berechnen eines Hash-Codes des nächsten zu bootenden Codes, wodurch ein Bild des Codes erzeugt wird, geschehen. In Block 2804 wird festgestellt, ob das Bild als gut bekannt ist. Falls dies der Fall ist, endet das Verfahren 2800 in Block 2806, wenn die IoT-Vorrichtung normale Operationen fortsetzt. Falls dies nicht der Fall ist, wird in Block 2808 festgestellt, ob das Bild als schlecht bekannt ist. Falls dies der Fall ist, endet das Verfahren 2800 in Block 2810 mit der Quarantäne des Codes und der Behebung des Problems.

[0187] Falls das Bild in Block 2808 nicht als schlecht bekannt ist, wird der Prozessfluss in Block 2812 fortgesetzt, wo festgestellt wird, ob das Bild unbekannt ist. Falls dies nicht der Fall ist, kann das Verfahren 2800 in Block 2814 enden, beispielsweise mit dem Status, dass es als nicht vertrauenswürdig aufgelistet wird. Falls dies der Fall ist, kann das Verfahren 2800 in Block 2816 enden, wo eine lokale Richtlinie konsultiert wird, um die anzuwendende Aktion zu bestimmen.

[0188] Um ein Bild für die Verwendung beim Vergleich in Block 2804 zu erhalten, kann in Block 2818 ein Seitenadministrator einen Referenz-Hash, beispielsweise aus einem Cloud-Ablagespeicher, erhalten. Der Hash kann von anderen Quellen, einschließlich anderer IoT-Vorrichtungen, Hersteller und dergleichen erhalten werden. In Block 2822 wird festgestellt, ob die Signatur am Hash gültig ist. Falls dies nicht der Fall ist, endet das Verfahren 2800 in Block 2822. In Block 2824 wird festgestellt, ob der Bild-Hash gleich dem Blockchain(BC)-Hash ist. Falls dies der Fall ist, signiert der Seitenadministrator in Block 2826 das Manifest für das Bild. In Block 2828 wird das Bild zur Whitelist hinzugefügt und wird die Whitelist zum Zugriff durch den Boot-Code an die Blockchain übergeben. Das Whitelist-Bild kann dann beim Vergleich in Block 2804 verwendet werden, beispielsweise durch eine IoT-Vorrichtung, die auf die Whitelist in der Blockchain oder in einem Bildablagenspeicher zugreift.

[0189] Falls der Bild-Hash in Block 2824 nicht mit dem BC-Hash übereinstimmt, wird in Block 2830 festgestellt, ob der Bild-Hash eine Angriffssignatur enthält. Falls dies der Fall ist, kann das Bild in Block 2832 zu einer Blacklist hinzugefügt werden, und die Blacklist kann an die Blockchain übergeben werden. Das Blacklist-Bild kann dann beim Vergleich in Block 2808 verwendet werden, beispielsweise durch eine IoT-Vorrichtung, die auf die Blacklist in der Blockchain oder in einem Bildablagenspeicher zugreift.

[0190] Falls in Block 2830 der Bild-Hash nicht mit einer bekannten Angriffssignatur übereinstimmt, kann das Bild in Block 2834 zu einer nicht klassifizierten Liste hinzugefügt werden. Die nicht klassifizierte Liste kann dann zur Blockchain hinzugefügt werden. Das nicht klassifizierte Bild kann dann beim Vergleich in Block 2812 verwendet werden, beispielsweise durch eine IoT-Vorrichtung, die auf die nicht klassifizierte Liste in der Blockchain oder in einem Bildablagenspeicher zugreift.

[0191] Die Angriffssignaturen können durch eine Anzahl von Techniken identifiziert werden. Beispielsweise kann in Block 2836 ein forensisches Labor den Angriff identifizieren und die Angriffssignatur für das Bild erzeugen. Wie hier verwendet, kann ein forensisches Labor ein kommerzieller Sicherheitsdienst sein, der Malware, Viren und anderen in Umlauf befindlichen problematischen Code identifiziert. In Block 2838 kann das forensische Labor die Angriffssignatur für das Bild in die Blockchain schreiben. Bei einigen Beispielen kann der Seitenadministrator die Angriffssignatur von einem kommerziellen forensischen Labor erhalten und sie in die Blockchain schreiben. In Block 2840 kann die Angriffssignatur von der Blockchain für die Verwendung in Block 2830 erhalten werden.

[0192] Wie hier beschrieben, kann der sichere Boot-Prozess erweitert werden, so dass er die Verwendung einer Blockchain zum Erhalten und Validieren von Referenzmessungen, zum Formulieren einer Whitelist, einer Blacklist oder einer nicht klassifizierten Liste, die zur Beurteilung lokaler Messungen verwendet werden kann, aufweist. Normalerweise geschieht eine sichere Boot-Erzwingung. Demgemäß kann die Blockchain Informationen für Erzwingungspunkte für die Netzquarantäne bereitstellen, welche dem Paketfluss zu und von Vorrichtungen Firewallbeschränkungen auferlegen kann, wenn eine bekannte schlechte oder unbekannte Konfiguration gefunden wird. Ferner kann die Blockchain Softwareaktualisierungsserver informieren, die versuchen können, Referenzmessungen von einer zuverlässigen Quelle zu erhalten.

[0193] **Fig. 29** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **2900** zum sicheren Booten vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **2900** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0194] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0195] Der Massenspeicher **808** kann einen Vertrauens-Root-Messer bzw. Root-of-Trust Measurer (RTM) **2902** aufweisen, der das erste ladbare Objekt in einem System misst und verifizieren kann. Ein Vertrauens-Root-Speichermanager bzw. Root-of-Trust Storage Manager (RTS) **2904** kann von anderen Sicherheitssystemen in der Art des RTM **2902** und eines Vertrauens-Root-Reporters bzw. Root-of-Trust Reporter (RTR) **2906** erzeugte und verbrauchte Werte speichern. Der RTR **2906** kann Werte in der Vertrauens-Root für die Speicherung bzw. Root of Trust for Storage (RTS) **F08** und die Umgebung, welche die RTM **2404**, RTR **2406** und RTS **2408** implementiert, attestieren. Ein Vertrauens-Root-Archivierer bzw. Root-of-Trust Archiver (RTA) **2910** kann eine zusätzliche Archivierungsfähigkeit zu RTC-Knoten hinzufügen, welche nicht die Fähigkeiten zum Aufrechterhalten einer vollständigen Chain-Geschichte **2912** haben können.

[0196] Verschiedene historische Datenbanken können in der IoT-Vorrichtung **2900** aufrechterhalten werden, oder es kann auf anderen Mesh-Vorrichtungen **812** auf sie zugegriffen werden. Beispielsweise kann eine Blockchain-Logik **2914** eine Chain-Geschichte **2912** aufrechterhalten, welche die Blöcke der Blockchain aufweist. Ferner kann die Blockchain-Logik **2914** Änderungen an andere Mesh-Vorrichtungen **812** übergeben oder von anderen Mesh-Vorrichtungen **812** an der Blockchain vorgenommene Änderungen akzeptieren und validieren. Eine Whitelist-Geschichte **2916** kann die Whitelist und an den Whitelist-Elementen vorgenommene Änderungen halten, beispielsweise bevor die Änderungen an die Chain-Geschichte **2912** übergeben werden. Ferner kann die Whitelist-Geschichte **2916** andere Listen und Änderungen halten, wie die Blacklist und die nicht klassifizierte Liste. Eine Messgeschichte **2918** kann aktuelle und frühere Messungen, die während des Boot-Prozesses vorgenommen wurden, beispielsweise zum Vergleich mit den Bildern, halten.

[0197] **Fig. 30** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **3000**, das Code aufweist, um einen Prozessor **902** anzuweisen, sicher zu booten, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **3000** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf den Prozessor **802** und den Bus **806** aus **Fig. 8** beschrieben ausgewählt werden. Das nichtflüchtige maschinenlesbare Medium **3000** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0198] Das nichtflüchtige maschinenlesbare Medium **3000** kann einen Code **3002** aufweisen, um den Prozessor **902** anzuweisen, ein Codeobjekt zu messen, bevor das Codeobjekt ausgeführt wird. Ein Code **3004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Messung mit einer Liste bekannter guter Bilder zu vergleichen. Ein Code **3006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, das Objekt mit einer Liste bekannter schlechter Bilder zu vergleichen. Ein Code **3008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, das Bild zu klassifizieren und ein Vertrauensniveau zu bestimmen, wodurch es dem Prozessor beispielsweise erlaubt wird, in eine vertrauenswürdige Ausführungsumgebung zu booten, es dem Prozessor erlaubt wird, in eine nicht vertrauenswürdige Umgebung zu booten, oder ein Booten blockiert wird und ein Seitenadministrator alarmiert wird. Ein Code **3010** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Blockchain **3014** zu unterhalten, beispielsweise unter anderem eine Transaktion an eine Chain-Geschichte zu übergeben, Transaktionsänderungen zu anderen IoT-Vorrichtungen weiterzuleiten oder

Änderungen von anderen IoT-Vorrichtungen zu validieren. Ein Code **3012** kann aufgenommen sein, um Vertrauens-Roots aufrechtzuerhalten, beispielsweise wie mit Bezug auf **Fig. 24** für die RTM **2404**, die RTR **2406**, die RTS **2408**, die RTC **2410** und die RTA **2414** beschrieben. Das maschinenlesbare Medium **3000** kann auch die Blockchain in der Art der Chain-Geschichte **2412** speichern, wie mit Bezug auf **Fig. 24** beschrieben.

[0199] **Fig. 31** ist eine Schemazeichnung **3102**, worin eine Zusammenarbeitsfähigkeit zwischen öffentlichen Domains **3102**, privaten Domains **3104** und öffentlich-privaten Domains **3106** dargestellt ist, gemäß einigen Ausführungsformen. Die Netztechnologie kann in einem kontinuierlichen Änderungszustand sein, wodurch ein Versuch, permanente Abbildungen zu erhalten, unmöglich gemacht wird. Dementsprechend können IoT-Vorrichtungen die Backbone-Ressourcen in der Art von Domain-Nameservern (DNS) verwenden, um Pakete zwischen Domains zu senden. Die Pakete können durch den Internet-Backbone, die als Router **3108** dargestellt sind, zwischen den Domains **3102**, **3104** und **3106** geleitet werden.

[0200] Bei einigen Aspekten stellen die Router **3108** die Randverbindungen bereit, welche die Domains miteinander koppeln. Wie hier beschrieben, kann eine beliebige Anzahl von Diensten an den Rändern der Domains **3102**, **3104** und **3106** bereitgestellt werden, um die Zwischenverbindbarkeit zu verbessern. Beispielsweise können Verbindungen zwischen der öffentlichen Domain **3102** und den privaten Domains **3104** unter anderem Gelegenheiten für Mikrobezahungen für den Domain-Zugriff, zu einer expliziten Erlaubnis und Verfolgung für den Domain-Zugriff und für die Trennung öffentlichen und privaten Verkehrs bereitstellen. Ähnlich können Verbindungen zwischen der öffentlichen Domain **3102** und der öffentlich-privaten Domain **3106** unter anderem Gelegenheiten für Dienste in der Art eines zeitbasierten Leasings, Ressourcenmarktplätze und verteilter Identitätsserver bereitstellen. Verbindungen zwischen den privaten Domains **3104** und den öffentlich-privaten Domains **3106** können unter anderem Gelegenheiten für Inline-Dienst-Zwischenverbindungen, eine verhaltensbasierte Gefahrenanalyse und einen Herkunftsbeweis bereitstellen.

[0201] **Fig. 32** ist eine Schemazeichnung der Zusammenarbeitsfähigkeit zwischen einem heterogenen Netz **3200** aus festverdrahteten Netzen **3202** und Drahtlosnetzen **3204** und **3206** gemäß einigen Ausführungsformen. Die Drahtlosnetze **3204** und **3206** können durch Vorrichtungen im festverdrahteten Netz **3202** kommunikativ gekoppelt sein. Dies stellt Gelegenheiten für Effizienzverbesserungen bei Kommunikationen zwischen Vorrichtungen in den Drahtlosnetzen **3204** und **3206** sowie Verbesserungen bei Kommunikationen zwischen Vorrichtungen in einem Drahtlosnetz **3204** oder **3206** und einer Vorrichtung im festverdrahteten Netz **3202** bereit. Beispielsweise kann eine Edge-Vorrichtung **3208**, die ein erstes Drahtlosnetz **3204** mit dem festverdrahteten Netz **3202** koppelt, eine Daten-zu-Informationen-Transformation zur Verringerung der Größe des Nutzinhalts bereitstellen. Ferner kann die Edge-Vorrichtung **3208** ein Permissionierungssystem aufweisen, das Pakete vom ersten Drahtlosnetz **3204** durchlässt, während es nicht erlaubte Pakete blockiert. Das Permissionierungssystem kann Systeme zur Vornahme von Mikrozahlungen aufweisen, um zu ermöglichen, dass sich die Informationen über das festverdrahtete Netz **3202** bewegen. Beispielsweise kann das erste Drahtlosnetz **3204** ein Bodenfeuchtigkeitssensorfeld an einem landwirtschaftlichen Ort sein. Die Mitteilungsfrequenz kann von der Änderungsrate abhängen, wodurch die Kosten infolge der Notwendigkeit ansteigen können, Bandbreite zu kaufen, um der höchsten Mitteilungsrates zu entsprechen. Demgemäß kann ein Mikrozahlungssystem die Kosten verringern, indem es ermöglicht wird, dass Transaktionen nach Bedarf bezahlt werden.

[0202] **Fig. 33** ist eine Schemazeichnung eines Inline-Routingsystems **3300**, das zwei verschiedene Fog- oder Cloud-Entitäten in der Art einer Cloud A **3302** mit einer Cloud B **3304** verbindet, gemäß einigen Ausführungsformen. Das Inline-Routing kann eine IoT-Vorrichtungen **3306** als Kanäle zwischen mehreren Quellen und Zielen verwenden, wo die kombinierte Wirkung der als Inline-Router wirkenden IoT-Vorrichtungen **3306** etwas bildet, was gegenwärtig als Gateway bezeichnet werden kann.

[0203] Das Inline-Routing zwischen verbundenen IoT-Vorrichtungen **3306** in einem Netz kann unter Verwendung einer Stack-Popping-Technik ausgeführt werden, die an Schicht 3 und Schicht 4 eines Netzstapels anschließt. Diese Technik kann die Inline-Latenz und den Bedarf an einer anwendungsspezifischen Verkehrsbehandlung verringern. Die Technik kann direkt in ein Platinenunterstützungspaket bzw. Board Support Package (BSP) für verschiedene Vorrichtungen in der Art von Vorrichtungen der MCU-Klasse aufgenommen werden.

[0204] Untersysteme können in die IoT-Vorrichtungen **3306** aufgenommen werden, um die Routing-Funktion auszuführen, beispielsweise ein Anwendungs-/Dienstmanager **3308** zur Verbindung mit Anwendungen und zur Behandlung des Netzstapels für Kommunikationen zwischen den IoT-Vorrichtungen **3306**. Ein Zahlungs-/Kreditmanager **3310** kann Mikrozahlungen für den Transaktionszugriff auf andere Netze und andere Zwecke behandeln. Beispielsweise kann der Zahlungs-/Kreditmanager **3310** Systeme bezahlen, um eine Informationsübertragung über ein Netz zu ermöglichen oder den Zugang zu Informationen zu ermöglichen, wie lokale Wet-

terinformationen, Verkehrsflussmuster und dergleichen. Ein Dienstbarkeitssystem **3312** kann eine Netzschicht steuern, die zur Bereitstellung von Zugangsrechten verwendet wird, um zu ermöglichen, dass Verkehr durch eine bestimmte IoT-Vorrichtung **3306** fließt.

[0205] Ein Netzstapel **3314** ist dafür ausgelegt, die verschiedenen Schichten des Kommunikationsstapels bereitzustellen. Es können zusätzliche Schichten bereitgestellt werden, um die hier beschriebene Funktionalität zu implementieren. Beispielsweise kann das Dienstbarkeitssystem **3312** eine Dienstbarkeitsschicht steuern, wie hier beschrieben. Ein Zeitsteuerungs-Untersystem **3316** kann verwendet werden, um den Kommunikationen eine Zeitsteuerungsfunktionalität hinzuzufügen. Beispielsweise können Kommunikationen über eine Dienstbarkeitsschicht während eines begrenzten Zeitfensters für einen bestimmten Satz von Berechtigungsnachweisen erlaubt werden. Nachdem die Zeit verstrichen ist, können neue Berechtigungsnachweise oder weitere Bezahlungen erforderlich sein, um die Kommunikationen erneut zu öffnen. Das System kann ein Stromversorgungs-Untersystem **3318** aufweisen, um der Vorrichtung Strom zuzuführen.

[0206] Fig. 34 ist eine Schemazeichnung **3400** eines Inline-Routings, worin ein implizites Durchlass-Routing durch eine IoT-Vorrichtung **3306** dargestellt ist, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 33 beschrieben wurden. Die IoT-Vorrichtung **3306** kann als eine Edge-Vorrichtung zwischen zwei Domains wirken, beispielsweise die Kommunikationen zwischen verschiedenen Protokollen übersetzen.

[0207] Eine erste Entität **3402** kann mit der IoT-Vorrichtung **3306** kommunizieren, beispielsweise unter Verwendung eines ersten Protokolls **3404**. Bei diesem Beispiel beschränkt die IoT-Vorrichtung **3306** den Verkehr nicht, sondern leitet ihn ohne weitere Erlaubnisse durch. Die IoT-Vorrichtung **3306** kann die Pakete in ein zweites Protokoll **3406** übersetzen, wenn sie zur zweiten Entität **3408** weiter gesendet werden.

[0208] Fig. 35 ist eine Schemazeichnung eines expliziten permissionierten Routens durch eine IoT-Vorrichtung **3306** gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 33 und Fig. 34 beschrieben wurden. Dieser Ansatz adressiert einen möglichen Sicherheitsmangel in Systemen, in denen das Durchlassen implizit ist. Wie mit Bezug auf Fig. 34 beschrieben, kann die IoT-Vorrichtung **3306** als eine Edge-Vorrichtung zwischen zwei Domains wirken, beispielsweise um die Kommunikationen zu übersetzen.

[0209] Eine erste Entität **3402** kommuniziert mit der IoT-Vorrichtung **3306** beispielsweise unter Verwendung eines ersten Protokolls **3404**. Bei diesem Beispiel kann die IoT-Vorrichtung **3306** Aufforderungen **3502** zu einer oder beiden Vorrichtungen übertragen, um Erlaubnisse zu bestimmen, bevor der Verkehr durchgeleitet wird. Wie hier verwendet, ist das permissionierte Durchleiten ein Verfahren zum Geltendmachen eines Rechts zum Durchleiten einer Inline-Zwischenverbindung auf der Grundlage einer Schlüssel- oder Token-Aufforderung, einer zeitlich begrenzten Dienstbarkeit, einer verhaltens- oder reputationsbasierten Erlaubnis, eines monetären oder anderen Austausches einer elektronischen Währung oder Kombinationen davon.

[0210] Beispielsweise kann die IoT-Vorrichtung **3306** einen Identifikationsschlüssel von einer Entität **3402** oder **3408** akzeptieren und auf eine Blockchain zugreifen, um zu bestätigen, dass die Vorrichtung autorisiert ist, mit der anderen Entität **3408** oder **3402** zu kommunizieren. Die IoT-Vorrichtung **3306** kann zulassen, dass Kommunikationen während eines vorgegebenen Zeitraums, beispielsweise 1 Sekunde (s), 500 Millisekunden (ms), 100 ms oder weniger, fort dauern. Die IoT-Vorrichtung **3306** kann eine Mikrozahlung fordern, um das Fortdauern der Kommunikationen zu erlauben. Ferner kann die IoT-Vorrichtung **3306** die Pakete in ein zweites Protokoll **3406** übersetzen, wenn sie zur zweiten Entität **3408** weiter gesendet werden.

[0211] Fig. 36 ist eine Schemazeichnung einer Dienstbarkeitsschicht **3602**, **3604** und **3606** für das Inline-Routing, welche für die Durchlassrichtlinienkontrolle verwendet wird. Gleich nummerierte Elemente entsprechen jenen, die anhand der Fig. 33 und Fig. 34 beschrieben wurden. Wie hier verwendet, beschreibt der Begriff Dienstbarkeit ein nicht besitzergreifendes Recht, Informationen durch einen Knoten zu leiten, ohne den Knoten zu besitzen. Der Prozess stellt einen Kanal für Kommunikationen über mehrere Knoten bereit, die der Verwendung ihrer Ressourcen zustimmen, ohne dass Zwischenknoten über das Eintreten oder Austreten von Paketen oder Inhalten, beispielsweise in einer Dienstbarkeitsschicht unterhalb der Anwendungsschicht, Bescheid wissen.

[0212] Die Dienstbarkeitsschicht **3602**, **3604** oder **3606** kann einen Teil eines Netzstapels oberhalb der Routing-Schicht **3608**, **3610** oder **3612** aufweisen. Eine Anwendung **3614** in einer ersten Entität **3402** kann ein Paket zu einer Anwendung **3616** in einer zweiten Entität **3404** senden. Das Paket durchläuft die Dienstbar-

keitsschicht **3608** in der ersten Entität **3402**, wo es mit den geeigneten Informationen, die durch eine zweite Dienstbarkeitsschicht **3604**, beispielsweise in der IoT-Vorrichtung 3306 zu übertragen sind, gepackt wird.

[0213] Sobald die Erlaubnis zur Übertragung von Paketen, beispielsweise auf der Grundlage einer Identität, einer Autorisierung, einer Zahlung oder der Zeit, durch ein Dienstbarkeitssystem bestätigt wurde, beispielsweise als ein im Speicher wirkender Codeblock, kann die Paketübertragung durch eine Dienstbarkeitsschicht **3604** in einem Netzstapel in der IoT-Vorrichtung **3306** stattfinden. Das Protokoll des Pakets kann in der Dienstbarkeitsschicht **3604** in das zweite Protokoll **3406** konvertiert werden. Demgemäß wird das Paket unterhalb der Anwendungsschicht **3618** in der IoT-Vorrichtung 3306 behandelt, die seinen Inhalt nicht kennt oder nicht einmal weiß, dass ein Paket durchgeleitet wurde.

[0214] Das Paket wird dann zur zweiten Entität **3408** weitergeleitet. In der Dienstbarkeitsschicht **3606** in der zweiten Entität **3408** werden die Dienstbarkeitsinformationen aus dem Paket entfernt, und das Paket wird zum Verbrauch durch eine Anwendung zur Anwendungsschicht **3616** weitergeleitet. Ein ähnlicher Prozess geschieht für von der zweiten Entität **3408** zur ersten Entität **3402** gesendete Pakete in entgegengesetzter Richtung.

[0215] Das schematische Diagramm **3600** zeigt eine einzige Zwischenentität, die IoT-Vorrichtung **3306**. Wie in Fig. 33 dargestellt ist, können jedoch mehrere Zwischenentitäten verwendet werden, um das Paket von der ersten Entität **3402** zur zweiten Entität **3408** zu übertragen.

[0216] Fig. 37 ist ein Leiterdiagramm, das ein beispielhaftes Verfahren **3700** zum expliziten Durchlass-Routing auf der Grundlage von Erlaubnissen zeigt, gemäß einigen Ausführungsformen. Das Verfahren **3700** aus Fig. 37 kann durch die mit Bezug auf Fig. 39 beschriebene IoT-Vorrichtung **3900** implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 33 beschrieben wurden. In diesem Fall beruhen die Erlaubnisse auf elektronischen Guthaben für Mikrozahlungen, ähnliche Erlaubnisse können jedoch auf Schlüsseln, Einheitsidentifikationen, Reputationsschlüsseln und dergleichen beruhen.

[0217] Beim Verfahren **3700** kann ein Dienstbarkeitssystem **3312** eine Routing-Anforderung **3702** zu einem Vorrichtungsregistrator **3704** senden, um festzustellen, ob ein Paket durch die Dienstbarkeit durchgeleitet werden kann. Der Vorrichtungsregistrator **3704** oder ein Treuhänder kann sich innerhalb des als expliziter Durchlassknoten wirkenden Knotens oder außerhalb als ein gemeinsamer Agent für eine Gruppe von Knoten befinden. Die Routing-Anforderung **3702** weist die Identifikation oder Schlüssel in Zusammenhang mit dem Antragsteller auf, welche vom Vorrichtungsregistrator **3704** verwendet werden können, um festzustellen, ob die Durchlassanforderung gewährt oder verweigert werden sollte.

[0218] Bei einem zahlungsbasierten Beispiel wird auch die Fähigkeit des Antragstellers festgestellt, zu zahlen. Dies kann beispielsweise durch den Vorrichtungsregistrator **3704** geschehen, indem er die Identifikation oder die Schlüssel für die Vorrichtung in einer Blockchain nachschlägt **3706**. Eine Token-Guthabenprüfung **3708** kann an Transaktionen in der Blockchain ausgeführt werden, um festzustellen, ob ausreichend Guthaben existiert, um die Transaktion zu bezahlen. Die Token-Guthabenprüfung **3708** kann auf einem in die Durchlassanforderung eingebetteten Token oder auf einem in einer Blockchain aufgezeichneten Guthabenbetrag beruhen.

[0219] Die Zulassung-oder-Verweigerung-Entscheidung **3710** kann dann auf der Grundlage des Guthabens, der Identifikation, von Schlüsseln oder Kombinationen davon vorgenommen werden. Eine Antwort **3712** kann dann zum Dienstbarkeitssystem **3312** gesendet werden, um es über das Ergebnis der Zulassung-oder-Verweigerung-Entscheidung **3710** zu informieren. Falls die Entscheidung darin bestand, die Paketdurchleitung zu verweigern, kann das Dienstbarkeitssystem **3312** das Paket ohne weitere Aktion löschen oder den Absender über die Verweigerung des Zugangs informieren sowie andere Ereignisse ausführen.

[0220] Falls die Zulassung-oder-Verweigerung-Entscheidung 3710 das Paket durchlassen soll, kann das Dienstbarkeitssystem **3312** das Paket **3714** zur Zielvorrichtung weiterleiten. Die Feststellung der Erlaubnis, beispielsweise der Mikrozahlung, kann mit der Durchleitung durch eine oder mehrere der IoT-Vorrichtungen 3306 oder nur durch Edge-Vorrichtungen in der Art von Routern, die verschiedene Domains koppeln, geschehen. Sobald das Paket weitergeleitet wurde, kann das Dienstbarkeitssystem **3312** eine Benachrichtigung **3716** zum Vorrichtungsregistrator **3704** senden, dass die Weiterleitung abgeschlossen ist. Der Vorrichtungsregistrator **3704** kann dann eine Mikrozahlung **3718** freigeben und die Zahlung **3720** zum Dienstbarkeitssystem **3312** senden. Falls ein Blockchain-Datensatz verwendet wird, kann der Vorrichtungsregistrator **3704** die Zahlung in einem Block aufzeichnen und den Block an die Blockchain übergeben, um die Änderung des verbleibenden Guthabenbetrags aufzuzeichnen.

[0221] Fig. 38 ist ein Leiterdiagramm eines beispielhaften Verfahrens 3800 eines zeitlich begrenzten Mietansatzes für ein explizites Durchlassen gemäß einigen Ausführungsformen. Das Verfahren 3800 aus Fig. 38 kann durch die mit Bezug auf Fig. 39 beschriebene IoT-Vorrichtung 3900 implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 33 und Fig. 37 beschrieben wurden. Bei diesem Beispiel empfängt der Vorrichtungsregistrator 3704 in Schritt 3802 eine Durchlassanforderung, welche die Identifikation oder Schlüssel in Zusammenhang mit dem Antragsteller und eine angeforderte Durchlassdauer aufweist. Die Berechnungsnachweise und die Dauer können vom Registrator verwendet werden, um festzustellen, ob die Anforderung gewährt oder verweigert werden sollte. Dies kann beispielsweise geschehen, indem der Vorrichtungsregistrator 3704 veranlasst wird, die Identifikation und Schlüssel und die erlaubte Kommunikationsdauer für die Vorrichtung in einer Blockchain nachzuschlagen 3804.

[0222] In Schritt 3806 wird die zulässige Dauer für die Kommunikationen bestimmt. Falls die angeforderte Durchlassdauer nicht unterstützt werden kann 3808, wird in einer Antwort 3810 eine maximal zulässige Durchlassdauer spezifiziert. Sobald das Paket weitergeleitet wurde 3812, kann das Dienstbarkeitssystem 3312 eine Benachrichtigung 3814 zum Vorrichtungsregistrator 3704 senden, dass die Weiterleitung abgeschlossen ist. Der Vorrichtungsregistrator 3704 kann dann feststellen, ob die Mietzeit verstrichen ist 3816. Falls dies der Fall ist, macht der Vorrichtungsregistrator 3704 die Miete in Schritt 3818 ungültig und sendet in Schritt 3820 eine Benachrichtigung 3820 zum Dienstbarkeitssystem 3312, um es über das Auslaufen der Miete zu informieren.

[0223] Fig. 39 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 3900 zur Erzeugung von Koalitionsgruppen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3, Fig. 8, Fig. 33 und Fig. 38 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 3900 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0224] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0225] Der Massenspeicher 808 kann einen Anwendungs-/Dienstmanager 3308 aufweisen, der eine Anwendungsbetriebsumgebung in der Art eines Betriebssystems und einer Programmierschnittstelle zur Übermittlung von Paketen zu anderen Mesh-Vorrichtungen 812, Vorrichtungen in der Cloud 302 und dergleichen bereitstellen kann. Ein Zahlungs-/Kreditmanager 3310 kann Mikrozahlungen für Kommunikationen behandeln, wie mit Bezug auf Fig. 33 beschrieben. Ein Dienstbarkeitssystem 3312 kann eine Dienstbarkeitsschicht steuern, um Zugriffsrechte für die Paketübertragung bereitzustellen, wie mit Bezug auf die Fig. 37 und Fig. 38 beschrieben. Ein Netzstapel 3314 kann die Netzschichten für die Verpackung und die Übertragung von Paketen bereitstellen. Der Netzstapel 3314 kann eine Dienstbarkeit aufweisen, wie beispielsweise mit Bezug auf Fig. 36 beschrieben. Ein Zeitsteuerungssystem 3316 kann aufgenommen sein, um eine Zeitsteuerungsfunktionalität für Kommunikationen bereitzustellen, beispielsweise zum Herunterzählen der Zeit, bevor eine Miete ausläuft. Eine Blockchain-Logik 3902 kann aufgenommen sein, um auf eine Blockchain von Kommunikationserlaubnissen, -zahlungen oder -guthaben sowie anderer Elemente zuzugreifen und zu unterhalten.

[0226] Fig. 40 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums 4000, das Code aufweist, um einen Prozessor 902 anzuweisen, Kommunikationen zwischen Vorrichtungen durch Dienstbarkeiten zu übertragen. Der Prozessor 902 kann über einen Bus 904 auf das nichtflüchtige maschinenlesbare Medium 4000 zugreifen. Der Prozessor 902 und der Bus 904 können ähnlich wie der Prozessor 902 und der Bus 904, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium 4000 kann für den Massenspeicher 808 aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0227] Das nichtflüchtige maschinenlesbare Medium 4000 kann einen Code 4002 aufweisen, um den Prozessor 902 anzuweisen, eine Routing-Anforderung zu einem Vorrichtungsregistrator zu senden, um festzustellen, ob die Kommunikationen zulässig sind. Ein Code 4004 kann aufgenommen sein, um den Prozessor 902 anzuweisen, ein Paket zwischen Vorrichtungen auf der Grundlage der Reputation, einer Erlaubnis, von Mikrozahlungen und dergleichen weiterzuleiten. Der Code 4004 kann den Prozessor 902 anweisen, das Paket in ein neues Protokoll zu übersetzen, beispielsweise wenn in eine neue Domain eingetreten wird.

[0228] Ein Code **4006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Abschlussbenachrichtigung zum Vorrichtungsregistrator zu senden, um ihn zu informieren, dass Kommunikationen in der Art einer Gruppe eine Nachricht bildender Pakete übertragen wurden. Ein Code **4008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Zahlung vom Vorrichtungsregistrator für die Kommunikationen zu akzeptieren, die beispielsweise vom Vorrichtungsregistrator auf den Abschluss der Übertragung wartend zurückgehalten wurde.

[0229] Weil eine IoT-Vorrichtung 3306 (**Fig. 33**) als Vorrichtungsregistrator wirken kann, kann ein Code **4010** aufgenommen werden, um den Prozessor **902** anzuweisen, einen Antragsteller nachzuschlagen, beispielsweise durch Zugriff auf eine Blockchain, um Erlaubnisse und Identitäten festzustellen. Ein Code **4012** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Guthaben oder eine Zahlung zu prüfen, um festzustellen, ob sie für die Kommunikationen ausreicht. Der Code **4012** kann den Prozessor **902** anweisen, eine zulässige Kommunikationszeit zu berechnen. Ein Code **4014** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Erlaubnisentscheidung zu einem Dienstbarkeitssystem **3312** zu senden, wodurch es dem Dienstbarkeitssystem **3312** erlaubt wird, das Paket oder die Kommunikation zu übertragen oder zu löschen. Ein Code **4016** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine zeitbasierte Miete ungültig zu machen, beispielsweise falls die letzte abgeschlossene Kommunikation die zulässige Mietzeit überschritten hat. Der Code **4016** kann den Prozessor **902** anweisen, eine Nachricht zum Dienstbarkeitssystem **3312** zu senden, wodurch die Miete ungültig gemacht wird.

[0230] Die Verfolgung der Vorrichtungen, die ein Frame durchlässt, kann ein erhöhtes Sicherheitsniveau bereitstellen, weil Kommunikationen von einer Quellvorrichtung zu einer Zielvorrichtung verschiedene Domains durchlaufen. Dies kann dabei helfen, Angriffe zu verhindern, die Quellvorrichtungen imitieren, wie unter anderem Mittelsmannangriffe bzw. Man-in-the-Middle-Angriffe. Bei einem Mittelsmannangriff fängt eine Vorrichtung für eine andere Vorrichtung vorgesehene Kommunikationen ab und ändert die Kommunikationen, um die Sicherheit zu beeinträchtigen.

[0231] **Fig. 41** ist eine Schemazeichnung der Verwendung einer Frame-Struktur zur Übertragung von Herkunftsnachweis- bzw. Proof-of-Provenance(PoP)-Informationen durch Vorrichtungen in einem Netz gemäß einigen Ausführungsformen. Die Vorrichtungen können IoT-Vorrichtungen, Internetvorrichtungen, Edge-Vorrichtungen oder Kombinationen davon einschließen. Der Herkunftsnachweis stellt eine Verfolgbarkeit von Netzwerkverkehr durch eine Mehrverbindungskette bereit. Das PoP-Verfahren erzeugt ein Verkehrs-Audit-Trail und kann, wie hier beschrieben, zur Identifikation und zum Blockieren eines Mittelsmannangriffs verwendet werden. Der PoP kann verwendet werden, wenn eine Entität 1 **4102** Kommunikationen mit einer anderen Entität 4 **4104** einleitet, beispielsweise durch Senden eines Pakets **4106**. Das Paket **4106** kann eine Anzahl von Platzhaltern **4108** zum Identifizieren von Vorrichtungen, die das Paket **4106** durchlaufen hat, aufweisen.

[0232] Bei diesem Beispiel kann das Paket **4106** zu einer ersten Vorrichtung A **4110** oder einem Router in einer ersten Nachricht **4112** gesendet werden. Die Vorrichtung A **4110** kann das Paket **4106** modifizieren, um den Verlauf des Pakets durch die Vorrichtung A **4110** anzugeben, beispielsweise durch Schreiben **4114** eines Durchleitungscode PoP1 **4116** in die Platzhalter **4108**. Die Erzeugung des Durchleitungscode wird weiter mit Bezug auf **Fig. 42** erörtert.

[0233] Das Paket **4106** kann dann in einer zweiten Nachricht **4120** zu einer anderen Vorrichtung, der Entität 2 **4118** weitergeleitet werden. Die zweite Nachricht **4120** kann sich in einem von jenem der ersten Nachricht **4112** verschiedenen Protokoll befinden, beispielsweise in einer Vorrichtung A **4110** in das zweite Protokoll übersetzt werden. Die Entität 2 **4118** kann das Paket in einer dritten Nachricht **4124** zu einer anderen Vorrichtung B **4122** weiterleiten und das Paket auch in ein neues Protokoll übersetzen.

[0234] Die Vorrichtung B **4122** kann das Paket **4106** modifizieren, um das Durchleiten des Pakets **4106** durch die Vorrichtung B **4122** anzugeben, beispielsweise durch Schreiben **4126** eines zweiten Durchleitungscode PoP2 **4128** in die Platzhalter **4108**. Die Vorrichtung B **4122** kann das Paket dann in einer anderen Nachricht **4132** zu einer anderen Vorrichtung, der Entität 3 **4130**, weiterleiten. Bei diesem Beispiel kann die Entität 3 **4130** das Paket **4106** in das anfängliche Protokoll zurück übersetzen, bevor es in einer Nachricht **4136** zur Vorrichtung C **4134** weitergeleitet wird.

[0235] Die Vorrichtung C **4134** kann das Paket **4106** modifizieren, um das Durchleiten des Pakets **4106** durch die Vorrichtung C **4122** anzugeben, beispielsweise durch Schreiben **4138** eines dritten Durchleitungscode, PoP3 **4140**, in die Platzhalter **4108**. Die Vorrichtung C **4134** kann das Paket dann in einer anderen Nachricht

4142 zu einer anderen Vorrichtung, der Entität **4 4140**, weiterleiten. Die Vorrichtung C 4134 kann das Paket **4106** in ein anderes Protokoll übersetzen oder dies nicht tun, bevor es zur Entität **4 4140** weitergeleitet wird.

[0236] Jeder Knoten in der Kette kann nur Kenntnis über den vorhergehenden Knoten und den nächsten Knoten in der Kette haben. Demgemäß stellt die PoP-Sequenz im Paket **4106** einen Pfad von Vorrichtungen bereit, die das Paket **4106** durchlaufen hat. Bei einigen Beispielen werden die Platzhalter **4108** nicht explizit verwendet, wobei die PoP-Codes in das Paket **4106** eingebracht oder daran angehängt werden.

[0237] **Fig. 42** ist ein schematisches Diagramm **4200** einer Prozedur, die zur Erzeugung eines PoP-Durchleitungscodes oder -Schlüssels verwendet werden können, gemäß einigen Ausführungsformen. Für ein Ursprungspaket **4204** kann eine PoP-Sequenz **4206** durch Ausführen einer Transformation an einer Seed-Byte-Sequenz erzeugt werden. Die Seed-Byte-Sequenz kann durch einen Zufallszahlengenerator auf der Grundlage einer Vorrichtungskennung oder durch eine Anzahl anderer Techniken erzeugt werden. Der Transformationsprozess kann eine Exklusiv-ODER-Operation (XOR) oder ein anderes binäres Berechnungsverfahren einschließen.

[0238] Ein Knoten in der Kette empfängt die PoP-Sequenz und behandelt sie als Eingangsschlüssel **4208**. Der Knoten führt dann eine Transformation **4210** unter Verwendung der eingehenden Eingangssequenz aus, um seine eigene PoP-Sequenz **4212** zu erzeugen. Beispielsweise kann der Eingangsschlüssel **4208** unter Verwendung einer Exklusiv-ODER-Funktion **4216** mit einem Vorrichtungsschlüssel **4214** kombiniert werden, um die PoP-Sequenz **4212** zu erzeugen. Der Vorrichtungsschlüssel **4214** kann ein in der Vorrichtung gespeicherter Herstellerschlüssel, eine Vorrichtungskennung, ein Zufallszahlengenerator auf der Grundlage einer Vorrichtungskennung und dergleichen sein. Die PoP-Sequenz **4212** kann dann als PoP-Schlüssel **4218** zum Paket **4204** hinzugefügt werden. Die gleiche Sequenz kann als Ausgangsschlüssel **4220** hinzugefügt werden, der von der nächsten Stufe des PoP zur Erzeugung des nächsten PoP-Schlüssels verwendet wird. Bei einigen Beispielen kann der Ausgangsschlüssel **4220** unbenutzt gelassen werden und kann der PoP-Schlüssel **4218** selbst für die nächste Stufe verwendet werden. Eine Nullsequenz oder ein anderes Standardsequenzformat kann verwendet werden, um anzugeben, dass eine PoP-Transformation von der vorhergehenden Stufe nicht ausgeführt wurde.

[0239] **Fig. 43** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **4300** zur Erzeugung eines PoP-Schlüssels. Das Verfahren **4300** aus **Fig. 43** kann durch die mit Bezug auf **Fig. 45** beschriebene IoT-Vorrichtung **4500** implementiert werden. Block **4302** repräsentiert beispielsweise den Fall, in dem eine Vorrichtung von einer anderen Vorrichtung ein Paket empfängt, das zur Übertragung zu einer dritten Vorrichtung vorgesehen ist. In Block **4304** wird der Eingangsschlüssel beispielsweise durch Lesen des von der letzten PoP-Stufe erzeugten PoP-Schlüssels erhalten. In Block **4306** wird ein anderer PoP-Schlüssel berechnet, beispielsweise unter Verwendung einer XOR-Funktion des vorhergehenden PoP-Schlüssels und des Vorrichtungsschlüssels in der Art des privaten Schlüssels oder des gemeinsamen Schlüssels, der vom Knoten verwendet wird.

[0240] In Block **4308** wird der neu erzeugte PoP-Schlüssel an das Paket angehängt, wobei jegliche zuvor erzeugte PoP-Schlüssel intakt gelassen werden. In Block **4310** wird das Paket zur nächsten Vorrichtung weitergeleitet. In Block **4312** wird festgestellt, ob das Paket das Ziel erreicht hat. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **4302** zurück, um den Prozess am nächsten Knoten fortzusetzen. Falls das Paket das Ziel erreicht hat, endet das Verfahren **4300**. Es sei bemerkt, dass nicht alle Vorrichtungen einen PoP-Schlüssel in das Paket einfügen oder an dieses anhängen müssen. Einige Vorrichtungen in der Art der Entität **3 4130**, wie mit Bezug auf **Fig. 41** beschrieben, können Pakete weiterleiten, ohne einen PoP-Schlüssel in das Paket einzutragen. Diese Vorrichtungen können generische Router und andere Vorrichtungen innerhalb einer einzelnen Domain sein.

[0241] **Fig. 44** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **4400** zur Überprüfung der PoP-Schlüssel in einem Paket gemäß einigen Ausführungsformen. Der Zielknoten am Ende einer kontinuierlichen PoP-Kette empfängt eine gültige Sequenz von PoP-Schlüsseln, falls das Paket eine vollständige Kette durchlaufen hat. Vorrichtungsschlüssel von den Knoten in der Kette können für den Zielknoten oder eine andere Überprüfungsentität über eine Datenbank, eine Blockchain oder einen anderen Mechanismus verfügbar sein. Die Vorrichtungsschlüssel können verwendet werden, um die vollständige Kette unter Verwendung eines Sequenztransformationsprozesses zu überprüfen, wenn Kenntnis über das Eintreten und PoP-Schlüssel für alle Knoten besteht. Dieser Prozess kann verwendet werden, um festzustellen, ob und wo eine Unterbrechung in der PoP-Kette aufgetreten ist, wenn die iterative Überprüfungsberechnung zu einem PoP-Wert führt, der nicht mit der mitgeteilten PoP-Sequenz übereinstimmt.

[0242] Block **4402** repräsentiert beispielsweise den Fall, in dem eine Validierungsvorrichtung ein Paket empfängt. In Block **4404** wird die verkettete Liste von PoP-Schlüsseln aus dem zu überprüfenden Paket extrahiert. In Block **4406** können die PoP-Schlüssel tokenisiert werden, um die Zeichenkette in die einzelnen Tokens/Schlüssel zu segmentieren. In Block **4408** kann ein aktueller PoP-Schlüssel überprüft werden. In Block **4410** kann der aktuelle PoP-Schlüssel unter Verwendung eines XOR-Prozesses mit einem gemeinsamen geheimen Schlüssel kombiniert werden. In Block **4412** kann das XOR-Ergebnis mit dem nächsten Token verglichen werden, welches der Eingangsschlüssel für den nächsten Knoten auf dem Weg ist. Falls das XOR-Ergebnis und der nächste Eingangsschlüssel nicht übereinstimmen, kann der Prozessfluss mit Block **4416** fortgesetzt werden, um eine Fehlerbedingung mitzuteilen.

[0243] Falls es in Block **4412** eine Übereinstimmung gibt, wird in Block **4414** festgestellt, ob alle Schlüssel in der PoP-Sequenz getestet wurden. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **4408** zurück, um den nächsten Schlüssel zu testen. Falls alle Schlüssel getestet wurden, wird in Block **4416** ein erfolgreiches Ergebnis mitgeteilt.

[0244] **Fig. 45** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **4500** zur Verfolgung eines Herkunftsnachweises in Paketen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **4500** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0245] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0246] Der Massenspeicher **808** kann einen Kommunikator **4502** aufweisen, der Pakete von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** akzeptiert und die Pakete zu anderen Mesh-Vorrichtungen **812**, Vorrichtungen in der Cloud **302** und dergleichen weiterleitet. Der Kommunikator **4502** kann andere Funktionen ausführen in der Art einer Übersetzung von Paketen zwischen Protokollen, eines Akzeptierens von Mikrozahlungen und dergleichen. Ferner kann der Kommunikator **4502** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0247] Eine Eingangsschlüssel-Berechnungseinrichtung **4504** kann verwendet werden, um den Eingangsschlüssel zu bestimmen. Falls das Paket einen angehängten PoP-Schlüssel aufweist, kann dieser gelesen und als Eingangsschlüssel identifiziert werden. Falls die Eingangsschlüssel-Berechnungseinrichtung **4504** ein Nullzeichen oder ein anderes Zeichen, das angibt, dass die IoT-Vorrichtung **4500** die erste Vorrichtung in einer Kette sein kann, findet, kann die IoT-Vorrichtung einen Zufallszahlengenerator verwenden, um einen Eingangsschlüssel zu berechnen. Dieser kann auf einem Vorrichtungsschlüssel **4506** oder einem anderen Seed beruhen. Der Vorrichtungsschlüssel **4506** kann ein von einem Hersteller gespeicherter Wert sein oder ein Schlüssel sein, der erzeugt und der IoT-Vorrichtung **4500** zur sicheren Kommunikation in einer Gruppe zugewiesen wurde, wie unter anderem mit Bezug auf **Fig. 20** beschrieben wurde.

[0248] Eine PoP-Schlüsselberechnungseinrichtung **4508** kann verwendet werden, um PoP- oder Ausgangsschlüssel anhand des Vorrichtungsschlüssels **4506** und der Eingangsschlüssel zu berechnen. Dies kann so ausgeführt werden, wie mit Bezug auf **Fig. 42** erörtert wurde.

[0249] Ein Paketerzeuger **4510** kann verwendet werden, um das neue Paket zu bilden, beispielsweise sobald eine Übersetzung in verschiedene Protokolle ausgeführt wurde. Der Paketbilder kann den PoP-Schlüssel für den aktuellen Knoten nach jeglichen anderen Schlüsseln an das Ende des Pakets anhängen. Ferner kann der Paketbilder **4510** Teil einer Dienstbarkeitsschicht sein, wobei jegliche zur Übertragung des Pakets durch Dienstbarkeitsschichten in anderen Vorrichtungen verwendete Dienstbarkeitsinformationen hinzugefügt werden, wie mit Bezug auf **Fig. 36** beschrieben wurde.

[0250] Ein PoP-Sequenz-Überprüfer **4512** kann aufgenommen werden, um PoP-Sequenzen zu analysieren, wie mit Bezug auf **Fig. 44** beschrieben wurde. Die Überprüfung der PoP-Sequenz kann feststellen, dass die PoP-Schlüssel in einem Paket einem erlaubten Weg durch eine Kette von Vorrichtungen folgen und dass das Paket nicht durch eine illegitime Vorrichtung hindurchgelaufen ist. Hierdurch kann vor Mittelsmannangriffen geschützt werden.

[0251] Fig. 46 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **4600**, das Code aufweist, um einen Prozessor **902** anzuweisen, Kommunikationen zwischen Vorrichtungen durch Dienstbarkeiten zu übertragen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **4600** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **4600** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0252] Das nichtflüchtige maschinenlesbare Medium **4600** kann einen Code **4602** aufweisen, um den Prozessor **902** anzuweisen, ein empfangenes Paket zu lesen, um einen Eingangsschlüssel zu bestimmen, beispielsweise durch Erkennen eines angehängten PoP-Schlüssels. Ein Code **4604** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Vorrichtungsschlüssel zur Erzeugung anderer Schlüssel zu erzeugen oder auf andere Weise bereitzustellen. Der Vorrichtungsschlüssel kann unter anderem ein in der Vorrichtung gespeicherter Herstellerschlüssel oder ein vorrichtungsspezifischer Kommunikationsschlüssel, worauf von einer Blockchain zugegriffen wird, sein.

[0253] Ein Code **4606** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen neuen PoP-Schlüssel zu berechnen, beispielsweise anhand des Eingangsschlüssels und des Vorrichtungsschlüssels. Ein Code **4608** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den neuen PoP-Schlüssel an das Ende eines Pakets anzuhängen. Ein Code **4610** kann aufgenommen sein, um den Prozessor **902** anzuweisen, das Paket zur nächsten Vorrichtung zu senden.

[0254] Ein Code **4612** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Sequenz von an ein Paket angehängten PoP-Schlüsseln zu verifizieren, um sicherzustellen, dass das Paket durch legitime Vorrichtungen gelaufen ist. Ein Code **4614** kann aufgenommen sein, um den Prozessor **902** anzuweisen, über das Ergebnis der Verifikationsprozedur zu berichten, beispielsweise einer anderen Vorrichtung oder Vorrichtungsgruppe mitzuteilen, dass das Paket beeinträchtigt war. Ein Benutzer kann auch vor dem illegitimen Paket gewarnt werden.

[0255] Fig. 47 ist eine Schemazeichnung **4700** eines Beispiels eines Pakets **4702**, das Mikrozahlungsinformationen in einem Token-Bucket **4704** aufweist. Der Token-Bucket **4704** ist ein Bereich des Pakets, der eine Bitsequenz aufweist, die Tokens für Mikrozahlungen codiert. Die Verwendung des Token-Buckets **4704** kann es ermöglichen, dass ein Paket für die Durchleitung durch Systeme zahlt, die keinen Zugang zu einer gemeinsamen Datenbank oder Blockchain haben. Der Token-Bucket **4704** kann einen verschlüsselten Guthabenbetrag aufweisen, der durch eine oder mehrere übertragende Vorrichtungen dekrementiert werden kann. Bei einigen Beispielen kann ein Token im Token-Bucket eine oder mehrere mit einem ersten Schlüssel verschlüsselte Sequenzen sein, wobei eine oder mehrere der Edge-Vorrichtungen **4706** - **4710** einen Schlüssel aufweisen können, der die Token-Datei entschlüsselt und eine Sequenz oder ein Token entfernt.

[0256] Wenn das Paket **4702** an einer Edge-Vorrichtung **4706** empfangen wird, kann auf das Token-Bucket **4704** zugegriffen **4712** werden und kann er entschlüsselt werden, um den im Token-Bucket **4704** verbliebenen Guthabenbetrag zu bestimmen. Die Kosten für die Durchleitung durch die Edge-Vorrichtung **4706** werden abgezogen, und der Token-Bucket wird wieder verschlüsselt, und es wird darauf zugegriffen **4712**, um ihn im Frame zu speichern. Die Schritte werden wiederholt, wenn das Paket andere Edge-Vorrichtungen **4708** und **4710** durchläuft. Bei einem anderen Beispiel weist der Token-Bucket **4704** eine Anzahl identischer verschlüsselter Bitsequenzen auf. Jede Sequenz repräsentiert ein Token, das entfernt wird, wenn das Paket die Edge-Vorrichtung **4706** - **4710** durchläuft. Falls der Token-Bucket **4704** leer ist oder der Guthabenbetrag ungenügend ist, kann das Paket gelöscht oder zum Absender zurückgeführt werden.

[0257] Das Token-Bucket **4704** kann eine erhöhte Priorität für das Senden der Nachricht erlauben. Die Kosten können sich direkt auf die einem Paket gegebene Priorität beziehen. Ferner können die Tokens selbst diese Informationen übertragen. Falls ein Token beispielsweise einen ersten Guthabenbetrag hat, können die Edge-Vorrichtungen **4706** - **4710** das Paket nach einer normalen First-in-First-out-Sequenz übertragen. Falls ein Token jedoch einen höheren Guthabenbetrag hat, kann die Edge-Vorrichtung **4706** - **4710** das Paket vor anderen Paketen in der Warteschlange senden. Dies kann verwendet werden, um Kommunikationen von einer IoT-Vorrichtung zu priorisieren, beispielsweise um Alarmer mit einer hohen Priorität zu senden, während normale Datennachrichten mit einer niedrigeren Priorität gesendet werden.

[0258] Fig. 48 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **4800** zur Verwendung eines Token-Buckets zur Übertragung von Mikrozahlungen zu übertragenden Systemen gemäß einigen Ausführungs-

formen. Das Verfahren **4800** aus **Fig. 48** kann durch die mit Bezug auf **Fig. 49** beschriebene IoT-Vorrichtung **4900** implementiert werden. Block **4802** repräsentiert beispielsweise den Fall, in dem ein übertragendes System ein Paket empfängt. In Block **4804** werden die Paket-Frame-Metadaten geparkt, um beispielsweise den Token-Bucket zu lokalisieren. Falls in Block **4806** festgestellt wird, dass die geparkten Metadaten nicht korrekt sind, kann ein Absender alarmiert werden **4808**, beispielsweise durch Senden einer Fehlerhafte-Weiterleitung-Mitteilung. In Block **4810** wird festgestellt, ob fortgesetzt werden soll. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **4802** zurück.

[0259] In Block **4812** wird die Zahlung für den Abschluss der Übertragung berechnet. Dies kann durch einfaches Multiplizieren der Nutzinhaltsgröße mit der Gebühr pro weitergeleiteter Byte geschehen. Bei einigen Beispielen kann eine Prioritäts-Weiterleitung mit einer höheren Rate pro Byte verrechnet werden.

[0260] In Block **4814** wird die Zahlung für die Weiterleitung aus dem Token-Bucket extrahiert. Dies kann durch Dekrementieren des Token-Felds geschehen. Ein lokaler Zahlungsteilbetrag kann inkrementiert werden. Der lokale Zahlungsteilbetrag kann verwendet werden, um einen Zahlungsspeicher, beispielsweise in einer Datenbank, zu aktualisieren. Der Zahlungsspeicher kann eine Blockchain sein, welche die Zahlungsdatensätze aufweist.

[0261] In Block **4816** können die Metadaten für einen abgehenden Frame unter Verwendung des neuen Token-Felds vervollständigt werden. In Block **4818** kann das Paket dann vom Übertrager weitergeleitet werden. In Block **4820** wird festgestellt, ob fortgesetzt werden soll. Dies kann unter anderem auf der Grundlage davon geschehen, ob eine Sequenz von Kommunikationen weitergeht. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **4802** zurück, um fortzufahren.

[0262] **Fig. 49** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **4900** zur Verwendung von Token-Buckets zum Erleichtern von Zahlungen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **4900** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0263] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0264] Der Massenspeicher **808** kann einen Kommunikator **4902** aufweisen, der Pakete von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** akzeptiert und die Pakete zu anderen Mesh-Vorrichtungen **812**, Vorrichtungen in der Cloud **302** und dergleichen weiterleitet. Ein oder mehrere Pakete können eine Kombination zwischen Vorrichtungen bilden. Zusätzlich zu den mit Bezug auf **Fig. 49** beschriebenen Funktionen, wie mit Bezug auf **4502** beschrieben, kann der Kommunikator **4902** andere Funktionen ausführen, wie eine Übersetzung von Paketen zwischen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **4902** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0265] Ein Datenparser **4904** kann die Metadaten für ein empfangenes Paket parsen, um den Token-Bucket zu identifizieren. Dies kann das Entschlüsseln des Token-Buckets, beispielsweise unter Verwendung eines von der IoT-Vorrichtung **4900** gespeicherten öffentlichen oder privaten Schlüssels, einschließen. Eine Zahlungsberechnungseinrichtung **4906** kann verwendet werden, um die für die Übertragung einer Kommunikation erforderliche Zahlung zu berechnen. Die Berechnung kann einen Multiplikator aufweisen, der ein bestimmtes Prioritätsniveau berücksichtigt, wobei die Zahlung beispielsweise mit dem Multiplikator multipliziert wird, falls ein Kanal höherer Bandbreite ausgewählt wird. Eine Zahlungsextrahiereinrichtung **4908** kann die Zahlung vom Token-Bucket abziehen. Dies kann durch Abziehen eines Betrags von einem im Token-Bucket aufgezeichneten Guthabenbetrag geschehen. Bei einigen Beispielen kann der Token-Bucket eine Anzahl diskreter Tokens in der Art verschlüsselter Tokens aufweisen, die diskret entfernt werden können. Ein Frame-Erzeuger **4910** kann den Frame mit den Metadaten neu bilden, beispielsweise den Token-Bucket verschlüsseln und den Paketnutzinhalt und Metadaten zusammenstellen, um den Frame zu bilden. Der Kommunikator **4902** kann dann verwendet werden, um das zusammengestellte Paket zu übertragen. Bei einigen Beispielen kann eine Zahlungsakzeptiereinrichtung **4912** verwendet werden, um eine Zahlung von einer anderen Quelle zu akzeptieren, beispielsweise einer durch eine Bitsequenz im Token-Bucket identifizierten Blockchain.

[0266] Fig. 50 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **5000**, das Code zum Anweisen eines Prozessors **902** aufweist, Kommunikationen zwischen Vorrichtungen auf der Grundlage von Zahlungen von einem Token-Bucket zu übertragen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **5000** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **5000** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0267] Das nichtflüchtige maschinenlesbare Medium **5000** kann einen Code **5002** aufweisen, um den Prozessor **902** anzuweisen, Metadaten für einen Frame zu parsen, beispielsweise ein Token-Bucket aus den Metadaten zu extrahieren. Ein Code **5004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Zahlung zur Übertragung einer Nachricht zu berechnen. Die Zahlung kann beispielsweise auf der Größe der Nachricht und/oder der Priorität für die Nachrichtenübertragung beruhen. Ein Code **5006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Zahlung aus dem Token-Bucket zu extrahieren. Dies kann durch Verringern eines Guthabenbetrags im Token-Bucket oder durch Entfernen einer einem Token entsprechenden Bitsequenz aus dem Token-Bucket geschehen. Ein Code **5008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Frame zu aktualisieren, wobei beispielsweise der geänderte Guthabenbetrag für den Token-Bucket aufgenommen wird. Der Code **5008** kann auch den Prozessor **902** anweisen, den Token-Bucket zu verschlüsseln, bevor der Frame wieder zusammengestellt wird. Ein Code **5010** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Frame zu einem Ziel weiterzuleiten, beispielsweise durch Weitersenden des Frames zu einer nachfolgenden Vorrichtung mit einer das Endziel angegebenden Adresse.

[0268] Ein Code **5012** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Zahlung zu akzeptieren, um den Guthabenbetrag im Token-Bucket zu erhöhen. Ein Code **5014** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Priorität der Übertragung auf der Grundlage des bezahlten Betrags zu erhöhen. Falls der Token-Bucket beispielsweise diskrete Bitsequenzen für das Token verwendet, kann die übertragende Vorrichtung die Priorität zumindest teilweise auf der Grundlage der Beträge der einzelnen Tokens festlegen.

[0269] Fig. 51 ist eine Zeichnung einer heterogenen Netzinfrastruktur (hetnet-Infrastruktur) **5100**, welche an mehreren Stufen IP-Domains **5102** mit Nicht-IP-Domains **5104** verbindet, gemäß einigen Ausführungsformen. Bei diesem Beispiel steht eine LPWA-Domain **5106** in Kommunikation mit einem IEEE-802.15.4g-Mesh-Netz **5108**. Das Mesh-Netz **5108** steht in Kommunikation mit einem IEEE-802.15.4-Mesh-Netz **5110**. Ein Zugangspunkt **5112** stellt eine Kommunikationsverbindung **5114** (beispielsweise eine Ethernetverbindung) zu einem Kernnetz **5116** bereit. Das Kernnetz **5116** kann mit einem zweiten Zugangspunkt **5118** gekoppelt sein. Der zweite Zugangspunkt **5118** kann Kommunikationen einem zweiten LPWA-Netz **5120** bereitstellen. Die verschiedenen Permutationen von IP-Domains **5102** und Nicht-IP-Domains **5104**, gekoppelt mit der Bereitstellung von Unterstützung für eine erhebliche Anzahl von Vorrichtungen widerspricht zweckgebundenen Übersetzungsknoten. Ferner können dynamische Zwischenverbindungen für die Interaktion mit einer flüchtigen IoT-Infrastruktur nützlich sein, wobei Knoten Netzen beitreten können, Netze verlassen können und mobil sein können.

[0270] Beispielsweise können Daten von verschiedenen Domains durch die Verwendung einer Protokollverpackung effizient von einer Quelle zu einem Ziel übertragen werden. Bei diesem Beispiel kann ein Protokoll-Frame in einem ersten Protokoll in das Nutzinhaltsfeld eines Pakets in einem anderen Protokoll gepackt werden. Durch die Verwendung dieses Ansatzes bleiben beide Protokolle standardkompatibel. Wie ferner mit Bezug auf Fig. 52 beschrieben wird, kann ein an einem Gateway **5122** von einem Sensor **5124** in einer Fog-Vorrichtung in der Art eines LoRaWAN-Netzes **5106** empfangener LoRaWAN-Frame in einen IEEE-802.15.4-Frame gepackt werden, bevor er weitergesendet wird. Eine weitere Protokollverpackung kann am ersten Zugangspunkt **5112** ausgeführt werden. An der Zielvorrichtung, beispielsweise einem zweiten Zugangspunkt **5118**, kann die Verpackung entfernt werden und kann der Frame zu einer Zielvorrichtung **5124** weitergesendet werden. Dies kann für Kommunikationen in Fog-Vorrichtungen verwendet werden, die ferne Vorrichtungen aufweisen, auf welche über verschiedene Netze oder ein Kernnetz **5116** zugegriffen wird.

[0271] Fig. 52 ist eine Schemazeichnung **5200** einer Protokollverpackung, die für die Verpackung von Frames von einem Protokoll in ein anderes Protokoll verwendet wird, gemäß einigen Ausführungsformen. Beim in der Schemazeichnung **5200** dargestellten Beispiel wird ein LoRaWAN-Frame **5202** in das Nutzinhaltsfeld eines Pakets **5204** gepackt, das in einem IEEE-802.15.4-MAC-Frame **5206** enthalten ist. Der MAC-Frame **5206** hat die Header **5208**, die einen Übertragungs-Frame für die Übertragung zum Ziel bilden.

[0272] Eine beliebige Anzahl anderer Datenverbindungsschichten und Transporteinkapselungsziele kann durch die Verwendung dieses Verfahrens unterstützt werden. Beispielsweise können Frames, die dem Data-Over-Cable-Service-Interface-Specification(DOCSIS)-Protokoll folgen, in Pakete des AX.25-Protokolls eingekapselt werden. DOCSIS wird für Übertragungen mit hoher Datenrate über Kabel- und drahtlose Systeme verwendet. Es wird in erster Linie für Übertragungen mit hoher Datenrate in der Art von Breitbandinternet- und Fernsehdiensten verwendet. AX.25 wurde im Jahre 1996 mit einer Aktualisierung im Jahre 1998 von den Organisationen Tucson Amateur Packet Radio (TAPR) und American Radio Relay League (ARRL) entwickelt. AX.25 ist ein Datenverbindungsschichtprotokoll, dass vom AX.25-Protokoll abgeleitet ist und hauptsächlich für die Verwendung in gestörten schmalbandigen Drahtlosnetzen, in erster Linie in Amateurfunkbändern, ausgelegt ist.

[0273] **Fig. 53** ist eine Schemazeichnung **5300** einer Protokollverpackung, die zur Verpackung eines LoRaWAN-Frames **5302** in einem IEEE-802.11 (oder Wi-Fi®)-MAC-Frame **5304** verwendet wird, gemäß einigen Ausführungsformen. Wie in der Schemazeichnung **5300** dargestellt ist, kann der LoRaWAN-Frame **5302** als Netzdaten **5306** in den IEEE-802.11-MAC-Frame **5304** eingefügt werden. Wenn der IEEE-802.11-MAC-Frame **5304** das Ziel erreicht hat, in der Art eines zu einem LPWA-Netz führenden Gateways, kann der LoRaWAN-Frame **5302** aus dem IEEE-802.11-MAC-Frame **5304** lesen und zu einem Ziel übertragen werden.

[0274] **Fig. 54** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **5400** zur Protokollverpackung für die Übertragung eines Frames gemäß einigen Ausführungsformen. Das mit Bezug auf **Fig. 54** beschriebene Verfahren **5400** kann durch die mit Bezug auf **Fig. 55** beschriebene IoT-Vorrichtung **5500** implementiert werden. Block **5402** repräsentiert beispielsweise den Fall, in dem Daten bereit für das Aussenden sind.

[0275] In Block **5404** werden die verfügbaren Quell- und Zielprotokolle identifiziert. Ein Inventar verfügbarer Protokolle kann im Gateway oder Zugangspunkt erzeugt und gespeichert werden, und es werden die Eingangs- und Ausgangsprotokolle für die Protokollverpackung identifiziert. Die Nutzinhaltsgrößen und -randbedingungen in Zusammenhang mit jedem Protokoll, beispielsweise die erforderlichen Frame-Feld-Informationen in der Art von Adressen, Flags und dergleichen, werden identifiziert.

[0276] In Block **5406** werden die Nutzinhaltsrandbedingungen in Bezug auf den Nutzinhalt, beispielsweise den zu übertragenden Source-Frame, geprüft. In Block **5408** wird festgestellt, ob ein Quell-Frame in den Nutzinhalt des Zielprotokolls passt. Falls dies nicht der Fall ist, wird der Nutzinhalt in Block **5410** in mehrere Nutzinhalte fragmentiert. Dies kann beispielsweise durch Zerlegen des Nutzinhalts in N Bytesequenzen geschehen. Jede Bytesequenz kann in einen getrennten Zielprotokoll-Frame gegeben werden, und die Paketsequenzen werden nummeriert.

[0277] In Block **5414** werden die Nutzinhalts- und Vorrichtungsmetadaten in das Zielfeld geschrieben. In Block **5416** wird der Frame ans Ziel abgesendet. In Block **5418** wird festgestellt, ob alle Fragmente der Daten verarbeitet wurden. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **5414** zurück, um das nächste Fragment zu schreiben und zu senden.

[0278] In Block **5420** wird festgestellt, ob mehr Daten zu senden sind, beispielsweise ob ein anderer Eingangs-Frame empfangen wurde. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **5404** zurück, um den nächsten Frame zu verarbeiten. Falls dies nicht der Fall ist, endet das Verfahren **5400** in Block **5422**, wobei die Vorrichtung an diesem Punkt auf den Empfang eines anderen Frames wartet.

[0279] **Fig. 55** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **5500** zur Verpackung von Frames in einem ersten Protokoll in Frames eines anderen Protokolls vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **5500** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0280] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0281] Der Massenspeicher **808** kann einen Kommunikator **5502** aufweisen, der Frames von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** akzeptiert und die Frames zu anderen Mesh-Vorrichtungen

812, Vorrichtungen in der Cloud **302** und dergleichen weiterleitet. Zusätzlich zu den mit Bezug auf **Fig. 55** beschriebenen Funktionen, wie mit Bezug auf **4502** beschrieben, kann der Kommunikator **5502** andere Funktionen ausführen, wie eine Übersetzung von Frames zwischen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **5502** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0282] Ein Protokollbibliothekserzeuger **5504** kann feststellen, welche Protokolle verfügbar sind, und eine Protokollbibliothek **5506** bilden, welche die Protokolle und Formate für die jeweiligen speichert. Die Formate können Randbedingungen in der Art der Datenfeldlänge und dergleichen aufweisen, welche verwendet werden können, um festzustellen, wie der Frame zu formatieren ist, beispielsweise den Eingangs-Frame in Fragmente zur Übertragung in mehrere Ausgangs-Frames aufteilen.

[0283] Ein Frame-Analysator **5508** kann verwendet werden, um den von der sendenden Vorrichtung empfangenen Eingangs-Frame **5510** zu analysieren, um die Länge, das Verpackungsprotokoll und andere Randbedingungen zu bestimmen. Ein Frame-Erzeuger **5512** kann einen Ausgangs-Frame **5514** unter Verwendung der bestimmten Randbedingungen bilden. Beispielsweise kann der Frame-Erzeuger **5512** mehrere Ausgangs-Frames **5514** bilden, falls der Eingangs-Frame **5510** zu groß ist, um in das Nutzinhaltsfeld für den Ausgangs-Frame **5514** zu passen. Sobald der Ausgangs-Frame **5510** oder die Ausgangs-Frames gebildet wurden, kann der Kommunikator **5502** sie zum Ziel senden.

[0284] **Fig. 56** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **5600**, das Code aufweist, um einen Prozessor **902** anzuweisen, Frames in einem ersten Protokoll in Frames eines anderen Protokolls zu verpacken. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **5600** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **5600** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0285] Das nichtflüchtige maschinenlesbare Medium **5600** kann einen Code **5602** aufweisen, um den Prozessor **902** anzuweisen, ein Inventar möglicher Eingangs- und Ausgangsprotokolle zu erzeugen. Ein Code **5604** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Eingangs-Frame zu analysieren und die Größe und andere Randbedingungen zu bestimmen. Ein Code **5606** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Protokoll für einen Ausgangs-Frame zu bestimmen. Ein Code **5608** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Eingangs-Frame zu fragmentieren, falls er zu groß ist, um in das Nutzinhaltsfeld des Ausgangs-Frames zu passen. Die Codes **5606** können den Prozessor anweisen, jedes Fragment mit einer Sequenznummer für die korrekte Neuzusammensetzung an einem Ziel zu bezeichnen. Ein Code **5610** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Ausgangs-Frame zu schreiben, beispielsweise durch Eintragen des Eingangs-Frames oder eines Fragments des Eingangs-Frames mit einer zugeordneten Sequenznummer in das Nutzinhaltsfeld des Ausgangs-Frames. Ein Code **5612** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Ausgangs-Frame zu einem Ziel weiterzuleiten, beispielsweise durch Weitersenden des Frames zu einer nachfolgenden Vorrichtung mit einer das Endziel angegebenden Adresse.

[0286] Zusätzlich zur Verpackung von Frames verschiedener Protokolle innerhalb eines anderen Protokolls weist die zunehmende Verwendung komplexer Datenstrukturen in Kommunikationen mit einem geringen Overhead in der Art von LoRaWAN auf die Nützlichkeit eines höher entwickelten Framings hin. Eine neue Framing-Struktur wird mit Bezug auf **Fig. 57** beschrieben.

[0287] **Fig. 57** ist eine Zeichnung einer Nutzinhaltsstruktur **5700**, die als Nutzinhalt in einem Niederleistungs-Weitbereichs- bzw. Low-Power-Wide-Area(LPWA)-Frame **5702** in der Art eines LoRaWAN-Frames verwendet werden kann, gemäß einigen Ausführungsformen. Die Nutzinhaltsstruktur **5700** kann für mehrmodige Daten verwendet werden, einschließlich beispielsweise Informationen von einer oder mehreren Quellen, die aggregiert, fragmentiert, verschachtelt oder auf andere Weise gebildet werden können. Diese Typen mehrmodiger Daten können häufig zu Datengrößen führen, die mehr als einen LPWA-Frame **5702** zur Absendung zum vorgesehenen Ziel erfordern.

[0288] Die Spezifikation der Nutzinhaltsstruktur **5700** stellt einen minimalen Frame-Overhead bereit. Der Frame-Overhead kann wie in Tabelle 1 beschrieben sein.

Tabelle 1: Gesamter Frame-Overhead: 10 Bytes

Nutzzinhaltsstyp	4 Bits
Reserviert (rsvd)	1 Bit
Reserviert (rsvd)	1 Bit
Reserviert (rsvd)	1 Bit
Verschlüsselung (encr)	1 Bit
Vorrichtungskennung	4 Bytes
Stapelgröße	2 Bytes
Sequenznummer	2 Bytes
Länge	1 Byte
Nutzzinhalt	N-10 Bytes (Feld variabler Länge)

[0289] Bei einigen Aspekten identifiziert die Nutzzinhaltsstypkennung den Typ in der Nutzzinhaltsstruktur **5700** übertragener Daten unter anderem in der Art eines Bilds, eines 24-Bit-GPS-Berichts oder eines 3x2-Byte-Sensorberichts. Beispiele von Werten, die für die Nutzzinhaltsstypkennung verwendet werden können, sind in Tabelle 2 angegeben. Weil es sich hierbei um ein Vier-Bit-Feld handelt, können 15 mögliche Kennungen verwendet werden.

Tabelle 2: Nutzzinhaltsstypkennungen

Wert	Nutzzinhaltsstyp
0x0	Ungültig
0xA	Bild (JPEG)
0xB	Bild (PNG)
0x1	GPS-Bericht (24-Bit-Format)
0x2	GPS-Bericht (IEEE-754-Format)
0x3	3x2-Byte-Sensor (Temperatur/Druck/Regenfall)
0x4	Batterie + 2-Byte-Breite/Länge- GPS-Bericht

[0290] Bei einigen Aspekten kann der Frame-Overhead ein Verschlüsselungs-Hinweiszeichen aufweisen, um anzugeben, ob der Nutzzinhalt verschlüsselt ist, beispielsweise Encr = 0x1 für verschlüsselt und Encr = 0x0 für nicht verschlüsselt. Die Vorrichtungskennung kann eine Zeichenkette (beispielsweise eine Vier-Byte-Zeichenkette) aufweisen, welche die eindeutige Kennung für die sendende Vorrichtung angibt. Hierbei kann es sich unter anderem um die unter Verwendung der hier beschriebenen Blockchain-Verfahren zugewiesene Kennung oder um einen vom Hersteller zugewiesenen Namen handeln. Das Vorrichtungskennungsfeld kann beispielsweise verwendet werden, um eine Anzahl von Endpunktkennungen pro Funkkopf zu unterstützen. Bei einigen Beispielen kann eine Zwei-Byte-Kennung verwendet werden, um unter der Annahme, dass die Kennung null ungültig ist, bis zu 65535 Endpunktkennungen pro Funkkopf zu unterstützen.

[0291] Die Stapelgröße gibt die Anzahl der in eine einzelne Nachricht aufgenommenen Nutzzinhalte an. Die Sequenznummer gibt die Position der bestimmten Nutzzinhaltsstruktur **5700** in einer Sequenz zur Neuzusammensetzung an. Die Länge des Nutzzinhaltsfelds wird im Längenfeld übertragen. Die LoRa-Frame-Einkapselung kann eine Nachrichtenintegritätsprüfung bzw. Message Integrity Check (MIC) für Uplink-Nachrichten bereitstellen. Falls dies nicht der Fall ist, kann ein getrenntes MIC-Feld in den vorstehenden Frame **5702** aufgenommen werden.

[0292] Bei einigen Aspekten kann der Nutzzinhalt für längere Nachrichten über mehrere Frames fragmentiert werden müssen. Diese Frames brauchen nicht sequenziell in einem IoT-Netz gesendet zu werden, sondern können über parallele Funkkanäle gesendet werden, um die Übertragungszeit zu verringern und die Übertragungseffizienz zu verbessern. Diese als Netzgetrenntlagemultiplexierung bzw. Network Division Multiplexing (NDM) bezeichnete Technik ist ein Netzprotokollagnostisches Verfahren zum Zerlegen von Daten in mehre-

re unabhängige parallele Datenuntermengen und zum Befördern von ihnen über mehrere Netzwege vor der Rekombination am Ziel. Die Technik nutzt die Fähigkeit aus, mehrere Datenstreams parallel über verschiedene miteinander verbundene Netzwege und Infrastruktur zu überlagern, beispielsweise unter Verwendung unterschiedlicher Protokolle. NDM unterstützt mehrere Gleiches-Netz-Wege in der Art einer Anzahl von LP-WA-Wegen oder mehrere verschiedene Netzinfrastrukturwege in der Art einer Anzahl von LPWA-Wegen in Zusammenhang mit einer Anzahl von IEEE-802.15.4g-Strecken.

[0293] Die Ausrichtung des Datenstreams und verlustbehaftete Netzeigenschaften, die einen oder mehrere der NDM-Wege beeinflussen, können die Rekombination am Ziel beeinträchtigen. Diese Probleme können jedoch durch die Integration adaptiver verzögerungs-/störungstoleranter Protokolle in der Art des Licklider-Übertragungsprotokolls bzw. Licklider Transmission Protocol (LTP) verringert werden, das als Konvergenzschicht für ein fehlertolerantes Protokoll in der Art des Bundle-Protokolls (BP), das in <http://www.rfc-editor.org/rfc/pdf/rfc5050.txt.pdf> (zuletzt am 25 August 2016 zugegriffen) beschrieben ist, verwendet werden kann.

[0294] Beim LTP können Daten als wichtig und weniger wichtig identifiziert werden. Wichtige Daten in der Art von Headern müssen genau übertragen und empfangsbestätigt werden, bevor die Daten durch die sendende Einheit verworfen werden. Weniger wichtige Daten in der Art eines einzelnen Pixels in einem Bild können anhand der Übertragung wiederherstellbar oder weniger wichtig sein, falls sie verloren gehen, so dass die Daten verworfen werden können, nachdem sie gesendet wurden. Infolge der sehr hohen Latenz werden vor einleitenden Kommunikationen keine Verhandlungen ausgeführt. Die folgenden Figuren beschreiben die Übertragung von Daten unter Verwendung der NDM-Technik.

[0295] Fig. 58 ist eine Schemazeichnung 5800 von Übertragungsdaten-Nutzzinhalt 5802, der vor dem Senden in eine Anzahl von Sub-Blöcken 5804 fragmentiert wird, gemäß einigen Ausführungsformen. Jeder der Sub-Blöcke 5804 kann eine variable Länge L_i aufweisen. Die Sub-Blöcke 5804 können N Netzwegen zugewiesen werden, wo ein oder mehrere Netzprotokolle oder PHYs verwendet werden.

[0296] Sub-Header-Daten können dann an jeden Sub-Block 5804 angehängt werden, wobei beispielsweise jeder Frame in einem Daten-Sub-Stream mit Header-Daten verkapselt werden kann, um die Sub-Stream-Reihenfolge im Haupt-Datennutzzinhalt anzugeben, um die Rekombination am Ziel zu unterstützen. Die Headerdaten können auch eine maximale Durchleitzeit, beispielsweise eine Lebenszeit sowie eine Prioritätsreihenfolge- und Neuversuchrichtlinie aufweisen. Sobald die Headerinformationen angehängt wurden, können die Sub-Blöcke 5804 in die verschiedenen für die Übertragung verwendeten Protokoll-Frames gepackt werden, wie beispielsweise vorstehend mit Bezug auf Fig. 54 beschrieben wurde.

[0297] Fig. 59 ist eine Schemazeichnung 5900 einer NDM-Seriell-zu-Parallel-Übertragung gemäß einigen Ausführungsformen. Die Sub-Blöcke 5804 können für einen synchronisierten Absendemodus zur Zeit T_{tx} 5902 abgesendet werden. In einigen Fällen können die Sub-Blöcke 5804 in einem synchronisierten Pro-Weg-Absendemodus zu Zeiten $T_i \times [i]$ gesendet werden, wobei jedes $[i]$ einen Übertragungsweg repräsentiert.

[0298] Fig. 60 ist eine Schemazeichnung 6000 des Empfangs der Sub-Blöcke 5804 gemäß einigen Ausführungsformen. Am Ziel oder einem anderen vorgesehenen Abgriffspunkt können die Sub-Blöcke 5804 in einer anderen Reihenfolge als in der, in der sie abgesendet wurden, und mit anderen Zeitversätzen empfangen werden. Die Sub-Blöcke 5804 können entpackt werden, falls sie in einem anderen Protokoll verpackt sind, und analysiert werden, um die Anzahl und die Reihenfolge der Sub-Blöcke 5804, die für die Nachricht erwartet werden, zu bestimmen. Die Sub-Blöcke 5804 können dann gehalten werden, bis alle Teile empfangen wurden, bevor sie zum TX-Datennutzzinhalt 5802 aus Fig. 58 neu zusammengesetzt werden.

[0299] Fig. 61 ist eine Schemazeichnung 6100 der Rekombination der Sub-Blöcke 5804 zur Bildung des empfangenen Datennutzzinhalts 6102 gemäß einigen Ausführungsformen. Die Umwandlung von der parallelen in die serielle Blockform geschieht unter Verwendung von Headerdaten in jedem Sub-Block 5804, um die Blockreihenfolge zu identifizieren. Abhängig von den Anweisungen im Header kann die Neuzusammensetzung selbst dann ausgeführt werden, wenn Sub-Blöcke 6102 fehlen.

[0300] Fig. 62 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 6200 zum Fragmentieren und Absenden eines Nutzzinhalts über mehrere parallele Kommunikationskanäle gemäß einigen Ausführungsformen. Das Verfahren 6200 aus Fig. 62 kann durch die mit Bezug auf Fig. 64 beschriebene IoT-Vorrichtung 6400 implementiert werden. Das Verfahren 6200 beginnt in Block 6202, wenn der Datennutzzinhalt beispielsweise für die Übertragung bereit ist.

[0301] In Block **6204** werden die verfügbaren Netzwerke und zugeordneten Protokolle entdeckt. Diese können in einer Bibliothek in der IoT-Vorrichtung gespeichert sein und periodisch getestet werden, um zu bestätigen, dass die Verbindbarkeit noch vorhanden ist. Die entdeckten Informationen können auch Daten über die zulässigen Nutzinhaltsgroßen für die unterstützten Protokolle, Übertragungsgeschwindigkeiten und dergleichen aufweisen.

[0302] In Block **6206** wird festgestellt, ob ein Nutzinhalt abzusenden ist. Dies kann geschehen, wenn ein Nutzinhalt bereit ist und die Verbindbarkeit mit einem oder mehreren Netzen vorhanden ist.

[0303] In Block **6208** wird der Nutzinhalt fragmentiert, beispielsweise auf der Grundlage der verfügbaren Netzwerke und der maximal verfügbaren Nutzinhaltsgroßen, die von den zugeordneten Protokollen unterstützt werden. Die Fragmentierung kann andere Parameter der Kommunikationskanäle in der Art der Übertragungsraten, der Priorität und dergleichen berücksichtigen. Die Fragmentierung kann die mit Bezug auf die **Fig. 57 - Fig. 61** beschriebenen Sub-Blöcke **5804** bilden.

[0304] In Block **6210** werden die Fragmente indexiert. Dies kann durch Zuweisen von Sequenznummern zu den Fragmenten und anschließendes Bilden von Fragment-Headern, welche die Sequenznummern aufweisen, geschehen. Die Fragmente werden mit den Headern verkettet, um die Sub-Blöcke zu bilden. Die einzelnen Sub-Blöcke können dann für die Übertragung über die verschiedenen Wege in die Protokoll-Frames gepackt werden, wie beispielsweise mit Bezug auf die **Fig. 52 - Fig. 56** beschrieben wurde. In Block **6212** werden die Sub-Blöcke oder Fragmente des Nutzinhalts entlang den verschiedenen Übertragungswegen gesendet.

[0305] **Fig. 63** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **6300** zum Empfangen und Rekombinieren unter Verwendung einer NDM-Technik gesendeter Pakete gemäß einigen Ausführungsformen. Das Verfahren **6300** aus **Fig. 63** kann durch die mit Bezug auf **Fig. 64** beschriebene IoT-Vorrichtung **6400** implementiert werden. Das Verfahren **6300** beginnt in Block **6302**, wenn Fragmente über eine Anzahl verschiedener Kommunikationskanäle von einer sendenden Vorrichtung empfangen werden.

[0306] Dies kann durch Erkennen des Empfangs von Frames auf eine Anzahl verschiedener Wege und in einer Anzahl verschiedener Protokolle geschehen. Falls die Frames in verschiedenen Protokollen verpackt sind, wird der Nutzinhalt aus dem Protokoll-Frame entfernt und analysiert. Die Analyse parst den Frame und entfernt den Header. Das Fragment des Nutzinhalts wird in einen lokalen Speicher geschoben, und die Sequenznummer wird aufgezeichnet.

[0307] In Block **6304** wird festgestellt, ob alle Fragmente empfangen wurden. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **6302** zurück, um weiter auf Fragmente zu warten. Die IoT-Vorrichtung muss möglicherweise nicht warten, bis alle Fragmente empfangen wurden, bevor sie damit beginnt, die Daten zusammenzustellen. Beispielsweise könnte ein Befehl in einem der Fragmente angeben, dass ein fehlendes Fragment weniger wichtige Daten enthält und dass die Neuzusammensetzung nicht unterbrochen werden sollte.

[0308] In Block **6306** können die Fragmente umgeordnet und kombiniert werden. Beispielsweise kann jedes Fragment durch die Sequenznummer und die Länge an den neu zusammengesetzten Datennutzinhalt angehängt werden, um den empfangenen Datennutzinhalt zu bilden. In Block **6308** wird der rekombinierte Nutzinhalt beispielsweise an den Verbraucherprozess ausgegeben.

[0309] **Fig. 64** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **6400** zum Fragmentierung von Nutzgehalten für die Übertragung entlang mehreren parallelen Wegen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **6400** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0310] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0311] Der Massenspeicher **808** kann einen Kommunikator **6402** aufweisen, der Frames von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** über eine oder mehrere Kommunikationsstrecken, beispielsweise unter anderem durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine

NIC **816** sendet und empfängt. Zusätzlich zu den mit Bezug auf **Fig. 64** beschriebenen Funktionen, wie mit Bezug auf **4502** beschrieben, kann der Kommunikator **6402** andere Funktionen ausführen, wie eine Übersetzung von Frames zwischen Protokollen, eine Verpackung von Frames in anderen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **6402** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0312] Der Kommunikator **6402** kann von einem Netzentdecker **6504** zur Identifikation verfügbarer Netze und Protokolle für die Kommunikation zwischen der IoT-Vorrichtung **6400** und einer Zielvorrichtung verwendet werden. Der Netzentdecker **6504** kann eine Liste verfügbarer Netzkommunikationswege und Protokolle, die für parallele NDM-Kommunikationen zu verwenden sind, bilden und aufrechterhalten.

[0313] Ein Nutzinhalt **6406** kann von der IoT-Vorrichtung **6400** gebildet werden, beispielsweise anhand von den Sensoren **820** erhaltener Messungen. Bei einigen Beispielen kann der Nutzinhalt **6406** von einer anderen IoT-Vorrichtung in der Mesh-Vorrichtung **812** in der Art einer entfernteren Vorrichtung übergeben werden. Bei diesem Beispiel kann die IoT-Vorrichtung **6400** als Gateway zur Weiterleitung von Kommunikationen, einschließlich des Nutzinhalts, zu anderen Vorrichtungen arbeiten.

[0314] Ein Nutzinhaltsfragmentierer/-paketierer **6408** kann auf der Grundlage der Kommunikationsgeschwindigkeit, der Zuverlässigkeit, der Leistungsverfügbarkeit oder einer Anzahl anderer Faktoren und Kombinationen von Faktoren den Nutzinhalt und verfügbare Kommunikationskanäle analysieren, um die Kanalkombinationen zu bestimmen, die wahrscheinlich zu einer optimalen Kommunikation des Nutzinhalts führen. Der Nutzinhaltsfragmentierer/-paketierer **6408** kann dann den Nutzinhalt in Sub-Objekte zur Übertragung fragmentieren. Header und andere identifizierende und Sequenzinformationen können für die Übertragung angehängt werden. Abhängig von den ausgewählten Kommunikationen können die Sub-Objekte in die Datenfelder verschiedener Protokoll-Frames gepackt werden und dann vom Kommunikator **6402** über die ausgewählten Kommunikationskanäle gesendet werden.

[0315] Bei einigen Aspekten können die Kommunikationen bidirektional sein. Ein Nutzinhaltsempfänger/-analysator **6410** kann Frames von anderen Vorrichtungen empfangen, die Protokollverpackung entfernen und die Frames analysieren, um Nachrichten- und Sequenzinformationen zu identifizieren. Der Nutzinhaltsempfänger/-analysator **6410** kann feststellen, dass die Datenfelder empfangener Frames Sub-Objekte von Nutzinhalten sind. Der Nutzinhaltsempfänger/-analysator **6410** kann die Sub-Objekte und Sequenznummern speichern, bis verschiedene Bedingungen erfüllt sind, und dann die Sequenznummern und Speicherinformationen zu einem Nutzinhaltsdefragmentierer **6412** weiterleiten. Die Bedingungen können eine Feststellung, dass alle Sub-Objekte in einem Nutzinhalt empfangen wurden, oder eine Feststellung, dass jegliche fehlende Sub-Objekte in einer Sequenz weniger wichtige Daten aufweisen und dass die Zusammensetzung fortschreiten sollte, aufweisen.

[0316] Der Nutzinhaltsdefragmentierer **6412** kann die Nutzinhalte zum endgültigen Nutzinhaltsobjekt zusammenstellen, wie beispielsweise anhand **Fig. 63** erörtert. Der Nutzinhalt kann dann von der IoT-Vorrichtung **6400** verwendet oder zu einem Datenverbraucher weitergesendet werden.

[0317] **Fig. 65** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **6500**, das Code aufweist, um einen Prozessor **902** anzuweisen, Nutzinhalte zu fragmentieren und entlang mehreren parallelen Wegen zu übertragen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **6500** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **6500** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0318] Das nichtflüchtige maschinenlesbare Medium **6500** kann einen Code **6502** aufweisen, um den Prozessor **902** anzuweisen, verfügbare Netzwerke und -protokolle zu einer Empfangsvorrichtung zu entdecken. Ein Code **6504** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Nutzinhalt zu fragmentieren, so dass er in die Datenfelder von Frames für die Protokolle passt, die für die Kommunikationen ausgewählt wurden. Der Code **6504** kann den Prozessor **902** anweisen, Headerinformationen anzuhängen, welche unter anderem die Sequenznummer des Pakets aufweisen. Ein Code **6506** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Fragmente abhängig von den ausgewählten Kommunikationen in unterschiedliche Protokoll-Frames zu packen. Ein Code **6508** kann aufgenommen sein, um den Prozessor **902** anzuweisen,

die Frames in Richtung der Zielvorrichtung über die verschiedenen ausgewählten Kommunikationskanäle zu übertragen.

[0319] Ein Code **6510** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Fragmente zu empfangen, beispielsweise in Frames unterschiedlicher Protokolle. Ein Code **6512** kann aufgenommen sein, um die Nutzinhalt aus den verschiedenen Protokoll-Frames zu entpacken, und dann die Headerinformationen zu parsen, um einen Nutzinhalt und die Sequenznummer zu identifizieren. Der Code **6512** kann den Prozessor **902** anweisen, zu bestimmen, wann die Neuzusammenstellung des Nutzinhalts versucht werden sollte, beispielsweise bevor alle Fragmente empfangen wurden. Ein Code **6514** kann aufgenommen sein, um den Prozessor anzuweisen, den Nutzinhalt auf der Grundlage der Sequenznummer neu zusammenzustellen.

[0320] Die vorstehend beschriebenen Kommunikationstechniken können verwendet werden, um Kommunikationen mit fernen IoT-Vorrichtungen zu ermöglichen oder zu verbessern, beispielsweise Wettersensoren, industriellen Einheiten und dergleichen. Positionsbestimmungen für diese Vorrichtungen können ein Problem sein, insbesondere in Netzen, die batteriebetriebene Einheiten aufweisen. Techniken, die es ermöglichen, dass Vorrichtungen ihre Position anhand von anderen Vorrichtungen in einem Mesh-Netz übermittelten Informationen bestimmen, können es ermöglichen, dass zumindest ein Teil der Vorrichtungen Strom spart. Eine Technik, um dies auszuführen, wird anhand der **Fig. 66 - Fig. 70** beschrieben.

[0321] **Fig. 66** ist eine Schemazeichnung eines Überlagerungsbakensystems **6600**, bei dem ein Bakenknoten **6602** einer nahe gelegenen IoT-Vorrichtung **6606** eine Ortsnachricht **6604** bereitstellt, gemäß einigen Ausführungsformen. Für IoT-Installationen in Bereichen ohne eine Infrastruktur ist der einzelne IoT-Knoten oder Bakenknoten **6602** mit einem satellitenbasierten Positionsbestimmungsempfänger versehen und wirkt als Geolokationsbake zur Übermittlung von Orts- und Zeitdaten zu benachbarten verbundenen Knoten. Dies kann über mehrere heterogene Netze durch Senden eines Orts-Nutzinhalts in einem für jeden Kommunikationsstreckentyp geeigneten Frame als Teil der Nutzdaten geschehen. Der Bakenknoten **6602** kann eine IoT-Vorrichtung sein, die mit einem GPS-Modul in der Art eines Satellitenempfängers für den Empfang von Signalen vom Satellitensystem des globalen Positionsbestimmungssystems (GPS), vom globalen Navigationssatellitensystem bzw. Global Navigation Satellite System (GLONASS) oder von anderen globalen Navigationssatellitensystemen (GNSS) versehen ist.

[0322] Bei einigen Aspekten kann der Bakenknoten **6602** seine Position durch Erfassen eines Signals **6608** von drei oder mehr Satelliten **6610** des globalen Positionsbestimmungssystems bestimmen. Der Bakenknoten **6602** kann die von den Satelliten empfangenen Daten wandeln, beispielsweise aus Sätzen der National Marine Electronics Association (NMEA) in einen für die Übertragung geeigneten Datentyp. Es kann ein Orts-Nutzinhalt **6612** erzeugt werden, der Positionsdaten beispielsweise in einem IEEE754-Paketformat aufweist. Bei diesem Format können vier Bytes verwendet werden, um die Breite **6614** zu repräsentieren, vier Bytes verwendet werden, um die Länge **6616** zu repräsentieren, und vier Bytes ein angehängter Zeitstempel **6618** sein. Der Bakenknoten **6602** kann den Orts-Nutzinhalt **6612** zur Übertragung zu anderen Vorrichtungen als Ortsnachricht **6604** in einen Protokoll-Frame packen. Wie hier beschrieben, kann abhängig von den verfügbaren Kommunikationskanälen eine beliebige Anzahl von Protokollen verwendet werden.

[0323] In regelmäßigen Zeitintervallen oder ereignisbasiert oder auf der Grundlage anderer Auslöser überträgt der Bakenknoten **6602** die Ortsnachricht **6604**. IoT-Vorrichtungen oder andere Knoten innerhalb der Reichweite des Bakenknotens **6602**, beispielsweise innerhalb von einigen zehn bis einigen hundert Metern vom Bakenknoten **6602**, können die Ortsnachricht **6604** empfangen und den Geolokalisierungs-Nutzinhalt **6612** für ihre eigenen Nachrichten- oder andere Zwecke verwenden. Beispielsweise können Zeitkorrekturen für eine lokale Vorrichtung unter Verwendung des Zeitstempels **6618** vom Bakenknoten **6602** ausgeführt werden.

[0324] **Fig. 67** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **6700** zur Erzeugung eines Orts-Nutzinhalts gemäß einigen Ausführungsformen. Das Verfahren **6700** aus **Fig. 67** kann durch die mit Bezug auf **Fig. 69** beschriebene IoT-Vorrichtung **6900** implementiert werden. Block **6702** repräsentiert beispielsweise den Fall, in dem eine Vorrichtung mit Strom versorgt wird oder auf andere Weise angewiesen wird, den Geolokalisierungsprozess einzuleiten. In Block **6704** wird eine Positionsfeststellung erhalten. Es wird ein Befehl zu einem GPS-Modul gesendet, um eine Positionsfeststellung zu erhalten. In Block **6706** wird eine Wartezeit für eine erste Feststellung implementiert, beispielsweise 10 Sekunden, 30 Sekunden, 60 Sekunden oder länger. Nach Abschluss der Wartezeit wird in Block **6708** festgestellt, ob eine Feststellung erhalten wurde. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **6706** zurück, um auf ein anderes Inkrement zu warten. Falls eine Feststellung erhalten wurde, kehrt der Prozessfluss mit den Ortsdaten vom GPS-Modul zu Block **6704** zurück.

[0325] In Block **6710** werden die Ortsdaten geparkt. Dies kann in Block **6712** durch Extrahieren der Länge, der Breite, der Zeit und anderer Daten in der Art der Höhe implementiert werden. Die extrahierten Daten werden in einem lokalen Speicher **6714** gespeichert.

[0326] In Block **6716** wird ein Orts-Nutzzinhalt anhand der Daten im lokalen Speicher **6714** gebildet. Der Nutzzinhalt kann in Block **6718** durch Einfügen der Positionsbestimmung in ein Paket, beispielsweise unter Verwendung der IEEE754-Format-4-Byte-Repräsentation der Breiten- und Längenpositionsdaten, gebildet werden. In Block **6720** kann eine Bakenkennung in das Paket eingefügt werden, und in Block **6722** kann ein Zeitstempel angehängt werden. Der Zeitstempel kann von einer lokalen Uhr abgeleitet werden oder eine aus den Satellitendaten extrahierte Zeitdateneinheit sein. Der Nutzzinhalt kann für die Übertragung in einen Frame gepackt werden. Das Protokoll des Frames kann auf den zu verwendenden Kommunikationskanälen, unter anderem in der Art von Ethernet, LoRaWAN von 4G, beruhen.

[0327] In Block **6724** wird der Frame an die umgebenden IoT-Vorrichtungen ausgestrahlt. Dies kann mit einer Aktivierung eines Senders und der Übertragung des Frames als Nachricht über eine Funkübertragung einhergehen. Bei einigen Beispielen kann der Frame über ein festverdrahtetes Netz in der Art des Ethernets gesendet werden. Bei einigen Beispielen kann eine einfache Paketkonstruktion verwendet werden, beispielsweise durch Anhängen eines Headers an den Nutzzinhalt, um das Paket als Ortsinformationen zu identifizieren, und Aussenden des Pakets an umgebende Vorrichtungen ohne eine direkte oder zielgerichtete Kommunikation.

[0328] In Block **6726** kann eine Wartezeit implementiert werden, bevor der Prozess wiederholt wird. Dies kann durch Aktivieren eines Schlafbefehls für einen vorgegebenen Zeitraum geschehen. In Block **6728** wird festgestellt, ob der Orts-Bakenprozess fortgesetzt werden soll. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **6704** zurück, um die nächste Positionsbestimmung zu erhalten. Falls dies nicht der Fall ist, endet der Prozess in Block **6730**.

[0329] Fig. 68 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **6800** zum Parsen eines Frames, der einen Orts-Nutzzinhalt aufweist, gemäß einigen Ausführungsformen. Das Verfahren **6800** aus Fig. 67 kann durch die mit Bezug auf Fig. 69 beschriebene IoT-Vorrichtung **6900** implementiert werden. Block **6802** repräsentiert beispielsweise den Fall, in dem eine IoT-Vorrichtung einen Bakenknoten entdeckt. In Block **6804** wird ein Baken-Frame oder ein Ortspaket vom Bakenknoten empfangen. In Block **6806** wird die Bakenkennung geprüft, um die Identität des Bakenknotens zu bestätigen. In Block **6808** wird eine Frame-Integritätsprüfung ausgeführt, um festzustellen, ob der Frame oder das Ortspaket gültig ist. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **6804** zurück, um auf den Empfang eines anderen Frames oder Ortspakets zu warten.

[0330] Falls ein gültiger Frame oder ein gültiges Ortspaket empfangen wurde, werden in Block **6812** Positionsdaten, beispielsweise ein Orts-Nutzzinhalt, extrahiert. In Block **6812** kann der Orts-Nutzzinhalt geparkt werden. Dies kann durch Extrahieren der Breite und der Länge und der Höhe, falls enthalten, aus dem Nutzzinhalt in Block **6814** geschehen. Der Zeitstempel kann in Block **6818** extrahiert werden. Die Informationen können in einem lokalen Speicher **6816** gespeichert werden. Die IoT-Vorrichtung kann dann die Informationen aus dem lokalen Speicher **6816** verwenden, beispielsweise zur Nachrichtenübertragung, zur Synchronisation oder für andere Zwecke.

[0331] In Block **6820** wird festgestellt, ob ein anderer Baken-Frame empfangen wurde. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **6804** zurück, um diesen Frame zu verarbeiten. Falls dies nicht der Fall ist, endet der Prozessfluss in Block **6822**.

[0332] Bei dieser Technik benötigt nicht jeder Knoten einen zweckgebundenen GPS-Empfänger, wodurch Kosten und Batteriestrom gespart werden. In Fällen, in denen genaue Orts- oder Wegespunktinformationen pro Vorrichtung nicht benötigt werden, können hierdurch IoT-Vorrichtungen ausreichend Informationen bereitgestellt werden, um ihren Installationsbereich zu identifizieren und orts- und/oder zeitabhängige Aufgaben auszuführen.

[0333] Fig. 69 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem Bakenknoten **6900** vorhanden sein können, um ein Bakenknotensystem zum Teilen von Ortsdaten einzurichten, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für den Bakenknoten **6900** andere Komponenten ausgewählt und verwendet werden können als für jene, die für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **802** und andere hier erörterte IoT-Vorrichtungen ausgewählt wurden.

[0334] Bei diesem Beispiel kann die IoT-Vorrichtung **6900** ein GPS-Modul **6902** zum Empfangen und Verarbeiten von Satellitenpositionsdaten aufweisen. Das GPS-Modul **6902** kann in einer Anzahl miteinander verbundener Mesh-Vorrichtungen **812** enthalten sein, jedoch nur in einer oder einigen aktiviert sein. Dies kann es dem System ermöglichen, eine gewisse Orts- und Zeitredundanz zu haben, falls der Bakenknoten **6900** ausfällt, beispielsweise infolge einer schwachen Batterie.

[0335] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Bakenfunktion aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0336] Der Massenspeicher **808** kann einen Paketkommunikator **6904** aufweisen, der Frames oder Ortspakete, welche einen Orts-Nutzzinhalt enthalten, über eine oder mehrere Kommunikationsstrecken von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** unter anderem durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816** sendet und empfängt. Wie mit Bezug auf 4502 beschrieben, kann der Paketkommunikator **6904** andere Funktionen ausführen, wie das Verpacken von Orts-Nutzzinhalt in verschiedenen Protokollen, das Ausführen von Herkunftsnachweisprüfungen und dergleichen. Ferner kann der Kommunikator **6904** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0337] Ein GPS-Lokalisierer **6906** kann mit dem GPS-Modul **6902** kommunizieren, um Ortsinformationen zu erhalten. Der GPS-Lokalisierer **6906** kann das GPS-Modul **6906** einschalten oder ausschalten, beispielsweise um die Batterieverwendung zu steuern oder ein GPS-Modul **6906** zu starten, wenn eine andere Mesh-Vorrichtung **812**, die ein GPS-Modul aufweist, ausfällt. Der GPS-Lokalisierer **6906** kann die GPS-Ortsdaten vom GPS-Modul **6906** sammeln und speichern.

[0338] Ein Datenparser **6908** kann den GPS-Ort parsen, um die Breite, die Länge, die Zeit und andere Parameter in der Art der Höhe zu bestimmen. Die geparsten Daten können zur Weiterverwendung gespeichert werden.

[0339] Ein Nutzzinhaltserzeuger **6910** kann die geparsten Daten verwenden, um einen Orts-Nutzzinhalt zu erzeugen. Dies kann wie mit Bezug auf **Fig. 66** beschrieben geschehen. Der Nutzzinhalt kann durch den Nutzzinhaltserzeuger **6910** in einen Frame eines bestimmten Protokolltyps gepackt werden. Der Frame kann dann durch den Paketkommunikator **6904** gesendet werden.

[0340] Wie hier beschrieben, ist die IoT-Vorrichtung nicht auf eine Funktionsweise als Bakenknoten beschränkt, sondern kann auch Ortsdaten empfangen. Dies kann nützlich sein, wenn ein GPS-Modul **6902** ausfällt oder nicht in der Lage ist, eine Position zu bestimmen. Bei einigen Beispielen kann die IoT-Vorrichtung **6900** kein GPS-Modul **6902** aufweisen, sondern nur als Ortsverbraucher wirken.

[0341] Ein Frame-Validierer **6912** kann verwendet werden, um von einem Bakenknoten empfangene Frames zu validieren, um festzustellen, ob die Pakete mit einer Bakenkennung übereinstimmen und gültige Daten enthalten. Der Paketvalidierer **6912** kann gültige Pakete zurückweisen oder ignorieren oder eine Neusendeanforderung senden, beispielsweise falls die Bakenkennung korrekt war, der Frame oder das Ortspaket jedoch beschädigt war.

[0342] Eine Paketextrahiereinrichtung **6914** kann einen empfangenen Frame analysieren, um den Orts-Nutzzinhalt zu extrahieren. Hierbei können die Daten bestimmt werden, welche die Breiten-, Längen- und Zeitinformationen sowie andere Informationen in der Art der Höhe angeben.

[0343] **Fig. 70** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **7000**, das Code aufweist, um einen Prozessor **902** anzuweisen, Orts-Nutzzinhalt zu senden und zu empfangen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **7000** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **7000** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0344] Das nichtflüchtige maschinenlesbare Medium **7000** kann einen Code **7002** aufweisen, um den Prozessor **902** anzuweisen, Positionsdaten von GPS-Satelliten, beispielsweise unter Verwendung eines GPS-Moduls, zu erhalten. Ein Code **7004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Positionsdaten

vom GPS-Modul zu parsen, um getrennte Werte für die Breite, die Länge und die Zeit zu erhalten. Der Code **7004** kann den Prozessor **902** anweisen, andere Werte in der Art der Höhe anhand der Positionsdaten zu bestimmen. Ein Code **7006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Orts-Nutzinhalt zu bilden, der die Breite, die Länge und die Zeit aufweist. Der Code **7006** kann den Prozessor **902** anweisen, den Orts-Nutzinhalt in einem Frame eines bestimmten Protokolls zu verpacken. Ein Code **7008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Nutzdaten über verschiedene Kommunikationskanäle zu übertragen. Ein Code **7012** kann aufgenommen sein, um den Prozessor anzuweisen, Orts-Nutzdaten aus einem von einer Bake empfangenen gültigen Frame zu extrahieren und die Nutzdaten für andere Zwecke zu speichern.

[0345] Fast alle Verfahren zum Speichern und Ausliefern von Informationen über ein Netz verwenden ein Push- oder Pull-Verfahren. Push kann häufig mit der Ausstrahlung eines Gateways oder einer Basisstation zu allen angeschlossenen Basisknoten gleichgesetzt werden. Dieser Modelltyp wird häufig auch in Publish/Subscribe-Modellen verwendet, in denen Vorrichtungen Daten über Kanäle als Mittel zum Senden von Daten senden. Ferner verwenden die meisten Modelle einen zentralen Server, von dem aus Endpunkte Daten ausstrahlen (Push), oder einen Inhaltsserver, aus dem sie ziehen (Pull). Die mit Bezug auf die **Fig. 71** bis **Fig. 74** beschriebenen Techniken verwenden eine Kombination von Push und Pull zum Verteilen von Inhalt über Netze.

[0346] **Fig. 71** ist eine Schemazeichnung eines verteilten Inhaltsverteilungssystems **7100** für heterogene Netze gemäß einigen Ausführungsformen. Die Verwendung des verteilten Inhaltsverteilungssystems **7100** kann die Verteilung über heterogene Netze ermöglichen, die verlustbehaftet sein können oder eine intermittierende Verbindbarkeit haben können. Ferner ermöglicht es die zustandslose Verteilung von Daten. Eine oder mehrere IoT-Vorrichtungen oder Knoten **7102** im verteilten Inhaltsverteilungssystem **7100** haben einen Datenmanager **7104**, der für die Behandlung von Daten am Knoten verantwortlich ist. Der Datenmanager **7104** weist eine Anzahl von Untersystemen auf, einschließlich eines Datenklassifizierers **7106**, der die einwärts gerichteten Daten **7108** und auswärts gerichteten Daten **7110**, die durch das verteilte Inhaltsverteilungssystem **7100** laufen, klassifizieren kann. Er verwendet drei Hauptklassifikationen für die Daten, nämlich einwärts gerichtet, auswärts gerichtet und Cache.

[0347] Bei einigen Aspekten ist ein Daten-Mapper **7112** für die Abbildung der klassifizierten Daten auf einen physikalischen Ort im System verantwortlich. Der Daten-Mapper **7112** kann einen Algorithmus in der Art einer Hash-Funktion verwenden, um den optimalen Ort der Daten zu bestimmen. Der Datenklassifizierer **7106** kommuniziert mit einem Ressourcenmanager **7114**, um die Klassifikation für auswärts gerichtete und Cache-Daten zu bestimmen. Einwärts gerichtete Daten **7108** sind Daten, die dafür vorgesehen sind, vom Knoten selbst verbraucht zu werden. Der Daten-Mapper **7112** überträgt die Daten in ein Eingangsfach **7116**, und der Ressourcenmanager **7114** überwacht das Eingangsfach **7116** auf Änderungen oder Aktualisierungen.

[0348] Auswärts gerichtete Daten **7110** können von einem Knoten **7102** über einen Abstand von mehr als einem Hop geteilt werden, und sie werden vom Ressourcenmanager **7114** bestimmt. Die auswärts gerichteten Daten **7110** können in einem Ausgangsfach **7118** gespeichert werden. Der Ressourcenmanager **7114** berechnet die Anzahl der Hops durch Berechnen der aktuellen RessourcenVerfügbarkeit am Knoten **7102** in der Art der Stromverfügbarkeit und des Netzknotenzählwerts.

[0349] Gecachte Daten werden in einem Cache **7120** gespeichert, und es handelt sich dabei um Daten, die als für den Knoten **7102** nützlich bestimmt wurden. Ein Datengeschichtsaufzeichner **7122** kann Daten verfolgen, die sich in den Knoten **7102** hinein und aus diesem heraus bewegen, wie einwärts gerichtete und auswärts gerichtete Datenanforderungen. Ein Protokollmanager **7124** kann die für eingehende und abgehende Frames verwendeten Protokolle behandeln, beispielsweise auf der Grundlage der für die bestimmten Frames verwendeten Kommunikationskanäle. Ein Netzmanager **7126** kann Netzkommunikationen auf den verschiedenen Kommunikationskanälen behandeln, beispielsweise das Hosten des Netzstapels. Ein Kommunikationsmanager **7128** kann Physikalisches-Niveau- oder PHY-Operationen in der Art von Funkeinrichtungen, Netz-schnittstellen-Steuereinrichtungen und dergleichen behandeln.

[0350] **Fig. 72** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **7200** für eine dispergierte Inhaltsverteilung gemäß einigen Ausführungsformen. Das Verfahren **7200** aus **Fig. 72** kann durch die mit Bezug auf die **Fig. 71** und **Fig. 73** beschriebene IoT-Vorrichtung **7102** implementiert werden. In Block **7202** werden die Daten klassifiziert. Dies geschieht durch Klassifizieren eines oder mehrerer Bestandteile einwärts gerichteter und auswärts gerichteter Daten, die das System durchlaufen, beispielsweise als einwärts gerichtete Daten, auswärts gerichtete Daten und Cache-Daten.

[0351] In Block **7204** werden die klassifizierten Daten in den korrekten physikalischen Ort im System abgebildet. Beispielsweise kann dies, wie in Block **7206** angegeben, unter Verwendung eines Algorithmus zur Erzeugung eines Hash-Codes, der den Ort einwärts gerichteter Daten identifiziert, geschehen.

[0352] In Block **7208** wird festgestellt, ob die Daten einwärts gerichtet sind. Falls dies der Fall ist, werden die Daten lokal in Block **7210** gespeichert. In Block **7212** wird der Hash-Schlüssel geprüft. In Block **7214** wird festgestellt, ob der Hash-Schlüssel im lokalen Speicher **7216** ist. Falls dies nicht der Fall ist, wird in Block **7218** das neue Datenfragment lokal gespeichert. Der Prozessfluss kehrt dann zu Block **7202** zurück.

[0353] Falls in Block **7214** festgestellt wird, dass der Schlüssel im lokalen Speicher **7216** ist, wird in Block **7220** festgestellt, ob die Informationen ignoriert werden sollten, beispielsweise ob sie mit den vorhergehenden Informationen im lokalen Speicher **7216** identisch sind. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **7202** zurück. Falls dies nicht der Fall ist, werden die Daten im lokalen Speicher **7216** mit dem neuen Fragment aktualisiert, und der Prozessfluss kehrt zu Block **7202** zurück.

[0354] Falls in Block **7208** festgestellt wird, dass die Daten auswärts gerichtete Daten sind, wird in Block **7224** die maximale Anzahl von Hops berechnet. Diese wird als Lebensdauer bzw. Time-to-Live (TTL) bezeichnet und kann durch Berechnen der aktuellen Ressourcenverfügbarkeit am Knoten in der Art des verfügbaren Stroms und des Netzknotenzählwerts bestimmt werden. In Block **7226** werden die Daten zum Zielknoten abgesendet oder gepusht.

[0355] Ein Zielknoten kann auch Daten pullen, indem er Daten von einem Knoten eines Hops anfordert. Die Daten-Pull-Anforderung kann eine in Bezug auf den Hop-Zählwert, d. h. die Anzahl der Hops, die ein Paket ausführt, wenn es ein Netz durchläuft, wobei die TTL nach jedem Hop dekrementiert wird, gemessene TTL aufweisen. Wenn die TTL einen Null-Zählwert erreicht, wird das Datenfragment ungültig gemacht. Die TTL kann in einer absoluten Zeit gemessen werden, beispielsweise in Sekunden, Minuten oder Stunden, wobei das Datenfragment ungültig gemacht wird, wenn ein Zeitablauf auftritt. Falls er innerhalb des Zeitablaufs keine Pull-Anforderung erhält, kann er eine Anforderung zum Knoten pushen, die dann durch das System weitergeleitet werden kann.

[0356] In Block **7226** wird festgestellt, ob das verteilte Teilen von Inhalt fortgesetzt werden soll. Falls dies der Fall ist, wird der Prozessfluss in Block **7202** wiederaufgenommen.

[0357] Jeder Knoten kann im Datengeschichtsaufzeichner empfangene einwärts gerichtete und auswärts gerichtete Anforderungen verfolgen. Ein Cache-Fenster kann für alle Anforderungen aufrechterhalten werden. Die Frequenz kann durch eine Anzahl von Faktoren bestimmt werden, wie die Anzahl von Anforderungen über einen Zeitraum.

[0358] Die Vorrichtung behandelt auch selbstständig ihre Cache-Größe durch Anwenden eines Zählers und Zeitgebers, worauf zugegriffen wurde, um festzustellen, wie häufig auf die gecachten Daten zugegriffen wird. Falls häufig auf die Daten zugegriffen wird, kann sie den Cache vergrößern, und falls weniger häufig auf sie zugegriffen wird, kann sie den Cache verkleinern. Jeder Knoten stellt auch fest, ob er Daten über den Datenmanager pushen oder pullen kann.

[0359] **Fig. 73** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **7300** zur Implementation eines verteilten Inhaltsverteilungssystems vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3**, **Fig. 8** und **Fig. 71** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **7300** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0360] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0361] Der Massenspeicher **808** kann die mit Bezug auf **Fig. 71** beschriebenen Module aufweisen. Datenspeicher **7304** in der Art des Eingangsfachs **7116**, des Ausgangsfachs **7118**, des Cache **7120** und des Datengeschichtsaufzeichners **7122** können in den Massenspeicher **808** aufgenommen sein oder an anderen Stellen in der Art eines Speichers einer anderen Vorrichtung gespeichert sein.

[0362] Der Massenspeicher **808** kann einen Kommunikator **7302** aufweisen, der Pakete über eine oder mehrere Kommunikationsstrecken, unter anderem beispielsweise durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** sendet und Frames von diesen empfängt. Zusätzlich zu den mit Bezug auf **Fig. 73** beschriebenen Funktionen, wie mit Bezug auf **4502** aus **Fig. 45** beschrieben, kann der Kommunikator **7302** andere Funktionen ausführen, wie eine Übersetzung von Paketen zwischen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **7302** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0363] **Fig. 74** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **7400**, das Code aufweist, um einen Prozessor **902** anzuweisen, ein verteiltes Inhaltsverteilungssystem zu implementieren, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **7400** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **7400** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0364] Das nichtflüchtige maschinenlesbare Medium **7400** kann einen Code **7402** zum Klassifizieren der Daten, die das verteilte Inhaltsverteilungssystem durchlaufen, als einwärts gerichtete Daten, auswärts gerichtete Daten oder Cache-Daten aufweisen. Ein Code **7404** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die klassifizierten Daten auf einen physikalischen Ort am System abzubilden. Der Code **7404** kann den Prozessor **902** anweisen, den optimalen Ort der Daten zu bestimmen. Der Code **7406** kann den Prozessor **902** anweisen, eine Hash-Funktion der Daten zu berechnen. Ein Code **7408** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob der Hash-Schlüssel im lokalen Speicher ist.

[0365] Ein Code **7410** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein neues Datenfragment lokal zu speichern. Ein Code **7412** kann aufgenommen sein, um ein lokal gespeichertes Datenfragment zu aktualisieren. Ein Code **7414** kann aufgenommen sein, um den Prozessor anzuweisen, die Lebenszeit für ein Datenfragment, beispielsweise in der Anzahl der Hops vor der Löschung oder als Zeitdauer vor der Löschung oder als beides zu berechnen. Ein Code **7416** kann aufgenommen sein, um Daten zu anderen Knoten, beispielsweise in Frames, zu übertragen. Das Protokoll für die Frames kann auf der Grundlage des für das Senden der Frames verwendeten Kommunikationskanals ausgewählt werden.

[0366] **Fig. 75** ist eine Schemazeichnung eines Drahtlosspeichersystems **7500** gemäß einigen Ausführungsformen. Das Drahtlosspeichersystem **7500** verwendet einen Kommunikationskanal **7502** zwischen zwei oder mehr verbundenen Knoten in der Art eines Ursprungsknotens **7504** und eines Empfangsknotens **7506** als Speichermedium **7508**. Dies ist im Wesentlichen ein drahtloses Speichersystem mit sequenziellem Zugriff, wobei das Funksignal selbst als Speichermedium **7508** für die zwischen den Knoten **7504** und **7506** übertragenen Daten wirkt. Demgemäß können die Knoten **7504** und **7506** Speicherplatz gegen Kommunikationsbandbreite austauschen.

[0367] Im Drahtlosspeichersystem **7500** werden am Ursprungsknoten **7504** ankommende Daten **7510** in einer Schleife zurück übertragen **7512**, um sie zu einem anderen Knoten in der Art des Empfangsknotens **7506** zu senden. Die am Empfangsknoten **7506** ankommenden Daten **7514** werden dann in einer Schleife zurück übertragen **7516** und zum Ursprungsknoten **7504** zurück gesendet. Bei einigen Beispielen können mehrere Knoten eine Kette zum Empfangen und Senden der Daten bilden. Durch Wiederholen des Prozesses verbleiben Daten auf dem Weg, und der Kommunikationskanal **7502** wirkt als Speichermedium.

[0368] **Fig. 76** ist eine andere Schemazeichnung des Drahtlosspeichersystems **7500** gemäß verschiedenen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 75** beschrieben wurden. Bei diesem Beispiel sind der Netzstapel **7602** für den Ursprungsknoten **7504** und der Netzstapel **7604** für den Empfangsknoten **7506** dargestellt. Daten **7606**, die von der Dienstbarkeitsschicht **7608** im Ursprungsknoten **7504** ankommen, können getaggt, gesichert und zur Speicherung als Übertragung **7514** zum Empfangsknoten **7506** übertragen werden. Der Empfangsknoten **7506** kann die Daten in der Übertragung **7514** jedoch nicht zur Anwendungsschicht **7514** weiterleiten, sondern kann eine Rückverzweigungsoperation **7516** in der Netz-/Weiterleitungsschicht **7612** ausführen, um die empfangenen Daten als weitere Übertragung **7510** auszusenden, beispielsweise zum Ursprungsknoten **7504** zurück.

[0369] Die Rundlauf-Speicherzeit M_{tm} ist gegeben durch:

$$T_{O_{\text{Stapel}}} + T_{TX} + T_{1_{\text{Stapel}}} + T_{RX}$$

In dieser Gleichung bezeichnet $T_{O_{\text{Stapel}}}$ die Zeit, die der Speichernutzzinhalt benötigt, um die Netz-/Weiterleitungsschicht **7614** des Ursprungsknotens **7504** zu durchlaufen und durch Drahtlosübertragung auszutreten. T_{TX} bezeichnet die Im-Verlauf-Übertragungszeit vom Ursprungsknoten **7504** zum Empfangsknoten **7506**. $T_{1_{\text{Stapel}}}$ bezeichnet die Zeit, die der Im-Stapel-Rücklauf **7516** im Empfangsknoten **7506** in Anspruch nimmt, und T_{RX} ist die Im-Verlauf-Übertragungszeit vom Empfangsknoten **7506** zum Ursprungsknoten **7504** zurück.

[0370] Fig. 77 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **7700** zum Fragmentierten und Speichern von Daten in einer Übertragungsschleife zwischen Vorrichtungen gemäß einigen Ausführungsformen. Das Verfahren **7700** aus Fig. 77 kann durch die mit Bezug auf Fig. 79 beschriebene IoT-Vorrichtung **7900** implementiert werden. Das Verfahren **7700** beginnt in Block **7702**, wenn das System eingeschaltet wird. In Block **7704** wird das Kommunikationsundersystem initialisiert und werden Kommunikationskanäle zwischen den verschiedenen Vorrichtungen eingerichtet.

[0371] In Block **7706** wird ein Weiterleitungsvorgang zwischen Vorrichtungen eingeleitet. Der Weiterleitungsvorgang kann eine Datensende- oder eine Datenspeicheranforderung sein. In Block **7708** wird festgestellt, ob die Weiterleitungsanforderung eine Datenspeicheranforderung ist. Falls dies nicht der Fall ist, werden die Daten weitergeleitet und kehrt der Prozessfluss zu Block **7706** zurück, um auf eine andere Weiterleitungsanforderung zu warten.

[0372] In Block **7710** werden die zu speichernden Daten fragmentiert, um beispielsweise in einzelne Frames oder in eine andere geeignete Paketierung zu passen. In Block **7712** werden die Daten in ein Speicherpaket eingekapselt. Wenn die Daten nicht über die Kommunikationskanäle übertragen werden, sondern lediglich zurückgesendet werden, kann die Verpackung einfach sein, wobei das Speicherpaket durch Anhängen eines Headers, der angibt, dass es sich um gespeicherte Daten handelt, und einer Sequenznummer gebildet werden kann. Hierdurch kann der Overhead verringert werden, wodurch die zu speichernde Datenmenge erhöht werden kann. In Block **7714** wird das Speicherpaket sequenziert, um eine Neuzusammensetzung oder eine Identifikation des Anfangspunkts und des Endpunkts für die Daten zu ermöglichen.

[0373] In Block **7716** wird das Speicherpaket über den Kommunikationskanal gesendet. In Block **7718** wird festgestellt, ob alle Speicherpakete gesendet worden sind. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **7716** zurück, um ein anderes Speicherpaket abzusenden. Falls alle Pakete abgesendet wurden, kehrt der Prozessfluss zu Block **7706** zurück.

[0374] Fig. 78 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **7800** zur Datenspeicherung und zum Zugriff unter Verwendung eines Kommunikationskanals zur Speicherung gemäß einigen Ausführungsformen. Das Verfahren **7800** aus Fig. 78 kann durch die mit Bezug auf Fig. 79 beschriebene IoT-Vorrichtung **7900** implementiert werden. Block **7802** repräsentiert beispielsweise den Fall, in dem die Vorrichtung hochgefahren wird. In Block **7804** wird ein Kommunikationsundersystem initialisiert und werden Kommunikationskanäle mit anderen Vorrichtungen eingerichtet.

[0375] In Block **7806** findet ein Weiterleitungsvorgang statt, beispielsweise wenn ein Paket oder ein Frame von der Vorrichtung empfangen wird. In Block **7808** wird festgestellt, ob ein Speicherpaket empfangen wurde. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **7806** zurück, um die Weiterleitung abzuschließen und darauf zu warten, dass ein anderes Paket oder ein anderer Frame empfangen wird.

[0376] Falls in Block **7808** ein Speicherpaket identifiziert wurde, wird in Block **7810** festgestellt, ob das Paket weiter gespeichert werden sollte. Falls dies der Fall ist, wird das Paket in Block **7812** zum Speicherprozess zurückgeführt, wobei es beispielsweise zur absendenden Vorrichtung zurückgeleitet wird.

[0377] Falls das Paket in Block **7814** nicht mehr gespeichert ist, wird der Nutzzinhalt aus dem Paket entnommen und gespeichert. In Block **7816** wird die Sequenznummer anhand der Headerinformationen bestimmt und zur Datenneuzusammensetzung gespeichert. In Block **7818** wird festgestellt, ob alle Pakete empfangen worden sind. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **7806** zurück, um auf die nächsten Pakete oder den nächsten Frame zu warten.

[0378] Falls in Block **7818** festgestellt wird, dass alle Pakete empfangen wurden, werden die Nutzinhalte in Block **7820** neu zusammengesetzt, um die Daten zu bilden. In Block **7822** werden die Daten vom Verbraucher verwendet.

[0379] **Fig. 79** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **7900** zum Speichern von Daten in Übertragungskanälen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **7900** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0380] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0381] Ein Nutzinhaltsfragmentierer **7902** kann eine Speicheranforderung empfangen und die Daten auf der Grundlage des Kommunikationskanals, der Größe der Daten und dergleichen in Nutzinhalte fragmentieren. Der Nutzinhaltsfragmentierer **7902** kann eine zugeordnete Sequenznummer für den Nutzinhalt im Datenstream bestimmen. Eine Einkapselungseinrichtung **7904** kann Nutzinhalt in ein Paket einkapseln, wobei ein Header das Paket als Speicheranforderung identifiziert. Der Header kann auch die Sequenznummer für den Nutzinhalt enthalten. Abhängig vom ausgewählten Kommunikationskanal können die Pakete in die Datenfelder von Protokoll-Frames gepackt werden, wenngleich der Overhead die Verwendung einer einfacheren VerkapSELung nahe legen kann.

[0382] Der Massenspeicher **808** kann einen Kommunikator **7906** aufweisen, der Pakete über eine oder mehrere Kommunikationsstrecken, unter anderem beispielsweise durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** sendet und von diesen empfängt. Zusätzlich zu den mit Bezug auf **Fig. 79** beschriebenen Funktionen, wie mit Bezug auf **4502** in **Fig. 45** beschrieben, kann der Paketkommunikator **7902** andere Funktionen ausführen, wie die Übersetzung von Paketen zwischen Protokollen, die Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Paketkommunikator **7902** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems **3312** sein.

[0383] Ein Router **7908** kann Pakete und Frames untersuchen, die empfangen werden, um festzustellen, ob sie Teil einer Speicheranforderung sind. Pakete, die gespeicherte Daten aufweisen, können neu übertragen werden, beispielsweise von einem Netz-/Weiterleitungsniveau in einem Kommunikationsstapel. Falls eine Abrufanforderung empfangen wird, kann der Router Pakete, die gespeicherte Daten enthalten, zur Extraktion abfangen. Der Router **7908** kann auch Daten von einer Anwendung empfangen und feststellen, ob sie zu speichern oder zu übertragen sind.

[0384] Eine Nutzinhaltsextrahiereinrichtung **7910** kann vom Speicherstream extrahierte Pakete nehmen und einen Nutzinhalt und eine Sequenznummer aus den Paketen extrahieren. Ein Datenzusammensteller **7912** kann die abgerufenen Daten zur Verwendung durch die Vorrichtung dann neu zusammenstellen. Falls einige Pakete fehlen, kann der Datenzusammensteller **7912** den Router anweisen, weiter nach diesen Paketen zu suchen.

[0385] **Fig. 80** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **8000**, das Code aufweist, um einen Prozessor **902** anzuweisen, Daten in Übertragungskanälen zu speichern, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **8000** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **8000** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0386] Das nichtflüchtige maschinenlesbare Medium **8000** kann einen Code **8002** aufweisen, um den Prozessor **902** anzuweisen, Kommunikationskanäle mit anderen Vorrichtungen einzurichten. Ein Code **8004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob eine Weiterleitungsanforderung eine Datenspeicheranforderung ist. Ein Code **8004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Daten für eine Speicheranforderung in Nutzinhalte zu fragmentieren. Ein Code **8008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Nutzinhalte in Speicherpakete einzukapseln.

[0387] Ein Code **8010** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Speicherpakete über den Kommunikationskanal weiterzuleiten. Der Code **8010** kann den Prozessor **902** anweisen, festzustellen, ob Speicherpakete zu einer anderen Vorrichtung weitergesendet oder zur Verwendung abgefangen werden sollten.

[0388] Ein Code **8012** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Nutzinhalte aus den Speicherpaketen zu entpacken und dann die Headerinformationen zu parsen, um die Sequenznummer zu identifizieren. Ein Code **8014** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Daten auf der Grundlage der Sequenznummer aus den Nutzinhalten neu zusammenzusetzen.

[0389] **Fig. 81** ist eine Zeichnung einer Wellenform **8100**, die zur dynamischen Signalisierung verwendet werden kann, gemäß einigen Ausführungsformen. Wie hier beschrieben, kann eine Präambel-Wellenform **8102** und **8104** vor eine übertragene Wellenform **8106** gehängt werden, wobei die übertragene Wellenform eine Anzahl überlappender Frames in einzelnen Kanälen aufweist. Die Präambel-Wellenform **8102** und **8104** kann von einer Basisstation verwendet werden, um die Anzahl der Datenkanäle dynamisch zu bestimmen, die von einer Client-Vorrichtung für einen Uplink verwendet werden. Die Verwendung der Präambel-Wellenform **8102** und **8104** kann die Verwendung von Außerband-Steuernachrichtenübertragungen oder anderen Synchronisationsinformationen zur Information der Basisstation über die Datenkanäle unnötig machen.

[0390] Bei einigen Aspekten kann die Präambel-Wellenform **8102** und **8104** eine unter Verwendung einer verschobenen Zadoff-Chu(ZC)-Sequenz gebildete analoge Wellenform sein. ZC-Sequenzen sind eine Familie so genannter Null-Autokorrelations-Sequenzen mit konstanter Amplitude bzw. Constant-Amplitude-Zero-Autocorrelation(CAZAC)-Sequenzen mit Auto- und Kreuzkorrelationseigenschaften, die sie für Synchronisationszwecke sehr attraktiv machen. Sie werden vorherrschend im Long-Term-Evolution(LTE)-Standard für schnelle Drahtloskommunikationen verwendet.

[0391] Bei einigen Aspekten ist eine ZC-Sequenz eine komplexwertige mathematische Sequenz, die, wenn sie auf Funkkanäle angewendet wird, zu einem elektromagnetischen Signal konstanter Amplitude führt, wobei zyklisch verschobene Versionen der Sequenz bei der Null-Kreuzkorrelation miteinander am Empfänger einem Signalergebnis überlagert werden. Eine erzeugte Sequenz, die nicht verschoben wurde, ist als Root-Sequenz bekannt.

[0392] ZC-Sequenzen haben auch die Eigenschaft einer konstanten Amplitude. Wenn sie in einem Kommunikationssignal verwendet werden, verbessert dies die Effizienz des Leistungsverstärkers im Sender. Dies bietet eine Gelegenheit für die Verwendung kostengünstigerer Leistungsverstärker als sie typischerweise für ein System hoher Linearität verwendet werden würden in der Art eines orthogonalen Frequenzgetrenntlagemultiplex(OFDM)-Systems. Die Fähigkeit, ein linear verstärktes Signal unter Verwendung einer ZC- oder anderen Sequenz mit konstanter Einhüllenden zu verwenden, ist weniger komplex und weniger kostspielig als die Verwendung einer Sequenz mit einem sich schnell ändernden Amplitudenprofil. Falls die Linearität beeinträchtigt wird, kann die Signalqualität verschlechtert werden. CAZAC-Sequenzen weisen gute Antirauscheigenschaften auf und können selbst dann erkannt werden, wenn das Signal-Rausch-Verhältnis lediglich -10 dB ist. Zusammengefasst machen all diese Eigenschaften ZC-Sequenzen sehr attraktiv, wenn sie zur Präambel-Signalisierung und Synchronisation in Kommunikationssystemen verwendet werden.

[0393] Der Ansatz wurde entwickelt, um Client- und Basisstations-Signalisierungswellenformen vor Frames zu hängen, die ausgetauscht werden, wobei der für einen bestimmten Frame verwendete Kanal angegeben wird. Weil er für Drahtlossysteme entwickelt wurde, bei denen dynamische Spektrumszugriff- und adaptive Bandbreitenansätze verwendet werden, ist er für Systeme besonders geeignet, die keine Steuerkanäle oder geplante Uplink(UL)/Downlink(DL)-Zeitschlitze verwenden oder benötigen.

[0394] Bei einem kanalisiertem System kann es die Präambel-Struktur **8102** und **8104** Client-Vorrichtungen ermöglichen, eine empfangende Basisstation über die Anzahl der Kanäle zu informieren, die zur Übermittlung des UL-Datennutzinhalts von der Client-Vorrichtung zu einer Basisstation verwendet werden, bevor der Client seine UL-Nutzinhaltsnachricht absendet. Eine beispielhafte Verwendung ergibt sich bei Niederleistungs-Weitbereichs-Drahtloskommunikationen, die für IoT-Systeme mit geringem Overhead verwendet werden. Die Techniken ermöglichen es diesen Vorrichtungen, die Bandbreite in einem Fall dynamisch zu ändern, in dem mehr Daten im UL-Modus zu einer Basisstation übertragen werden müssen, und, falls variable Datenlängen von einer Basisstation zu fernen Client-Vorrichtungen gesendet werden müssen, beispielsweise Über-die-Luft-Firmware- und Konfigurationsaktualisierungen zu unterstützen.

[0395] Fig. 82 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 8200 zur Übertragung von Daten unter Verwendung einer ZC-Präambel-Struktur gemäß einigen Ausführungsformen. Das Verfahren 8200 aus Fig. 82 kann durch die mit Bezug auf Fig. 85 beschriebene IoT-Vorrichtung 8500 implementiert werden. In Block 8202 bestimmt eine Client-Vorrichtung die Anzahl der verfügbaren oder möglichen Kanäle N . Dies kann durch eine Informationstheorieanalyse der die Vorrichtungen koppelnden Kommunikationskanäle geschehen. Die verfügbare Anzahl von Kanälen kann beispielsweise in einer Einzelkanalnachricht zu einer Empfangsvorrichtung gesendet werden, um Kommunikationen zu initialisieren.

[0396] In Block 8204 wird die Menge von N ZC-Sequenzen, die der Anzahl N der verfügbaren Kanäle entsprechen, durch Erzeugen einer Menge K ganzzahliger von Null verschiedener und eindeutiger ZC-Verschiebungswerte in Zusammenhang mit jedem Kanal K_c erzeugt, wobei c die Kanalnummer bezeichnet. Es sei bemerkt, dass alle Drahtlosvorrichtungen, einschließlich der Basisstation, Kenntnis von K haben, wobei sie beispielsweise ihre eigene Kopie der Kanalinformationen erzeugen. Die ZC-Sequenzen für jeden Verschiebungswert K_c werden nach der folgenden Formel erzeugt:

$$x_{K_c}(n+1) = \exp^{-j\pi K_c n(n+1)/N_{ZC}}$$

Der komplexe Wert an jeder Position (n) jeder Root-ZC-Sequenz (u) ist gegeben durch:

$$x_u(n+1) = \exp^{-j\pi u n(n+1)/N_{ZC}},$$

wobei $0 \leq n \leq N_{ZC}$ und N_{ZC} die Länge der Sequenz ist.

[0397] Die Vorrichtung kann dann die Anzahl der Kanäle k bestimmen, die sie verwenden möchte, um UL-Daten bei einer Kommunikation zu einer Basisstation zu senden. Beispielsweise kann die Vorrichtung nicht alle möglichen Kanäle verwenden müssen, sondern eine geringere Anzahl verwenden, um das Signal-Rausch-Verhältnis für die Übertragung zu erhöhen.

[0398] In Block 8206 wählt die drahtlose Client-Vorrichtung eine Sequenz K_c entsprechend der Anzahl c der für die Übertragung der Wellenform zu verwendenden Kanäle. Die Client-Vorrichtung erzeugt dann die ZC-Präambel in Block 8208. In Block 8210 hängt die Drahtlosvorrichtung die einzelne ZC-Sequenz x_{K_c} vor die komplexwertige Basisbandfunktion, die von der Vorrichtung verwendet wird, um die modulierten Frames zu senden. In Block 8212 wandelt die drahtlose Client-Vorrichtung die Basisbandwellenform dann aufwärts und überträgt sie.

[0399] Fig. 83 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 8300 zum Empfangen von Daten über mehrere Kanäle unter Verwendung von ZCverschobenen Sequenzen gemäß einigen Ausführungsformen. Das Verfahren 8300 aus Fig. 83 kann durch die mit Bezug auf Fig. 85 beschriebene IoT-Vorrichtung 8500 implementiert werden. Das Verfahren 8300 beginnt in Block 8302, wenn die empfangende Vorrichtung die Anzahl der Kanäle bestimmt, die von der sendenden Vorrichtung verwendet werden. Dies kann durch eine Autokorrelation an eingehenden komplexwertigen Sequenzen geschehen, um eine Präambel zu erkennen. ZC-Sequenzen sind mit einer Periode N_{ZC} periodisch, falls N_{ZC} eine Primzahl ist. Wenn N_{ZC} eine Primzahl ist, ist auch die diskrete Fouriertransformation (DFT) einer ZC-Sequenz eine ZC-Sequenz. Die Autokorrelation einer ZC-Sequenz mit primzahliger Länge mit einer zyklisch verschobenen Version von sich selbst ist null. Die Präambel kann auch durch Ausführen einer Kreuzkorrelation mit jeder der verschobenen ZC-Sequenzen an der empfangenden Vorrichtung erkannt werden. Falls eine Sequenz arbeitet, ist die Signalintensität für diese Sequenz viel höher als die anderen, wie mit Bezug auf Fig. 84 beschrieben.

[0400] Falls in Block 8304 die Präambel erkannt wird, wird die Anzahl der Kanäle, die für die Verwendung durch die Client-Vorrichtung vorgesehen sind, anhand der Kreuzkorrelation der empfangenen ZC-Präambel gegenüber einer bekannten Menge möglicher ZC-verschobener Sequenzen bestimmt. Der Empfänger benötigt A-priori-Kenntnisse über die ZC-Sequenzlänge N_{ZC} und die Menge der Verschiebungswerte und kann die folgende Gleichung verwenden:

$$\hat{R} = \left\{ \begin{array}{l} \sum_{n=0}^{N-m-1} x_{n+m} y_n^*, \quad m \geq 0, \\ \hat{R}_{yx}^*(m-N), \quad m = 1, 2, \dots, N-1 \end{array} \right\},$$

wobei $x(n)$ und $y(n)$ die beiden korrelierten Sequenzen sind und die Korrelationsausgabe durch R bezeichnet ist. Die Kreuzkorrelation zwischen zwei ZC-Sequenzen mit primzahliger Länge, beispielsweise mit unterschiedlichen u -Werten, ist konstant und gleich:

$$\frac{1}{\sqrt{N_{ZC}}}.$$

[0401] Die beim empfangenen Signal verwendete Sequenz wird durch die Korrelationsergebnisse bestimmt. Die Null-Autokorrelation-ZC-Sequenzeigenschaften ermöglichen, dass dies mit einem hohen Vertrauensgrad erreicht wird. Falls in Block **8304** keine ZC-Präambel erkannt wird, kehrt der Prozessfluss zu Block **8304** zurück, um mit dem nächsten Frame zu wiederholen.

[0402] In Block **8306** wird eine Umkehrabbildung ausgeführt, um die Anzahl der Kanäle zu bestimmen, die der im UL-Signal verwendeten erkannten ZC-Sequenz entspricht. In Block **8310** bereitet die Basisstation ihre Empfangskette vor, um den kombinierten i-Kanal-Nutzzinhalt von der Client-Vorrichtung zu empfangen und zu demodulieren, welcher der ZC-basierten Signalisierungswellenform unmittelbar folgt. In Block **8312** werden die Nutzdaten jeder der N Kanaldateneinheiten demoduliert, beispielsweise unter Verwendung einer Kreuzkorrelationstechnik an der Nutzzinhaltswellenform mit einer einem Kanal entsprechenden verschobenen ZC-Sequenz.

[0403] **Fig. 84** ist eine Reihe von Auftragungen, welche den in der Gleichung für R für jede der durch K gegebenen Sequenzen detaillierten Korrelationsprozess zeigen, gemäß einigen Ausführungsformen. Hierdurch wird festgestellt, welche Sequenz K_c zum größten Korrelations-Peak geführt hat. Die erste Auftragung zeigt, dass die erste Sequenz, wo $u = 19$, $k = 19$ ist und wo u dem Kanal c entspricht, zur größten Korrelationsausgabe geführt hat. Insbesondere wird die korrekte Sequenz Z_{Cd} durch Finden des Maximums der folgenden Gleichung bestimmt:

$$ZC_d = \max(\max(x_{Kc})),$$

wobei x_{Kc} die Ausgabe des Kreuzkorrelationsprozesses ist.

[0404] **Fig. 85** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 8500 zur Verwendung von ZC-Sequenzen für das Senden von Daten in mehreren gleichzeitigen Kanälen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **8500** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0405] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0406] Der Massenspeicher **808** kann eine Kanalkennung **8502** aufweisen, welche die maximale Anzahl der verfügbaren Kanäle festlegt. Ein Sequenzgenerator **8504** kann die ZC-Sequenzen für jeden der Kanäle erzeugen. Ein Präambel-Generator kann die Präambel-Wellenform erzeugen, welche die Anzahl der bei der Kommunikation verwendeten Kanäle angibt. Ein Kommunikator **8506** kann die modulierte Wellenform für jeden der Frames in Zusammenhang mit einem Kanal unter Verwendung der diesem Kanal zugeordneten ZC-Sequenz erzeugen. Der Kommunikator **8506** kann dann die modulierten Wellenformen überlagern, die Präambel-Wellenform vorne anhängen und die sich ergebende Wellenform an einen Sender in der Art des Mesh-Sendeempfängers **810** übergeben.

[0407] Bei einigen Aspekten sind die Kommunikationen bidirektional. Eine Indexkennung **8510** kann eine von einer anderen Vorrichtung empfangene Wellenform analysieren und eine Kreuzkorrelation ausführen, um festzustellen, ob eine Präambel vorhanden ist. Falls dies der Fall ist, kann die Indexkennung ein Nachschlagen ausführen, um die Anzahl der Kanäle im Nutzzinhalt zu bestimmen. Ein Kanaldemodulator **8512** kann die Informationen in jedem der Kanäle demodulieren, um den ursprünglichen in diesem Kanal gesendeten Frame wiederherzustellen.

[0408] Fig. 86 ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **8600**, das einen Code aufweist, um einen Prozessor **902** anzuweisen, über unter Verwendung ZC-verschobener Sequenzen modulierte Kanäle zu kommunizieren, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **8600** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **8600** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0409] Das nichtflüchtige maschinenlesbare Medium **8600** kann einen Code **8602** aufweisen, um den Prozessor **902** anzuweisen, die Anzahl der verfügbaren Kanäle zu bestimmen. Ein Code **8604** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ZC-Sequenzen für jeden der Kanäle zu erzeugen. Ein Code **8606** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine ZC-Präambel für eine modulierte Wellenform zu erzeugen. Ein Code **8608** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die ZC-Präambel vor der modulierten Wellenform anzuhängen. Ein Code **8610** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die ZC-Präambel und die modulierte Wellenform zu übertragen.

[0410] Ein Code **8612** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Kreuzkorrelation an einer empfangenen Wellenform auszuführen, um festzustellen, ob eine ZC-Präambel vorhanden ist, und, falls dies der Fall ist, wie viele Kanäle repräsentiert werden. Ein Code **8614** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Empfänger für die Anzahl der vorhandenen Kanäle zu konfigurieren. Ein Code **8616** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Kanäle zu demodulieren, um die Daten von jedem Kanal abzurufen.

[0411] Fig. 87 ist eine Schemazeichnung eines Mehrfunk-Koexistenzsystems **8700** in einer IoT-Vorrichtung **8702** gemäß einigen Ausführungsformen. Die IoT-Vorrichtung **8702** kann ein Gateway oder Koordinator sein, der Kommunikationen mit einer Cloud oder mit anderen IoT-Vorrichtungen in einer Fog-Vorrichtung ermöglicht. Wie hier beschrieben, ermöglicht das Mehrfunk-Koexistenzsystem **8700** Kommunikationen unter Verwendung mehrerer Funksysteme **8704** mit Funksystemen in anderen Knoten **8706**, um eine wirksamere Verwendung eines Spektrums zu ermöglichen. Dies kann die Koexistenz zwischen verschiedenen Funktechnologien sowie primären und sekundären Benutzern des Frequenzspektrums ermöglichen.

[0412] Ein Ansatz besteht darin, einen Lizenzausnahmezugang durch Konformität mit vorübergehend unbenutzten Teilen des Frequenzspektrums unter Verwendung kognitiver Funkeinrichtungen als Zugangssystem eines Sekundärnutzers, der auf spezifische Frequenzbänder beschränkt ist, zu ermöglichen. Kognitive Funkeinrichtungen (CR) können erkennen, welche Kommunikationskanäle in Gebrauch sind, und Kommunikationen zu freien Kanälen verschieben, während belegte vermieden werden. Vorrichtungen, die auf diesen Frequenzbändern arbeiten, halten sich an einen definierten Regelsatz, beispielsweise zum Schützen der Primärnutzer und zur Koexistenz mit anderen Benutzern.

[0413] Während CR die Verwendung des White-Space(WS)-Spektrums abdecken, betreffen die hier beschriebenen Techniken die Koexistenz von Funksendeempfängern unter Verwendung eines Standards in der Art von Funkeinrichtungen **8704**, die mit IEEE 802.11x, IEEE 802.15.4 übereinstimmen und von nicht standardmäßigen Funkeinrichtungen in der Art von LoRa. Die Kommunikation zwischen Knoten **AL06** kann verwendet werden, um Informationen über die Koexistenz zwischen Funksystemen zu teilen. Beispielsweise kann ein Koexistenzmanager **8708** die Funkeinrichtungen **8704**, die für eine bestimmte Kommunikation verwendet werden, verfolgen und die Informationen in einem Kanaldatenspeicher **8710** speichern.

[0414] Bei einigen Aspekten kann eine universelle Koexistenzschnittstelle **8712** auf die Informationen vom Koexistenzmanager **8708** zugreifen und feststellen, welche Kommunikationen zu welchen Zeitpunkten verwendet werden können. Funkstandards auf der Grundlage von IEEE 802.22 und IEEE 802.11 unterstützen bereits Verfahren für die Koexistenz. Beispielsweise verwendet IEEE 802.22 ein zeitbasiertes Verfahren für die Koexistenz, während IEEE 802.19.1 Mechanismen zum Teilen von Koexistenzdaten nur zwischen TV-WS-Frequenzen bereitstellt. Die universelle Koexistenzschnittstelle **8712** kann auch die Modifikation von Betriebsparametern in der Art des Codierungsschemas und der Modulation und der Übertragungsleistung einzelner Funkeinrichtungen ermöglichen.

[0415] Die Kommunikationskanäle, die von der universellen Koexistenzschnittstelle **8712** ausgewählt werden, können an eine Protokollabstraktions-API (Anwendungsprogrammierschnittstelle) **8714** übergeben werden, um Frames für die bestimmten Kommunikationskanäle zu bilden. Die Protokollabstraktions-API **8714** kann auf

einen Protokolldatenspeicher **8716** zugreifen, um die Protokolle zu erhalten, die für die ausgewählten Kommunikationskanäle verwendet werden können. Die Frames können dann zu den anderen Knoten **8706** gesendet oder von diesen empfangen werden, wie durch Linien **8718** angegeben ist.

[0416] Fig. 88 ist ein Leiterdiagramm eines beispielhaften Verfahrens **8800** zur Steuerung und Behandlung mehrerer koexistierender Funkeinrichtungen gemäß einigen Ausführungsformen. Das Verfahren **8800** aus Fig. 88 kann durch die mit Bezug auf die Fig. 87 und Fig. 89 beschriebene IoT-Vorrichtung **8702** implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 87 beschrieben wurden. In Schritt **8802** sendet ein Koexistenzmanager **8708** eine Anforderung **8802** für Kommunikationen verfügbarer Bänder zu einem lokalen Sicherheitsstellen-Domain-Broker bzw. Local Security Authority Domain Broker (LSADB) **8804**. Der LSADB **8802** antwortet mit einer Nachricht **8806**, welche die verfügbaren Bänder bereitstellt.

[0417] Der Koexistenzmanager **8708** berechnet dann **8808** einen anfänglichen Bandplan und bildet **8810** eine Nachbarkarte, welche die Identität eines oder mehrerer Nachbarn aufweist. Diese Informationen können im mit Bezug auf Fig. 87 beschriebenen Kanaldatenspeicher **8710** gespeichert werden. Der Koexistenzmanager **8708** kann dann eine Anforderung **8812** zu einem Über-Domains-Informationsteilungssystem bzw. Cross-Domain Information Sharing System (CDIS) **8814** senden, um die Kommunikationskanäle oder -bänder zu identifizieren, die für die Kommunikation mit den Nachbarn verwendet werden können. Das CDIS **8814** kann mit einer Nachricht **8815** antworten, welche die Kommunikationskanäle identifiziert, die mit einem Nachbarn verwendet werden können. Diese Informationen können vom Koexistenzmanager **8708** verwendet werden, um ein anfängliches Koexistenzmodell zu bestimmen **8816**, das sowohl Nachbarn als auch zugeordnete Kommunikationskanäle identifiziert. An diesem Punkt wartet der Koexistenzmanager **8708** auf weitere Kommunikationen, um das System einzurichten.

[0418] Zum Abschluss der Initialisierung kann ein Funksystem **8704** eine Nachricht **8818** zur Protokollübersetzungs-API **8714** senden, wodurch die in der IoT-Vorrichtung verfügbaren Funktypen aufgezählt werden. Die Protokollübersetzungs-API **8714** überprüft dann **8820**, dass die Standards unter anderem in der Art von Protokollen für Frames, die für die vorhandenen Funktypen verwendet werden, beispielsweise im mit Bezug auf Fig. 87 beschriebenen Protokolldatenspeicher **8716** verfügbar sind. Falls dies nicht der Fall ist, kann die Protokollübersetzungs-API **8714** die geeigneten Standards von der Cloud herunterladen. In Schritt **8822** bestätigt die Protokollübersetzungs-API **8714** dann, dass die Funkeinrichtungen den Standards **8822** folgen, und sendet eine Subskriptionsanforderung für die Funkeinrichtungen zur universellen Koexistenzschnittstelle **8712**.

[0419] In Schritt **8826** weist die universelle Koexistenzschnittstelle **8712** eine Funkbehandlungsidentifikation einem oder mehreren der Funktypen zu. Die universelle Koexistenzschnittstelle **8712** sendet dann eine Subskriptionsanforderung **8828** zum Koexistenzmanager **8708**, der die Behandlungskennung für die Funktypen aufweist.

[0420] Nach dem Empfang der Subskriptionsanforderung **8828** bestimmt **8830** der Koexistenzmanager **8708** ein aktives Koexistenzmodell durch Aktualisieren oder Ersetzen des anfänglichen Koexistenzmodells. Falls irgendwelche der Funktypen nicht im anfänglichen Koexistenzmodell vorhanden sind, beispielsweise weil sie nicht im CDIS **8814** vorhanden sind, sendet der Koexistenzmanager **8708** eine Anforderung **8832** einer Subskription für die neue Funkeinrichtung. Das CDIS **8814** antwortet **8834** beispielsweise mit einer Nachricht, die angibt, dass die Funkeinrichtung registriert ist. Der Koexistenzmanager **8708** sendet dann eine Benachrichtigung **8836** zur universellen Koexistenzschnittstelle **8712**, dass die neue Funksubskriptionsanforderung akzeptiert wurde.

[0421] Sobald die Protokollübersetzungs-API **8714** der universellen Koexistenzschnittstelle die Funktypen angegeben hat, kann Sie eine Nachricht **8838** zum Funksystem **8704** senden, um anzugeben, dass die Funktion abgeschlossen wurde. Die Nachricht **8838** kann die der universellen Koexistenzschnittstelle **8712** angegebenen Funktypen auflisten.

[0422] Das Funksystem **8704** kann eine Nachricht **8840** zur Protokollübersetzungs-API **8714** senden, um die Funkinitialisierung für eine oder mehrere der Funkeinrichtungen neu einzuleiten. Die Protokollübersetzungs-API **8714** kann wieder die Standards für die Funktypen validieren **8842** und feststellen **8844**, ob irgendwelche der Funkeinrichtungen den Standards nicht folgen. Die Protokollübersetzungs-API **8714** kann dann eine Nachricht **8846** zum Funksystem **8704** senden, um die konfigurierbaren Parameter für jede der Funkeinrichtungen festzulegen. Das Funksystem **8704** kann mit einer Nachricht **8848** antworten, wodurch die für die Funkeinrichtungen festgelegten Parameter bestätigt werden. Die Protokollübersetzungs-API **8714** kann dann einen Pa-

parameterabbildungssatz für die verwendeten Funkeinrichtungen erzeugen und eine Nachricht **8852** zum Funksystem **8704** senden, die angibt, dass die Aufzählung der Funktypen abgeschlossen ist, und die Kommunikationen mit dem Funksystem **8704** werden initialisiert.

[0423] Falls das CDIS **8814** eine Koexistenzverletzung erkennt, beispielsweise dass ein lizenzierte Benutzer eine Frequenz belegt, wodurch die Verwendung durch die IOT-Vorrichtung blockiert wird, kann es eine die Verletzung ankündigende Nachricht **8854** zum Koexistenzmanager **8708** senden. In Schritt **8856** kann der Koexistenzmanager **8708** die Koexistenzverletzung überprüfen, beispielsweise durch Feststellen, ob die zugeordnete Funkeinrichtung ein Blockiersignal empfängt, und dann ein die Verletzung angegebendes Hinweiszeichen setzen **8858**. Er kann dann eine Nachricht **8860** zur universellen Koexistenzschnittstelle **8712** senden, um eine Rekonfiguration der Kommunikationsparameter zu fordern.

[0424] Die universelle Koexistenzschnittstelle **8712** kann den Funktyp für die Koexistenzverletzung überprüfen **8862** und eine Nachricht **8864** mit einem neuen Parametersatz zum Funksystem **8704** senden, beispielsweise zum vorübergehenden Deaktivieren einer bestimmten Funkeinrichtung, zum Übergang zu einer anderen Frequenz und dergleichen. Das Funksystem **8704** kann mit einer Nachricht **8866** antworten, welche die Änderung in den Funkparametern bestätigt. Das Funksystem **8704** kann dann die aktive Typliste mit der Protokollübersetzungs-API **8714** rekonfigurieren, beispielsweise durch Senden der Nachricht **8840** zur Protokollübersetzungs-API **8714**, um eine Funkinitialisierung für die Funkeinrichtungen anzugeben. Das Funksystem **8704** kann dann die Funkinitialisierungssequenz **8868** durch Senden einer Nachricht **8840** zur Protokollübersetzungs-API **8714** neu einleiten. Die Protokollübersetzungs-API **8714** kann dann die Initialisierungssequenz **8868** durch die Nachricht **8852** an das Funksystem **8704** schrittweise durchlaufen, um anzugeben, dass die Aufzählung der Funktypen abgeschlossen ist, und die Kommunikationen mit dem Funksystem **8704** werden initialisiert.

[0425] Der Koexistenzmanager **8708** kann auf einer wiederkehrenden Basis eine Gute-Nachbarn-Prüfung **8870** ausführen, um festzustellen, ob andere Knoten noch mit der IoT-Vorrichtung kommunizieren. Falls Kommunikationen geändert wurden, kann der Koexistenzmanager eine Nachbarbefehlslistenänderung feststellen **8872** und eine lokale Änderung an der Befehlsliste vornehmen **8874**. Der Koexistenzmanager **8708** kann dann eine Rekonfigurationsnachricht **8876** mit einem neuen Parametersatz zum Funksystem **8704** senden. Das Funksystem **8704** kann mit einer Nachricht **8878** antworten, welche die Akzeptanz der Parameter bestätigt. Wie hier beschrieben, kann das Funksystem **8704** dann die Initialisierungssequenz **8868** mit der Protokollübersetzungs-API **8714** wiederholen.

[0426] Zusätzlich zur Auslösung von Änderungen durch wiederholte Prüfungen durch den Koexistenzmanager **8708** können andere Einheiten eine Änderung in den Kommunikationsparametern fordern. Beispielsweise kann der Koexistenzmanager **8708** feststellen **8880**, dass eine Anforderung einer Änderung von einem Nachbarn empfangen wurde. Der Koexistenzmanager **8708** kann dann die Rekonfigurationsanforderung **8882** mit den vorgeschlagenen Parametern zum Funksystem **8704** senden. Das Funksystem **8704** kann dann mit einer Nachricht **8884** antworten, wodurch bestätigt wird, dass die Parameter akzeptiert wurden. Das Funksystem **8704** kann dann die Initialisierungssequenz **8868** mit der Protokollübersetzungs-API **8714** wiederholen.

[0427] Fig. 89 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **8900** zur Verwendung mehrerer koexistierender Funkeinrichtungen für die Kommunikation mit anderen Knoten vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3, Fig. 8 und Fig. 87 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **8900** andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können. Das mit Bezug auf Fig. 87 beschriebene Funksystem **8704** kann den für den Mesh-Sendeempfänger **810**, den Uplink-Sendeempfänger **814** oder beide verwendeten Funkeinrichtungen entsprechen. Die anderen Knoten **8706** können Mesh-Vorrichtungen **812**, Vorrichtungen in der Cloud **302** oder beide aufweisen.

[0428] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0429] Der Massenspeicher **808** kann einen Koexistenzmanager **8708** zum Steuern der Verwendung mehrerer koexistierender Funkeinrichtungen aufweisen. Eine universelle Koexistenzschnittstelle **8712** kann sich mit den Funkeinrichtungen verbinden, beispielsweise durch eine Protokollübersetzungs-API **8714**. Die Proto-

kollabstraktions-API **8714** kann die verschiedenen in Gebrauch befindlichen Kommunikationskanäle verfolgen und Daten in Frames für die bestimmten benötigten Protokolle verpacken. Ein Kanaldatenspeicher **8710** kann die durch den Koexistenzmanager bestimmten aktiven Kommunikationspläne und Funkeinrichtungen halten. Ein Protokolldatenspeicher **8716** kann die verfügbaren Protokolle für die Protokollübersetzungs-API **8714** speichern. Ein Kommunikator **8902** kann die Frames unter Verwendung der geeigneten Funkeinrichtung, beispielsweise im Mesh-Sendeempfänger **810** oder im Uplink-Sendeempfänger **814** von der Protokollübersetzungs-API **8714** zu einer anderen Vorrichtung senden.

[0430] Fig. 90 ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **9000**, das Code aufweist, um einen Prozessor **902** anzuweisen, mehrere koexistierende Funkeinrichtungen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **9000** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **9000** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0431] Das nichtflüchtige maschinenlesbare Medium **9000** kann einen Code **9002** aufweisen, um den Prozessor **902** anzuweisen, einen anfänglichen Bandplan festzulegen und eine Nachbarkarte zu bilden, welche die Identität der Nachbarn und die Kommunikationskanäle, die mit den identifizierten Nachbarn verwendet werden können, aufweist. Der anfängliche Bandplan kann anhand einer Liste von Funkeinrichtungen für die Vorrichtung bestimmt werden und angeben, von welchen Funkeinrichtungen erwartet wird, dass sie verfügbar sind, und ihre zugeordneten Protokolle aufweisen. Der Bandplan kann dann nach Bestimmen der Funkeinrichtungen, die tatsächlich verfügbar sind, fertig gestellt werden. Ein Code **9004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein anfängliches Koexistenzmodell zu bestimmen, das sowohl Nachbarn als auch zugeordnete Bänder identifiziert. Das Koexistenzmodell kann fertig gestellt werden, nachdem die Initialisierung der Funksysteme die Feststellung ermöglicht hat, welche Funkeinrichtungen und Bänder betriebsbereit sind. Ein Code **9006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob eine Koexistenzverletzung vorhanden ist, beispielsweise nachdem er durch eine außenstehende Einheit oder durch Erkennen einer störenden Übertragung informiert wurde. Ein Code **9008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Kommunikationen zu rekonfigurieren, beispielsweise falls eine Koexistenzverletzung erkannt wurde, falls eine wiederholte Prüfung benachbarter Einheiten angibt, dass die Parameter angepasst werden müssen, oder bei einer Anforderung von einer benachbarten Einheit.

[0432] Ein Code **9010** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Protokollübersetzungen einzuleiten, beispielsweise durch ein Funksystem, das eine Protokollübersetzungs-API über die verfügbaren Kommunikationskanäle informiert. Ein Code **9012** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Daten in Frames für spezifische Kommunikationskanäle zu packen. Ein Code **9014** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Frames über die zugeordneten Kommunikationskanäle zu übertragen. Ein Code **9016** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Protokollübersetzungsfunktion neu zu initialisieren, nachdem Funkoperationen modifiziert wurden.

[0433] Fig. 91 ist ein schematisches Diagramm einer Netzdienste-Overlay-Funktion über ein heterogenes Netz (HetNet) **9100** gemäß einigen Ausführungsformen. Die Technik ermöglicht die Erzeugung von Dienstketten über heterogene Netze, wodurch ein automatisches Bereitstellen und Rekonfigurieren von IoT-Vorrichtungen in einem Fog- oder Mesh-Netz ermöglicht werden kann. Beispielsweise können IoT-Vorrichtungen funktionell geclustert werden, um einen Dienst zu bilden, wie bei einer temporären virtuellen oder Fog-Vorrichtung, wie mit Bezug auf Fig. 4 beschrieben. Beim HetNet-Netz **9100** können Domains **9102** und **9104** IoT-Vorrichtungen aufweisen, die zusammen gruppiert werden können, um eine bestimmte Funktion auszuführen, in der Art einer Verkehrsregelungsfunktion an einer Kreuzung. Die Vorrichtungen können durch eine Anzahl drahtgebundener und drahtloser Verbindungsstrecken **9106** miteinander und mit der Cloud **302** verbunden werden.

[0434] Eine Netzdomain **9102** oder **9104** kann eine Netzdomain-Steuereinrichtung (NDC) **9108** oder einen Dienstkoordinator aufweisen, die oder der auf einer Vorrichtung innerhalb der Netzdomain **9102** oder **9104** läuft. Die NDC **9108** kann dynamisch zu einer Netzdomain **9102** oder **9104** bewegt werden oder vor der Installation auf der Vorrichtung vorinstalliert werden. Die NDC **9108** kann mit einem Orchestrierungssystem höherer Ebene **9110** kommunizieren. Die NDC **9108** kann als ein Dienstkoordinator wirken, wodurch Einheiten oder Komponenten identifiziert werden, die am Dienst teilnehmen können. Es sei bemerkt, dass auch andere Vorrichtungen als Dienstkoordinator wirken können, wie Endpunkt-IoT-Vorrichtungen, Daten-Aggregatoren, Vorrichtungen in der Cloud **302** oder Vorrichtungen in anderen Netzdomains **9102** oder **9104**.

[0435] Dienstbehandlungsanforderungen zur Ausführung eines Dienstes oder zur Erzeugung einer Fog-Vorrichtung zur Ausführung eines Dienstes können von einem Orchestrator **9112** an die NDC **9108** übergeben werden. Wenngleich er als Teil des Orchestrierungssystems höherer Ebene **9110** dargestellt ist, kann sich der Orchestrator **9112** auch in einer anderen Einheit in der Cloud in der Art einer Gateway-Schnittstelle zur Domain **9102** oder **9104** oder eines Servers **9114**, der als Datenverbraucher wirkt, oder in der NDC **9108** befinden.

[0436] Management-Anwendungen im Orchestrator **9112** können die Erzeugung, Aktualisierung, Löschung und Migration von Netzdienst-Overlays **9116** einschließen. Die Netzdienst-Overlays **9116** können als Mikroprogramme wirken, beispielsweise Codesegmente, die für die Ausführung einer spezifischen Aufgabe ausgelegt sind, wie unter anderem das Erhalten einer Temperatur von einem Ort oder das Erhöhen des Verkehrsflusses in einer Richtung entlang einer Straße. Ferner können die Netzdienst-Overlays **9116** auf höheren Ebenen wirken, einschließlich Codesequenzen für einen Dienst, die eine Anzahl von Aufrufen von Netzdienst-Overlays **9116** einer niedrigeren Ebene aufweisen.

[0437] Der Orchestrator **9112** kann den Dienst oder das virtuelle Dienstnetz in Netzdienstelemente zerlegen, die durch zugeordnete Netzdienst-Overlays **9116** vervollständigt werden können. Eine NDC **9108**, die mit dem Orchestrator **9116** registriert ist, kann eine Provideranforderung an den Orchestrator **9112** übergeben, um die Ressourcen in der Art von Netzdienst-Overlays oder Vorrichtungen in der anderen Domain **9102** oder **9104** bereitzustellen, um eines oder viele der Dienstelemente für eine Dienstbehandlungsanforderung zu erfüllen.

[0438] Nachdem die NDC **9108** durch den Orchestrator **9112** als Dienstkoordinator bestätigt wurde, ist sie für die Erfüllung der Dienstanforderung verantwortlich, beispielsweise das Behandeln der den Dienst bereitstellenden Netzdienstelemente. Wie hier verwendet, kann ein Netzdienstelement eine codebetriebene Komponente eines Systems zur Bereitstellung von Daten für den Dienst sein. Mehrere Netzdienstelemente können zusammen gruppiert werden, um einen Dienst bereitzustellen, wobei es sich um eine Fog-Vorrichtung **402** handeln kann, wie mit Bezug auf **Fig. 4** beschrieben. Es sei bemerkt, dass ein Netzdienstelement einen Knoten **9118** oder **9120**, einen einzelnen Sensor von einem Knoten **9118** oder **9120**, ein auf einer Einheit laufendes Programm in der Art eines Daten-Aggregators **406** oder eine Anzahl anderer physischer oder virtueller Vorrichtungen oder Systeme aufweisen kann.

[0439] Eine NDC **9108** in der ersten Domain **9102** kann auch mit einer NDC **9108** in der zweiten Domain **9104** kommunizieren, beispielsweise wenn ein Dienst Vorrichtungen von mehreren Netzdomains aufweist. Die NDC **9108** kann eine Datenbank **9122** zur Speicherung von Daten und Metadaten in der Art von Ressourcen von an einer bestimmten Domain **9102** oder **9104** registrierten Knoten **9118** oder **9120**, einschließlich angeschlossener Vorrichtungen und Fähigkeiten, verwenden. Die NDC **9108** kann auch einen geteilten virtuellen Ablagespeicher **9124** unterhalten, wo sie Netzdienstelemente ankündigt, die Maßnahmen benötigen, und Identitäten von Netzdienstelemente bereitstellenden Dienstkomponenten speichert.

[0440] Die NDC **9108** kann eine Maschinelernen(ML)-Engine **9126** verwenden, die sie verwendet, um auszuwählen, welche Knoten **9118** oder **9120** oder welche Kombination von Knoten **9118** oder **9120** verwendet wird, um die Anforderungen des Dienstes zu erfüllen. Die ML-Engine **9126** kann Simulationen, neuronale Netze, eine statistische Analyse und eine Anzahl anderer Techniken verwenden, um festzustellen, welche Komponenten ein Netzdienstelement vervollständigen können.

[0441] Die NDC **9108** kann eine Vielzahl von Kriterien verwenden, um auszuwählen, welche Knoten **9118** oder **9120** oder andere Vorrichtungen Netzdienstelemente hosten. Die Auswahlkriterien können Latenzanforderungen, spezifische Bandbreitenanforderungen oder Zuverlässigkeitsmetriken einschließen. Die Daten werden in der Datenbank **9122** gespeichert und können auf historischen Funktionsweisedaten beruhen. Die NDC **9108** kann auch als Vermittler wirken, wenn sich mehrere Endknoten anbieten, eine Werbeanforderung für das gleiche Netzdienstelement zu erfüllen. Die NDC **9108** ist für das Veröffentlichen der Komponenten oder Aufgaben, die ihr durch den Orchestrator **9112** zugewiesen wurden, verantwortlich.

[0442] Ein Netz-Client **9128** kann sich auf jeder Vorrichtung oder jedem Knoten **9118** oder **9120** in der Netzdomain **9102** oder **9104** befinden. Er kann bei der NDC **9108** oder einem anderen Dienstkoordinator registriert sein, um Informationen über den Knoten **9118** oder **9120** und jegliche angeschlossene Elemente in der Art von Sensoren, Kameras, Betätigungselementen und dergleichen bereitzustellen. Der Informationstyp, den er bereitstellt, kann Funktionsweise- und Systemtelemetrieinformationen in der Art von Strom-, Funktionsweise- und Zuverlässigkeitsmessungen einschließen. Der Netz-Client **9128** ermöglicht auch die Steuerung durch die NDC **9108** oder einen anderen Dienstkoordinator, um den Betrieb oder die Konfiguration des Knotens **9118** oder **9120** zu ändern und zu gewährleisten, dass Funktionsweisekriterien erfüllt sind. Beispielsweise kann ei-

ne NDC **9108** das Tastverhältnis zum Sammeln von Daten von einem angeschlossenen Sensor modifizieren. Die NDC **9108** kann auch die Netz- und Transporteinstellungen des Endknotens **9118** oder **9120**, der mit der Netzdomain **9102** oder **9104** kommuniziert, in der Art eines mit Bezug auf die **Fig. 3** und **Fig. 4** beschriebenen Gateways **310** konfigurieren. Der Netz-Client **9118** kann sich beim geteilten virtuellen Ablagespeicher **9124** anmelden oder diese auf jegliche Netzdienstelemente, die er vervollständigen kann, abfragen.

[0443] Der virtuelle geteilte Ablagespeicher **9124** kann eine Liste aller Aufgaben aufweisen, beispielsweise Netzdienstelemente, die eine Ausführung erfordern. Ein Knoten **9118** oder **9120** kann seine Fähigkeit ankündigen, eine Aufgabe auszuführen, und die Aufgabenzuweisung anfordern. Die NDC **9108** führt ein Nachschlagen nach dem anfordernden Knoten **9118** oder **9120** aus, um zu gewährleisten, dass er nicht zuvor verletzt wurde oder eine Funktion nicht ausgeführt hat. Falls die NDC **9108** entscheidet, die Aufgabe dem Knoten **9118** oder **9120** zuzuweisen, markiert sie die Aufgabe im virtuellen geteilten Ablagespeicher **9124** als zugewiesen. Der virtuelle geteilte Ablagespeicher **9124** kann Teil der Datenbank **9122** oder ein autonomes System sein.

[0444] Der Dienst und das Netzdienstelement sind nicht auf einen einzelnen Knoten **9118** oder **9120** oder auch nur auf eine einzelne Domain **9102** oder **9104** beschränkt. Beispielsweise kann ein Dienst eine Fog-Vorrichtung **9130** sein, der die Knoten **9118** und **9120** in beiden Domains **9102** und **9104** zugewiesen sind. Wie dargestellt ist, kreuzt die Fog-Vorrichtung **9130** mehrere Domains **9102** und **9104** und wird unter der Anleitung der NDC **9108** in der ersten Domain **9102** und der NDC **9108** in der zweiten Domain **9104** für die Knoten **9118** und **9120** bereitgestellt. Auf eine dritte Netzdomain **9132** kann über die Cloud **302** zugegriffen werden, und sie kann beispielsweise eine Datenbank **9134** zum Bereitstellen einer langzeitigen Speicherung von Daten als Netzdienstelement aufweisen. Die Komponenten in der Art der Knoten **9118** oder **9120** und der Datenbank **9134**, die sich in anderen Domains **9102**, **9104** oder **9132** befinden, können durch den Orchestrator **9112** identifiziert werden, und sie können in eine geteilte virtuelle Domain aufgenommen werden, um Ressourcen zu teilen, wie mit Bezug auf die **Fig. 110** - **Fig. 114** beschrieben wird.

[0445] Die Netzdienst-Overlays **9116** können in einem geteilten Ablagespeicher **9136** von Aufgaben und Komponenten gespeichert werden, die auch andere vom Orchestrator **9112**, von der NDC **9108** oder von anderen Komponenten angeforderte Elemente aufweisen können. Zusätzlich dazu, dass Netzdienst-Overlays **9116** an Knoten **9118** und **9120** übergeben werden, um eine Fog-Vorrichtung **9130** zu bilden, können die Knoten **9118** und **9120** auch Netzdienst-Overlays **9116** anfordern oder ziehen, um eine Aufgabe zu beenden, wie ein Netzdienstelement, wofür sie Code oder andere Konfigurationsinformationen benötigen.

[0446] **Fig. 92** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **9200** zur Behandlung neuer Anforderungen für einen Dienst gemäß einigen Ausführungsformen. Das Verfahren **9200** aus **Fig. 90** kann durch die mit Bezug auf **Fig. 94** beschriebene IoT-Vorrichtung **9400** implementiert werden. Das Verfahren **9200** beginnt in Block **9202**, wenn die Orchestrierungsanforderung empfangen wird, beispielsweise an einer Netzdomain-Steuereinrichtung oder einem anderen Dienstkoordinator. In Block **9204** wird festgestellt, ob die Dienst-anforderung neu ist, um beispielsweise einen neuen Dienst oder eine neue Fog-Vorrichtung zu bilden. Falls dies nicht der Fall ist, wird in Block **9206** die Orchestrierungsanforderung an einen existierenden Dienstkoordinator übergeben. Beispielsweise kann die Dienstanforderung eine Anforderung von Daten oder Informationen sein, wobei es sich gegenwärtig um einen Zweck des Dienstes oder der Fog-Vorrichtung handelt, oder sie kann die Fog-Vorrichtung einem anderen Zweck zuweisen, um andere Informationen bereitzustellen. Falls dies der Fall ist, kann der Dienstkoordinator den Dienst durch Hinzufügen oder Fortnehmen von Knoten modifizieren. Ferner kann der Dienstkoordinator oder können Dienstkomponten das Herunterladen von Netzdienst-Overlays anfordern, um die Vervollständigung von Netzdienstelementen zu ermöglichen.

[0447] Falls sich die Orchestrierungsanforderung auf einen neuen Dienst bezieht, kann in Block **9208** ein Dienstkoordinator identifiziert werden. Der Dienstkoordinator kann eine NDC sein, die sich in einer Domain befindet, welche sich auf die Dienstanforderung bezieht, beispielsweise die NDC, welche die größte Anzahl von Knoten bedient, die Informationen für die Dienstanforderung bereitstellen würden.

[0448] In Block **9210** kann ein Dienstmodell präpariert werden. Das Dienstmodell kann als virtuelle Teilleiste für eine Fog-Vorrichtung oder einen Dienst, die oder der zu verwenden ist, um die Dienstanforderung zu erfüllen, angesehen werden. Das Dienstmodell kann identifizieren, welche Typen von Netzdienstelementen, Endknoten und anderen Dienstanbietern für den Dienst benötigt werden. Das Dienstmodell kann am Dienstkoordinator gebildet werden oder an einem Orchestrator präpariert und zum Dienstkoordinator heruntergeladen werden. In Block **9212** kann der Dienstkoordinator die Netzdienstelemente präparieren. Dies können die Teile des Dienstes sein, welche die spezifischen Datenanforderungen, Aktionen und dergleichen identifizieren.

Die Netzdienstelemente können bereits in einem Datenspeicher am Dienstkoordinator vorhanden sein oder Netzdienst-Overlays sein, die aus einem anderen Speicher in der Art der Cloud gezogen werden.

[0449] In Block **9214** kann der Dienstkoordinator Kandidatendienstkomponenten identifizieren, wie einzelne Endpunktknoten, Datenquellen, Code und dergleichen, die in der Lage sind, spezifische Netzdienstelemente bereitzustellen. Die einzelnen Endpunktknoten können IoT-Vorrichtungen sein, die ihre Identität und ihre Fähigkeit bei der NDC registriert haben, wie mit Bezug auf **Fig. 93** beschrieben. In Block **9216** kann der Dienstkoordinator Subskriptionsanforderungen für Netzdienstelemente an die Dienstkomponenten versenden, die identifiziert wurden.

[0450] In Block **9218** kann die Dienstkomponente die Subskriptionsanforderung validieren. Dies kann durch Vergleichen der Dienstanforderung mit den Sensoren und anderen Vorrichtungen, die in der Dienstkomponente vorhanden und betriebsfähig sind, geschehen, um zu gewährleisten, dass die Dienstkomponente in der Lage ist, das Netzdienstelement in der Dienstanforderung auszuführen. In Block **9220** wird festgestellt, ob die Dienstanforderung unterstützt wird. Falls dies nicht der Fall ist, wird in Block **9222** eine Ablehnungsnachricht zum Dienstkoordinator gesendet. Der Dienstkoordinator kann dann die Dienstkomponente aus der Liste von Vorrichtungen entfernen, die in der Lage sind, dieses Netzdienstelement zu erfüllen, und nach einer anderen Vorrichtung suchen, die in der Lage ist, das Netzdienstelement bereitzustellen.

[0451] Falls die Dienstkomponente in der Lage ist, die Dienstanforderung durch Bereitstellen der Daten oder Aktionen für das Netzdienstelement zu erfüllen, kann sie in Block **9224** eine Bestätigungsnachricht zum Dienstkoordinator senden und sie zur Liste von Vorrichtungen hinzufügen. Wie hier beschrieben, kann eine Blockchain-Transaktion verwendet werden, um die Dienstkomponente in einer Transaktion aufzuzeichnen, und kann eine Gruppenidentifikation erteilt werden, um zu ermöglichen, dass die Dienstkomponente als Teil der Gruppe kommuniziert. Die Dienstkomponente kann einen Netzdienst-Overlay aufweisen, um das Netzdienstelement in einem lokalen Speicher zu implementieren, oder den Netzdienst-Overlay vom Dienstkoordinator oder von einem Speicher in der Cloud herunterladen.

[0452] In Block **9226** kann die Dienstkomponente die Aktion für das Netzdienstelement ausführen. Dies kann die Sammlung von Daten von einem Sensor in der Art der Temperatur, der Windgeschwindigkeit, des Niederschlags und dergleichen in Zusammenhang mit der Dienstkomponente sein. Bei einigen Beispielen kann das Netzdienstelement durch die Dienstkomponente vervollständigt werden, die eine Aktion ausführt, wie das Ein- oder Ausschalten einer Lampe, das Aktivieren eines Kompressors zum Verringern der Temperatur und dergleichen.

[0453] In Block **9228** gibt die Dienstkomponente Daten oder eine Bestätigung an den Dienstkoordinator zurück. Dies können die Daten in Zusammenhang mit einer Sensorablesung oder eine Bestätigung, dass eine Aktion ausgeführt wurde, sein.

[0454] **Fig. 93** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **9300** zum Registrieren einer Endpunkt- oder Dienstkomponente bei einer NDC oder einem anderen Dienstkoordinator gemäß einigen Ausführungsformen. Das Verfahren **9300** aus **Fig. 93** kann durch die mit Bezug auf **Fig. 94** beschriebene IoT-Vorrichtung 9400 implementiert werden. Block **9302** repräsentiert beispielsweise den Fall, in dem eine Dienstkomponente in der Art einer IoT-Vorrichtung oder eines Endpunktknotens einen lokalen Dienstkoordinator nachschlägt. Hierbei kann es sich um eine NDC handeln, die in der Netzdomain arbeitet, welche die Dienstkomponente aufweist. In Block **9304** sendet die Dienstkomponente eine Verbindungsanforderung zum Dienstkoordinator. Nach dem Empfang einer Bestätigung vom Dienstkoordinator kann die Dienstkomponente in Block **9306** einen geteilten Schlüssel oder andere identifizierende Informationen in der Art eines von einer Blockchain erzeugten Schlüssels zum Dienstkoordinator senden. Nach dem Empfang einer Bestätigung, dass die Dienstkomponente am lokalen Dienstkoordinator registriert wurde, kann die Dienstkomponente in Block **9308** dem Dienstkoordinator die Vorrichtungssperipheriedaten, beispielsweise von angeschlossenen Sensoren, Betätigungselementen und dergleichen, senden. In Block **9310** wird festgestellt, ob die Dienstkomponente noch registriert ist. Falls dies nicht der Fall ist, kann der Prozessfluss zu Block **9302** zurückkehren, um die Vorrichtung erneut zu registrieren. In Block **9312** kann eine Subskriptionsanforderung von der Dienstkomponente empfangen werden. Sobald die Dienstkomponente auf die Subskription eingewirkt hat, kann sie zu Block **9312** zurückkehren, um festzustellen, ob die Vorrichtung noch registriert ist. Falls die Dienstkomponente nicht mehr registriert ist, kann der Prozessfluss zu Block **9302** zurückkehren, um den Prozess zu wiederholen.

[0455] **Fig. 94** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 9400 zum Koordinieren oder Erfüllen von Dienstanforderungen vorhanden sein können, gemäß einigen Ausführungsfor-

men. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3**, **Fig. 8** und **Fig. 91** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 9400 andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können. Die IoT-Vorrichtung 9400 kann ein Orchestrator, eine NDC, ein Endpunktnoten oder eine Funktion als Kombination dieser Systeme sein.

[0456] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0457] Der Massenspeicher **808** kann einen Orchestrator **9112** zur Übermittlung von Dienstanforderungen an andere Einheiten in der Art von Dienstkoordinatoren aufweisen. Eine Datenbank **9122** kann Daten, Metadaten und Ressourcen von an einer speziellen Domain registrierten Knoten, einschließlich angeschlossener Vorrichtungen und Fähigkeiten, speichern. Ein virtueller geteilter Ablagespeicher **9124** kann verwendet werden, um Netzdienstelemente bekannt zu machen, die Tätigkeiten erfordern, und Identitäten von Dienstkomponenten, die Netzdienstelemente bereitstellen, speichern. Eine Maschinenlern-Engine **9126** kann verwendet werden, um auszuwählen, welche Dienstkomponenten in der Art von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** verwendet werden können, um die Anforderungen des Dienstes zu erfüllen. Ein Client **9128** kann sich beim Dienstkoordinator registrieren und Informationen über angeschlossene Vorrichtungen und Fähigkeiten bereitstellen. Der Client **9128** kann die Fähigkeit der IoT-Vorrichtung 9500, ein Netzdienstelement **9402** zu erfüllen, ankündigen. Der Client **9128** kann auf eine Dienstanforderung mit einer Bestätigung antworten, dass die IoT-Vorrichtung 9400 die Aktionen für das Netzdienstelement **9402** ausführen kann, oder eine Ablehnung senden, die den Dienstkoordinator informiert, dass sie die Aktionen nicht ausführen kann. Der Client **9128** kann auf den Dienstkoordinator zugreifen, um jegliche Netzdienst-Overlays zu erhalten, die erforderlich sind, um das Netzdienstelement **9402** auszuführen, oder er kann direkt auf einen Speicher in der Cloud **302** zugreifen, um die benötigten Netzdienst-Overlays herunterzuladen.

[0458] **Fig. 95** ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **9500**, das Code aufweist, um einen Prozessor **902** oder Prozessoren anzuweisen, Dienstanforderungen zu koordinieren oder zu erfüllen, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **9500** zugreifen. Das nichtflüchtige maschinenlesbare Medium **9500** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0459] Das nichtflüchtige maschinenlesbare Medium **9500** kann Code **9502** aufweisen, um den Prozessor **902** anzuweisen, einen Dienstkoordinator in der Art einer Netzdomain-Steuereinrichtung in der lokalen Domain zu identifizieren. Ein Code **9504** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Netzdienstelemente für eine Dienstanforderung zu präparieren. Ein Code **9506** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Kandidatendienstkomponenten zu identifizieren, die spezifische Netzdienstelemente bereitstellen können. Ein Code **9508** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Subskriptionsanforderung zu validieren. Ein Code **9510** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Aktion für ein Netzdienstelement auszuführen. Ein Code **9512** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Daten oder eine Bestätigung an einen Dienstkoordinator zurückzugeben. Ein Code **9514** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Verbindungsanforderung zum Dienstkoordinator zu senden. Ein Code **9516** kann aufgenommen sein, um den Prozessor **902** anzuweisen, dem Dienstkoordinator die Vorrichtungssperipheriedaten in der Art von Daten von angeschlossenen Sensoren, Stellgliedern und dergleichen zu senden. Ein Code **9518** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Subskriptionsanforderungen zu anderen Einheiten zu senden. Es sei bemerkt, möglicherweise dieser Einheiten in jeder Vorrichtung vorhanden sein können. Beispielsweise kann ein Endpunktnoten nicht als ein Dienstkoordinator oder Orchestrator wirken und würde bei diesem Beispiel nicht die Codeblöcke **9502**, **9504**, **9506** und **9518**, die diese Funktionen ausführen, aufweisen.

[0460] Mesh-Netze sind in Umgebungen nützlich, in denen ein Akteur mit sehr geringer Latenz Zugriff auf Daten wünschen kann oder wenn die Daten in einer zeitlichen oder räumlichen Weise eng miteinander gekoppelt sein könnten. Um das Beispiel einer Verkehrskreuzung zu verwenden, wie mit Bezug auf **Fig. 4** erörtert, sei bemerkt, dass ein Datenaustausch an Kreuzungen zwischen Straßenbenutzern in der Art von Fahrzeugen, Fußgängern, Fahrradfahrern und Verkehrsampeln oder anderen straßenseitigen Einheiten mit einem sehr hohen Volumen, einer sehr hohen Geschwindigkeit und einer sehr hohen Vielfalt auftreten kann. Die Umgebung

der Kreuzung kann bei sich schnell bewegendem Fahrzeugverkehr, einer minimalen festen Infrastruktur und Signalausbreitungsbedingungen, die schwierig sein können, besondere Herausforderungen bieten.

[0461] Ein Ansatz zur Verbesserung der Leistungsfähigkeit und zum Verbessern der Latenz besteht darin, eine Netzdatenspeicherung und Inhaltsauslieferung ganz am Rand des Netzes auszuführen. Dies kann das Caching von Daten, die einen sehr schnellen Zugriff erfordern, in Vorrichtungen, die in der Lage sind, die Daten zu halten, und Anwendungen, welche die Daten benötigen, nahe sind, einschließen.

[0462] Eine Kreuzung kann eine Anzahl von Anwendungen zu Verkehrsregelungen und anderen Zwecken verwenden. Die Anwendungen können unter anderem Kollisionsverhinderungssysteme, Notdienste, die Modifikation von Operationen auf der Grundlage von Umgebungsdaten, Einzelhandels- und Werbedienste einschließen. Ein großer Anteil dieser Daten ist zeitlich und räumlich abhängig. Beispielsweise sind die Daten ein Verkehrsstau eng mit dem Ort und der Zeit gekoppelt. Überdies sollen die Daten häufig von nahen Gruppen in der Art von Verkehrsregelungssystemen, Fahrzeugen, Radfahrern, Fußgängern, Polizei, Notdiensten und dergleichen verbraucht werden.

[0463] Infolge der IP-Adressbestimmungs-, Framing-, Handshaking- und anderer Kommunikationsanforderungen können IP-basierte Technologien in Umgebungen schlecht funktionieren, in denen die Kontaktzeit zwischen Vorrichtungen sehr gering sein kann. Die Datensemantik und andere hier beschriebene Technologien können ein Mittel zum Vereinfachen des Zugriffs und der Verwendung von Daten bereitstellen. Die Technologien können es Benutzern von Daten ermöglichen, auf die Daten von Vorrichtungen in der unmittelbaren Nähe zuzugreifen. Fahrzeuge können auch als Träger zum Bewegen von Daten, beispielsweise zwischen verschiedenen Kreuzungen und Systemen, wirken. Eine Technik, die für diese Kommunikationstypen verwendet werden kann, ist die Verwendung verteilter Hash-Tabellen zum Beschreiben, Speichern und Veröffentlichen von Daten in einem Netz.

[0464] Fig. 96 ist ein schematisches Diagramm 9600 der Ad-hoc-Bildung eines umgekehrten Verteilte-Hash-Tabelle- bzw. Distributed-Hash-Table(DHT)-Netzes für IoT-Dienste gemäß einigen Ausführungsformen. Die IoT-Dienste können das Senden oder Speichern von Daten einschließen. DHT-Netze können durch die Erzeugung eines eindeutigen Hashes zum Beschreiben oder Veröffentlichen einer Datei innerhalb eines Netzes als zu verschiedenen Knoten gesendete Fragmente gebildet werden. Knoten, die Fragmente der Datei empfangen haben, werden zur Quelle neuer Kopien der Datei. Knoten, die auf diese Datei zugreifen möchten, laden Dateifragmente von anderen Knoten herunter, bis sie die gesamte Datei haben. Wenn Knoten beginnen, Fragmente der Datei zu empfangen, können Sie diese Fragmente teilen oder auf andere Knoten, welche die Datei erhalten möchten, seeden. Dies erzeugt ein Ad-hoc-Netz ohne zentrale Steuerung, das in der Lage ist, viele Kopien einer Datei über die Peers in einem Netz zu verteilen und es neuen Peers zu ermöglichen, die Dateien schnell zu erhalten, wenn sie Fragmente von vielen statt einer Quelle herunterladen.

[0465] Bei einigen Aspekten kann eine Umkehrversion hiervon auch auf die Datenübertragung angewendet werden, um die Wahrscheinlichkeit eines Datenverlusts während Ereignissen zu verringern. Bei diesem Beispiel erkennt ein Sensorknoten in der Art des Knotens 1 im Diagramm einen Ereignissturm, der ein hohes Datenvolumen erzeugt. Der Ereignissturm kann einem Notfall, einem hohen Verkehrsfluss an einer Kreuzung, einem verlängerten Zeitraum der Datensammlung und dergleichen entsprechen. Der Knoten 1 sendet die Daten über einen ersten Kommunikationskanal 9602 zur Cloud 302 oder zu einem Fog-Knoten, einem Gateway oder einer anderen nachgeschalteten Datensenke. Der erste Kommunikationskanal 9602 hat jedoch eine begrenzte Netzbandbreite und wird schnell verstopft, so dass ein Rückstau von Nachrichten, die sich am Knoten 1 aneinander reihen, erzwungen wird. Es kann eine erhebliche Zeit in Anspruch nehmen, um die Warteschlange zu leeren, wodurch Nachrichten zum Ziel verzögert werden oder sogar der Verlust von Daten hervorgerufen wird, falls die Warteschlange überläuft.

[0466] Bei den für ein Umkehr-DHT-Netz verwendeten Techniken kann der Knoten 1 jedoch die Netzkapazität seiner Nachbarknoten verwenden, um eine zu hohe Nachrichtenlast zum selben Endziel zu senden. Der Knoten 1 kann die Nachbarknoten 1 ... n beispielsweise über staufreie Netzverbindungsstrecken 9606 entdecken. Für die Peer-Entdeckung können verschiedene Techniken verwendet werden, einschließlich Mesh-Netzprotokollen in der Art der Spezifikationen vom Openfog-Konsortium, der Alljoyn-Spezifikation und anderer, welche verschiedene Open-Source-Ad-hoc-Netzprotokolle einschließen. Ferner weisen die IP-Protokolle für IPv6 eine native Unterstützung zur Peer-Entdeckung auf, wie es bei anderen Netzprotokollen der Fall ist.

[0467] Die staufreien Netzverbindungsstrecken 9606 können eine Anzahl von Netzverbindungsstrecken unter anderem in der Art von WiFi oder Bluetooth einschließen. Bei einigen Beispielen, wie mit Bezug auf Fig. 3

beschrieben, können die Knoten durch ein festverdrahtetes Netz gekoppelt sein, wenngleich jeder einzelne Knoten auch eine Drahtlosverbindungsstrecke zu einem Gateway aufweisen kann. Physikalisch nahe Knoten können Verbindungen mit höherer Bandbreite als ferne aufweisen, so dass ein überlasteter Knoten Dateifragmente zu umgebenden Peers senden kann und sie veranlassen kann, die Nachrichten zu ihrem Ziel weiterzuleiten.

[0468] Die Datensenke, die sich bei diesem Beispiel in der Cloud **302** befindet, kann die empfangenen Nachrichten dem Knoten, der sie gesendet hat, bestätigen, wobei beispielsweise, falls Knoten 2 einen Teil der Daten zur Datensenke sendet, die Datensenke Knoten 2 den Empfang bestätigen kann. Knoten 2 würde dann Knoten 1, der Datenquelle, den Empfang der Daten durch die Datensenke bestätigen. Sobald Knoten 1 eine Bestätigungsnachricht (ACK) von einer Quelle in der Art des Knotens 2 oder anderer Peers bei der Auslieferung einer bestimmten Nachricht empfängt, kann er die Nachricht als ausgeliefert ansehen und sie aus der Warteschlange entfernen.

[0469] Die Rate, mit der die ACK von der Datenquelle zurück empfangen werden, kann eine Flusssteuerung ermöglichen. Beispielsweise würde, falls Knoten 2 jede Sekunde eine Bestätigung sendet und Knoten 3 alle zwei Sekunden eine Bestätigung sendet, dieser angeben, dass Knoten 2 in der Lage ist, eine höhere Nachrichtenlast zu behandeln, so dass Knoten 1 seine Arbeitsweise anpassen würde, um mehr Nachrichten zu Knoten 2 als zu Knoten 3 zu senden.

[0470] Der Quellsensorknoten oder Knoten 1 beim Beispiel kann den Nachrichtenfluss verfolgen und Mechanismen implementieren, um Peers Rechnung zu tragen, die eine Nachricht nicht ausliefern. Falls beispielsweise ein Peer-Knoten in der Art von Knoten 4 keine ACK für eine Nachricht zurückgibt, die er gesendet hat, kann der Quellknoten das Leiten von Nachrichten zu ihm während eines konfigurierbaren Zeitraums unterbrechen. Ferner können jegliche zu Knoten 4 gesendete Nachrichten als verloren gegangen angesehen werden und durch andere Peers erneut gesendet werden oder direkt vom Sensorknoten zur Vorrichtung in der Cloud **302** gesendet werden.

[0471] Fig. 97 ist ein schematisches Diagramm **9700** eines Prozesses zur Verfolgung, welche Knoten für das Speichern oder Übertragen von Dateidaten **9702** verwendet werden können, gemäß einigen Ausführungsformen. Die Dateidaten **9702** können in Fragmente **9704 - 9708** zerlegt werden. Eine Hash-Funktion **9710 - 9714** kann an jedem der Fragmente **9704 - 9708** ausgeführt werden, um einen Schlüssel **9716 - 9720** zu erzeugen. Der Schlüssel **9716 - 9720** kann dann verwendet werden, um festzustellen, welcher Knoten zum Speichern oder Senden eines Datenfragments in einem dispergierten Mesh **9722** von Knoten verwendet werden sollte.

[0472] Fig. 98 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **9800** zur Vorgabe von Speicher- oder Sendeknoten gemäß einigen Ausführungsformen. Das Verfahren **9800** aus Fig. 90 kann durch die mit Bezug auf Fig. 100 beschriebene IoT-Vorrichtung **10000** implementiert werden. Block **9802** repräsentiert beispielsweise den Fall, in dem ein Knoten eine Datei zum Speichern oder zur Übertragung erzeugt. In Block **9804** wird eine neue Dateispeicher- oder -übertragungsanforderung für die Datei erzeugt. In Block **9806** wird die Datei, beispielsweise abhängig von der Nutzinhaltsgröße von Paketen oder Frames, die zur Übertragung der Daten verwendet werden, in N Fragmente unterteilt. In Block **9808** kann ein Hash für jedes Fragment berechnet werden.

[0473] In Block **9810** kann der Ziel-Speicher- oder Übertragungsknoten für jedes Fragment identifiziert werden. Dies kann durch Extrahieren der ersten M Bytes aus dem Hash-Code für das Fragment geschehen, wobei die Anzahl M durch die Länge der im Mesh-Netz verwendeten Knotenkennungen bestimmt werden kann. Die Bytes vom Hash können dann zur Bestimmung eines Zielknotens durch Identifizieren des Zielknotens als beste Übereinstimmung zwischen den ersten M Bytes der Knotenkennung und den ersten M Bytes des Dateifragment-Hashes verwendet werden. Die Knotenkennung kann dann einen Teil der Dateiadresstabelle bilden. Mit bekannter Knotenkennung kann der Bereich von Dateifragmenten bestimmt werden, der für das Speichern oder Übertragen verantwortlich ist. Für Knoten mit gut übereinstimmenden Kennungen kann die Technik in Fällen Redundanz einbauen, in denen die Knotenverfügbarkeit nicht garantiert ist, beispielsweise in flüchtigen Netzumgebungen. In Block **9812** wird das Fragment, in ein Paket oder einen Frame eingepackt, zum Ziel-Speicher- oder Übertragungsknoten gesendet.

[0474] In Block **9814** wird festgestellt, ob der Prozess fortgesetzt werden soll. Der Prozess kann ändern, wenn die Datenspeicherung abgeschlossen wird, oder er kann warten, bis ACK für alle von anderen Knoten gesendete Fragmente empfangen werden.

[0475] Fig. 99 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 9900 zum Speichern oder Übertragen von Daten unter Verwendung einer verteilten Hash-Tabelle bzw. Distributed Hash Table (DHT) gemäß einigen Ausführungsformen. Das Verfahren 9900 aus Fig. 99 kann durch die mit Bezug auf Fig. 100 beschriebene IoT-Vorrichtung 10002 implementiert werden. Block 9902 repräsentiert beispielsweise den Fall, in dem eine Vorrichtung eingeschaltet wird und sich einem Ad-hoc-Netz anschließt. In Block 9904 wird eine Knotenkennung mit einer Länge von n Bits berechnet oder erhalten. Dies kann unter anderem eine von einer Blockchain zugewiesene Kennung, wie hier beschrieben, eine von einem Hersteller zugewiesene Kennung oder eine von einem Zufallszahlengenerator berechnete Kennung sein.

[0476] In Block 9906 wird ein Wartezeitraum für eine eingehende Speicher- oder Übertragungsanforderung implementiert. Sobald der Wartezeitraum abgeschlossen ist, wird in Block 9908 festgestellt, ob Daten zur Speicherung oder Übertragung empfangen wurden. Falls dies nicht der Fall ist, kehrt der Prozessfluss für einen weiteren Wartezeitraum zu Block 9906 zurück.

[0477] Falls in Block 9908 Daten empfangen wurden, wird in Block 9910 ein Schlüssel für die empfangenen Daten erzeugt. Dies kann durch Berechnen einer Hash-Funktion für die empfangenen Daten geschehen.

[0478] In Block 9912 werden der Schlüssel und die Daten lokal gespeichert oder übertragen. Dies kann durch Speichern oder Übertragen der Daten im Knoten und anschließendes Hängen der Knotenkennung vor den Schlüssel geschehen. Der sich ergebende Schlüssel kann in einem lokalen Speicher gespeichert werden, beispielsweise in einer Liste, einer Tabelle, einer Warteschlange oder einer Datenbank sowie anderen Datenstrukturen.

[0479] In Block 9914 wird festgestellt, ob der Prozess fortgesetzt werden soll. Dies kann auf einer Feststellung beruhen, ob mehr Daten für die aktuelle Sequenz erwartet werden. Falls dies der Fall ist, kehrt der Prozessfluss zu Block 9908 zurück, um festzustellen, ob Daten zur Speicherung oder Übertragung empfangen wurden. Falls dies nicht der Fall ist, endet der Prozess in Block 9916.

[0480] Fig. 100 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 10000 zum Koordinieren oder Erfüllen von Dienstanforderungen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 10000 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können. Die anderen Knoten können Mesh-Vorrichtungen 812, Vorrichtungen in der Cloud 302 oder beide aufweisen.

[0481] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher 808 kann einen Datenfragmentierer 10002 zum Fragmentieren einer Datendatei, so dass sie in das Nutzinhaltfeld von Paketen oder Frames passt, aufweisen. Eine Hash-Berechnungseinrichtung 10004 kann die Hash-Schlüssel berechnen, um Speicher- und Übertragungsknoten für die Daten zu identifizieren.

[0482] Ein Nachrichtengenerator 10006 kann die Hash-Schlüssel verwenden, um Knotenkennungen zum Speichern oder zur Übertragung der Daten zu bestimmen. Der Nachrichtengenerator 10006 kann auch ein Datenfragment zum Senden zu einem anderen Knoten zum Speichern oder zur Übertragung formatieren, beispielsweise das Datenfragment in ein Nutzinhaltfeld eines Pakets oder Frames packen.

[0483] Ein Kommunikator 10008 kann das verpackte Datenfragment zum Speichern oder zur Übertragung zu einem anderen Knoten senden. Zur Übertragung kann der Kommunikator 10008 auch eine Bestätigung von einem anderen Knoten in der Art einer Mesh-Vorrichtung 812 empfangen und feststellen, ob die Bestätigung von einem vorgeschalteten Ziel in der Art einer Vorrichtung in der Cloud 302 kam. Ein Daten-Tracker 10010 kann die Bestätigungen verwenden, um festzustellen, ob Daten vom sendenden Knoten zur Zielvorrichtung gesendet worden sind oder erneut gesendet werden müssen. Der Daten-Tracker 10010 kann auch eine Flusssteuerung implementieren, beispielsweise auf der Grundlage der Rate von anderen Knoten eingehender Bestätigungen. Zur Speicherung kann ein Datenspeicher 10012 einen Schlüssel mit dem Ort und der Identität eines Datenfragments speichern. Der Schlüssel kann den Hash-Code für das Fragment vor die Kennung des Knotens oder der Mesh-Vorrichtung 812, welche die gespeicherten Daten hält, hängen.

[0484] Fig. 101 ist ein Blockdiagramm eines beispielhaften nichtflüchtigen maschinenlesbaren Mediums **10100**, das Code aufweist, um einen Prozessor **902** oder Prozessoren anzuweisen, Dienstanforderungen zu koordinieren oder zu erfüllen, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 9 beschrieben wurden. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **10100** zugreifen. Das nichtflüchtige maschinenlesbare Medium **10100** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0485] Das nichtflüchtige maschinenlesbare Medium **10100** kann einen Code **10102** aufweisen, um den Prozessor **902** anzuweisen, eine Datei in Fragmente zu segmentieren. Ein Code **10104** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Hash-Code für jedes Fragment zu berechnen. Ein Code **10106** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Ziel-Speicher- oder Übertragungsknoten für jedes Fragment zu identifizieren. Ein Code **10108** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Datenfragment zu einem Zielknoten zu senden. Ein Code **10110** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Knotenkennung zu berechnen. Ein Code **10112** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Schlüssel für die Speicherung zu erzeugen. Ein Code **10114** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Schlüssel und den Hash zu speichern.

[0486] Die Kommunikationen durch die verschiedenen Knoten brauchen nicht über die gleichen Typen von Kommunikationskanälen zu erfolgen. Verschiedene Kommunikationswege, -frequenzen und dergleichen können abhängig von den Anwendungsanforderungen verwendet werden. Wie mit Bezug auf die Fig. 102 bis Fig. 105 erörtert, können geeignete Verkehrswege für Daten ausgewählt werden. Die Techniken ermöglichen die Auswahl verschiedener Wege oder Kombinationen von Wegen für die Verwendung bei der Adressierung von Anwendungsanforderungen, die von einer hohen zu einer geringen Latenz und von verbindungsorientiert zu verbindungslos reichen. Dies erweitert Transportauswahl- und -verwendungskriterien über das Mehrweg-Übertragungssteuerprotokoll bzw. Transmission Control Protocol (TCP) hinaus.

[0487] Fig. 102 ist ein schematisches Diagramm eines Mehrweg-Kommunikationssystems, worin drei beispielhafte Wege zwischen zwei Endpunkten **10202** und **10204**, die für eine mögliche Verwendung verfügbar sein können, dargestellt sind, gemäß einigen Ausführungsformen. Bei diesem Beispiel umfassen die Wege einen Satelliten-Uplink **10206**, einen Internet-Backbone oder eine drahtgestützte Verbindung **10208** und eine LTE-Verbindung **10210**. Es kann jedoch eine beliebige Anzahl drahtgestützter oder drahtloser Verbindungen zwischen den beiden Endpunkten verwendet werden, einschließlich jeder der mit Bezug auf Fig. 8 erörterten Funkverbindungen sowie optischer Faser-, Mikrowellen- und anderer Verbindungen.

[0488] Die Auswahl des Datenwegs kann von der zu übertragenden Datenmenge, der Zuverlässigkeit des Datenwegs, der Geschwindigkeit der Kommunikationen und dergleichen abhängen. Falls beispielsweise eine festverdrahtete Verbindung **10208** verloren geht oder nicht verfügbar ist, kann ein Endpunkt **10202** einen alternativen Kommunikationsweg **10206** oder **10210** auf der Grundlage der Anwendungsanforderungen auswählen.

[0489] Fig. 103 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **10300** zur Auswahl eines Kommunikationswegs gemäß einigen Ausführungsformen. Das Verfahren **10300** aus Fig. 103 kann durch die mit Bezug auf Fig. 104 beschriebene IoT-Vorrichtung **10400** implementiert werden. Es sei bemerkt, dass mehrere Kommunikationswege parallel verwendet werden können. Block **10302** repräsentiert beispielsweise unter anderem den Fall, in dem eine Vorrichtung eingeschaltet wird oder ein Netzdienst-Overlay, das die Verwendung mehrerer Kommunikationswege ermöglicht, heruntergeladen wird.

[0490] In Block **10306** werden an der Vorrichtung verfügbare Netzschnittstellen entdeckt. Dies kann unter Verwendung von Konfigurationsbefehlen in der Art von Ifconfig oder ipconfig geschehen, die verwendet werden können, um angeschlossene Netzschnittstellen aufzulisten. Der erste, Ifconfig, ist ein Unix-Befehl, der von einigen Betriebssystemen verwendet werden kann, um Netzschnittstellen zu initialisieren. Der zweite, ipconfig, ist ein Befehl, der bei einer Anzahl von Betriebssystemen verwendet werden kann, einschließlich beispielsweise Systemen von Apple, Microsoft und anderen, wodurch alle gegenwärtigen TCP/IP-Konfigurationswerte, einschließlich der Schnittstellen in einem System, angezeigt werden können.

[0491] In Block **10306** werden verfügbare Wege zwischen den Endpunkten über die aktiven Netzschnittstellen entdeckt. Die Anzahl der verfügbaren Wege kann infolge der möglicherweise verschiedenen Topologien, Sicherheitsberechtigungs-nachweise und Protokolle, die von den jeweiligen einzelnen Netzschnittstellen unterstützt werden, größer als die Anzahl der Netzschnittstellen sein. Dies kann beispielsweise durch Entdecken logischer Kanäle, Erhalten letzter bekannter aktiver Wege und dergleichen geschehen. Die Wege, die entdeckt

werden, können in einer Wegdatenbank gespeichert werden. Die Wegdatenbank kann sich unter anderem lokal an der Vorrichtung, auf einem fernen Server oder auf einer anderen Vorrichtung in einem Mesh-Netz befinden. Eine fern angeschlossene Datenbank kann von einer oder mehreren Vorrichtungen verwendet werden, die sich an der Vorrichtung oder einer anderen Vorrichtung in einem Mesh-Netz befinden kann.

[0492] In Block **10308** kann eine Rangbewertung der Wege ausgeführt werden. Der Rangbewertungsprozess kann eine Einordnung der Wege beispielsweise durch die erwartete oder historische Dienstqualität, Kapazität, Kosten, Latenz, einschließen. Die aktualisierten Weginformationen können in der Wegdatenbank gespeichert werden.

[0493] In Block **10310** wird festgestellt, ob eine Weiterleitungsanforderung empfangen wurde. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **10306** zurück. Falls eine Weiterleitungsanforderung empfangen wurde, wird in Block **10312** eine Weiterleitungsstrategie berechnet. Die Weiterleitungsstrategie kann durch lokale Richtlinienziele in der Art einer Bestes-QoS-Weiterleitung, einer Geringste-Kosten-Weiterleitung, einer Geringste-Latenz-Weiterleitung und dergleichen angeleitet werden.

[0494] Die Weiterleitungsstrategie kann in Block **10314** mit einer Feststellung beginnen, ob ein einziger bester Weg verwendet werden sollte. Falls dies der Fall ist, wird in Block **10316** der einzige Weg ausgewählt, beispielsweise auf der Grundlage gewünschter Wegeigenschaften in der Art der Kosten, Zuverlässigkeit, Latenz usw., wie anhand der Ranginformationen in der Weiterleitungsdatenbank bestimmt. Falls eine Mehrwegstrategie auszuwählen ist, können in Block **10318** mehrere Wege in der Weiterleitungsdatenbank für die Verwendung ausgewählt werden, beispielsweise auf der Grundlage der Gesamtdaten für die Übertragung, der Zuverlässigkeit und dergleichen.

[0495] In Block **10320** werden Pakete oder Frames für die Installation über die ausgewählten Wege präpariert. Dies kann durch Fragmentieren der Daten in Größen geschehen, die in die Nutzinhalt der Pakete oder Frames passen, die für die ausgewählten Wege verwendet werden. Die Fragmente werden dann in die Pakete oder Frames für die ausgewählten Wege gepackt.

[0496] In Block **10322** werden Pakete über die ausgewählten Wege zum Ziel gesendet. In Block **10324** können Funktionsweisestatistiken für die verschiedenen verwendeten Wege gesammelt werden, und in Block **10326** können die Wegrangbewertungen in der Weiterleitungsdatenbank auf der Grundlage der Funktionsweisestatistiken aktualisiert werden. Die Funktionsweisestatistiken können beispielsweise die Frage, ob die Übertragung auf einem gegebenen Weg erfolgreich war und die Paketfehlerrate, Latenz, Wegzuverlässigkeit, Anzahl von Neuversuchen, übertragenen Gesamtdaten und dergleichen für den Weg einschließen.

[0497] In Block **10328** wird festgestellt, ob der Prozess fortgesetzt werden soll. Beispielsweise kann ein Orchestrator oder Dienstkoordinator feststellen, dass die Mehrwegkommunikationen nicht mehr erforderlich sind. Falls Mehrwegkommunikationen fortgesetzt werden sollen, kehrt der Prozessfluss zu Block **10310** zurück, um auf eine andere Weiterleitungsanforderung zu warten. Falls dies nicht der Fall ist, kann das Verfahren **10300** in Block **10330** enden.

[0498] **Fig. 104** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **10400** zum Senden von Daten über mehrere Kommunikationskanäle vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **10400** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0499] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Wegbestimmung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0500] Der Massenspeicher **808** kann eine Wegentdeckungseinrichtung **10402** aufweisen, welche die vorhandenen Netzverbindungen identifiziert und die Wege oder Kommunikationskanäle von der IoT-Vorrichtung **10400** zu einer Endvorrichtung in der Art einer Mesh-Vorrichtung **812** oder einer Vorrichtung in der Cloud **302** bestimmt. Eine Weg-Rangbewertungseinrichtung **10404** kann die Wege, beispielsweise auf der Grundlage von Übertragungsraten, der Zuverlässigkeit und anderer Funktionsweisestatistiken, bewerten. Eine Wegdatenbank **10408** kann die Wege, Rangbewertungen und zugeordneten Funktionsweisestatistiken speichern. Die Weg-

datenbank **10408** kann verwendet werden, um andere verwandte Informationen zu speichern, wie Protokolle für Kommunikationskanäle, Zugangsinformationen und Berechtigungsnachweise für Kanäle und dergleichen.

[0501] Eine Wegberechnungseinrichtung **10406** kann einen Weg oder Wege zum Senden von Daten von der IoT-Vorrichtung **10400** zu einem Endpunkt bestimmen. Die Wegberechnungseinrichtung kann auf den Wegen gespeicherte Informationen und Rangbewertungen in der Wegdatenbank **10408** verwenden.

[0502] Eine Datenpräpariereinrichtung **10410** kann die Informationen von der Wegberechnungseinrichtung **10406** nehmen und Daten in der Art von Paketen oder Frames, die über den ausgewählten Weg oder die ausgewählten Wege zu senden sind, präparieren. Die Präparation kann das Fragmentieren der Daten, so dass sie in die Nutzinhaltsfelder von Paketen oder Frames in Zusammenhang mit den verschiedenen Wegen passen, und das Verpacken der Daten in den Paketen oder Frames einschließen. Ein Kommunikator **10412** kann die Pakete oder Frames über einen Sender in der Art des Mesh-Sendeempfängers **810** oder des Uplink-Sendeempfängers **814** oder über das Internet über eine Netzschnittstellen-Steuereinrichtung **816** zur Zielvorrichtung senden.

[0503] Eine Funktionsweise-Überwachungseinrichtung **10414** sammelt Funktionsweisedaten für die Kommunikationskanäle. Die Funktionsweise-Überwachungseinrichtung **10414** kann die in der Wegdatenbank **10408** gespeicherten Wegrangbewertungen aktualisieren. Die Funktionsweise-Überwachungseinrichtung **10414** kann auch bemerken, wenn ein Weg ausgefallen ist, beispielsweise durch Bemerken des Fehlens einer Bestätigung von einer Zielvorrichtung innerhalb eines vorgegebenen Zeitraums, und dann den Frame neu senden und den Weg als möglicherweise ungeeignet in der Wegdatenbank **10406** auszeichnen. Die Wegentdeckungseinrichtung **10402** kann den ausgezeichneten Weg periodisch prüfen, um festzustellen, ob er neu eingerichtet wurde.

[0504] Fig. 105 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **10500**, das Code aufweist, um einen Prozessor **902** anzuweisen, Daten über mehrere Kommunikationskanäle zu senden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **10500** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf Fig. 9 beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **10500** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0505] Das nichtflüchtige maschinenlesbare Medium **10500** kann einen Code **10502** aufweisen, um den Prozessor **902** anzuweisen, Netzschnittstellen zu entdecken. Ein Code **10504** kann aufgenommen sein, um den Prozessor **902** anzuweisen, verfügbare Wege zu einer Endpunktvorrichtung unter Verwendung der entdeckten Netzschnittstellen zu entdecken. Der Code **10504** kann durch verschiedene Funktionsweisedaten in der Art der Kommunikationsgeschwindigkeit, der Zuverlässigkeit und dergleichen eine Rangbewertung der Wege ausführen. Ein Code **10506** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Weiterleitungsstrategie für eine Datendatei über einen Weg oder eine Gruppe von Wegen, unter anderem beispielsweise unter Verwendung der Rangbewertungen, zu berechnen. Ein Code **10508** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Daten für die Installation zu präparieren, beispielsweise durch Fragmentieren einer Datendatei in Größen, die in Datenfelder für Pakete oder Frames in Zusammenhang mit den Kommunikationskanälen passen. Ein Code **10510** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Pakete oder Frames über den ausgewählten Weg oder die ausgewählten Wege zur Zielvorrichtung zu übertragen.

[0506] Ein Code **10512** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Wegfunktionsweisedaten zu sammeln, einschließlich der Zeit für das Empfangen und Bestätigen, des Nichtempfangens einer Bestätigung und dergleichen. Ein Code **10514** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Rangbewertungen auf der Grundlage der für die Wege gesammelten Funktionsweisedaten zu aktualisieren. Hier offenbarte Techniken können dabei helfen, eine Staubehandlung in IoT-Netzen und anderen Typen von Mesh-Netzen bereitzustellen.

[0507] Fig. 106 ist eine Schemazeichnung eines IoT-Gateways **10600** für sichere Kommunikationen und Übersetzungen zwischen Domains gemäß einigen Ausführungsformen. IoT-Netze sind häufig heterogen, wobei verschiedene Domains und Protokolle auf mehreren Schichten wirken. Für die Kommunikation über diese Domains können IoT-Frameworks Sitzungs- und Anwendungsebenenprotokoll- und Datenmodelle definieren. Der Ende-zu-Ende-Datenschutz verwendet jedoch Verschlüsselungs- und Signaturformate, die häufig mit IoT-Frameworks inkompatibel sind, und sie verwenden einen Gateway für die Übersetzung. Gateway-Strategien können Plugins verwenden, die eine Protokollübersetzungslogik enthalten. IoT-Gateways können jedoch nicht

die Protokollübersetzung, Sicherheitsübersetzung und semantische Übersetzung in einen einzigen IoT-Gateway kombinieren.

[0508] Eingangsdaten **10602** können von einer ersten Domain in der Art des Internets oder der Cloud in den IoT-Gateway **10600** eintreten. Bei hier beschriebenen Beispielen können die Eingangsdaten **10602** in einem ersten Protokoll **P1 10604** mit einem ersten Sicherheitsniveau **10606** vorliegen und unter Verwendung einer ersten semantischen Repräsentation **10608** Daten und Vorrichtungen repräsentieren. Um mit nachgeschalteten Vorrichtungen kompatibel zu sein, können die den IoT-Gateway **10600** verlassenden Ausgangsdaten **10610** in einem zweiten Protokoll **10612** mit einem zweiten Sicherheitsniveau **10614** vorliegen und eine zweite semantische Repräsentation **10616** verwenden.

[0509] Der IoT-Gateway **10600** kann eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) **10618** zum Schützen vorgeschalteter und nachgeschalteter Vorrichtungen vor durch Angriffe auf den IoT-Gateway **10600** hervorgerufenen beschädigten Daten verwenden. Die TEE **10618** kann eine Anzahl verschiedener Sicherheitstechniken aufweisen, die eine sichere Boot-Umgebung, eine Validierung von Code vor der Ausführung, eine isolierte Ausführung, eine Fernattestierung, eine sichere Speicherung, eine sichere Bereitstellung, einen vertrauenswürdigen Weg und dergleichen implementieren können. Beispielsweise kann eine Anzahl von Systemen verwendet werden, die mit den Spezifikationen kompatibel sind, die von der Trusted Computing Group (TCG) herausgegeben wurden.

[0510] Die Eingangsdaten **10602** können in einem Protokollübersetzer **10620** verarbeitet werden, um den Nutzinhalt aus einem Eingangspaket oder -frame zu entfernen. Dies kann dadurch geschehen, dass der Gateway **10600** ein erstes Protokoll-Plugin **10622** verwendet. Die Eingangsdaten können dann zur Verarbeitung an einen Sicherheitsübersetzer **10624** weitergeleitet werden.

[0511] Im Sicherheitsübersetzer **10624** kann ein erstes Sicherheits-Plugin **10626** verwendet werden, um die Eingangsdaten für einen biometrisch verschlüsselten Nutzinhalt zu analysieren. Falls die Sicherheitsübersetzungsrichtlinie **10628** eine Entschlüsselung des biometrisch codierten Pakets ermöglicht, kann der Nutzinhalt entschlüsselt werden. Das erste Sicherheits-Plugin **10626** ist nicht auf biometrische Nutzinhalte beschränkt. Beispielsweise kann ein Nutzinhalt in den Eingangsdaten **10602** auf einem ersten Sicherheitsniveau **10606** in der Art von 1024 Bits verschlüsselt werden. Falls es die Sicherheitsübersetzungsrichtlinie **10628** erlaubt, kann der Nutzinhalt zur späteren Verschlüsselung unter Verwendung eines zweiten Sicherheitsniveaus **10614** in der Art von 512 Bits, 2048 Bits und dergleichen für die Ausgangsdaten **10610** entschlüsselt werden. Die Sicherheitsübersetzungsrichtlinie **10628** kann die Schlüssel für die Entschlüsselungs- und Verschlüsselungsprozesse aufweisen und Grenzen für die maximale Abweichung des Sicherheitsniveaus festlegen, beispielsweise unter anderem eine 1024-zu-2048-Bit-Änderung zwischen Eingangsdaten **10602** und Ausgangsdaten **10610** erlauben, jedoch eine 2048-zu-512-Bit-Änderung blockieren.

[0512] Ein Semantikübersetzer **10630** kann verwendet werden, um die semantische Repräsentation des Nutzinhalts in den Eingangsdaten **10602** in die für den Nutzinhalt in den Ausgangsdaten **10610** verwendete semantische Repräsentation zu übersetzen. Hierbei kann beispielsweise ein erstes Datensemantik-Plugin **10632** zum Umwandeln des Nutzinhalts von einer ersten semantischen Repräsentation **10608** in der Art von HTML in einen Zwischenzustand verwendet werden und dann ein zweites Datensemantik-Plugin **10634** zum Umwandeln der Daten aus dem Zwischenzustand in eine zweite semantische Repräsentation **10616** in der Art von XML verwendet werden. Viele verschiedene semantische Repräsentationen können verwendet werden, und die Plugins können auf der Grundlage der erforderlichen Übersetzungen ausgewählt werden. Bei einigen Beispielen kann ein einziges erstes Datensemantik-Plugin **10632** mit zwei verschiedenen zweiten Datensemantik-Plugins **10634** gepaart werden. Dies kann nützlich sein, falls Daten von verschiedenen IoT-Netzen den IoT-Gateway **10600** von einer gemeinsamen Quelle oder Senke in der Art einer Datenbank in der Cloud durchlaufen. Die Übersetzung **10636** kann abhängig vom Ursprung und vom Ziel der Daten bidirektional sein. Der Semantikübersetzer **10630** kann ein einziges Plugin aufweisen, wodurch die direkte Wandlung zwischen den semantischen Repräsentationen **10608** und **10616** ermöglicht wird.

[0513] Sobald die semantische Übersetzung abgeschlossen ist, läuft der Nutzinhalt für die Ausgangsdaten **10610** zur Umwandlung in das Sicherheitsniveau **S2 10614** für die Ausgangsdaten **10610** zum Sicherheitsübersetzer **10624**. Hierbei kann der Nutzinhalt in einem zweiten Sicherheits-Plugin **10638** in das Ausgangsdaten-Sicherheitsniveau **10614** verschlüsselt werden, wie es von der Sicherheitsübersetzungsrichtlinie **10628** erlaubt wird. Beispielsweise kann der Nutzinhalt auf einer anderen Bitebene verschlüsselt werden als die Eingangsdaten **10602**. Falls eine biometrische Markierung für den Schutz des Nutzinhalts verwendet wurde, können die Daten unter Verwendung einer simulierten biometrischen Markierung erneut geschützt werden.

[0514] Nachdem die Sicherheitsübersetzung **10640** abgeschlossen ist, wird der Nutzinhalt für die Ausgangsdaten **10610** an den Protokollübersetzer **10620** übergeben. Am Protokollübersetzer **10620** wird ein zweites Protokoll-Plugin **10642** zur Verpackung des Nutzinhalts im Protokoll **P2 10616** für die Ausgangsdaten **10610** verwendet. Hierbei können die Ausgangsdaten **10610** im Nutzinhaltsfeld eines Pakets oder Frames, das oder der mit dem Protokoll für den Übertragungsweg zu nachgeschalteten Vorrichtungen übereinstimmt, verpackt werden. Falls der Nutzinhalt der Ausgangsdaten **10610** für das Nutzinhaltsfeld des bestimmten Pakets oder Frames zu groß ist, kann er in mehrere Pakete oder Frames fragmentiert oder gepackt werden. Die Protokollübersetzung **10644** vervollständigt die Verarbeitung der Eingangsdaten **10602** zur Bildung der Ausgangsdaten **10610**. Die Ausgangsdaten **10610** werden dann zur nachgeschalteten Vorrichtung übertragen.

[0515] Andere Einheiten können in der IoT-Vorrichtung **10600** vorhanden sein, um Anpassbarkeit bereitzustellen. Beispielsweise kann ein Plugin-Installer **10646** Plugins validieren und im geeigneten Übersetzer **10620**, **10624** oder **10630** installieren. Beim Validierungsprozess kann eine Messung (beispielsweise Berechnen eines Hash-Codes) des Plugins unter Verwendung eines vertrauenswürdigen Plattformmoduls bzw. Trusted Platform Module (TPM) oder eines anderen Systems vorgenommen werden und kann die Messung mit einer Whitelist akzeptierter Werte verglichen werden.

[0516] Ein Skript-Compiler **10648** kann aufgenommen sein, um Skripts, die in Plugins verwendet werden, zu kompilieren. Dies kann für Anwendungen unter anderem in der Art von Video-Feeds, die hohe Geschwindigkeits- und Bandbreitenanforderungen aufweisen, nützlich sein.

[0517] Es sei bemerkt, dass nicht jeder Nutzinhalt auf allen drei Ebenen verarbeitet werden kann. Beispielsweise kann ein Nutzinhalt in den Eingangsdaten **10602** verarbeitet werden, um den Nutzinhalt aus einem Eingangs-Frame zu entfernen, und dann direkt verarbeitet werden, um in einem Ausgangs-Frame verpackt zu werden, und als Ausgangsdaten **10610** ausgesendet werden. Bei anderen Beispielen kann eine semantische Übersetzung nicht verwendet werden. Bei diesen Beispielen kann eine Sicherheitsübersetzung **10640** ausgeführt werden, nachdem der Nutzinhalt aus einem Eingangs-Frame entfernt wurde, und kann der Nutzinhalt zum Protokollübersetzer **10620** zurückgeführt werden, um ihn in einem Ausgangs-Frame zu verpacken, bevor er als Ausgangsdaten ausgesendet wird. Jegliche Kombinationen der Übersetzungsfunktionen können abhängig von den Parametern der Eingangs- und Ausgangsnetze verwendet werden.

[0518] Es sei bemerkt, dass ein Ressourcenaufteilungsschema mehrere virtuelle Clients aufweisen kann. Die für die Verarbeitung einer Gateway-Übersetzung mit einem Daten- und Steuer-Eingang und -Ausgang von einem ersten virtuellen IoT-Client zu einem ersten virtuellen IoT-Server verwendeten Ressourcen können von Ressourcen zur Steuerung der Datenübertragung von einem zweiten virtuellen IoT-Client zu einem zweiten virtuellen IoT-Server isoliert werden. Dieses Plattformaufteilungsschema kann das Austauschen von Schlüsseln zwischen einem Client und einem Server innerhalb von TEE-„Kanälen“ aufweisen. Im Allgemeinen verhandelt ein virtueller Client auf einer Seite eines Kanals Schlüssel, die vom virtuellen Server auf der anderen Seite desselben Kanals verwendet werden. Auf diese Weise können die Sicherheitssemantik und das Sicherheitsniveau über unähnliche IoT-Netze, die überbrückt werden, bewahrt werden. Die Ressourcenaufteilung und -zuweisung kann mit dem Gateway-Begriff von „Kanälen“ in Einklang gebracht werden, wobei ein virtueller Client und ein virtueller Server interagieren können.

[0519] Dieses Beispiel weist eine Gateway-Abstraktion auf, wobei ein virtueller Client und ein virtueller Server über einen sicheren Kanal, der ein Sicherheitsniveau aufweist, kommunizieren können. Verschiedene Systemressourcenaufteilungs- und -zuweisungsschemata können verwendet werden, beispielsweise auf der Grundlage der Virtualisierungstechnologie (VT-X und VT-d) von Intel oder einer Intel SGX (Softwareschutzerweiterungen bzw. Software Guard Extensions). Die Ressourcenaufteilung und -zuweisung kann mit dem Gateway-Kanal in Einklang gebracht werden, so dass alle Schlüssel, die gesamte Kommunikation und die gesamte Verarbeitung auf dem geeigneten Sicherheitsniveau ausgeführt werden.

[0520] Ferner kann eine Sicherheitsrichtlinie, welche Sicherheitsniveaus beschreibt, Eingangs- und Ausgangskanälen zugeordnet werden, wobei ein erster Kanal auf einem ersten Sicherheitsniveau ist und ein zweiter Kanal auf einem zweiten Sicherheitsniveau ist. Beispielsweise können auf einem ersten Sicherheitsniveau Daten unter Verwendung einer 1024-Bit-Verschlüsselung verschlüsselt werden, während auf einem zweiten Sicherheitsniveau Daten unter Verwendung einer 256-Bit-Verschlüsselung verschlüsselt werden können. Das Hardwareaufteilungsschema kann ausreichen, um eine Kanalisolation auf dem zugewiesenen Sicherheitsniveau oder darüber zu erzwingen.

[0521] Zusätzlich kann das sichere oder vertrauenswürdige Boot-Schema gewährleisten, dass das Hardware-aufteilungsschema in Kraft und verifizierbar und entsprechend einem vertrauenswürdigen Plattformmodul bzw. Trusted Platform Module (TPM) oder einem anderen vertrauenswürdigen Ausführungsumgebungsschema in der Art von Intel TXT, Intel SGX oder ARM Trustzone attestierbar ist.

[0522] **Fig. 107** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **10700** zur Übersetzung von Arbeitslasten in einem sicheren IoT-Gateway gemäß einigen Ausführungsformen. Das Verfahren **10700** aus **Fig. 107** kann durch die mit Bezug auf **Fig. 108** beschriebene IoT-Vorrichtung **10800** implementiert werden. Block **10702** repräsentiert beispielsweise den Fall, in dem eine Arbeitslast zur Übersetzung am Gateway ankommt. In Block **10704** werden die Eingangsdaten unter Verwendung des Eingangsprotokolls verarbeitet, um den Eingangs-Nutzzinhalt zu erhalten.

[0523] In Block **10706** wird festgestellt, ob der Eingangs-Nutzzinhalt ein biometrischer Stream ist. Falls dies der Fall ist, kann die Biometrie in Block **10708** auf datenschutzempfindlichen Inhalt analysiert werden.

[0524] In Block **10710** wird festgestellt, ob der Eingangs-Nutzzinhalt datenschutzempfindlich ist. Falls dies der Fall ist, kann in Block **10712** eine Eingangs-Datenschutzrichtlinie angewendet werden, beispielsweise um den Eingangs-Nutzzinhalt zu decodieren. Beispielsweise kann ein IoT-Objekt mit einer ersten Identität in der Art von UUID eine Richtlinie aufweisen, welche die Offenbarung der Identität beschränkt, weil sie von einem fremden System verwendet werden kann, um mehrere Interaktionen unter Verwendung seines oder eines anderen Gateways zu verfolgen, so dass die Verfolgungsinformationen verwendet werden können, um einen Verbindungsgraphen oder eine Maschinenlernrepräsentation zu bilden, der oder die unter Verwendung differenzieller Analyse weiter analysierbar ist. Die Datenschutzrichtlinie kann den Übersetzer anweisen, die erste UUID durch eine zweite zufällig erzeugte UUID zu ersetzen, um eine Verfolgung, Zusammenarbeit oder andere Analytik auf der Grundlage der UUID zu verhindern.

[0525] Die Richtlinie kann ferner Objektdatenwerte erkennen, von denen ein Fingerabdruck genommen werden kann, so dass eine weitere Analyse der Daten unter Verwendung einer differenziellen Analyse statistische Korrelationen mit anderen von anderen Objekten erhaltenen Datenwerten offenbaren könnte. Dies kann zum Blockieren von Ableitungen verwendet werden, die über das innere Verhalten, innere Prozesse und innere Entscheidungen vorgenommen werden können. Das Datenschutzrichtlinienmodul kann veranlassen, dass der zu ersetzende fragliche Datenwert durch einen anderen mit einer geringeren Granularität, einer geringeren Auflösung, einer höheren Abstraktion oder einer höheren Allgemeinheit ersetzt wird. Das Datenschutzrichtlinienmodul kann sich einfach weigern, den angeforderten Datenwert über die Gateway-Schnittstelle anzuliefern.

[0526] Die Datenschutzrichtlinie kann ferner die vom Gateway verwendeten Berechtigungsnachweise anweisen, die Daten für die Verwendung eines privaten EPID-Schlüssels zu authentifizieren, zu autorisieren oder auf andere Weise zu schützen. Die Auswahl des EPID-Schlüssels kann mit einer Person oder Gruppe oder Gemeinschaft von Dingen, Personen, Vorrichtungen oder Konzepten, die nummeriert sein können, wobei die Anzahl der Instanzen eine Datenschutzrichtlinie erfüllt, in Einklang gebracht werden. Beispielsweise kann der Anonymitätsgrad auf der Grundlage einer Population nummerierter Instanzen verhindern, dass eine differenzielle Datenschutzanalyse bei einer statistisch relevanten Ableitung von Wissen konvergiert.

[0527] In Block **10714** wird festgestellt, ob der Eingangs-Nutzzinhalt sicherheitsempfindlich ist, beispielsweise verschlüsselt ist. Falls dies der Fall ist, kann in Block **10716** eine Eingangs-Sicherheitsrichtlinie angewendet werden, beispielsweise um den Nutzzinhalt für die Verarbeitung zu entschlüsseln.

[0528] In Block **10718** wird festgestellt, ob der Eingangs-Nutzzinhalt mit dem Ausgangsnetz semantisch inkompatibel ist. Falls dies der Fall ist, wird in Block **10720** eine semantische Übersetzung ausgeführt, beispielsweise um den Eingangs-Nutzzinhalt in eine semantisch zwischenstehende oder IoT-Gateway-Repräsentation zu übersetzen und dann die semantisch zwischenstehende Repräsentation in die Ausgangsrepräsentation zu übersetzen. Bei einigen Beispielen kann die Eingangsrepräsentation direkt in die Ausgangsrepräsentation umgewandelt werden, um die Umwandlungszeiten zu verkürzen. Dies kann unter anderem für Daten mit einer hohen Bandbreite oder wenn beide Formate sehr gewöhnlich sind, geschehen.

[0529] In Block **10722** wird festgestellt, ob der Ausgangs-Nutzzinhalt sicherheitsempfindlich ist, beispielsweise ob die Richtlinie die Verschlüsselung für das Ausgangsnetz vorschreibt. Falls dies der Fall ist, wird in Block **10724** die Ausgangs-Sicherheitsrichtlinie angewendet. Der Ausgangs-Nutzzinhalt kann auf der von der Sicherheitsrichtlinie geforderten Bitebene verschlüsselt werden.

[0530] In Block **10726** wird festgestellt, ob der Ausgangs-Nutzzinhalt datenschutzempfindlich ist. Falls dies der Fall ist, wird in Block **10728** die Ausgangs-Datenschutzrichtlinie angewendet. In Block **10730** wird festgestellt, ob der Ausgangs-Nutzzinhalt Teil eines biometrischen Streams ist. Falls dies der Fall ist, wird ein entsprechend der Datenschutzrichtlinie erzeugter synthetischer biometrischer Stream verwendet, um den Nutzzinhalt zu schützen.

[0531] In Block **10734** wird der Ausgangs-Nutzzinhalt entsprechend dem Ausgangsprotokoll verarbeitet, wobei er beispielsweise in das Datenfeld eines Ausgangspaketes oder -frames gepackt wird. Das Ausgangspaket oder der Ausgangs-Frame wird dann über das Ausgangsnetz zur Zielvorrichtung gesendet.

[0532] Das Verfahren **10700** kann auch eine Anzahl von Aktionen zur Präparation des IoT-Gateways für die Übersetzungsfunktionen aufweisen. In Block **10736** kann ein Plugin-Installer ein Übersetzungsprofil zwischen Eingangs- und Ausgangsnetzen, einschließlich für Protokoll-, Datenschutz-, Sicherheits- und semantische Übersetzungen zu verwendender Plugins, identifizieren. Die Identifikation und Entdeckung der Eingangs- und Ausgangsnetze kann im IoT-Gateway erfolgen oder an einer anderen Vorrichtung erfolgen in der Art einer Datenbank, die sich in der Cloud befindet. Falls eine andere Vorrichtung einen Netzkatalog hat, kann der Plugin-Installer im IoT-Gateway eine Liste von Plugins akzeptieren und dann die Plugins herunterladen und versuchen, sie im IoT-Gateway zu installieren.

[0533] In Block **10738** wird jedes Plugin validiert, beispielsweise unter anderem in Bezug auf die Authentizität, Integrität, Version und Kompatibilität mit dem IoT-Gateway. Falls in Block **10740** festgestellt wird, dass ein Plugin nicht validiert werden konnte, kehrt der Prozessfluss zu Block **10736** zurück, um einen Ersatz für dieses Plugin zu finden. Falls kein Ersatz gefunden wird, kann eine Warnnachricht gesendet werden.

[0534] In Block **10742** kann ein Skript-Compiler oder ein anderes Richtlinienensystem feststellen, welche Übersetzungsroutinen unter Verwendung interpretierten Codes ausgeführt werden können und welche durch die Kompilation in binären Code beschleunigt werden können. In Block **10744** wird festgestellt, ob jedes der Übersetzungsskripts kompiliert werden kann. Falls dies der Fall ist, werden in Block **10746** die Übersetzungsskripts kompiliert und als Binärcodedateien gelinkt. In Block **10748** werden die Sicherheits- und Datenschutzrichtlinien konfiguriert. Der Prozessfluss kehrt zu Block **10702** zurück, um mit der Übersetzung zu beginnen.

[0535] Der Prozessfluss kann nach Bedarf zu den Konfigurationsblöcken **10736** bis **10748** zurückkehren, um das System zu rekonfigurieren. Dies kann geschehen, wenn ein neues Netz mit dem IoT-Gateway gekoppelt wird oder ein älteres Netz vom IoT-Gateway entkoppelt wird. Ferner kann eine Rekonfiguration geschehen, wenn neue Datenschutz-, Sicherheits- oder semantische Übersetzungen für Nutzzinhalte verwendet werden.

[0536] **Fig. 108** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem IoT-Gateway **10800** zur Übersetzung von Arbeitslasten zwischen Domains vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3**, **Fig. 8** und **Fig. 106** beschrieben wurden. Es sei bemerkt, dass für den IoT-Gateway **10800** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0537] Der IoT-Gateway **10800** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) **10802** aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel ist. Das TPM **10802** kann einen kryptographischen Prozessor (CP) **10804**, einen nichtflüchtigen Speicher (NVM) **10806** und einen sicheren Speicher (SM) **10808** aufweisen. Der CP **10804** kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM **10806** kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM **10808** kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datensegment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung an einem Boot-Codesegment können die Messungen zum Einrichten einer vertrauenswürdigen Ausführungsumgebung **10618**, wie mit Bezug auf **Fig. 106** beschrieben, durch Erzeugen einer Vertrauenskette vom anfänglichen Booten verwendet werden.

[0538] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Übersetzungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind,

ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0539] Der Massenspeicher **808** kann eine sichere Boot-/Messeinrichtung **10806** aufweisen, die Messungen an Code und Daten ausführt. Wie hier beschrieben, kann eine Messung die Erzeugung eines Hash-Codes des Codes oder der Daten sein. Der Hash-Code kann mit einem erwarteten Messwert verglichen werden, bevor der Code oder die Daten ausgeführt oder übertragen werden. Eine anfängliche Boot-Messung kann vom Prozessor **802** ausgeführt werden, um die sichere Boot-/Messeinrichtung **10806** einzurichten, zusätzliche Messungen auszuführen. Ein Protokollübersetzer **10620** kann Übersetzungen zwischen einem Eingangsprotokoll und einem Ausgangsprotokoll ausführen. Ein Sicherheitsübersetzer **10624** kann Sicherheitsübersetzungen zwischen einem Eingangssicherheitsniveau und einem Ausgangssicherheitsniveau ausführen. Ein Datenschutzübersetzer **10808** kann Datenschutzübersetzungen unter anderem beispielsweise auf der Grundlage biometrischer Berechtigungsnachweise ausführen. Ein Semantikübersetzer **10624** kann eine Datensemantikübersetzung zwischen einem Eingangs-Nutzinhalt und einem Ausgangs-Nutzinhalt ausführen. Ein sicherer Datenspeicher **10810** kann Plugins, Sicherheitsrichtlinien, Datenschutzrichtlinien, einen Skript-Compiler und dergleichen speichern.

[0540] Diese Ressourcen können einem Gateway-Verarbeitungs-„Kanal“ entsprechend einem von den Sicherheitsrichtlinienmodulen verwalteten zugewiesenen Sicherheitsniveau zugewiesen werden. Eingangs- und Ausgangsrichtlinien können eine Sicherheitsniveauerhöhung beispielsweise von nicht geheim zu geheim oder eine Sicherheitsniveauperiode von geheim zu nicht geheim autorisieren. Die Richtlinien können eine Datenintegritätsklassifizierung, beispielsweise von gesäubert zu nicht gesäubert oder von Herstellung zu Entwicklung oder gemäß einem anderen Datenkategorisierungsschema autorisieren.

[0541] Fig. 109 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **10900**, das Code aufweist, um einen Prozessor **902** anzuweisen, eine Arbeitslast zwischen einem Eingangsnetz und einem Ausgangsnetz zu übersetzen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **10900** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **10900** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0542] Das nichtflüchtige maschinenlesbare Medium **10900** kann einen Code **10902** aufweisen, um den Prozessor **902** anzuweisen, einen Eingangs-Nutzinhalt nach einem Eingangsprotokoll zu verarbeiten. Ein Code **10904** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Eingangs-Nutzinhalt nach einer Eingangs-Datenschutzrichtlinie zu verarbeiten. Ein Code **10906** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Eingangs-Nutzinhalt nach einer Eingangs-Sicherheitsrichtlinie zu verarbeiten. Ein Code **10908** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Datensemantik eines Eingangs-Nutzinhalts in die Datensemantik eines Ausgangs-Nutzinhalts zu übersetzen.

[0543] Ein Code **10910** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Ausgangs-Nutzinhalt nach einer Ausgangs-Sicherheitsrichtlinie zu verarbeiten. Ein Code **10912** kann aufgenommen sein, um einen Ausgangs-Nutzinhalt nach einer Ausgangs-Datenschutzrichtlinie zu verarbeiten. Ein Code **10914** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Ausgangs-Nutzinhalt nach einer Ausgangs-Protokollrichtlinie zu verarbeiten. Ein Code **10916** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Softwaremessungen von Plugins und Richtlinien auszuführen, beispielsweise vor der Installation. Ein Code **10918** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Plugins zu überprüfen und zu installieren. Ein Code **10920** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Plugins zu kompilieren.

[0544] IoT-Netze können als eine Sammlung eine Fog-Vorrichtung bildender Vorrichtungen angesehen werden. Die einzelnen Vorrichtungen können sich über eine Vielzahl von Netztransport-, Sitzungs- und Anwendungsschicht-Kommunikationswegen verbinden. Ein Eigentümer des IoT-Netzes in der Art eines Benutzers, einer Organisation oder einer Gruppe hat ein gemeinsames Interesse und eine gemeinsame Teilnahme im IoT-Netz. Der Eigentümer kann feststellen, dass Vorrichtungen zu einer Organisation gehören, weil er die Zusammenarbeit zwischen den verschiedenen Vorrichtungen verwaltet, gesetzlich besitzt oder orchestriert.

[0545] Eine Vorrichtung kann in ein IoT-Netz eingegliedert werden, um es einem Eigentümer zu ermöglichen, den Besitz der Vorrichtung zu ergreifen und sie dadurch als besessene Vorrichtung beim Eigentümer zu registrieren. Wie hier verwendet, gibt das Eingliedern an, dass Aktivitäten zur Hinzufügung einer Vorrichtung,

wie der Austausch von Beitrittsanforderungen und die Überprüfung von Identitäten sowie die Erzeugung von Vorrichtungsressourcen, stattgefunden haben. Eine Vorrichtung kann wiederum die Eigentümerschaft in der Domain durch Aufzeichnen der Eigentümer-/Domain-Informationen in Vorrichtungsressourcen bestätigen. Eine Vorrichtung kann mehrere Eigentümer zulassen oder aufweisen. Bei einigen Beispielen können die Vorrichtungen in mehreren Domains existieren, wodurch die Erkennung der Vorrichtungen durch jeweilige andere Vorrichtungen komplizierter gemacht wird.

[0546] Fig. 110 ist ein schematisches Diagramm 11000 von Vorrichtungen, die durch verschiedene Domains eingegliedert werden, welche von einer geteilten Domain aufgenommen werden, die erzeugt wurde, um es den Vorrichtungen zu ermöglichen, als Komponenten einer neuen Domain teilzunehmen, gemäß einigen Ausführungsformen. Im schematischen Diagramm 11000 wird eine erste Vorrichtung 11002 durch ein Eingliederungswerkzeug (OBTA) 11006 in eine erste Domain A 11004 eingegliedert. Eine zweite Vorrichtung 11008 wird durch ein zweites Eingliederungswerkzeug (OBTB) 11012 in eine zweite Domain B 11010 eingegliedert. Bei diesem Beispiel können sich die Vorrichtungen 11002 und 11008 als Mitglieder der Domains A 11004 bzw. B 11010 ansehen.

[0547] Interaktionen zwischen den Vorrichtungen D1 11002 und D2 11008 können unter den Sicherheitsniveaus zugelassen werden, beispielsweise falls die Domains Teil einer Familie sind, können jedoch in manchen Fällen nicht zugelassen werden, weil die verschiedenen OBTA 11006 und OBTB 11012 eine Unterteilung zwischen den Ressourcen 11014 oder 11016 in den Netzen einrichten. Demgemäß kann das OBTA 11006 für die Domain A 11004 eine Vorrichtung, die in eine fremde Domain B 11010 eingegliedert ist, nicht erkennen oder ihr nicht vertrauen. Dies könnte beispielsweise darauf zurückzuführen sein, dass sich die jeweiligen Eingliederungswerkzeuge nicht eine gemeinsame Ressource 11014 oder 11016 teilen, die eingegliederte und daher vertrauenswürdige Vorrichtungen 11002 und 11008 enthält.

[0548] Bei den hier beschriebenen Techniken kann, wenn Vertrauen zwischen den Eingliederungswerkzeugen 11006 und 11012 in den jeweiligen Domains 11004 und 11010 hergestellt wird, eine neue Domain 11018 erzeugt werden, welche eine geteilte Ressource 11020 aufweist. Die geteilte Ressource 11020 kann Informationen von Ressourcen 11014 oder 11016 in den einzelnen übergeordneten Domains 11004 und 11010 aufweisen. Dies wird weiter mit Bezug auf Fig. 111 erörtert.

[0549] Fig. 111 ist ein schematisches Diagramm 11100 einer beispielhaften Erzeugung einer geteilten Ressource, um es einer Vorrichtung zu ermöglichen, über Domains teilzunehmen, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 110 beschrieben wurden. Wie anhand Fig. 110 beschrieben, führt die Entdeckung lokaler Eingliederungsressourcen R1 11012 und R2 11016 in einer anderen Domain zur Erzeugung einer geteilten Ressource R3 11020, so dass in R1 11014 enthaltene Datensätze in R3 11020 gespeichert werden, wodurch der Zugriff durch das Eingliederungswerkzeug OBTB 11012 in Domain B 11010 ermöglicht wird. Ähnlich werden in R2 11016 enthaltene Datensätze in R3 11020 gespeichert, und es kann durch das Eingliederungswerkzeug OBTA 11014 in der Domain A 11004 darauf zugegriffen werden. Ferner kann die geteilte Ressource R3 11020 unter anderen Konflikten Namenskonflikte lösen, beispielsweise wenn ein von OBTA 11006 angenommener Domainname der gleiche ist wie ein von OBTB 11012 angenommener Domainname.

[0550] Die Techniken finden oder erzeugen eine neue Domainkennung für die Vereinigung der Domains 11004 und 11010, beispielsweise eine neue UUID, so dass die geteilte Ressource R3 11020 eine Domainkennung in einer lokalen Ressource R1 11014 und R2 11016 synchronisiert. Eine Subdomainkennung 11102 in R1 11014 kann sich von einer Subdomainkennung 11104 in R2 RP16 unterscheiden, so dass jede Subdomain zu einer jeweiligen Subdomain der neu gebildeten Domain 11018 wird. Die geteilte Ressource R3 11020 synchronisiert mit den jeweiligen lokalen Ressourcen R1 11014 und R2 11016 zur Belegung der vereinigten Ressource, welche die mehreren Subdomainkennungen aufweist.

[0551] Die Eingliederungswerkzeuge OBT-A 11006 und OBT-B 11012 werden ähnlich mit der geteilten Ressource 11020 synchronisiert, wobei sie jeweils als Mitglieder einer gemeinsamen Domain 11018 eingerichtet werden. Ähnlich werden die Vorrichtungen D1 11002 und D2 11008 mit der geteilten Ressource 11020 synchronisiert, wodurch sie jeweils als Mitglied derselben gemeinsamen Domain 11018 eingerichtet werden, jedoch jeweils die Mitgliedschaft in der jeweiligen Subdomain 11004 oder 11010 behalten können, welche die Vorrichtung 11002 oder 11008 ursprünglich eingegliedert hat.

[0552] Fig. 112 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 11200 zur Einrichtung einer kombinierten IoT-Domain mit geteilten Ressourcen gemäß einigen Ausführungsformen. Das Verfahren 11200

aus **Fig. 112** kann durch die mit Bezug auf **Fig. 113** beschriebene IoT-Vorrichtung **11300** implementiert werden. Wie hier verwendet, können die geteilten Ressourcen virtualisierte Ressourcen, Speicherressourcen, Kommunikationsressourcen, Eingliederungsressourcen, Dienstanbieterressourcen und dergleichen einschließen. Die Ressourcen können auf der Domänebene, der Subdomänebene oder der Vorrichtungsebene existieren. Block **11202** repräsentiert beispielsweise den Fall, in dem sich ein Eingliederungswerkzeug einer ersten Vorrichtung in einer ersten Netzdomain anschließt. In Block **11204** fügt das erste Eingliederungswerkzeug die Vorrichtung einer lokalen Ressourcen hinzu, beispielsweise als Mitglied oder in Besitz befindliche Vorrichtung.

[0553] In Block **11206** fügt ein zweites Eingliederungswerkzeug eine zweite Vorrichtung zu einer zweiten Netzdomain hinzu. In Block **11208** fügt das zweite Eingliederungswerkzeug die Vorrichtung einer lokalen Ressource hinzu, beispielsweise als Mitglied oder in Besitz befindliche Vorrichtung.

[0554] In Block **11210** entdecken die Eingliederungswerkzeuge einander auf einem Netz und stellen Vertrauen zwischen ihnen her. Dies kann beispielsweise durch wechselseitige Attestierung, individuelles Paaren, durch eine administrative Konsole oder durch eine Blockchain, wie hier beschrieben, geschehen.

[0555] In Block **11212** erzeugen die Eingliederungswerkzeuge eine geteilte Ressource, wobei sie Teilhaber an der Ressource sind. In Block **11214** binden die Eingliederungswerkzeuge ihre jeweiligen Ressourcen an die geteilte Ressource. Dadurch sind die Ressourcen der ersten Vorrichtung für das zweite Eingliederungswerkzeug zugänglich und sind die Ressourcen der zweiten Vorrichtung für das erste Eingliederungswerkzeug zugänglich. In Block **11216** wird eine neue Domain gebildet, die auf der Vereinigung der beiden Vorrichtungsdomains beruht. Die Domänenkennung für die neue Domain wird in der geteilten Ressource aufgezeichnet.

[0556] In Block **11218** wird festgestellt, ob die Subdomänenkennung in der ersten Domain der Subdomänenkennung in der zweiten Domain gleicht oder dieser ähnelt. Falls dies der Fall ist, wird in Block **11220** eine neue Subdomänenkennung für die Subdomänenkennung in der zweiten Ressource gewählt und werden alle Ressourcen, die auf diese Subdomänenkennung zugreifen, mit dem neuen Namen aktualisiert.

[0557] In Block **11222** wird festgestellt, ob die OBT-Kennung oder Eingliederungswerkzeugkennung in der ersten Domain gleich der OBT-Kennung in der zweiten Domain ist. Falls dies der Fall ist, wird in Block **11224** eine neue OBT-Kennung für die OBT-Kennung in der zweiten Ressource gewählt und werden alle Ressourcen, die auf diese OBT-Kennung zugreifen, mit dem neuen Namen aktualisiert.

[0558] In Block **11226** wird festgestellt, ob die Vorrichtungskennung in der ersten Domain gleich der Vorrichtungskennung in der zweiten Domain ist. Falls dies der Fall ist, wird in Block **11228** eine neue Vorrichtungskennung für die Vorrichtungskennung in der zweiten Ressource gewählt und werden alle Ressourcen, die auf diese Vorrichtungskennung zugreifen, mit dem neuen Namen aktualisiert.

[0559] Wenngleich das Verfahren für zwei Vorrichtungen und Domains dargestellt ist, kann eine beliebige Anzahl von Vorrichtungen, die von überlappenden Domains aufgenommen werden müssen, verwendet werden. Beispielsweise können zwei Domains mit mehreren Vorrichtungen durch eine geteilte Domain vereinigt werden, die durch Eingliederungswerkzeuge in beiden Domains erzeugt wurde. Bei einem anderen Beispiel können Vorrichtungen in drei oder mehr Domains durch eine geteilte Domain vereinigt werden.

[0560] **Fig. 113** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **11300** zur Erzeugung geteilter Ressourcen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **11300** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0561] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Teilung von Ressourcen über Domains implementieren. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0562] Der Massenspeicher **808** kann ein Eingliederungswerkzeug **11302** aufweisen, das Vorrichtungen zur Domain der IoT-Vorrichtung **11300** hinzufügt und einen Speicher von Vorrichtungsressourcen **11304** für die Vorrichtungen erzeugt. Ein Vorrichtungsentdecker **11306** kann Vorrichtungen in anderen Domains identifizieren, die als Teil einer Fog-Vorrichtung mit Vorrichtungen in der aktuellen Domain verwendet werden können.

Der Vorrichtungsentdecker **11306** kann von einem Orchestrator bereitgestellte Informationen verwenden, um andere Vorrichtungen zu entdecken, wie hier beschrieben. Ein Vertrauensbilder **11308** kann verschiedene Techniken zum Herstellen von Vertrauen zwischen dem Eingliederungswerkzeug **11302** und einem Eingliederungswerkzeug in einer anderen Domain verwenden. Der Vertrauensbilder **11308** kann Attestierungsinformationen oder Identifikationsschlüssel austauschen oder ein von einer Administrator-Workstation zugewiesenes Vertrauenszertifikat verwenden. Bei einigen Beispielen kann der Vertrauensbilder **11308** eine Blockchain-Vertrauens-Root verwenden, wie hier beschrieben.

[0563] Ein Geteilte-Domain-Erzeuger **11310** kann das Eingliederungswerkzeug dabei unterstützen, mit Eingliederungswerkzeugen von den anderen Domains zu arbeiten, um eine geteilte Domain zu erzeugen. Die geteilte Domain kann ein geteiltes Ressourcenverzeichnis **11312** aufweisen, das für alle Eingliederungswerkzeuge über die verschiedenen Domains zugänglich ist oder in jede der Eingliederungswerkzeuge aufweisenden IoT-Vorrichtungen gespiegelt ist.

[0564] Fig. 114 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **11400**, das Code aufweist, um einen Prozessor **902** anzuweisen, geteilte Ressourcen über Domains einzurichten, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **11400** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **11400** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0565] Das nichtflüchtige maschinenlesbare Medium **11400** kann einen Code **11402** aufweisen, um den Prozessor **902** anzuweisen, eine Vorrichtung zu einer Domain hinzuzufügen. Ein Code **11404** kann aufgenommen sein, um den Prozessor **902** anzuweisen, lokale Ressourcen für die Vorrichtung in der Domain zu erzeugen. Ein Code **11406** kann aufgenommen sein, um den Prozessor **902** anzuweisen, relevante Vorrichtungen in anderen Domains zu entdecken, einschließlich beispielsweise Eingliederungswerkzeuge in diesen Domains. Ein Code **11408** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Ressourcen für lokale Vorrichtungen mit Ressourcen in anderen Domains zu verbinden. Ein Code **11410** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine geteilte Domain zu erzeugen, welche die geteilten Ressourcen für alle Vorrichtungen hält. Ein Code **11412** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob es jegliche Namens- oder Kennungsüberlappungen zwischen Domains, Eingliederungswerkzeugen und Vorrichtungen gibt. Ein Code **11414** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Namensüberlappungen durch Umbenennen von Domains, Eingliederungswerkzeugen oder Vorrichtungen, die zuletzt eingegliedert wurden, zu korrigieren und die neuen Namen zu allen relevanten Ressourcen zu übertragen.

[0566] Die vorstehend beschriebenen Netzkommunikations- und -authentifizierungssysteme stellen eine Anzahl von Aspekten zur Implementation von IoT-Netzen für bestimmte Anwendungen bereit. Bei einem Beispiel kann ein verteiltes Netz verwendet werden, um die Verfolgbarkeit von Endprodukten in der Art von Nahrungsmitteln, Pharmazeutika oder industriellen Produkten zu implementieren.

[0567] Für ein Lebenszyklusverfolgungssystem ergibt sich die Frage, wie die Teilnehmer an dem System Vertrauen entwickeln, dass sich das System entsprechend einem erwarteten Verhaltensmodell und nicht entsprechend etwas anderem, das sich außerhalb des Modells befindet, verhält. Die Herausforderung besteht darin, dass die Entität, die gutes Verhalten definiert, nicht vertrauenswürdig sein kann. Für diesen Zweck haben ein provinzielles Vertrauen, beispielsweise von Vorrichtung zu Vorrichtung, und institutionelle Vertrauensmechanismen, die beispielsweise von zentralen Stellen gesteuert werden, Schwächen. Ein Infrastrukturvertrauen kann jedoch eine zuverlässigere Form der Vertrauenserzwingung sein, und diese Blockchain ist eine Technologie zur Implementation von Infrastrukturvertrauen. Daher kann die Aufnahme von Vorrichtungen in andere Domains, wie mit Bezug auf Fig. 110 beschrieben, die Bildung von Vorrichtungen aus Gruppen von IoT-Vorrichtungen und die Herstellung von Vertrauen zwischen diesen Vorrichtungen ermöglichen. Dies kann unter Verwendung verschiedener Systeme zur Herstellung von Vertrauen in der Art der mit Bezug auf Fig. 20 erörterten Blockchain-Vertrauens-Roots geschehen.

[0568] In Fällen, in denen ein „Datensatzschlüssel“ verwendet wird, wird ein Verfahren zum Einrichten von Vertrauenseigenschaften des Datensatzschlüssels bereitgestellt. Für die Verwendung von IoT zur Entwicklung eines Verfolgbarkeitssystems, wodurch viele Industrien berührt werden, einschließlich neuer Industrien, die noch nicht entstanden sind, kann das Framework in der Art des hier beschriebenen Blockchain-Vertrauens für die Entwicklung von Vertrauen in das Verfolgbarkeitssystem nützlich sein.

[0569] Ein beispielhaftes IoT-Verfolgbarkeitssystem wird ferner mit Bezug auf die **Fig. 115 - Fig. 120** beschrieben. Das IoT-Verfolgbarkeitssystem kann unter Verwendung eines IoT-Netzes, wie mit Bezug auf **Fig. 3** beschrieben, oder einer Fog-Vorrichtung, wie beispielsweise mit Bezug auf **Fig. 4** beschrieben, implementiert werden. Das IoT-Verfolgbarkeitssystem kann Domains in der Art öffentlicher und privater Domains, IP- und IoT-Netzdomains und Domains, die von verschiedenen Lieferanten oder kommerziellen Entitäten gesteuert werden, durchlaufen.

[0570] **Fig. 115** ist ein Leiterdiagramm **11500**, das Stufen in einem Produktlebenszyklus für die Implementation eines Produktverfolgungssystems zeigt, gemäß einigen Ausführungsformen. Das Leiterdiagramm **11500** aus **Fig. 115** kann durch die mit Bezug auf **Fig. 119** beschriebene IoT-Vorrichtung **11900** implementiert werden. Die Verfolgbarkeitssysteme können die Aufzeichnung, Speicherung, Überprüfung, Sicherung und Abrufung von Befehlsketten- und Verfolgungsinformationen vom Ursprungsort **11502** des Produkts zur Verkaufsstelle **11504** implementieren, wobei es beispielsweise unter anderem eine Verarbeitungsanlage **11506** und ein Verteilungszentrum **11508** durchläuft. Die während der Übergänge **11510**, **11512** und **11514** von einer Stufe zur nächsten erzeugten Informationen sind auch relevant. Beispielsweise geschehen die Übergänge **11510**, **11512** und **11514**, wenn das Produkt vom Ursprungsort **11502** in der Art eines Bauernhofs zu einer Verarbeitungsanlage **11506** oder von der Verarbeitungsanlage **11506** zu einem Verteilungszentrum **11508** bewegt wird. Die Informationen können beispielsweise Inventarinformationen, wie das Versanddatum, Produkttypen und die Menge, und Transformationsinformationen, wie Erntedaten (für landwirtschaftliche Erzeugnisse), Produktionsdaten (für Waren), die Lagertemperaturgeschichte oder eine Anzahl anderer Parameter, welche Änderungen an den Produkten beeinflussen, einschließen.

[0571] Ein Datensatzschlüssel **11516**, **11518** und **11520** kann an einer oder mehreren Stufen erzeugt werden, um die Informationen zu speichern und auf diese zuzugreifen, welche hier als Verfolgbarkeitsdatensätze für diese Stufe bezeichnet werden. Es sei bemerkt, dass die Verfolgbarkeit ein Vertrauensaspekt ist. Die Verfolgbarkeit nimmt an, dass es ein Verfahren gibt, um festzustellen, ob die Verfolgungsgeschichten ein erwartetes gutes Verhalten gegenüber einem nicht erwarteten oder unangemessenen Verhalten angeben. In Fällen, in denen ein provinzielles und institutionelles Vertrauen beteiligt ist, ist es möglich, dass die Kriterien, die ein erwartetes Verhalten definieren, beispielsweise gut, oder ein unerwartetes Verhalten definieren, beispielsweise ein unangemessenes Verhalten, nicht wohldefiniert sind. Der Prozess des Verwendens eines Infrastrukturvertrauens, beispielsweise durch eine Blockchain, erfordert die Definition dieser Kriterien und die Verwendung einer Konsenswahrheit in Bezug auf ein erwartetes Verhalten. Daher kann das Infrastrukturvertrauen für einige Fälle der stärkere Ansatz sein. Die Verfolgbarkeitsdatensätze können in einem öffentlich zugänglichen Speicher, privaten Speichern oder in einer Kombination davon gespeichert werden. Ferner können Verfolgbarkeitsdatensätze teilweise in einem öffentlichen Speicher in der Art des Datensatzschlüssels gespeichert werden, während die restlichen Einzelheiten in privaten Speichern für den Zugriff bei der Fehlersuche bei späteren Geschehnissen gehalten werden.

[0572] **Fig. 116** ist eine Schemazeichnung der Verwendung privater Datenspeicher **11602**, wobei Datensatzschlüssel **11604** verwendet werden können, um auf die Verfolgbarkeitsdatensätze **11606** bis **11612** für eine Stufe **11614** bis **11620** zuzugreifen, gemäß einigen Ausführungsformen. Bei diesem Beispiel können die Verfolgbarkeitsdatensätze **11606** bis **11612**, die öffentlich zugänglich sind, keine detaillierten Prozesse oder Informationen in Zusammenhang mit dem Landwirt, Hersteller und dergleichen aufweisen.

[0573] Es kann nützlich sein, eine private/öffentliche Grenze zwischen grundlegenden Verfolgbarkeitsdaten in der Art des Datensatzschlüssels, der Zeit und des Ursprungsorts und den detaillierten proprietären Aufgabendaten von Herstellungsanlagen aufrechtzuerhalten. Hierdurch kann der wahrgenommene Datenschutz für Entitäten an einer Stufe **11614** bis **11620** erhöht werden, wodurch die Annahmerate erhöht wird. Bei diesem Beispiel werden zu den getrennten Datenspeichern **11602** gesendete Aufrufe **11622** bis **11628** verwendet, um die Verfolgbarkeitsdatensätze **11606** bis **11612** zu extrahieren. In einigen Fällen könnten die getrennten privaten Datenspeicher **11602** Sicherheits- und Zuverlässigkeitsprobleme präsentieren, wenn die privaten Datenspeicher **11602** von verschiedenen Entitäten verwaltet werden.

[0574] **Fig. 117** ist eine Schemazeichnung der Verwendung eines öffentlichen oder gemeinsamen Datenspeichers **11702** gemäß einigen Ausführungsformen. Bei diesem Beispiel können die Daten öffentlich gespeichert werden, jedoch unter den Datensatzschlüsseln **11704** für die verschiedenen Stufen verschlüsselt werden. Die Aufzeichnungsschlüssel **11704** können auch als Indizes für den gemeinsamen Datenspeicher **11702** wirken, wo die Befehlsketten- und Verfolgbarkeitsdaten extrahiert werden können. Ein Datensatzschlüssel **11704** kann auf die Verfolgbarkeitsdatensätze **11706** bis **11712** für eine Stufe zugreifen und sie entschlüsseln. Beispielsweise kann ein Schlüssel **11714** den Verfolgbarkeitsdatensatz **11706** lokalisieren und entschlüsseln.

[0575] Die Datensatzschlüssel **11704** können an eine Kette angehängt werden und entlang den Stufen weitergeleitet werden, entweder physikalisch oder durch Markierungen auf der Produktverpackung. Beispielsweise kann sich eine IoT-Vorrichtung in Zusammenhang mit der Verpackung des Produkts mit einem Mesh-Netz oder einer Fog-Vorrichtung an einer Einrichtung in der Art einer Verarbeitungsanlage bei der Auslieferung des Produkts an einem Ladedock verbinden.

[0576] Die IoT-Vorrichtung kann dem Produkt durch die Verarbeitungsanlage folgen und mit dem verarbeiteten Produkt beispielsweise in der zugeordneten Verpackung austreten. Wenn die IoT-Vorrichtung zur Auslieferung an eine andere Stufe in der Art von Einzelhandelsläden in einen Lastwagen geladen wird, kann das Mesh-Netz oder die Fog-Vorrichtung an der Anlage die IoT-Vorrichtung mit dem Datensatzschlüssel **11704** für diese Stufe programmieren. Sobald die IoT-Vorrichtung die Stufe verlässt, beispielsweise den Kontakt mit dem Mesh-Netz für die Verarbeitungsanlage verliert, kann der Verfolgbarkeitsdatensatz für diese Stufe an einem öffentlichen Datenspeicher **11702** oder an einem privaten Datenspeicher **11602**, wie mit Bezug auf **Fig. 116** beschrieben, oder an beiden gespeichert werden.

[0577] **Fig. 118** ist ein schematisches Diagramm eines beispielhaften Prozesses **11800** zur Implementation eines Verfolgbarkeitssystems gemäß einigen Ausführungsformen. Der Prozess **11800** aus **Fig. 118** kann durch die mit Bezug auf **Fig. 119** beschriebene IoT-Vorrichtung **11900** implementiert werden. Bei diesem Beispiel ist das Produkt ein landwirtschaftliches Erzeugnis in der Art von Getreide, Frucht und dergleichen. In diesem Fall betreffen die Stufen eine Aktion, die an dem Erzeugnis ausgeführt wird, wie Züchten, Versenden, Verarbeiten und dergleichen. Wenngleich die nachstehenden Stufen mit Bezug auf ein spezifisches Produkt erörtert werden, gilt das Beispiel für beliebige Produkttypen, ob gezüchtet oder hergestellt. Demgemäß können Aktionen in der ersten Stufe Aktionen betreffen, die beispielsweise in einer Herstellungsanlage stattfinden, wenn ein Produkt hergestellt wird.

[0578] Die erste Stufe **11804** betrifft den Prozesslebenszyklus in Zusammenhang mit dem Ursprung des Erzeugnisses. Die erste Stufe **11804** kann in Block **11806** mit dem Pflanzen des Erzeugnisses beginnen. Die Pflanzung kann durch ein IoT-Mesh-Netz in Zusammenhang mit der Pflanzvorrichtung in der Art eines Traktors verfolgt und überwacht werden. Viele Traktoren sind mit GPS und Computerbildschirmen ausgerüstet, um den Traktor auf dem wirksamsten Weg durch das Feld zu leiten. Das IoT-Mesh-Netz kann die Bordcomputer sowie Vorrichtungen, welche die Saatmaschine, welche die Saat pflanzt, den Saatvorratsbehälter, die Pflanztemperatur, das Bodenfeuchtigkeitsniveau und andere Pflanzparameter sowie Geräteparameter in der Art des Kraftstoffniveaus, des Kompressor-drucks für eine Luftsaatmaschine, Kraftstoffniveaus und dergleichen überwachen, einschließen. Die Daten **11808**, die erzeugt werden, können in einem Bauernhof-Datenspeicher **11810** in der Art eines zentralen Servers gespeichert werden. Der Traktor kann über ein LPWA-Netz, ein LTE-Netz, einen Satelliten-Uplink oder ein anderes Funknetz in kontinuierlicher Kommunikation mit dem Bauernhof-Speicher **11810** stehen oder die Daten für das Downloaden speichern, wenn der Traktor ein Hauptgebäude oder eine Scheune erreicht. Der Bauernhof-Datenspeicher **11810** kann sich an einer fernen Stelle, beispielsweise in der Cloud, befinden.

[0579] In Block **11812** kann das Anbauprodukt gedüngt werden, beispielsweise durch Verspritzen von flüssigem Ammoniak. Dies kann durch das dem Traktor zugeordnete IoT-Mesh-Netz überwacht werden, wobei zusätzliche Vorrichtungen den Nährstoffapplikator überwachen. Diese Vorrichtungen können den Vorratsbehälter wasserfreien Ammoniaks, den Spritzdruck und dergleichen überwachen. Andere Prozesse können zusätzlich zur Düngung oder an Stelle von dieser in diesem Block überwacht werden. Dies kann die Anwendung festen Düngers und einer Bewässerung einschließen. Die Anwendung festen Düngers kann durch das IoT-Mesh-Netz des Traktors überwacht werden. Die Bewässerung kann durch ein IoT-Mesh-Netz überwacht werden, das einem Bewässerungssystem in der Art von Pumpen, Strömungsmessern, Drucksensoren, Motoren, welche Ausleger bewegen, und dergleichen zugeordnet ist. Das Bewässerungsnetz kann durch eine Funkstrecke in der Art eines LPWA-Netzes dem Zentralsystem berichten. Daten **11808** von diesen Prozessen können auch im Bauernhof-Datenspeicher **11810**, beispielsweise in einer privaten Blockchain, gespeichert werden.

[0580] In Block **11814** kann das Anbauprodukt beispielsweise unter Verwendung eines Mähdreschers geerntet werden. Ebenso wie der Traktor haben Mähdrescher häufig ein umfangreiches Computersteuersystem, das GPS und eine Datensammlung aufweist. Ein IoT-Mesh-Netz kann am Mähdrescher installiert werden und mit dem Computersystem am Mähdrescher verbunden werden, um das Erzeugnis zu verfolgen, wenn es geerntet wird. Das Mesh-Netz kann Vorrichtungen aufweisen, die das Gewicht des geernteten Materials, die Feldorte für das geerntete Material und andere Parameter, die für die Verfolgbarkeit verwendet werden können, verfolgen.

[0581] Verglichen mit Technologien, die Prozesssteuerungstechnologien auf einzelne Einheiten in der Art eines Traktors oder eines Mähdreschers anwenden, kann das IoT-Mesh-Netz Vorrichtungen aufweisen, die sich an zusätzlichen Fahrzeugen befinden, wie Getreidelastwagen, Tanklastwagen und anderen Vorrichtungen. Diese können verwendet werden, um einen weiteren Datensatz des Produktorts und von Transformationen, wenn es sich durch mehrere Vorrichtungen bewegt, bereitzustellen. Ferner können Vorrichtungen an festen Orten am IoT-Mesh-Netz teilnehmen, wie Vorrichtungen in Erzeugniskühlern, Silos und anderen Erntegut-Lagerungsbereichen, welche die Menge des gespeicherten Produkts, den Wassergehalt, die Schädlingsaktivität, Staubmengen und dergleichen verfolgen. Diese Vorrichtungen können Teil des Bauernhofnetzes sein oder Teil eines Firmennetzes sein, das Informationen in erster Linie für den Verfolgbarkeitsdatensatz sammelt.

[0582] Es sei bemerkt, dass das Produkt eine minimale Verfolgbarkeitseinheit aufweisen kann, beispielsweise die in einem Mähdrescher, einem Erzeugnislastwagen, einem Getreidelastwagen und dergleichen gehaltene Menge. Eine Menge unterhalb der minimalen verfolgbaren Einheit kann nicht unterscheidbar sein, weil die Verfolgungsinformationen nur für die minimale verfolgbare Einheit gelten. Ferner kann eine gewisse Vermischung des Produkts bei der Lagerung, beispielsweise in einem Getreidesilo, erwartet werden, wenngleich die Flusseigenschaften zur Bestimmung des Grads der auftretenden Vermischung verwendet werden können. Das gleiche Problem kann für ein pharmazeutisches Produkt oder andere Produkte, in denen eine Losgröße die minimale verfolgbare Einheit bestimmen kann, gelten.

[0583] In Block **11816** kann das Produkt sortiert und verpackt werden. Dies kann beispielsweise geschehen, wenn es auf einen Lastwagen oder einen Schienenwagen geladen wird. In Block **11818** können zusätzliche Informationen für einen Verfolgbarkeitsdatensatz in der Art der Zeit, des Datums, des Orts der Herstellung, der Versandfirma und dergleichen hinzugefügt werden. Diese Informationen können in die öffentlichen Informationen aufgenommen werden oder im Bauernhof-Datenspeicher **11810** gehalten werden.

[0584] In Block **11820** kann das Erzeugnis für den Versand etikettiert werden. Hierbei kann eine IoT-Vorrichtung an der Erzeugnisverpackung angebracht werden oder eine IoT-Vorrichtung auf andere Weise dem Produkt zugeordnet werden, bevor es versandt wird. Die IoT-Vorrichtung kann einen Datensatzschlüssel vom Bauernhof-Datenspeicher **11810** speichern. Bei anderen Beispielen kann ein Strichcode vor dem Versand auf die Erzeugnisverpackung gedruckt werden. Das Etikett kann den Datensatzschlüssel für den Zugriff auf den Verfolgbarkeitsdatensatz vom Bauernhof-Datenspeicher **11810** aufweisen. Der Datensatzschlüssel kann auch zu einem öffentlichen Datenspeicher **11824** in der Art einer öffentlichen Blockchain gesendet **11822** werden. Vom öffentlichen Speicher **11824** können Datensatzschlüssel für eine Stufe zugänglich sein.

[0585] In Stufe 2 **11826** kann das Produkt vom Bauernhof zur Herstellungsanlage transportiert werden. In Block **11828** können die Zeit, das Datum und die Versandfirma sowie die Loskennungen des Produkts aufgezeichnet werden. Die Daten **11830** können zu einem Versandspeicher **11832** gesendet werden, der sich beispielsweise auf einem Computer oder Server, der dem Lastwagen zugeordnet ist, befindet. Der Versandspeicher **11832** kann sich auch an einem zentralen Versandbüro für die Speditionsfirma befinden. In diesem Fall können die Daten **11830** durch einen Funk-Uplink, beispielsweise eine LTE-Strecke oder eine Satellitenstrecke, zum zentralen Versandbüro hochgeladen werden.

[0586] In Block **11834** können IoT-Vorrichtungen in der Art einer IoT-Vorrichtung, die der Produktverpackung zugeordnet ist, oder andere IoT-Vorrichtungen an einem Lastwagen oder einer Versandplattform, Versandverlaufereignisse überwachen. Versandverlaufereignisse können beispielsweise unter anderem Temperaturschwankungen, Beschleunigungskräfte, die Transportzeit oder Verzögerungen beim Transport sein. Die Daten **11830** vom Block **11834** können auch zum Versandspeicher **11832** gesendet werden.

[0587] In Block **11836** kann der Lieferstatus des Produkts aufgezeichnet werden. Dieser kann die Zeit, den Ort und das Datum der Lieferung einschließen. Die Lieferdaten können zum Versandspeicher **11832** gesendet werden. Ein Datensatzschlüssel kann für Stufe 2 **11826** erzeugt und im Versandspeicher **11832** gespeichert werden und in Block **11838** an den Datensatzschlüssel für Stufe 1 **11804** angehängt werden. Der Datensatzschlüssel kann zum öffentlichen Speicher **11824** gesendet **11840** werden und in einer der Verpackung zugeordneten IoT-Vorrichtung gespeichert werden. Nach Abschluss von Stufe 1 **11826** kann sich das Produkt am Ladedock einer Verarbeitungsanlage befinden.

[0588] In Stufe 3 **11842** wird das Produkt für Verkaufsstellen verarbeitet. In Block **11844** werden die Zeit, das Datum, die Verpackungsfirma und Loskennungen des Materials aufgezeichnet. Die Daten **11846** werden zu einem Verarbeitungsdatenspeicher **11852** gesendet. Dies kann die Verpackungs- und die Vorverkaufsverarbeitung einschließen. Beispielsweise kann für einen frisch geschnittenen Salat die Verarbeitung das Waschen

der Inhaltsstoffe, das Zerhacken der Inhaltsstoffe, das Mischen der Inhaltsstoffe und das Verpacken der Inhaltsstoffe einschließen. In Block **11848** können, wenn die jeweiligen Prozesse ausgeführt oder abgeschlossen wurden, Daten über den Status zum Verarbeitungsdatenspeicher **11852** gesendet werden. Sobald die Verarbeitung abgeschlossen ist, wird in Block **11850** die Datensatzkennung für Stufe 3 **11842** an die Datensatzkennung für vorhergehende Stufen angehängt, im Verarbeitungsdatenspeicher **11852** gespeichert und zum öffentlichen Speicher **11853** gesendet. Das verpackte, verarbeitete Produkt wird dann zu einem Ladedock für Stufe 4 **11854** bewegt, um es zur Endverkaufsstelle zu transportieren.

[0589] An jeder Stufe können die angehängten Datensatzschlüssel in der Verpackung zugeordneten IoT-Vorrichtungen gespeichert werden. Wie hier beschrieben, können die IoT-Vorrichtungen auch Versandverlaufsergebnisse verfolgen. In Stufe 4 **11854** werden in Block **11856** die Zeit, das Datum, die Versandfirma und die Loskennungen für das Produkt aufgezeichnet. Die Daten **11858** können zu einem Versandspeicher **11866**, beispielsweise an Bord des Lastwagens oder an der Versandfirma, gesendet werden. In Block **11859** überwachen die IoT-Vorrichtungen die Versandparameter für das Produkt, wobei sie beispielsweise Temperaturschwankungen, Beschleunigungskräfte und andere Parameter, welche die Qualität des Produkts beeinträchtigen können, aufzeichnen. Die Daten **11858** von der Überwachung der Versandparameter können auch zum Versandspeicher **11866** gesendet werden.

[0590] In Block **11860** kommt das Produkt am Ziel, beispielsweise einer Einzelhandels-Verkaufsstelle, an. Der Lieferstatus wird aufgezeichnet, und in Block **11862** wird eine Datensatzkennung oder ein Datensatzschlüssel an die Datensatzkennung oder den Datensatzschlüssel für vorhergehende Stufen angehängt. Der angehängte Schlüssel kann von der IoT-Vorrichtung gespeichert werden und zum öffentlichen Speicher **11824** gesendet werden **11864**.

[0591] Die Endstufe **11868** ist bei diesem Beispiel der Verkaufspunkt. In Block **11870** werden die Zeit, das Datum und der Lagerort aufgezeichnet. Die Frische des Produkts kann in Block **11872** beispielsweise durch die Rotation älterer Waren, die durch die neueren Waren zu ersetzen sind, berücksichtigt werden. In Block **11874** kauft der Verbraucher das Produkt. Der Verbraucher kann eine kommerzielle Firma sein oder eine Privatperson sein, die das Produkt an einem Laden oder einem anderen Einzelhandelsgeschäft kauft.

[0592] Nach dem Kauf kann der Verbraucher oder andere Entitäten eine Verfolgbarkeitsprüfung am Produkt ausführen möchten. Beispielsweise kann auf den Verfolgbarkeitsdatensatz zugegriffen werden, um den Ursprung einer Verunreinigung im Produkt zu bestimmen. In Block **11876** wird festgestellt, ob eine Verfolgbarkeitsprüfung angefordert wurde. Falls dies der Fall ist, wird in Block **11878** auf den Datensatzschlüssel zugegriffen. In Block **11880** kann dies durch Erhalten des Datensatzschlüssels von einer IoT-Vorrichtung auf der Verpackung oder auf dem Versandbehälter oder durch Zugreifen auf die Daten **11882** vom öffentlichen Speicher **11824** unter Verwendung einer SKU-Nummer oder einer anderen Paketmarkierung geschehen. Der angehängte Datensatzschlüssel, der die verketteten Datensatzschlüssel für die Stufen umfasst, wird zuerst auf Kontinuität vom Ausgangsort bis zum Verkaufspunkt geprüft. Dieser Prozess kann eine zyklische Redundanzprüfung des Gesamtschlüssels, eine Exklusiv-ODER-Operation der einzelnen Datensatzschlüssel mit einer Prüfung auf Null, um die Gültigkeit anzugeben, oder einen Onlineüberprüfungsprozess, wobei ein Schlüssel in einer Datenbank oder im öffentlichen Speicher **11824** nachgeschlagen wird, einschließen. In Block **11884** kann der Datensatzschlüssel verwendet werden, um auf in den privaten Speichern **11810**, **11832**, **11852** oder **11866** gespeicherte Daten zuzugreifen **11886**.

[0593] Nach der Überprüfung können die Daten dann dem Antragsteller präsentiert werden. Verfügbare Optionen können das Anzeigen eines vollständigen oder teilweisen Wegs vom Ursprungsort bis zu Verkaufspunkten für den Antragsteller einschließen. Andere Operationen können das Anzeigen eines Werts oder einer Textnachricht in Bezug auf das Ergebnis des Verfolgbarkeitsschlüssels einschließen, was mit einer Aktivierung eines Tons, einer Farbe, eines Bilds oder eines anderen Typs eines sensorischen Alarms für den Antragsteller einhergehen kann.

[0594] Eine Blockchain kann ein inhärent verfolgbares System sein, so dass die IoT-Vorrichtungen die öffentlichen und privaten Blockchains als Weg zum Herstellen von Vertrauen in die Arbeitsweise des IoT-Netzes unter Verwendung verteilter Vertrauenssemantiken ausnutzen können.

[0595] Ein Auditierungssystem kann dabei helfen, zu gewährleisten, dass die Sammlung und Speicherung von Verfolgbarkeits-Hashes unter einem provinziellen Vertrauensverfahren korrekt behandelt wird. Ein System in der Art desjenigen auf der Grundlage einer TPM mit einer PKI zur Signierung von PCR-Messungen von Verfolgbarkeits-Hashes kann eine auf dem institutionellen Vertrauen beruhende Strategie sein. Ein Verfahren,

das den privaten Speicher um eine Blockchain oder eine Hierarchie von Blockchains bildet, verwendet das Infrastrukturvertrauen.

[0596] Fig. 119 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 11900 vorhanden sein können, um Verfolgbarkeitsdatensätze für ein Produkt bereitzustellen, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 11900 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0597] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der Verfolgbarkeit von Datensätzen für ein Produkt aufweisen, wie hier beschrieben. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0598] Der Massenspeicher 808 kann einen Datensatzschlüsselgenerator 11902 aufweisen, der einen Datensatzschlüssel an den Herstellungs- und Verkaufsstufen erzeugt, wobei der Datensatzschlüssel an einer Stufe erzeugt werden kann. Ein Privater-Speicher-Kommunikator 11904 kann mit einem privaten Speicher kommunizieren, um den Datensatzschlüssel und von der IoT-Vorrichtung 11900 und anderen IoT-Vorrichtungen 812 gesammelte Daten an den Herstellungs- und Verkaufsprozessstufen zu speichern. Ein Öffentlicher-Speicher-Kommunikator 11906 kann mit dem öffentlichen Speicher kommunizieren, um den Datensatzschlüssel im öffentlichen Speicher zu speichern. Der Öffentlicher-Speicher-Kommunikator 11906 kann andere Informationen im öffentlichen Speicher speichern, wie die Versandtemperatur, die Versandzeit und dergleichen. Der Öffentlicher-Speicher-Kommunikator 11906 kann die angehängten Datensatzschlüssel für alle Stufen aus einem Datensatzschlüsselspeicher 11908 im öffentlichen Speicher speichern. Eine Datenüberwachungseinrichtung 11910 kann Sensoren 820 in der Art eines Temperatursensors, von Beschleunigungssensoren und dergleichen über die Schnittstelle 818 überwachen, um festzustellen, ob es Versandprobleme in der Art von Temperaturextremwerten, Schockextremwerten und dergleichen für das Produkt gab. Eine Warneinrichtung 11912 kann ein Betätigungselement 822 über die Schnittstelle 818, beispielsweise eine Lampe, einen Tongeber oder eine andere Vorrichtung, aktivieren, falls eine Versandgrenze in der Art der Temperatur verletzt wurde. Die Warneinrichtung 11912 kann auch jegliche Versandverletzungen im öffentlichen Datenspeicher speichern, beispielsweise durch den Öffentlicher-Speicher-Kommunikator 11906.

[0599] Fig. 120 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums 12000, das einen Code aufweist, um einen Prozessor 902 anzuweisen, Ressourcen über Domains zu teilen, gemäß einigen Ausführungsformen. Der Prozessor 902 kann über einen Bus 904 auf das nichtflüchtige maschinenlesbare Medium 12000 zugreifen. Der Prozessor 902 und der Bus 904 können wie mit Bezug auf Fig. 9 beschrieben sein. Das nichtflüchtige maschinenlesbare Medium 12000 kann für den Massenspeicher 808 aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0600] Das nichtflüchtige maschinenlesbare Medium 12000 kann einen Code 12002 aufweisen, um den Prozessor 902 anzuweisen, Daten aufzuzeichnen, während sich ein Produkt in einer Verarbeitungs- oder Versandstufe befindet. Ein Code 12004 kann aufgenommen sein, um den Prozessor 902 anzuweisen, die Stufen- und Daten an einen privaten Speicher zu übermitteln. Ein Code 12006 kann aufgenommen sein, um den Prozessor 902 anzuweisen, einen Datensatzschlüssel für eine Stufe zu erzeugen. Ein Code 12008 kann aufgenommen sein, um den Prozessor 902 anzuweisen, den Datensatzschlüssel für die Stufe im privaten Speicher zu speichern. Ein Code 12010 kann aufgenommen sein, um den Prozessor 902 anzuweisen, den Datensatzschlüssel für die Stufe an vorhergehende Datensatzschlüssel für vorhergehende Stufen anzuhängen. Ein Code 12012 kann aufgenommen sein, um den Prozessor 902 anzuweisen, die angehängten Stufenschlüssel zu einem öffentlichen Speicher zu senden. Ein Code 12014 kann aufgenommen sein, um den Prozessor 902 anzuweisen, eine Warnung auszugeben, falls Datengrenzen unter anderem in der Art von Temperaturgrenzen während einer Stufe verletzt wurden. Ein Code 12014 kann aufgenommen sein, um den Prozessor 902 anzuweisen, die Datengrenzverletzung im öffentlichen Speicher zu speichern.

[0601] Richtlinien werden als ein Satz von Regeln zum Behandeln und Steuern des Zugriffs auf Netzressourcen definiert. Eine Richtlinie kann einen Satz von Ereignissen, Bedingungen, Aktionen, Gegenständen und Zielen aufweisen. Eine Richtlinie stellt die Ereignisse, Bedingungen, Aktionen, Gegenstände und Ziele zu einer

Richtlinienstruktur zusammen, die eine Vorrichtung oder ein Netz anweist, auf auftretende Bedingungen zu reagieren.

[0602] Für IoT-Mesh-Netze, beispielsweise in den verschiedenen Stufen des Herstellungsprozesses beim vorstehenden Beispiel, kann jedoch die Verbreitung von Richtlinien adressiert werden müssen. Ferner kann die Verwendung weit verteilter IoT-Netze die Relevanz von Richtlinien in der Art von Richtlinien zum Schützen der Sicherheit von Daten, zum Ändern der gesammelten Daten oder zur Erhöhung der Zugänglichkeit dieser Daten vergrößern.

[0603] Fig. 121(A) ist eine Schemazeichnung eines in Computernetzen verwendeten hierarchischen Richtlinien-Behandlungssystems **12100** gemäß einigen Ausführungsformen. Ein Ansatz für die Echtzeitbehandlung von Vorrichtungsrichtlinien ist eine hierarchische Rundsendearchitektur. Diese kann durch ein Veröffentlichungs-Subskriptions-Modell auf der Grundlage von Bloomfiltern, wie hier beschrieben, ersetzt werden. Der typische Fluss geht von einem Zentralsystem in der Art eines zentralisierten Cloud-Servers **12102** aus, der Richtlinien an Subeinheiten in der Art eines Gateways **12104** überträgt. Der Gateway **12104** kann die Richtlinie dann zu einer niedrigeren Ebene **12106**, einschließlich IoT-Endpunktvorrichtungen **12406**, übertragen. Eine der IoT-Endpunktvorrichtungen **12108** kann die Richtlinien dann zu einer niedrigeren Ebene **12110**, beispielsweise zu Sensorvorrichtungen oder anderen Einheiten übertragen.

[0604] Beim hierarchischen Richtlinien-Behandlungssystem **12100** sind die einzelnen Vorrichtungen direkt adressierbar. Durch seine Natur kann die Installation von Richtlinien bei dieser Architektur erfordern, dass der Administrator explizit die Adresse aller Zielknoten kennt und weiß, wie fehlerhafte Knoten oder Richtlinien zu ersetzen sind. Zusätzlich können Vorrichtungen häufig infolge von Ressourcenrandbedingungen eine begrenzte Anzahl von Richtlinien im lokalen Speicher speichern und die Richtlinien ersetzen, wenn zusätzliche Richtlinien implementiert werden.

[0605] Wie hier beschrieben, kann ein verteiltes richtlinienbasiertes Behandlungsframework implementiert werden, um Richtlinien in einem Netz zu speichern, zu lokalisieren, auf sie zuzugreifen und sie auszuführen. Dieses Framework kann einen Peer-to-Peer(P2P)-Richtlinienspeicher- und Installationsmechanismus zur Verwendung eines verfügbaren Speichers, beispielsweise im IoT-Mesh-Netz, verwenden. Dies kann zu einem Richtliniensystem führen, das bei einem Knotenversagen und einem Einzelpunktversagen hilft.

[0606] Fig. 121(B) ist eine Schemazeichnung der Richtlinienbehandlung in einem Peer-to-Peer(P2P)-Netz in der Art eines IoT-Mesh-Netzes gemäß einigen Ausführungsformen. Im P2P-Netz verteilt ein Koordinator **12112** in der Art eines Gateways Richtlinien **12114** an Nachbarn in der Art angeschlossener Knoten **12116**, welche die nächsten Nachbarn sein können. Diese Nachbarn können die Richtlinien dann zu anderen angeschlossenen Knoten weiterleiten.

[0607] Weil das IoT-Mesh-Netz naturgemäß skaliert und heterogen wird, kann es erforderlich sein, große Anzahlen von Richtlinien zu definieren und kontinuierlich zu ändern, um dabei zu helfen, zu gewährleisten, dass das IoT-Mesh-Netz Operationsziele erfüllt. Eine autonome Netzbehandlung in der Art einer verteilten Richtlinienbehandlung kann die Entscheidungsfindungsprozesse, die an optimierenden Netzoperationen beteiligt sind, automatisieren und verteilen. Dies kann es Administratoren ermöglichen, sich weniger auf Vorrichtungskonfigurationsprozesse niedriger Ebene zu konzentrieren. Die Aufnahme von Richtlinien in ein autonomes Behandlungssystem kann mit Verfahren und Algorithmen zur Richtlinienübersetzung, Codeerzeugung, Konfliktanalyse und Richtlinienenerzwingung einhergehen.

[0608] Fig. 122 ist ein schematisches Diagramm von Systemen in Knoten **12116** zur Implementation eines verteilten Richtlinien-Behandlungssystems **12200** gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 121 erörtert wurden. Jeder der Knoten **12116** kann eine Richtlinien-Entscheidungsmaschine **12202**, eine Richtlinien-Erzwingungsmaschine **12204**, einen Richtlinien-Ablagespeicher **12206** und eine Überwachungseinrichtung **12208** implementieren. Der Richtlinien-Ablagespeicher **12206** speichert die Richtlinien für den Knoten **12116**, der keine hohe Speicherkapazität benötigen kann. Die Richtlinien-Entscheidungsmaschine **12202** trifft Entscheidungen in Bezug darauf, welche Richtlinien erzwungen werden, welche zur Richtlinien-Erzwingungsmaschine **12204** weitergeleitet werden. Die Entscheidungen können auf den im Richtlinien-Ablagespeicher **12206** gespeicherten Richtlinien sowie auf von der Überwachungseinrichtung **12208** mitgeteilten Einschaltzustandsinformationen beruhen. Die Richtlinien-Entscheidungsmaschine **12202** interagiert mit anderen Knoten **12116**, um Richtlinien zu nicht konfigurierten Knoten zu verteilen. In einem nicht konfigurierten Knoten kann die Richtlinien-Entscheidungsmaschine **12202** mit anderen Knoten **12116** kommunizieren, um auf Richtlinien zuzugreifen.

[0609] Die Richtlinien-Erzwingungsmaschine **12204** implementiert von der lokalen Richtlinien-Entscheidungsmaschine **12202** bereitgestellte Richtlinienentscheidungen. Die lokale Richtlinien-Erzwingungsmaschine **12204** sammelt auch Informationen über ihren Zustand, über Netzverkehr, über Übertragungsfehler und Informationen, die ihr von der Überwachungseinrichtung **12208** mitgeteilt werden.

[0610] Die Überwachungseinrichtung **12208** verbindet sich mit der lokalen Richtlinien-Erzwingungsmaschine **12204** und mit Überwachungseinrichtungen **12208** in anderen Knoten **12116**. Die Überwachungseinrichtung **12208** sammelt Informationen in spezifischen Intervallen und speichert sie in einer Datenbank, beispielsweise im lokalen Richtlinien-Ablagespeicher **12206**. Beispiele von Informationen, die von der Überwachungseinrichtung **12208** gesammelt werden können, umfassen die aktuelle Vorrichtungskonfiguration, Fähigkeiten und Funktionen, die von jedem Knoten unterstützt werden. Andere Informationen, die von der Überwachungseinrichtung **12208** gesammelt werden können, umfassen Informationen über die Dienste, die angeboten werden, über Knotenanforderungen für das Netz, über die Ressourcenverfügbarkeitsschätzung, ausgelöste Ereignisse und dergleichen.

[0611] **Fig. 123(A)** ist ein Leiterdiagramm eines beispielhaften Verfahrens **12300** eines neuen nicht konfigurierten Knotens **12302**, der versucht, Richtlinien auf einem Netz zu entdecken, beispielsweise von einem Peer-Knoten **12304**, gemäß einigen Ausführungsformen. Das Verfahren **12300** aus **Fig. 123(A)** kann durch die mit Bezug auf **Fig. 126** beschriebene IoT-Vorrichtung **12600** implementiert werden. Wenn der neue nicht konfigurierte Knoten **12302** dem Netz beitrifft, leitet er eine Richtlinienentdeckungsaktion ein. Er kann eine Entdeckungsnachricht **12306** zu einem Peer-Knoten **12304** übertragen und warten, bis ein Entdeckungs-Zeitüberschreitungs-Zeitgeber **12308** abläuft. Falls er keine Antwort empfängt, sendet er die Entdeckungsnachricht **12306** erneut.

[0612] Die Rollen eines koordinierenden Knotens, von konfigurierten Knoten und von neuen nicht konfigurierten Knoten können unter Verwendung eines Pub-Sub-Benachrichtigungssystems unter Verwendung von Bloomfiltern modelliert werden, wie hier beschrieben. Bei diesem Beispiel kann ein Bloomfilter-„Router“-Knoten als Koordinatorknoten wirken, um dabei zu helfen, zu gewährleisten, dass neue nicht konfigurierte Knoten existierende konfigurierte Knoten finden können. Existierende konfigurierte Knoten sind Veröffentlicher der Richtlinienobjekte, die sie gegenwärtig implementieren. Neue nicht konfigurierte Knoten können an den Richtlinienobjekten der konfigurierten Knoten teilnehmen. Änderungen oder Aktualisierungen an Richtlinienobjekten konfiguierter Knoten können eine Kaskade von Benachrichtigungsverkehr erzeugen, welche das Netz durchdringen kann.

[0613] **Fig. 123(B)** ist ein Leiterdiagramm eines beispielhaften Verfahrens **12310** eines neuen nicht konfigurierten Knotens **12302**, der Richtlinien von einem konfigurierten Knoten **12312** entdeckt, gemäß einigen Ausführungsformen. Das Verfahren **12310** aus **Fig. 123(B)** kann durch die mit Bezug auf **Fig. 126** beschriebene IoT-Vorrichtung **12600** implementiert werden. Der konfigurierte Knoten **12312** hat eine Richtlinie hoher Ebene, die ein Ziel des Netzes erfüllt. Bei einem Beispiel kann die Richtlinie hoher Ebene einschließen, wie Vorrichtungen im Netz Kommunikationen behandeln sollen, um die Dienstqualität mit der Stromreserve in Einklang zu bringen. Es kann eine Anzahl anderer Richtlinien implementiert werden. Der neue nicht konfigurierte Knoten **12302** sendet eine Entdeckungsnachricht **12306** zum konfigurierten Knoten **12312**. Der konfigurierte Knoten **12312** antwortet mit einer Angebotsnachricht **12314**.

[0614] Nach dem Empfang der Angebotsnachricht **12314** prüft der nicht konfigurierte Knoten **12302** die Nachricht. Falls das Angebot akzeptiert wird, sendet er eine Akzeptanznachricht **12316** als Antwort. Andernfalls wird eine Ablehnungsnachricht zum konfigurierten Knoten **12312** zurückgesendet.

[0615] Nach dem Empfang der Akzeptanznachricht **12316** sendet der konfigurierte Knoten **12312** eine Richtlinien-einleitungsnachricht **12318** zum nicht konfigurierten Knoten **12302**. Die Richtlinien-einleitungsnachricht **12318** nimmt die zum nicht konfigurierten Knoten **12302** zu sendenden Richtlinien auf. Der nicht konfigurierte Knoten **12302** verarbeitet die Richtlinienobjekte, installiert die Richtlinien und aktualisiert **12320** seinen Zustand zu einem konfigurierten Knoten **12322**.

[0616] Eine aktualisierte Richtlinie kann beispielsweise von einem Koordinator **12324** in einer Aktualisierungsnachricht **12326**, die von einem konfigurierten Knoten **12312** empfangen wird, abgesendet werden. Der konfigurierte Knoten **12312** kann eine Aktualisierung **12328** an der sich in Kraft befindenden Richtlinie nach Validierungs- und Richtlinienintegritätsprüfungen ausführen.

[0617] Die Validierungsprüfung kann feststellen, ob die Richtlinie in Konflikt mit einem aktuellen Ziel steht. Beispielsweise kann eine Richtlinie, die alle Vorrichtungen anweist, Strom zu sparen, zu allen Knoten in einem Netz übertragen werden. Wie hier beschrieben, kann dies in Bezug auf ein Pub-Sub-System beschrieben werden, wobei eine Stromverbrauchsbehandlungs-Richtlinie als Pub-Sub-„Thema“ spezifiziert und übernommen wird. Beispielsweise kann eine Richtlinienanweisung, die aussagt, auf dem Stromverbrauchsniveau 4 zu arbeiten, für Teilnehmer des Stromverbrauchsbehandlungsthemas veröffentlicht werden. Ein effizientes bloomfilterbasiertes Nachrichtenübertragungssystem hilft dabei, zu gewährleisten, dass Teilnehmer des Stromverbrauchsbehandlungsthemas über die Richtlinienänderung benachrichtigt werden.

[0618] Falls das Richtlinienobjekt eine sicherheitskritische Funktion impliziert, kann dem Empfang der Themenbenachrichtigungsnachricht das Eröffnen einer sicheren Sitzung an einem Richtlinienentscheidungspunkt, wo der Knoten Ende-zu-Ende-Sicherheitsberechtigungs-nachweise authentifizieren und einrichten kann, bevor anhand der Benachrichtigung gehandelt wird, folgen. Die Knoten können jedoch bereits aktiv eine Richtlinie implementieren, welche fordert, dass die Vorrichtungen eine bestimmte Dienstqualität bzw. Quality of Service (QoS) beibehalten. Die Implementation der Stromsparrichtlinie könnte in Konflikt mit der QoS-Richtlinie gelangen. Daher kann die neue Richtlinie abgelehnt werden.

[0619] Falls die Richtlinie eine Validierungsprüfung nicht besteht, kann die Aktualisierung eine teilweise Ersetzung der sich in Kraft befindenden Richtlinie ausführen. Eine teilweise Ersetzung kann die Berechnung einer Differenz zwischen der aktuellen sich in Kraft befindenden Richtlinie und der aktualisierten Richtlinie einschließen. Die teilweise Aktualisierung kann möglicherweise den Einfluss einer vollständigen Richtlinienänderung verringern, indem sie nur die betroffenen Richtlinienparameter oder -bedingungen modifiziert. Dies wird weiter mit Bezug auf **Fig. 124** erörtert.

[0620] Die Aktualisierungsnachricht **12326** kann auch eine Verkettung von Richtlinien aufweisen. Dies ist insbesondere auf verteilte und dispergierte Netzumgebungen anwendbar, bei denen eine Basisebenen-Richtlinie durch zusätzliche Richtlinienregeln, die von benachbarten Knoten empfangen werden, erweitert wird. Dies wird weiter mit Bezug auf **Fig. 125** erörtert.

[0621] Falls ein konfigurierter Knoten **12312** eine Richtlinie aktualisiert oder ersetzt hat, kann eine Konfliktwarnnachricht **12340** zu einem anderen konfigurierten Knoten **12322** gesendet werden, um ihn vor dem Richtlinienkonflikt zu warnen. Richtlinienkonflikt-Analyseprozesse müssen wirksam und skalierbar sein, um der dynamischen Natur und Größe solcher Kommunikationsnetze Rechnung zu tragen. Ein Richtlinienauswahlprozess für die Richtlinienkonfliktanalyse kann eine Geschichte vorhergehender Richtlinienvergleiche in einer baumbasierten Datenstruktur zur Verringerung der Anzahl der in nachfolgenden Iterationen erforderlichen Vergleiche aufrechterhalten.

[0622] **Fig. 124** ist ein Leiterdiagramm eines beispielhaften Verfahrens **12400** eines konfigurierten Knotens **12322**, der mit einem Knoten **12402** kommuniziert, der eine aktualisierte Richtlinie zur Aktualisierung der Richtlinien des konfigurierten Knotens **12322** aufweist, gemäß einigen Ausführungsformen. Das Verfahren **12400** aus **Fig. 124** kann durch die mit Bezug auf **Fig. 126** beschriebene IoT-Vorrichtung **12600** implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 123** beschrieben wurden. Dies kann beispielsweise geschehen, wenn der konfigurierte Knoten **12322** eine Konfliktwarnnachricht **12340** vom anderen Knoten **12402** empfängt. Der konfigurierte Knoten **12322** kann eine Entdeckungsnachricht **12306** zum aktualisierten Knoten **12402** senden.

[0623] Der aktualisierte Knoten **12402** kann mit einer Angebotsnachricht **12314** antworten, welche den konfigurierten Knoten **12322** über die Richtlinienaktualisierung benachrichtigt. Der konfigurierte Knoten **12322** kann dann mit einer Akzeptanznachricht **12316** antworten, um dem aktualisierten Knoten **12402** anzugeben, dass er die aktualisierte Richtlinie senden kann. Die aktualisierte Richtlinie kann dann in einer Aktualisierungsnachricht **12404** vom aktualisierten Knoten **12402** zum konfigurierten Knoten **12322** gesendet werden. Nach Validierungs- und Richtlinienintegritätsprüfungen kann der konfigurierte Knoten **12322** dann eine vollständige oder teilweise Ersetzung einer sich in Kraft befindenden Richtlinie ausführen **12328**.

[0624] Um festzustellen, ob nur eine teilweise Ersetzung erforderlich ist, kann ein Verfahren zum Berechnen einer Differenz zwischen den Richtlinien implementiert werden. Beispielsweise kann ein Vergleich zwischen einzelnen Regeln in der neuen Richtlinie und der alten Richtlinie vorgenommen werden, um festzustellen, ob Regeln hinzugefügt, entfernt oder modifiziert wurden, beispielsweise durch Änderung eines Parameterwerts für die Regel. In einem Bloomfiltermodell sind die verschiedenen Übernehmer einer Richtlinie als Benachrichtigungen im Bloomfilter repräsentierbar. Änderungen an einer Richtlinienentscheidung werden zu Richtlinienener-

zwingungspunkten übertragen, welche die Teilnehmer an PDP sind, welche die Veröffentlicher sind. Die gleichen Effizienzaspekte, die von der Bloomfilter-Benachrichtigungsübertragung, wie hier beschrieben, geboten werden, können für die Implementation einer verteilten Richtlinienbehandlung ausgenutzt werden.

[0625] Wenn die Anzahl der IoT-Vorrichtungen skaliert, ist eine geeignete Differenztechnologie ein integraler Bestandteil eines verteilten Richtlinien-Behandlungssystems. Eine geringere Dateigröße für eine Differenzdatei kann die Größe der Aktualisierungsdatei, die über das Netz verteilt wird, verringern, wodurch weniger Zeit in Anspruch genommen wird und weniger Netzstaus hervorgerufen werden. Weil Richtlinienaktualisierungen in Bezug auf die Priorität, Komplexität und Größe variiert werden können, können durch Senden nur der Änderungen kleinere Dateien erzeugt werden. Diese Dateien würden die Differenz (oder den Differenzwert) zwischen der aktuellen Richtlinie und der neuen Richtlinie beispielsweise durch Auswählen einer adaptiven Differenzkompressionstechnik auf der Grundlage der Anforderungen oder Wünsche des Clients wirksam einkapseln.

[0626] Richtlinienaktualisierungen können auch Beschränkungen der Hardware auf der Clientseite berücksichtigen. Beispielsweise können in verschiedenen IoT-Mesh-Netzen in der Art elektronischer Steuereinheiten bzw. Electronic Control Units (ECUs) von Kraftfahrzeugen, eingebetteter Module und Maschine-zu-Maschine- bzw. Machine-to-Machine(M2M)-Vorrichtungen, die in Einrichtungen, Herstellungsanlagen und bei der Logistik verwendet werden, Vorrichtungen beschränkt sein. Eine komprimierte Datei, die ausgesendet wird, kann nur entsprechend der Kapazität der Hardware rekonstruiert werden. Sie kann durch CPU, flüchtigen Speicher und permanenten Speicher begrenzt sein. Falls die Empfangsvorrichtung nicht die Ressourcen hat, um eine Richtlinienänderung zu implementieren, kann der Absender dies vorwegnehmen. Diese Restriktionen können von Vorrichtung zu Vorrichtung variieren, so dass ein einstellbares und adaptives System möglicherweise in der Lage sein muss, entsprechend zu komprimieren.

[0627] Die Fähigkeit, historische Informationen in den Auswahlprozess aufzunehmen, kann durch einen Zweiphasenansatz im Konfliktanalysealgorithmus verwirklicht werden. Die erste Phase des Algorithmus initialisiert eine Beziehungsmustermatrix zwischen einer Kandidatenrichtlinie und einer installierten Richtlinie, und die zweite Phase vergleicht dieses Muster mit einer Konfliktsignatur. Einige Lösungen vergleichen Kandidatenrichtlinien sequenziell mit allen installierten Richtlinien. Der hier beschriebene beispielhafte Ansatz kann jedoch die bereits entdeckten Muster anhand vorhergehender Iterationen des Algorithmus wiederverwenden, um die Anzahl der Vergleiche zu verringern. Durch die Verwendung dieses Ansatzes können Funktionsweiseverbesserungen vorgenommen werden, der Grad dieser Verbesserung kann jedoch von der Natur der Beziehungen zwischen installierten Richtlinien abhängen.

[0628] Die Richtlinien können gestaffelt sein. Beispielsweise kann eine Richtlinie ein Hinweiszeichen aufweisen, das fordert, dass sie ohne eine Hypothesenprüfung implementiert wird. Umgekehrt könnte ein Knoten einen Richtlinienkompromiss vorschlagen, falls er eine Richtlinie nicht implementieren konnte. Dies könnte in einem mit einer geschlossenen Schleife arbeitenden System ausgeführt werden. Ein Beispiel kann eine Richtlinie sein, die fordert, dass die IoT-Vorrichtungen die Übertragungsintervalle von alle 5 Minuten zu alle 5 Stunden erhöhen. Falls implementiert, könnte diese Richtlinie die QoS-Anforderungen für die Vorrichtung brechen. Die Vorrichtung kann eine Übertragungsrate von einmal pro Stunde anbieten.

[0629] Es sei bemerkt, dass ein Satz von Richtlinien, wodurch die verfügbaren durch eine Ortsrichtlinie steuerbaren Parameter repräsentiert werden, unter Verwendung eines Satzes von Richtlinienobjektkennungen modelliert werden kann, die jeweils einer Benachrichtigung entsprechen, welche ferner durch ein Bloomfilter repräsentierbar ist, wie hier beschrieben. Eine existierende Benachrichtigungsübertragungsfähigkeit auf der Grundlage von Bloomfiltern kann ausgenutzt werden, um Benachrichtigungen zu übertragen, die von einer Netzverwaltungsentität auferlegten Richtlinienänderungen entsprechen. Wenn eine Richtlinienänderungsb Benachrichtigung empfangen wird, kann der Knoten eine sichere Verbindung zu einem Richtlinienserver öffnen, um weitere Anweisungen in Bezug auf Richtlinienerzwingungspunktanpassungen zu erhalten.

[0630] Nicht dateibasierte Richtlinien können für eine erhöhte Sicherheit implementiert werden. Ferner könnten nicht dateibasierte Systeme für das Speichern von Richtlinien in Vorrichtungen, die keinen Speicher außerhalb des RAM aufweisen, verwendet werden. Gemäß einigen Aspekten wird, wenn eine Vorrichtung eine Richtlinie empfängt, die Richtlinie nicht gespeichert, sondern werden stattdessen beispielsweise bestimmte Parameter im RAM aktualisiert und unmittelbar implementiert. Ferner können Richtlinienparameter im ROM gespeichert werden. In einer sicheren leichtgewichtigen Vorrichtung kann die Ausführung der Richtlinien aus dem ROM ausgeführt werden, wobei einige Parameter aus dem RAM gelesen werden. Demgemäß kann ein ROM als Kern wirken, wobei alle anderen Merkmale im RAM ausgeführt werden.

[0631] Fig. 125 ist ein Leiterdiagramm 12500 eines beispielhaften Verfahrens zur Verkettung von verschiedenen Knoten durch den konfigurierten Knoten erhaltener Richtlinien gemäß einigen Ausführungsformen. Das Verfahren 12500 aus Fig. 125 kann durch die mit Bezug auf Fig. 126 beschriebene IoT-Vorrichtung 12600 implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 123 beschrieben wurden. Bei diesem Beispiel hat ein erster Knoten 12502 eine aktualisierte Richtlinienkomponente A, während ein zweiter Knoten 12504 eine aktualisierte Richtlinienkomponente B hat. Der konfigurierte Knoten 12322 kann eine Konfliktwarnnachricht 12340 empfangen haben, die angibt, dass er Richtlinien im konfigurierten Knoten 12322 aktualisieren muss. Der konfigurierte Knoten 12322 sendet eine erste Entdeckungsnachricht 12506 zum ersten Knoten 12502. Der konfigurierte Knoten sendet auch eine zweite Entdeckungsnachricht 12508 zum zweiten Knoten 12504. Ansprechend darauf sendet der erste Knoten 12502 eine Richtlinienaktualisierungsnachricht 12510 zum konfigurierten Knoten 12322. Die Richtlinienaktualisierungsnachricht 12510 weist die Richtlinienkomponente A auf, die der konfigurierte Knoten 12322 an die aktuelle Richtlinie anhängt 12512.

[0632] Der zweite Knoten 12504 sendet eine Angebotsnachricht 12514 zum konfigurierten Knoten 12322 und lässt den konfigurierten Knoten 12322 wissen, dass der zweite Knoten 12504 die Richtlinienkomponente B hat. Der konfigurierte Knoten 12322 sendet eine Akzeptanznachricht 12516 zum zweiten Knoten 12504 und lässt ihn wissen, dass er die Aktualisierung akzeptiert. Der zweite Knoten 12504 sendet dann eine Richtlinienaktualisierungsnachricht 12518, welche die Richtlinienkomponente B aufweist, welche der konfigurierte Knoten 12322 an die aktuelle Richtlinie anhängt 12520. Dies führt zu einer Richtlinienkonfiguration für den konfigurierten Knoten 12322, welche eine Kombination der Richtlinienkomponenten von den verschiedenen anderen Knoten ist, wie in Tabelle 3 dargestellt ist.

[0633] Falls eine Bloomfilterstruktur für die Richtlinienverteilung verwendet wird, kann das Richtlinienobjekt eine Richtlinienobjektkennung (OID) mit Linienelementen in der Richtlinienstruktur in Zusammenhang bringen, wobei jede Richtlinien-OID einem Bit in einem Bloomfilter entsprechen kann. Bei diesem Beispiel kann jeder Knoten, der einen Satz von OID implementiert, das eine OID abdeckende Bloomfilter übernehmen. Folglich kann das gleiche Benachrichtigungssystem, welches das Pub-Sub-Routing implementiert, ausgenutzt werden, um ein verteiltes Richtlinienerzwingungsverfahren zu implementieren.

Tabelle 3: Richtlinien im konfigurierten Knoten

RICHTLINIE	
Basisebene	
	Richtlinienkomponente von Knoten 1
	Richtlinienkomponente von Knoten 2
	...
	Richtlinienkomponente von Knoten N

[0634] Die Knoten in einem Mesh-Netz sind nicht darauf beschränkt, alle die gleichen Richtlinien zu implementieren oder alle in der gleichen Weise zu implementieren. Beispielsweise kann ein Knoten, der eine schwache Batterie spürt, eine Richtlinie zum Sparen von Batterieenergie implementieren, während andere Knoten, welche diese Einschränkung nicht aufweisen, mit Richtlinien fortsetzen können, die eine QoS aufrechterhalten.

[0635] Fig. 126 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 12600 zur verteilten Behandlung von Richtlinien vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3, Fig. 8 und Fig. 122 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 12600 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0636] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0637] Der Massenspeicher 808 kann eine Richtlinien-Entscheidungsmaschine 12202 zum Feststellen, welche Richtlinien erzwungen werden, aufweisen. Eine Richtlinien-Erzwingungsmaschine 12204 implementiert die Richtlinienentscheidungen. Ein Richtlinien-Ablagespeicher 12206 speichert die Richtlinien für die IoT-Vor-

richtung 12600. Die Überwachungseinrichtung **12208** kommuniziert mit Überwachungseinrichtungen in anderen Knoten im Mesh-Netz **812** und sammelt Informationen, beispielsweise einschließlich der Vorrichtungskonfiguration, von Fähigkeiten und Funktionen, die von den Knoten unterstützt werden.

[0638] Ein Datensammler **12602** kann Daten durch die Schnittstelle **818** von den Sensoren **820** sammeln. Ein Kommunikator **12604** kann die vom Datensammler **12602** oder von anderen Einheiten in der Art der Überwachungseinrichtung **12208** oder der lokalen Richtlinien-Entscheidungsmaschine **12202** gesammelten Daten zu anderen Vorrichtungen im Mesh **812** oder in der Cloud **302** übertragen.

[0639] Fig. 127 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **12700**, das Code aufweist, um einen Prozessor **902** anzuweisen, Richtlinien in einem IoT-Netz in Zusammenarbeit mit anderen IoT-Vorrichtungen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **12700** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf Fig. 9 beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **12700** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0640] Das nichtflüchtige maschinenlesbare Medium **12700** kann einen Code **12702** aufweisen, um den Prozessor **902** anzuweisen, Richtlinien in anderen Knoten zu entdecken. Ein Code **12704** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Richtlinien von Nachrichten zu aktualisieren, die von den anderen Knoten gesendet wurden. Ein Code **12706** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die von mehreren Knoten erhaltenen Richtlinien zu verketteten. Ein Code **12708** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die von den anderen Knoten erhaltenen Richtlinien zu validieren. Ein Code **12710** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Differenz oder eine Änderung für Richtlinien gegenüber aktuellen Richtlinien zu berechnen.

[0641] Ein Code **12712** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Richtlinien abzulehnen, die in Konflikt mit Gruppenzielen stehen. Der Code **12712** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine teilweise Implementation von Richtlinien zu verhandeln, die in Konflikt mit Gruppenzielen stehen. Ein Code **12714** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Richtlinien zu ändern, die implementiert wurden, um mit aktuellen Bedingungen übereinzustimmen.

[0642] Zusätzlich zur Verteilung von Richtlinien und zur Ausführung von Funktionen ist das Aufrechterhalten der Verfügbarkeit von IoT-Vorrichtungen relevant, beispielsweise um dabei zu helfen, den Verlust von den IoT-Vorrichtungen gesammelter Daten zu verhindern. Eine Technik, welche die Verfügbarkeit von IoT-Vorrichtungen erhöhen kann, könnte Außerbandmechanismen verwenden, um ihre Verfügbarkeit zu gewährleisten.

[0643] Fig. 128 ist eine Zeichnung einer beispielhaften Netzsteckervorrichtung **12800**, die verwendet werden kann, um die Verfügbarkeit einer IoT-Vorrichtung zu verbessern, gemäß einigen Ausführungsformen. Bei diesem Beispiel ist die Netzsteckervorrichtung **12800** in einen 220-Volt-Wechselspannungsstecker eingebaut. Linux und andere Betriebssysteme implementieren Funktionen, die entweder mit Software- oder mit Hardware-Watchdogs interagieren können. Diese Watchdogs sind Bordfunktionen, und sie können in eingebetteten Vorrichtungen oder in IoT-Vorrichtungen nicht immer vorhanden sein. Die Netzsteckervorrichtung **12800** kann ein Außerbandmechanismus zum Behandeln der Verfügbarkeit von generischen oder Nutzenanwendungs-IoT-Vorrichtungen sein. Sie kann diese Funktion durch Kommunizieren mit einer IoT-Vorrichtung über eine oder mehrere ihrer verfügbaren Schnittstellen **12802** ausführen, um festzustellen, ob die IoT-Vorrichtung betriebsfähig ist und korrekt funktioniert. Falls dies nicht der Fall ist, kann die Netzsteckervorrichtung **12800** behebbare Aktionen ausführen, um die IoT-Vorrichtung in einen funktionstüchtigen Zustand zurückzuführen. Wie hier beschrieben, kann die Netzsteckervorrichtung **12800** als primäre Stromquelle für die IoT-Vorrichtung wirken. Auf diese Weise kann sie die Verfügbarkeit verbessern, indem sie in der Lage ist, die Stromzufuhr zu einer damit verbundenen behandelten Vorrichtung zyklisch zu unterbrechen.

[0644] Es kann ein Bereich von Außerband-Behandlungstechnologien implementiert werden. Beispiele von diesen können für Desktop- oder Client-Systeme vPro von Intel Corporation und für Server- oder Datenzentrumssysteme die intelligente Plattformbehandlungsschnittstelle bzw. Intelligent Platform Management Interface (IPMI) von Intel Corporation einschließen. Diese Technologien wurden jedoch in eingebetteten Vorrichtungen oder Vorrichtungen der IoT-Klasse nicht verfügbar gemacht. Demgemäß führen IoT-Vorrichtungen beim Erreichen der Verfügbarkeit eine neue Komplexitätsebene herbei, weil diese Vorrichtungen Herausforderungen für ihre Behandlung herbeiführen können.

[0645] Die Netzsteckervorrichtung **12800** kann eine standardmäßige Wandsteckdose als Stromquelle verwenden, oder ihr kann unter anderem durch Batterien oder über eine Strom-über-Ethernet-Verbindung Strom zugeführt werden. Sie kann ihr eigenes Betriebssystem in der Art eines einfachen OS, RTOS oder BIOS ausführen.

[0646] Benutzer oder Betreiber der Vorrichtung können eine Web- oder andere Schnittstelle verwenden, um die Vorrichtung zu konfigurieren. Die Konfiguration der Vorrichtung kann einen Anmeldungsdienst aufweisen, der Rollen für Administratoren und Betreiber aufweisen kann. Ein Einstellungsbildschirm für eine Schnittstelle kann angeben, ob etwas mit der Schnittstelle verbunden ist. Parameter können auf der Schnittstelle festgelegt werden, um als Regeln oder Richtlinien zu wirken, um vorzuschreiben, wann und welche Aktionstypen ausgeführt werden sollten, falls die Vorrichtung einem Betriebsproblem begegnet. Dieses kann beispielsweise vom Neustarten eines Dienstes oder des Betriebssystems der IoT-Vorrichtung bis zum zyklischen Unterbrechen der Stromzufuhr zur IoT-Vorrichtung reichen.

[0647] Die Netzsteckervorrichtung **12800** kann ihre eigenen Kommunikationskanäle aufweisen, die sie in die Lage versetzen, als ein Außerbandmechanismus zu wirken, um IoT-Vorrichtungen zu kontaktieren, die offline sind. Die Vorrichtung kann so hergestellt werden, dass sie den Anforderungen einer IoT-Installation entspricht, oder sie könnte allgemeiner mit einer Vielzahl von Schnittstellen **12802** hergestellt werden, um zu einem breiten Bereich von Vorrichtungen zu passen. Die Schnittstellen **12802** können beispielsweise USB-Verbindungen, Ethernet-Verbindungen, SPI/I2C/I3C, Sensor-HID, Magnetometer/Funkeinrichtungen oder JTAG einschließen. Diese Schnittstellen **12802** werden mit Bezug auf **Fig. 131** weiter erörtert.

[0648] **Fig. 129** ist eine Auftragung **12900** eines globalen Zustandsübergangs auf der Grundlage einer Selbstanpassung für die Netzsteckervorrichtung gemäß einigen Ausführungsformen. In der Auftragung **12900** kann festgestellt werden **12902**, ob eine IoT-Vorrichtung im Dienst oder außer Dienst ist. Überdies wird festgestellt **12904**, ob die behandelte IoT-Vorrichtung funktionstüchtig oder nicht funktionstüchtig ist. Wie hier beschrieben, führt eine funktionstüchtige Vorrichtung ihre zugewiesenen Funktionen und die Kommunikation mit einem Netz aus, während eine nicht funktionstüchtige Vorrichtung entweder keine Funktionen ausführt, nicht kommuniziert oder beides.

[0649] Die Netzsteckervorrichtung kann feststellen, dass die IoT-Vorrichtung außer Dienst ist **12906**, indem sie feststellt, dass der IoT-Vorrichtung kein Strom zugeführt wird, dass sie nicht in Kommunikation mit der Netzsteckervorrichtung ist oder auf andere Weise nicht auf die Netzsteckervorrichtung antwortet. Dies kann auf mehrere Arten geschehen, wobei beispielsweise, falls die Netzsteckervorrichtung durch eine SPI-, I2C- oder I3C-Schnittstelle mit der IoT-Vorrichtung verbunden ist, sie in der Lage sein kann, den Hardwarestatus von Komponenten an der IoT-Vorrichtung zu erkennen. Falls sie nicht korrekt funktionieren, kann eine korrigierende Aktion ausgeführt werden, wie das Rücksetzen der Vorrichtung, das zyklische Unterbrechen der Stromzufuhr und dergleichen.

[0650] Ferner kann die Netzsteckervorrichtung feststellen, ob die IoT-Vorrichtung in Dienst, jedoch nicht funktionstüchtig ist **12910**, indem festgestellt wird, ob das Betriebssystem auf der IoT-Vorrichtung läuft. Dies kann durch Kommunizieren mit dem Dienst getestet werden, um zu beweisen, dass das Betriebssystem antwortet, beispielsweise durch Mounten einer Platte auf der IoT-Vorrichtung, durch Senden einer Telnet-Nachricht zu einem Port oder durch Aktivieren eines anderen Dienstes sowie durch andere Tests. Die Netzsteckervorrichtung kann feststellen, ob Kommunikationen auf der IoT-Vorrichtung funktionieren. Dies kann durch Senden von Ping-Signalen zur IoT-Vorrichtung über eine Ethernet-Verbindung oder eine andere festverdrahtete Verbindung getestet werden. Dies kann auch geschehen, indem gemessen wird, ob die Funkeinrichtung auf der IoT-Vorrichtung sendet, oder indem ein Cloud-Fog-Dienst über die letzte von der IoT-Vorrichtung empfangene Nachricht abgefragt wird.

[0651] Die Netzsteckervorrichtung kann einen Letzter-Wille-Befehl von der IoT-Vorrichtung empfangen haben. Die IoT-Vorrichtung kann dafür ausgelegt werden, im Fall eines Systemcrashes einen Letzter-Wille-Befehl über jegliche oder alle ihrer Kommunikationswege auszugeben. Dies ist ein verfügbares Merkmal verschiedener Protokolle in der Art von MQTT, dies kann jedoch auf einer Hardwareebene durch den Entwickler der IoT-Vorrichtung implementiert werden, um ein Signal auf dem SPI-, I2C- oder einem I3C-Schnittstellenbus zu erzeugen, das die Netzsteckervorrichtung über eine SPI-, I2C- oder eine I3C-Schnittstelle empfangen kann. Der Letzter-Wille-Befehl kann andere Vorrichtungen informieren, dass die IoT-Vorrichtung ausgefallen oder gecrashed ist, oder er kann Anweisungen zu Aktionen aufweisen, die im Fall eines Ausfalls oder eines Crashes auszuführen sind. Ferner kann das Nichtvorhandensein einer periodischen „Ist-am-Leben“-Nachricht oder ei-

nes anderen Herzschlagsignals von einem Watchdog-Agenten in der IoT-Vorrichtung zur Netzsteckervorrichtung als Auslöser zum Rücksetzen der IoT-Vorrichtung verwendet werden.

[0652] Andere Vorrichtungen in einer Cloud oder einem Mesh können feststellen, dass die IoT-Vorrichtung entweder außer Dienst ist 12906 oder im Dienst, jedoch nicht funktionstüchtig ist 12910. Beispielsweise kann ein Cloud- oder Fog-Dienst, der feststellt, dass die IoT-Vorrichtung während eines konfigurierbaren Zeitraums keine Nachricht gesendet hat, einen fernen Befehl zur Netzsteckervorrichtung senden, um die Stromversorgung der IoT-Vorrichtung zurückzusetzen, um zu versuchen, sie zu einem funktionstüchtigen Zustand zurückzuführen. Die Netzsteckervorrichtung kann dann den Rücksetzbefehl für die IoT-Vorrichtung ausführen, beispielsweise durch Ziehen eines Rücksetzstifts auf den niedrigen Pegel, durch zyklisches Unterbrechen der Stromzufuhr und dergleichen.

[0653] Die Netzsteckervorrichtung kann ein Rücksetzen der Lebenszeit bzw. Time to Live (TTL) implementieren. Beispielsweise kann die Netzsteckervorrichtung die Stromzufuhr zur IoT-Vorrichtung regelmäßig periodisch unterbrechen. Bei vielen Installationen kann das regelmäßige zyklische Unterbrechen der Stromzufuhr zu einer größeren Aktivitätszeit führen. Das zyklische Unterbrechen der Stromzufuhr kann bei einem skriptgesteuerten Abschaltverfahren bevorzugter sein, während dessen ein nicht reagierender Prozess verhindern kann, dass das Betriebssystem heruntergefahren wird.

[0654] Durch die Verwendung dieser beispielhaften Techniken kann die Netzsteckervorrichtung die IoT-Vorrichtung über längere Zeiträume im funktionstüchtigen und dienstbereiten Quadranten **12908** halten. Dies kann verglichen mit einer Installation ohne die Netzsteckervorrichtung die Aktivitätszeit verlängern und damit die Zuverlässigkeit der IoT-Vorrichtung und des IoT-Netzes erhöhen.

[0655] Fig. 130 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **13000** zur Verwendung einer Netzsteckervorrichtung zur Erhöhung der Zuverlässigkeit einer IoT-Vorrichtung gemäß einigen Ausführungsformen. Das Verfahren **13000** aus Fig. 130 kann durch die mit Bezug auf Fig. 131 beschriebene IoT-Vorrichtung **13100** implementiert werden. Block **13002** repräsentiert beispielsweise den Fall, in dem die Netzsteckervorrichtung bootet. Während des Bootens lädt die Netzsteckervorrichtung ein Betriebssystem und dann benötigte Firmware und Treiber von einer internen Speichervorrichtung.

[0656] In Block **13004** entdeckt die Netzsteckervorrichtung eine Liste behandelter Vorrichtungen. Diese kann eine einzige in die Netzsteckervorrichtung eingesteckte Vorrichtung, mehrere Vorrichtungen oder ein IoT-Mesh-Netz oder eine Fog-Vorrichtung aufweisen. Diese Entdeckung kann dynamisch oder statisch erfolgen. Beispiele einer dynamischen Entdeckung würden darin bestehen, alle mit der Netzsteckervorrichtung verbundenen physikalischen Wege zu spezifizieren und abzufragen, was physikalisch verbunden ist, oder es könnten dabei die Funk- und Kommunikationsstrecken zyklisch durchlaufen werden, um nahe gelegene Vorrichtungen zu entdecken, die behandelt werden können.

[0657] Die Vorrichtung kann auch eine statische Liste in einer lokalen Konfigurationsdatei, die auf der Vorrichtung gespeichert ist, aufgelisteter behandelter Vorrichtungen aufweisen. Die Netzsteckervorrichtung fügt die behandelten Vorrichtungen zu einer lokal gespeicherten Liste hinzu. Die Entdeckung kann nach Bedarf wiederholt oder bei Erkennung einer neu angeschlossenen IoT-Vorrichtung implementiert werden. Beispielsweise ermöglicht das Vorhandensein von Sensor-HID und USB, eine Plug-n-Play-Erkennung behandelter Vorrichtungen vorzunehmen. Die Netzsteckervorrichtung kann Funk- und Kommunikationsstrecken periodisch zyklisch durchlaufen, um festzustellen, ob eine neue IoT-Vorrichtung vorhanden ist und zur Behandlungsliste hinzugefügt werden sollte. Statische Listen können der Vorrichtung von einem Fog- oder Cloud-Dienst oder durch direkte menschliche oder durch die Vorrichtung vorgenommene Interaktion mit der Netzsteckervorrichtung durch eine Schnittstelle, die dadurch verfügbar gemacht werden kann, wie eine RESTful-Benutzerschnittstelle oder Anwendungsprogrammierschnittstelle, bereitgestellt werden.

[0658] In Block **13006** initialisiert die Netzsteckervorrichtung eine Operation und kann diese beginnen. Die Netzsteckervorrichtung kann Richtlinien von einem fernen Host oder einer Orchestrierungsvorrichtung laden, welche Cloud- oder Fog-basiert sein kann. Richtlinien können unter Verwendung einer Schnittstelle, die auf der Vorrichtung läuft, zur Netzsteckervorrichtung geladen werden. Die Richtlinien können Regelsätze zum Identifizieren, welche behandelten Vorrichtungen mit welchen Richtlinien in Zusammenhang stehen, aufweisen. Dies kann es unterstützenden Betreibern ermöglichen, zu definieren, welche Arten von Aktionen für welche Arten von Vorrichtungen möglich sind. Dies kann auf dem Typ der Verbindung mit der Vorrichtung beruhen.

[0659] In Block **13008** überwacht die Netzsteckervorrichtung die behandelten IoT-Vorrichtungen über ihre jeweiligen Verbindungen. Die Funktionstüchtigkeitsprüfungen können die hier beschriebene Stromzufuhr- und Kommunikationsüberwachung, einen Bericht über die Funktionstüchtigkeit von einem Watchdog-Agenten in der IoT-Vorrichtung oder ein Nichtempfangen eines Funktionstüchtigkeitsberichts vom Watchdog-Agenten einschließen. Es kann eine beliebige Anzahl von Funktionstüchtigkeitsprüfungen ausgeführt werden, und Vorrichtungsentwickler können andere in diesem Dokument nicht früher beschriebene Verfahren implementieren. Dieser Teil des Betriebs führt zu periodischen Prüfungen der Funktionsweise der behandelten Vorrichtung.

[0660] In Block **13010** kann die Netzsteckervorrichtung den Status mitteilen. Falls die behandelte Vorrichtung beispielsweise nicht wie erwartet funktioniert, kann eine Warnrichtlinie verwendet werden, um festzustellen, ob eine Warnung gesendet werden sollte und welcher Warnungstyp zu senden ist. Warnungen können als Warnmeldungen, Fehler oder kritische Ausfälle klassifiziert werden. Verschiedene Nachrichtenübermittlungsmechanismen wie E-Mail, ein in ein Protokoll geschriebenes Ereignis, eine Text- oder SMS-Nachricht oder eine andere Warnungsart können den Warnungsniveaus zugeordnet werden. Jede von diesen oder Kombinationen davon können zu verschiedenen Endpunkten, Benutzern, Verteilungslisten und dergleichen gesendet werden.

[0661] In Block **13012** kann die Netzsteckervorrichtung versuchen, eine Aktion auszuführen, um einen Fehler zu korrigieren. Beispielsweise kann die Netzsteckervorrichtung, abhängig vom Verbindungstyp, versuchen, Dienste, die auf der fehlerhaften oder fehlerhaft behandelten IoT-Vorrichtung laufen, zu kontrollieren und Maßnahmen zu ergreifen, um den Fehler zu korrigieren, bevor versucht wird, die Vorrichtung wiederherzustellen.

[0662] Falls die behandelte IoT-Vorrichtung über eine Drahtlosverbindung angeschlossen ist, können Standardbefehle verwendet werden, um die Vorrichtung aus der Ferne neu zu starten oder zurückzusetzen. Dies kann jedoch nicht funktionieren, falls das Betriebssystem in der behandelten IoT-Vorrichtung nicht antwortet. Dementsprechend kann die Netzsteckervorrichtung, falls die IoT-Vorrichtung über ein Medium angeschlossen ist, das ihr auch Strom zuführt, wie Strom-über-Ethernet bzw. Power-over-Ethernet (PoE), USB oder Strom-Aus-Verbindungen, die Stromzufuhr zur behandelten IoT-Vorrichtung zyklisch unterbrechen, falls ein Problem erkannt wird.

[0663] Falls eine IoT-Vorrichtung über eine JTAG-Schnittstelle angeschlossen wird, wird es möglich, die IoT-Vorrichtung aus der Ferne zu debuggen, ein Flashen der Vorrichtungsfirmware bereitzustellen und dergleichen. Eine JTAG-Schnittstelle kann einen zweckgebundenen Debug-Port aufweisen, der eine serielle Kommunikationsschnittstelle implementiert. Die JTAG-Schnittstelle kann von der Netzsteckervorrichtung an einen Port an der IoT-Vorrichtung angeschlossen werden, der mit einem Testzugangsport bzw. Test Access Port (TAP) am Prozessor der IoT-Vorrichtung verbindet.

[0664] Die behandelten IoT-Vorrichtungen können dafür ausgelegt sein, eine Wachsamkeit der Netzsteckervorrichtungen auf dem Netz aufzunehmen. Falls eine beispielsweise weiß, dass ein nützliches Verfahren zur Wiederherstellung einer ausgefallenen Vorrichtung darin besteht, die Stromversorgung von ihr zyklisch zu unterbrechen, kann eine Mechanismen in die Boot-Sequenz der behandelten Vorrichtung implementieren, um Protokolle aufzuräumen, auf übliche Fehler zu prüfen und alle Prozesse automatisch neu zu starten, ohne dass ein menschlicher Eingriff stattfindet. Dieser Ansatz kann zusätzliche Funktionalität bereitstellen, selbst im Fall weniger wachsamer IoT-Vorrichtungen kann die Netzsteckervorrichtung jedoch die Stromzufuhr zyklisch unterbrechen, um zu versuchen, sie in einen funktionstüchtigen Zustand zurückzuführen.

[0665] **Fig. 131** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer Netzsteckervorrichtung **13100** vorhanden sein können, welche zur Erhöhung der Verfügbarkeit einer IoT-Vorrichtung verwendet werden können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden.

[0666] Die Netzsteckervorrichtung **13100** kann eine Stromversorgung **13102** aufweisen, die mit dem Stromnetz **13104** oder einer anderen Stromquelle in der Art einer Strom-über-Ethernet-Verbindung, einer Batterie und dergleichen gekoppelt ist. Die Stromversorgung **13102** stellt einer Stromversorgungs-Steuereinrichtung **13106** Strom bereit, die diesen Strom dann einem externen Stecker **13108** für eine IoT-Vorrichtung zuführen kann. Die Stromversorgungs-Steuereinrichtung **13106** kann über den Bus **806** mit dem Prozessor **802** gekoppelt sein, um zu ermöglichen, dass die Stromversorgung für eine IoT-Vorrichtung zyklisch unterbrochen wird.

[0667] Die Netzsteckervorrichtung **13100** kann eine USB-Schnittstelle **13110** aufweisen. Diese ermöglicht es, dass die Netzsteckervorrichtung **13100** als USB-Client einer IoT-Vorrichtung wirkt und der IoT-Vorrichtung

Strom über USB zuführt. Ein USB-Port **13112**, der mit der USB-Schnittstelle **13110** gekoppelt ist, kann sowohl eine Datenkopplung als auch eine Stromverbindung bereitstellen.

[0668] Die Netzschnittstellen-Steuereinrichtung (NIC) **816** kann durch einen Ethernet-Port **13114** eine direkte Verbindung mit IoT-Vorrichtungen bereitstellen, die einen Ethernet-Anschluss aufweisen. Der Ethernet-Port **13114** kann auch der IoT-Vorrichtung Strom bereitstellen, beispielsweise gemäß der Strom-über-Ethernet-Spezifikation, die in IEEE 802.3at Typ 1 für 12,95 Watt (W) oder IEEE 802.3at Typ 2 für 25,5 W dargelegt ist.

[0669] Die Netzsteckervorrichtung **13100** kann eine Schnittstelle **13116** für einen Serieller-Peripherieschnittstellen- bzw. Serial-Peripheral-Interface(SPI)-Bus, einen I2C-Bus oder einen I3C-Bus oder eine Kombination davon aufweisen. Durch diese Schnittstelle **13116** kann die Netzsteckervorrichtung direkt über Kommunikationsschnittstellen, die häufig an eingebetteten Vorrichtungen verfügbar sind, mit einer IoT-Vorrichtung verbunden werden. Die Schnittstelle **13116** kann durch einen oder mehrere Ports **13118**, die auf der Außenseite der Netzsteckervorrichtung bereitgestellt sind, mit einer IoT-Vorrichtung koppeln. Falls mehr als einer dieser Busse in der Netzsteckervorrichtung **13100** vorhanden ist, können Ports **13118** für jeden der verschiedenen Busse bereitgestellt werden.

[0670] Die Netzsteckervorrichtung **13100** kann eine HID-Transportschnittstelle **13120** aufweisen, um es der Netzsteckervorrichtung **13100** zu ermöglichen, sich mit bestimmten Klassen von Sensoren **820** zu verbinden. Die Sensoren **820**, die von der HID-Transportschnittstelle **13120** unterstützt werden können, umfassen beispielsweise unter anderem einen Umgebungstemperatursensor, einen Feuchtigkeitssensor, einen Näherungssensor oder einen Präsenzsensoren. Die HID-Transportschnittstelle **13120** kann durch einen als SensorHID bezeichneten HID-Klassentreiber im Betriebssystem unterstützt werden. SensorHID ist ein Funktionsblock, der den Plug-n-Play-Anschluss eines Sensors an die Netzsteckervorrichtung **13100** über die HID-Transportschnittstelle **13120** sowie von Sensoren, die durch die USB-Schnittstelle **13110** oder den SPI-, I2C- oder I3C-Bus **13116** angeschlossen werden können, unterstützt.

[0671] Die mit der Netzsteckervorrichtung **13110** gekoppelten Sensoren **820** können Sensoren für die Überwachung der Netzsteckervorrichtung **13110** selbst sowie Sensoren **820** für die Überwachung der Umgebung in der Art der Temperatur und Feuchtigkeit und Sensoren **8204**, welche die IoT-Vorrichtung überwachen, einschließen. Die Sensoren **820** zur Überwachung der IoT-Vorrichtung können Stromverbrauchssensoren, Magnetometer und dergleichen einschließen. Diese Sensoren können verwendet werden, um zu erkennen, ob behandelte IoT-Vorrichtungen senden.

[0672] Die Netzsteckervorrichtung **13100** kann eine JTAG-Schnittstelle **13124** zum Bereitstellen eines Zugriffs auf die IoT-Vorrichtung auf der Chipebene aufweisen, um zu ermöglichen, dass sie aus der Ferne debugged wird. Die JTAG-Schnittstelle **13124** kann durch einen Port **13126** an der Netzsteckervorrichtung **13100** mit der IoT-Vorrichtung koppeln. Die JTAG-Schnittstelle **13124** kann auch eine Stromversorgungs-Rücksetzfunktion für die IoT-Vorrichtung ermöglichen. Ferner kann die JTAG-Schnittstelle **13124** verwendet werden, um den nichtflüchtigen Speicher in IoT-Vorrichtungen aus der Ferne zu flashen.

[0673] Die Sendeempfänger **810** und **814** in der Netzsteckervorrichtung **13100** können verwendet werden, um unter Verwendung ihrer Kurzstrecken-Funkteinrichtungen mit nahe gelegenen IoT-Vorrichtungen zu kommunizieren oder um festzustellen, ob Vorrichtungen mit bestimmten Hardwaresignaturen noch Signale senden. Es kann jedoch eine physikalische Verbindung zur behandelten Vorrichtung erforderlich sein, damit die Netzsteckervorrichtung **13100** die Stromzufuhr für die behandelte IoT-Vorrichtung zyklisch unterbricht, falls die IoT-Vorrichtung nicht funktioniert.

[0674] Die Netzsteckervorrichtung **13100** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) **13128** aufweisen, das beispielsweise mit der von der Trusted Computing Group als ISO/IEC **11889 2009** herausgegebenen Spezifikation kompatibel ist. Das TPM **13128** kann einen kryptographischen Prozessor (CP) **13130**, einen nichtflüchtigen Speicher (NVM) **13132** und einen sicheren Speicher (SM) **13134** aufweisen. Der CP **13130** kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM **13132** kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM **13134** kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datensegment, das im Massenspeicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code.

[0675] Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execute Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. An Stelle des TPM **13128** kann die TEE auf anderen Technologien in der Art von EPID, SGX oder ähnlichen Technologien beruhen. Die TEE kann verwendet werden, um die IoT-Vorrichtung vor Angriffen von einer beschädigten Netzsteckervorrichtung **1310** zu schützen.

[0676] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Weil IoT-Vorrichtungen häufig sowohl in Bezug auf die Rechenleistung als auch auf andere Parameter beschränkt sind, können andere Vorrichtungen auch in einer einfacheren Schaltungsanordnung implementiert werden. Beispielsweise kann ein TPM als getrennter Sicherheitscoprozessor ein wichtigerer Prozessor sein als jene, die auf beschränkte Vorrichtungen für IoT abzielen. Daher ist es wahrscheinlich, dass ein stärker eingeschränktes „TPM“ verwendet wird als durch die TCG-TPM-Spezifikationen spezifiziert wird. Ferner können vertrauenswürdige Rechenelemente, beispielsweise für TPM-Operationen, für eine verbesserte Markteinführungszeit gegenüber einer ASIC-Lösung in einem FPGA implementiert werden.

[0677] Die hier beschriebene Koalitionsgruppen-Bildungslogik kann durch die TRE-Architektur in **13340** repräsentiert werden und einen Teil der TPM-Funktionalität, wie hier definiert, implementieren. Wiederum kann die Logik in einem FPGA, ASIC oder einem hybriden Ansatz instanziiert werden, um die Zeit bis zur Markteinführung, die Flexibilität und die Kosten zu optimieren.

[0678] Der Massenspeicher **808** kann ein einfaches Ein-/Ausgabesystem und ein Betriebssystem (OS/BIOS) **13122** aufweisen. Dieses weist das Betriebssystem (OS) der Netzsteckervorrichtung **13100** auf. Das OS/BIOS **13122** kann ein Echtzeit-OS (RTOS), ein erweitertes BIOS, ein Linux-OS, ein Windows-Kern oder ein HAL sein. Das OS kann abhängig von den Fähigkeiten des Prozessors, des flüchtigen Speichers und des permanenten Speichers ein vereinfachtes System oder ein vollständiges System sein.

[0679] Die FW/Treiber **13136** repräsentieren Software, die als Firmware (FW) verwendet wird, oder Treiber, welche die Netzsteckervorrichtung **13100** für den Betrieb benötigt. Die FW/Treiber **13136** können Treiber und Richtlinien zur Verbindung mit der IoT-Vorrichtung einschließen.

[0680] Der Massenspeicher **808** kann einen Software-Ablagespeicher (SW Repo) **13138** aufweisen, der als Ausführungspunkt für Über-die-Luft- bzw. Over-the-Air(OTA)-Verbesserungen an Vorrichtungen, die von der Netzsteckervorrichtung **13100** behandelt werden, wirken kann. Eine Überwachungseinrichtung **13140** kann aufgenommen sein, um die behandelten IoT-Vorrichtungen zu überwachen.

[0681] Der Massenspeicher **808** kann ein Betätigungselement **13142** zur Implementation der Aktionen, die auszuführen sind, um Operationen für die behandelten IoT-Vorrichtungen wiederherzustellen, wie unter anderem das zyklische Unterbrechen der Stromzufuhr, aufweisen. Ein Berichterstatter **13144** kann die Mitteilungsfunktion ausführen, welche über den Status der behandelten Vorrichtungen berichtet.

[0682] **Fig. 132** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **13200**, das Code aufweist, um einen Prozessor **902** anzuweisen, die Verfügbarkeit einer IoT-Vorrichtung zu erhöhen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **13200** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **13200** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0683] Das nichtflüchtige maschinenlesbare Medium **13200** kann einen Code **13202** aufweisen, um den Prozessor **902** anzuweisen, IoT-Vorrichtungen, die behandelt werden können, zu entdecken. Der Code **13202** kann den Prozessor **902** anweisen, eine Liste **13204** von IoT-Vorrichtungen zu bilden. Ein Code **13206** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Netzsteckervorrichtung zu initialisieren. Ein Code **13208** kann aufgenommen sein, um den Prozessor **902** anzuweisen, behandelte IoT-Vorrichtungen zu überwachen. Ein Code **13210** kann aufgenommen sein, um den Prozessor **902** anzuweisen, über IoT-Vorrichtungen zu berichten, beispielsweise wenn ein Fehler aufgetreten ist. Ein Code **13212** kann aufgenommen sein, um den Prozessor **902** anzuweisen, IoT-Vorrichtungen zu betätigen, um Operationen wiederherzustellen.

[0684] Zusätzlich zum Gewährleisten der Verfügbarkeit von IoT-Vorrichtungen werden Techniken zur Behandlung von Ausfällen von IoT-Vorrichtungen vorgestellt. Diese Techniken können das Warnen anderer IoT-Vorrichtungen über den Ausfall, beispielsweise durch die Verwendung von Blockchains, wie hier beschrieben, einschließen. Die IoT-Vorrichtungen, die über den Ausfall gewarnt werden, können eine IoT-Vorrichtung einschließen, die der ausgefallenen Vorrichtung ähnlich genug ist, um die Funktionalität dieser Vorrichtung zu übernehmen.

[0685] Fig. 133 ist ein schematisches Diagramm eines Ausfallsicherungsmechanismus 13300 für eine ausgefallene Vorrichtung 13302 gemäß einigen Ausführungsformen. Die ausgefallene Vorrichtung 13302 kann eine vertrauenswürdige Zuverlässigkeitsmaschine bzw. Trusted Reliability Engine (TRE) 13304 aufweisen, die eine unabhängige Stromversorgung 13306 aufweist. Die TRE 13304 kann eine Blockchain-Logik 13308 in Hardware unter anderem in der Art von ASIC, FPGA oder EC implementieren.

[0686] Eine Host-Umgebung 13310 kann einen Watchdog-Agenten (WA) 13312 aufweisen, der Watchdog-Nachrichten 13314 erzeugt, welche der TRE 13304 über die Funktionstüchtigkeit und den Betrieb der Host-Umgebung 13310 berichten. Die Host-Umgebung 13310 kann auf von der Hardware der TRE 13304 getrennter Host-Hardware 13316 laufen.

[0687] Die TRE kann ein MESH-Netz sein, das beispielsweise mehrere Instanzen von 13304 aufweist, die zusammenarbeiten, um eine allerletzte Ausfallsicherungsfunktion auszuführen, wenn erwartete Watchdog-Berichte nicht mehr vom lokalen Host eingehen. Ein Fehlen einer Watchdog-Nachricht 13314 kann ein Hinweis darauf sein, dass die Host-Umgebung 13310 gestorben ist oder auf andere Weise funktionsunfähig geworden ist. Ein Aspekt dieses Punkts besteht darin, dass eine Ausfallsicherungsnachricht erhalten wird, die ausgeliefert wurde, bevor der Knoten ausgegangen ist. Die TRE 13304 ist mit einer geringen Menge an Reserveenergie ausgelegt, welche beispielsweise ausreicht, um die Ausfallsicherungsfunktionen mit einer Peer-TRE auszuführen.

[0688] Der WA 13312 kann unabhängig Watchdog-Nachrichten 13314 zu einer Blockchain übermitteln, wo Blockchain-Beobachter das Muster empfangener Watchdog-Ereignisse analysieren können, um Schlussfolgerungen über die Funktionstüchtigkeit des Hosts zu ziehen. Intermittierende Verluste können ein Hinweis auf mögliche Ausfälle in der Host-Umgebung 13310 oder einer Netzumgebung sein. Diese können Funktionstüchtigkeitsbedingungen sein, die proaktiv korrigiert werden können, können jedoch keine Ausfallsicherungsaktionen auslösen.

[0689] Die Watchdog-Nachrichten 13314 können durch Blockchain-Transaktionen 13318 von der Blockchain-Logik 13308 in eine Blockchain 13320 geschrieben werden. Durch Schreiben der Watchdog-Nachrichten 13314 in die Blockchain 13320 können sie über IoT-Vorrichtungen, beispielsweise in einem Mesh- oder Fog-Netz, synchronisieren.

[0690] Einige der anderen IoT-Vorrichtungen im Mesh-Netz können eine ähnliche Funktionalität wie die ausgefallene Vorrichtung besitzen und freie Zyklen aufweisen, wodurch sie als Ausfallsicherungsziel wirken können. Beispielsweise kann eine Ausfallsicherungsvorrichtung 13322 oder eine Reparatur-/Ersatzdrohne 13324 die funktionelle Kompatibilität mit der ausgefallenen Vorrichtung 13302 unter Verwendung zusammengesetzter Objektidentitäten beurteilen, wie beispielsweise mit Bezug auf die Fig. 5 bis Fig. 9 definiert. Bei diesen Beispielen kann die Blockchain 13320 eine Geschichte ähnlicher Objekttypen, die als solche authentifiziert sein können, aufweisen.

[0691] Wenn eine Ausfallsicherungsbedingung existiert, können IoT-Vorrichtungen mit ähnlichen Objekttypen in der Art der Ausfallsicherungsvorrichtung 13322 durch periodisches Registrieren ihrer Kandidatur mit den TRE-Datensätzen, beispielsweise durch eine Transaktion 13326 zur Blockchain 13320, konkurrieren, um die Zielvorrichtung zu werden. Die TRE 13304 kann eine Liste möglicher anhand der Blockchain 13320 erhaltener 13328 Ausfallsicherungskandidaten aufrechterhalten, wenn sie periodische Registrierungen empfängt.

[0692] Wenn von der TRE 13304 ein Ausfall beobachtet wird, beispielsweise der Verlust von Watchdog-Nachrichten 13314 vom Watchdog-Agenten 13312 in der Host-Umgebung 13310, kann eine Ausfallsicherungsaktion angewendet werden. Zunächst kann die TRE 13304 eine lokale Strategie 13330 ausführen, um den Host wiederherzustellen. Dies kann unter der Annahme geschehen, dass die TRE 13304 durch das Ausfallereignis nicht beschädigt wurde. Die lokale Strategie 13330, die von der TRE 13304 verfolgt wird, kann das Wiederherstellen eines Host-Ersatzabbilds 13332 zur Host-Umgebung 13310 umfassen.

[0693] Eine TRE **13304** an einem geeigneten Ausfallsicherungsziel in der Art der Ausfallsicherungsanordnung **13322** kann die Watchdog-Aktivität in der Blockchain **13320** beobachten **13334** und das Nichtvorhandensein von dieser bemerken. Falls die lokale Strategie **13330** nicht erfolgreich ist, beispielsweise falls sie nicht innerhalb eines Zeitfensters verwirklicht wird, kann ein geeigneter Ausfallsicherungs-Peer in der Art der Ausfallsicherungsanordnung **13322** die Rolle der ausgefallenen Vorrichtung **13302** annehmen **13336**. Dies kann durch Übergeben einer Transaktion an die Blockchain **13320**, welche die Rechte als Ausfallsicherungsziel beansprucht, erreicht werden. Die Synchronisation der Blockchain **13320** zwischen IoT-Vorrichtungen gewährleistet, dass ein erster Beansprucher ausgewählt wird und nicht mit einem zweiten konkurriert.

[0694] Wenngleich die Ausfallsicherungsanordnung **13322** die ausgefallene Vorrichtung **13302** vorübergehend übernehmen kann, kann eine permanente Lösung erhalten werden. Eine Reparatur- oder Ersatzdrohne **13324** kann ausgesendet werden **13338**, um die ausgefallene Vorrichtung **13302** zu reparieren oder zu ersetzen. Die Reparatur- oder Ersatzdrohne **13324** kann sich automatisch auslösen, beispielsweise durch Überwachen der Blockchain **13320**, um festzustellen, dass eine Vorrichtung ausgefallen ist. Eine Ersatzdrohne kann ein direkter Ersatz sein, der durch eine Reparaturdrohne oder einen Servicetechniker an seinen Einsatzort bewegt wird. Bei einigen Beispielen kann die Ersatzdrohne eine autonome Einheit sein, die sich selbst an ihren Einsatzort bewegt. Sobald die Reparatur- oder Ersatzdrohne **13324** an ihrem Einsatzort ist, kann sie Funktionalität für die ausgefallene Vorrichtung **13302** übernehmen **13340**, so dass die Ausfallsicherungsanordnung **13322** normale Operationen wiederaufnehmen kann. An diesem Punkt kann die TRE **13304** in der ausgefallenen Vorrichtung **13302** die ausgefallene Vorrichtung **13302** außer Betrieb nehmen **13342**. Beobachter der Aktivität in der Blockchain **13320** können Ausfälle überwachen und eine Strategie zur Reparatur, zum Entfernen oder zum Ersetzen der ausgefallenen Vorrichtung **13302** planen.

[0695] Fig. **134** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **13400** zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine bzw. Trusted Reliability Engine (TRE) gemäß einigen Ausführungsformen. Das Verfahren **13400** aus Fig. **134** kann durch die mit Bezug auf Fig. **135** beschriebene IoT-Vorrichtung **13500** implementiert werden. Die TRE kann eine autonome Strategie implementieren, indem zuerst unter Verwendung der TRE auf einen Host-Ausfall überwacht wird, während auch eine Blockchain in Bezug auf den Funktionstüchtigkeitszustand der Vorrichtung informiert wird. Die erste autonome Strategie kann ein Ersatzabbild verwenden, um den beschädigten oder ausgefallenen Host wiederherzustellen, beispielsweise durch Ersetzen eines beschädigten Abbilds in einer ausgefallenen Vorrichtung. Eine zweite Strategie kann eine Ausfallsicherungsanordnung erkennen und die Vorrichtungsarbeitslast von der ausgefallenen Vorrichtung zur Ausfallsicherungsanordnung übertragen. Eine dritte Strategie kann eine Ersatzvorrichtung unter Verwendung einer automatisierten Versandvorrichtung in der Art einer Ersatz- oder Reparaturdrohne versenden. Eine vierte Strategie setzt die ausgefallene Vorrichtung außer Betrieb, um die Wahrscheinlichkeit unbekannter Verhaltensweisen zu verringern und das Risiko zu verringern, Ausfälle in umgebenden Netzvorrichtungen hervorzurufen. Die TRE kann auch Vertrauenswürdige-Ausführungsumgebungs- bzw. Trusted-Execution-Environment(TEE)-Funktionen, einschließlich des Speicherns und der Behandlung von Schlüssel-, Attestierungs- und kryptographischen Operationen, ausführen. Das Verfahren **13400** beginnt in Block **13402**, wenn die IoT-Vorrichtung, welche die TRE aufweist, eingeschaltet wird.

[0696] In Block **13404** überwacht die TRE die Host-Umgebung. Hierbei können unter anderem Operationen und die Funktionalität des Speichers, des Busses oder der CPU überwacht werden. Ferner überwacht die TRE die Host-Umgebung auf Watchdog-Nachrichten oder Pings, die bestätigen, dass die Host-Umgebung betriebsfähig ist. Beispielsweise weist die IoT-/Vorrichtungsattestierungsmessung die von den Watchdog(WD)-Pings erzeugte Herzschlagberichterstattung auf. Dies kann einen historischen Datensatz mehrerer Herzschläge oder des letzten mitgeteilten Herzschlags einschließen. Falls über einen ausgewählten Zeitraum, beispielsweise 1 Millisekunde (ms), 5 ms, 15 ms oder länger, keine Pings empfangen werden, kann die TRE feststellen, dass es einen Ausfall der Host-Umgebung gab.

[0697] In Block **13406** erzeugt die TRE eine WD-Pings aufweisende WD-Nachricht. Der TRE-Attestierungsschlüssel kann verwendet werden, um die WD-Nachricht ansprechend auf eine Attestierungsanforderung oder eine Anforderung zum Signieren der WD-Nachricht zu signieren. In Block **13408** kann die WD-Nachricht zu einer Überwachungs-Entität gesendet werden, wobei die WD-Nachricht beispielsweise als Transaktion zu einer Blockchain übermittelt wird. Die WD-Nachrichtenerzeugungslogik kann innerhalb der TRE geschützt bleiben, wodurch eine höhere Gewährleistung und Widerstandsfähigkeit gegen einen Einfluss durch Host-Ausfälle bereitgestellt wird. Knoten, welche die WD-Nachrichten in der Blockchain überwachen, können die Blockchain-Aktualisierungen über eine Vielzahl von Subnetzen, Vorrichtungen und Netzen beobachten.

[0698] In Block **13410** kann ein Ausfall der IoT-Vorrichtung lokal, beispielsweise durch die TRE, erkannt werden. Falls in Block **13410** kein lokaler Ausfall erkannt wird, kann eine ferne Vorrichtung in Block **13412** einen Ausfall erkennen. Falls bei **13412** keine Fernerkennung eines Ausfalls geschieht, kehrt die Überwachung in Block **13414** zu Block **13404** zurück.

[0699] Falls in Block **13412** ein ferner Ausfall erkannt wird, wird in Block **13416** eine Prozessausfallnachricht zur TRE in der lokalen Vorrichtung gesendet, beispielsweise durch die ferne Vorrichtung, die den Ausfall erkannt hat. Falls in Block **13410** eine Prozessausfallnachricht empfangen oder ein lokaler Ausfall erkannt wird, wird in Block **13418** eine Ausfallverarbeitung eingeleitet.

[0700] In Block **13420** wird festgestellt, ob der Host lokal wiederherstellbar ist. Dies kann beispielsweise dadurch festgestellt werden, dass bemerkt wird, dass der Host noch eingeschaltet ist und sich lediglich an einem bestimmten Codesegment aufgehängt haben kann. Falls dies der Fall ist, kann in Block **13422** ein Host-Ersatzabbild installiert werden, wobei beispielsweise der aktuelle Arbeitsspeicher der ausgefallenen Vorrichtung überschrieben wird. Die TRE kann dann einen Neustart der Host-Vorrichtung im Code des Host-Ersatzabbilds versuchen. Die TRE kann einen anfänglichen Neustart der Host-Umgebung vor der Installation des Host-Ersatzabbilds versuchen. Dies kann Zeit sparen, wenn der Ausfall nicht auf eine Beschädigung des Betriebscodes zurückzuführen ist, sondern beispielsweise auf einen Software-Crash oder Hänger zurückzuführen ist.

[0701] Falls die Host-Vorrichtung nicht lokal wiederherstellbar ist, wird in Block **13424** von der TRE festgestellt, dass eine Ausfallsicherungs Vorrichtung nahe ist. Falls eine Ausfallsicherungs Vorrichtung nahe ist, wird die Ausfallsicherungs Vorrichtung in Block **13426** dafür ausgelegt, mit der Ausführung der Host-Funktionen zu beginnen.

[0702] Falls in Block **13424** eine Ausfallsicherungs Vorrichtung nicht nahe ist, wird in Block **13428** festgestellt, ob ein Host ersetzbar oder reparierbar ist. Falls dies der Fall ist, kann in Block **13430** eine Ersatzvorrichtung oder eine Reparaturdrohne abgesendet werden, um die Reparatur oder den Ersatz der ausgefallenen Vorrichtung vorzunehmen. Selbst wenn in Block **13426** eine Ausfallsicherungs Vorrichtung identifiziert wurde und die Funktionen der ausgefallenen Vorrichtung übernommen hat, kann in Block **13430** noch eine Reparatur- oder Ersatzdrohne abgesendet werden, um zu ermöglichen, dass die Ausfallsicherungs Vorrichtung zu normalen Operationen zurückkehrt.

[0703] In Block **13432** wird festgestellt, ob der Ausfall gelöst wurde, beispielsweise ob die Funktionen der ausgefallenen Vorrichtung ausgeführt werden. Falls dies der Fall ist, endet das Verfahren **13400** in Block **13436**, beispielsweise durch Zurückkehren zu normalen Überwachungsoperationen in Block **13404**. Falls die ausgefallene Vorrichtung in Block **13432** nicht zu normalen Operationen zurückgekehrt ist, wird sie in Block **13434** außer Betrieb gesetzt. Die TRE in der ausgefallenen Vorrichtung kann in einen Schlafzustand versetzt werden. Bei diesem Beispiel hat die Ausfallsicherungs Vorrichtung oder die Ersatzvorrichtung die Funktion der ausgefallenen Vorrichtung übernommen und stellt weiter die Dienste der ausgefallenen Vorrichtung bereit. Das Verfahren **13400** endet dann in Block **13436**.

[0704] Bei einem Szenario, bei dem ein Host-Ausfall bösartig ist, kann das schädigende Ereignis nicht von normalen Anomalien oder von einem unerwarteten Verhalten unterscheidbar sein. Die TRE-Umgebung kann die Sicherheit einer Endpunktvorrichtung verbessern und die Wahrscheinlichkeit erhöhen, dass ein Angreifer nicht in der Lage ist, das Absetzen einer WD-„SOS“-Nachricht zu verhindern. Ferner kann ein Angreifer in der Fähigkeit begrenzt werden, Auditierungsverlaubbeweise zu verbergen, die während des normalen Verlaufs einer Sicherheitsauditierung gesammelt worden sein können.

[0705] Fig. 135 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **13500** zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3, Fig. 8 und Fig. 133 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **13500** andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0706] Die vertrauenswürdige Zuverlässigkeitsmaschine bzw. Trusted Reliability Engine (TRE) **13304** kann eine vertrauenswürdige Ausführungsumgebung bzw. Trust Execute Environment (TEE) bereitstellen, die Zuverlässigkeitslogik und Isolation enthält, beispielsweise durch ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) implementiert. Dementsprechend kann die TRE **13304** eine Anzahl funktioneller Einheiten aufweisen, die vor dem allgemeinen Zugriff geschützt sind. Diese funktionellen Einheiten können

andere funktionelle Einheiten in der IoT-Vorrichtung **13500** duplizieren. Diese können die TRE-Logik **13308**, das Host-Ersatzabbild **13332** und die Blockchain **13320**, wie hier erörtert, aufweisen. Zusätzlich kann die TRE **13304** einen Mikroprozessor **13502** und eine unabhängige Stromversorgung **13504**, einen TRE-Kommunikator **13506** und einen Speicher **13508** aufweisen. Die Stromversorgung **13504** kann mit der Stromversorgung vom Stromversorgungsblock **828** gekoppelt sein oder eine unabhängige Stromversorgung aufweisen, beispielsweise eine mit einem Ladegerät verbundene Batterie.

[0707] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation des Ausfallsicherungsmechanismus unter Verwendung der hier beschriebenen vertrauenswürdigen Zuverlässigkeitsmaschine aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0708] Der Massenspeicher **808** des Hosts kann einen Watchdog(WD)-Agenten **13312** aufweisen, der WD-Nachrichten über den Bus **806** zur TRE **13304** sendet. Wie hier beschrieben, kann die TRE **13304** eine Watchdog-Nachricht erzeugen und die Watchdog-Nachricht zur Blockchain **13320** übertragen. Die TRE-Logik **13308** kann die Blockchain **13320** über eine zusätzliche Kommunikationsstrecken, beispielsweise unter anderem über einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** übertragen. Die TRE **13304** kann durch den TRE-Kommunikator **13506**, der die Sendeempfänger **810** oder **814** oder die Netzschnittstellen-Steuereinrichtung **816** nach Bedarf mit Strom versorgen kann, auf die Kommunikationsstrecken zugreifen. Dies kann gewährleisten, dass die TRE **13304** Kommunikationen mit externen Vorrichtungen aufrechterhält, selbst wenn das Host-System in der IoT-Vorrichtung **13500** ausgefallen ist.

[0709] Gemäß einigen Aspekten ist nicht die gesamte Funktionalität des Systems in der TRE **13304** enthalten. Zusätzlich zum Watchdog-Agenten **13312** kann der Speicher **808** der IoT-Vorrichtung **13500** eine Anzahl anderer Blöcke, die dem System Funktionalität bereitstellen, enthalten. Beispielsweise kann der Massenspeicher **808** eine Host-Blockchain-Logik **13510** zum Unterhalten einer Host-Blockchain **13512** außerhalb der TRE **13304** aufweisen. Die Host-Blockchain **13512** kann alle Transaktionen in der Blockchain **13320** in der TRE **13304** aufweisen und einen umfangreicheren Satz von Transaktionen aufweisen. Beispielsweise kann die Blockchain im Massenspeicher **808** Identitätsblöcke, Peer-Vorrichtungsblöcke und andere Blöcke, die infolge von Speicherbeschränkungen nicht in der Blockchain **13320** in der TRE **13304** vorhanden sind, aufweisen.

[0710] Der Massenspeicher **808** der IoT-Vorrichtung **13500** kann einen Abbilderzeuger **13512** zum Kopieren eines Host-Abbilds **13514** und zum Senden von ihm zur TRE **13304** über den Bus **806**, um es als Host-Ersatzabbild **13332** zu speichern, aufweisen. Das Host-Abbild **13514** kann das Betriebssystem, Treiber und Funktionscode für die Host-Umgebung der IoT-Vorrichtung **13500** aufweisen.

[0711] Der Massenspeicher **808** kann einen Kommunikator **13518** aufweisen, der Pakete oder Frames von Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** akzeptiert und Pakete oder Frames zu anderen Mesh-Vorrichtungen **812**, Vorrichtungen in der Cloud **302** und dergleichen sendet. Wie mit Bezug auf **Fig. 45** beschrieben, kann der Kommunikator **13518** andere Funktionen ausführen, wie die Übersetzung von Paketen zwischen Protokollen, das Akzeptieren von Mikrozahlen und dergleichen.

[0712] **Fig. 136** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **13600**, das Code aufweist, um einen Prozessor **902** anzuweisen, einen Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine zu implementieren, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über den Bus **904** auf das nichtflüchtige maschinenlesbare Medium **13600** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **13600** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0713] Das nichtflüchtige maschinenlesbare Medium **13600** kann einen Code **13602** aufweisen, um den Prozessor **902** anzuweisen, die Host-Umgebung auf Herzschlagnachrichten oder Pings zu überwachen. Ein Code **13604** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Watchdog-Nachrichten, beispielsweise einschließlich der Herzschlagnachrichten, zu erzeugen. Ein Code **13606** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Watchdog-Nachrichten, beispielsweise als Transaktion, zu einer Blockchain zu übermitteln. Ein Code **13608** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Ausfälle in einer lokalen Vorrichtung in Zusammenhang mit der TRE zu erkennen. Ein Code **13610** kann aufgenommen sein,

um den Prozessor **902** anzuweisen, Ausfälle in einer fernen Vorrichtung, beispielsweise durch Untersuchen der Watchdog-Nachrichten in einer Blockchain, zu erkennen.

[0714] Ein Code **13612** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Host-Ersatzabbild an Stelle des gegenwärtig in einer Host-Umgebung gespeicherten zu installieren. Ein Code **13614** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Ausfallsicherungsvorrichtung zu konfigurieren. Ein Code **13616** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Reparatur- oder Ersatzdrohne auszusenden. Ein Code **13618** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine ausgefallene Vorrichtung außer Betrieb zu setzen.

[0715] Sicherheit ist bei IoT-Netzen eine Sorge, insbesondere wenn die Größe der Netze zunimmt. Die Speicherung privater Schlüssel, Aktualisierungen und ein Abfang im Verlauf, eine Erkennung bössartiger Schlüssel und eine schnelle neue Schlüsselerzeugung sind mögliche Befürchtungen. In vielen Fällen sind IoT-Vorrichtungen jedoch durch Speicher-, Verarbeitungsleistungs- und andere Probleme in der Art begrenzter Komponenten beschränkt. Ferner haben IoT-Netze eine begrenzte Bandbreite zur Teilung von Daten und aller anderen Funktionen. Demgemäß ist es nützlich, die Effizienz von Kommunikationen zwischen den Vorrichtungen zu maximieren.

[0716] Bei den hier beschriebenen Techniken brauchen IoT-Knoten in einem Netz möglicherweise nicht einen vollständigen privaten Schlüssel zu empfangen oder abzusenden, beispielsweise mit jeder Nachricht. Stattdessen können Sie fraktionelle Teile des Schlüssels absenden und empfangen. Zusätzlich zur Verbesserung der Effizienz von Kommunikationen kann hierdurch die Angriffsfläche für ein sicheres IoT-Netz verringert werden, weil kein individueller Knoten die vollständigen Schlüsselsequenzen im permanenten Speicher zu speichern braucht.

[0717] Fig. 137 ist ein schematisches Diagramm der Konstruktion eines Schlüssels **13702** unter Verwendung fraktioneller Schlüssel **13704** und **13706**, die zwischen Knoten in einem IoT-Netz ausgetauscht werden, gemäß einigen Ausführungsformen. Bei diesem Beispiel kann ein Wasserfüllansatz für die Konstruktion des Schlüssels **13702** unter Verwendung der fraktionellen Schlüssel **13704** und **13706** verwendet werden. Der Schlüssel **13702** kann in einem zirkulären Puffer **13708** zusammengesetzt werden. Jeder fraktionelle Schlüssel **13704** oder **13706** kann einen Offset **13710** aufweisen, der angibt, wo der Teil des Schlüssels **13712** in jedem fraktionellen Schlüssel **13704** oder **13706** in den zirkulären Puffer **13708** einzufügen ist. Der Schlüssel **13702** kann verwendet werden, um auf Dienste für das IoT-Netz zuzugreifen, mit anderen IoT-Netzen zu kommunizieren und dergleichen.

[0718] Wenngleich bei diesem Beispiel zwei fraktionelle Schlüssel **13704** und **13706** dargestellt sind, können mehrere fraktionelle Schlüssel verschiedener Größen im zirkulären Puffer gespeichert werden. Ein vollständiger Schlüssel kann identifiziert werden, wenn genügend fraktionelle Schlüssel hinzugefügt wurden, um den zirkulären Puffer zu füllen. Dieser Ansatz kann zu überlappenden Schlüsselindizes führen, was eine Teilschlüsselverifikation ermöglicht, weil Bytes überlappender fraktioneller Schlüssel identisch sein sollten. Ebenso ermöglicht dies die Erkennung bössartiger Vorrichtungen, bevor vollständige Schlüsselsequenzen konstruiert wurden. Falls jegliche Bytes überlappender fraktioneller Schlüssel nicht übereinstimmen, kann eine Warnung zu anderen Vorrichtungen im Mesh oder zu anderen Benutzern gesendet werden, welche angibt, dass eine Vorrichtung beschädigt worden sein kann.

[0719] Im Allgemeinen speichert gemäß einigen Aspekten keine einzelne Vorrichtung im IoT-Netz den vollständigen Schlüssel. Dementsprechend kann keine einzelne Vorrichtung durch die Verwendung eines Mikroskops zur Bestimmung des vollständigen Schlüssels angegriffen oder analysiert werden. Sobald der vollständige Schlüssel **13702** zusammengestellt wurde, kann er vom IoT-Netz oder der Fog-Vorrichtung verwendet werden, um auf andere Vorrichtungen, beispielsweise in der Cloud, zuzugreifen.

[0720] Fig. 138 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **13800** zur Zusammenstellung eines vollständigen Schlüssels aus in einzelnen Knoten in einem IoT-Netz gespeicherten fraktionellen Schlüsseln gemäß einigen Ausführungsformen. Das Verfahren **13800** aus Fig. 138 kann durch die mit Bezug auf Fig. 140 beschriebene IoT-Vorrichtung **14000** implementiert werden. Block **13802** repräsentiert beispielsweise den Fall, in dem ein vollständiger Schlüssel von einer Fog-Vorrichtung benötigt wird, um auf das System in der Cloud zuzugreifen.

[0721] In Block **13804** wird der erste Teil eines fraktionellen Schlüssels abgesendet. Dies kann geschehen, wenn ein Knoten einen Nutzinhalt bildet und eine drahtgestützte oder drahtlose Kommunikation einleitet, um

den Nutzinhalt einschließlich des fraktionellen Schlüssels zu einem Knoten zu senden, der ihn angefordert hat. Das Absenden des fraktionellen Schlüssels kann auch als Anforderung für andere Knoten, fraktionelle Schlüssel zu Peer-Knoten zu senden, wirken.

[0722] In Block **13806** empfängt der anfordernde Knoten einen Teil des fraktionellen Schlüssels von einem sendenden Knoten. In Block **13808** analysiert der anfordernde Knoten den Nutzinhalt, um festzustellen, ob er einen fraktionellen Schlüssel und einen Offset aufweist. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **13806** zurück.

[0723] Falls in Block **13808** festgestellt wird, dass ein Nutzinhalt einen fraktionellen Schlüssel aufweist, kann der anfordernde Knoten dann in Block **13810** den fraktionellen Schlüssel gegenprüfen, um festzustellen, ob der empfangene fraktionelle Schlüssel andere Teile überlappt. Dies kann auf eine Anzahl von Arten geschehen, einschließlich beispielsweise des Vornehmens eines Vergleichs des Pufferindex. Ferner kann der fraktionelle Schlüssel im zirkulären Puffer gespeichert werden, und falls irgendwelche Teile andere Schlüssel überlappen, können sie verglichen werden, um zu bestätigen, dass die überlappenden Teile übereinstimmen. Jede Nicht-übereinstimmung überlappender Teile kann darauf hinweisen, dass die Vorrichtung beschädigt wurde. Falls dies der Fall ist, kann der Zusammensetzungsprozess unterbrochen werden und kann eine Warnung ausgesendet werden.

[0724] Weitere Sicherheit kann durch andere Techniken bereitgestellt werden. Beispielsweise kann ein „schmutziges Bit“ für jede „Zelle“ im zirkulären Schlüsselpuffer gehalten werden, das für die Verwendung durch einen fraktionellen Schlüssel zugeordnet werden kann. Eine Sicherheitsschwäche kann herbeigeführt werden, wenn eine zuvor verwendete Zelle als Element einer nachfolgenden Schlüsselfraktion ausgewählt wird. Um diese mögliche Schwäche zu korrigieren, kann das schmutzige Bit bei der ersten Zuordnung festgelegt werden und bei der nachfolgenden Überlappungsüberprüfung geprüft werden. Falls eine Überlappungsüberprüfung das schmutzige Bit offenbart, wird die Offset-Berechnung für den zirkulären Puffer wiederholt, um festzustellen, ob dies zu einer nicht schmutzigen Zelle führt. Der Prozess wird wiederholt, bis genug jungfräuliches Schlüsselmaterial für das Schlüsselerzeugungsverfahren gefunden wird.

[0725] In Block **13812** kann festgestellt werden, ob alle fraktionellen Schlüssel empfangen wurden. Falls dies nicht der Fall ist, kann der Prozessfluss zu Block **13806** zurückkehren. Falls alle fraktionellen Schlüssel empfangen wurden, kann in Block **13814** der vollständige Schlüssel gebildet werden.

[0726] Das Verfahren **13800** endet in Block **13816**. Dies kann beispielsweise stattfinden, wenn der vollständige Schlüssel im Auftrag einer Fog-Vorrichtung einer anderen Vorrichtung bereitgestellt wird.

[0727] Fig. 139 ist ein schematisches Diagramm der Zusammenstellung eines vollständigen Schlüssels **13902** aus von fünf Knoten A - E bereitgestellten fraktionellen Schlüsseln gemäß einigen Ausführungsformen. Bei diesem Beispiel tauschen die fünf Knoten A - E ihre fraktionellen Schlüssel miteinander aus. Jeder Knoten A - E kann den vollständigen Schlüssel durch Anordnen der empfangenen Schlüssel am festgelegten Offset in einem zirkulären Puffer bilden. Der Offset kann durch $\{N : x, 0 : y\}$ bezeichnet werden, wobei x die Anzahl der Bytes N im fraktionellen Schlüssel ist und y der Anfangsindex oder Offset O des fraktionellen Schlüssels im vollständigen Schlüssel **13902** ist.

[0728] Falls sich beispielsweise ein zirkularer Puffer **13904** im Knoten A befindet, kann sich der fraktionelle Schlüssel A **13906** vom Knoten A bereits im zirkulären Puffer **13904** befinden. Der fraktionelle Schlüssel B **13908** kann dann vom Knoten B empfangen werden. Bei diesem Beispiel überlappt das erste Byte des fraktionellen Schlüssels B **13908** das letzte Byte des fraktionellen Schlüssels A **13906**, und ein Bytevergleich **13910** kann vorgenommen werden, um zu gewährleisten, dass das überlappende Byte zwischen den beiden fraktionellen Schlüsseln **13906** und **13908** übereinstimmt. Falls der Bytevergleich **13910** feststellt, dass das überlappende Byte zwischen den beiden fraktionellen Schlüsseln **13906** und **13908** übereinstimmt, kann der fraktionelle Schlüssel vom Knoten B in den zirkulären Puffer **13904** geladen werden.

[0729] Knoten A kann dann den fraktionellen Schlüssel C **13912** von Knoten C empfangen. Weil der fraktionelle Schlüssel C **13912** keinen der vorhergehenden fraktionellen Schlüssel **13906** und **13908** überlappt, kann er ohne Bytevergleich in den Puffer geladen werden. Der fraktionelle Schlüssel C **13912** kann einen Offset und eine Länge haben, welche das Ende des zirkulären Puffers **13904** überlappt, so dass dementsprechend das letzte Byte der fraktionellen Schlüssel C **13912** rotiert werden kann, so dass es an den Anfang des zirkulären Puffers **13904** fällt, wie durch den Pfeil **13914** angegeben ist.

[0730] Knoten A kann dann den fraktionellen Schlüssel D **13916** von Knoten D empfangen. Weil das letzte Byte des fraktionellen Schlüssels D **13916** das erste Byte des fraktionellen Schlüssels C **13912** überlappt, kann ein Bytevergleich **13918** ausgeführt werden, um zu gewährleisten, dass die beiden Bytes übereinstimmen. Sobald dies bestätigt wurde, kann der fraktionelle Schlüssel D **13916** in den zirkulären Puffer **13904** geladen werden.

[0731] Knoten A kann dann den fraktionellen Schlüssel E **13920** von Knoten E empfangen. Weil es eine erhebliche Byteüberlappung zwischen den fraktionellen Schlüsseln D und E **13916** und **13920** gibt, kann ein Bytevergleich **13922** an jedem dieser Bytes ausgeführt werden, um zu gewährleisten, dass sie übereinstimmen. Falls dies der Fall ist, kann dann der fraktionelle Schlüssel E **13920** des Knotens E in den zirkulären Puffer **13904** geladen werden, um den vollständigen Schlüssel **13902** zu bilden.

[0732] Wenn Überlappungen auftreten, geschieht eine Byteverifizierung, um zu bestätigen, dass die überlappenden fraktionellen Teile übereinstimmen. Falls dies nicht der Fall ist, kann der Prozess beendet werden und kann die Möglichkeit eines beschädigten Knotens mitgeteilt werden. Die überlappenden Bytes können auch in Fällen Redundanz bereitstellen, in denen ein oder mehrere Knoten nicht in der Lage sein können, ihre fraktionellen Schlüssel mit anderen Knoten im Netz auszutauschen. Diese Situation kann andernfalls dazu führen, dass alle Knoten nicht in der Lage sind, den vollständigen Schlüssel **13902** zu bilden, falls alle fraktionellen Schlüssel orthogonal sind, beispielsweise keine Byteüberlappungen hatten.

[0733] Fig. 140 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **14000** zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz **812** zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **14000** andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0734] Der Massenspeicher **800** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0735] Der Massenspeicher **808** kann einen Kommunikator **14002** aufweisen, der Pakete über eine oder mehrere Kommunikationsstrecken, unter anderem beispielsweise durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** sendet und von diesen empfängt. Zusätzlich zu den mit Bezug auf Fig. 140 beschriebenen Funktionen, wie mit Bezug auf Fig. 45 beschrieben, kann der Kommunikator **14004** andere Funktionen ausführen, wie eine Übersetzung von Paketen zwischen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **14004** Teil eines mit Bezug auf Fig. 33 beschriebenen Dienstbarkeitssystems sein.

[0736] Ein Fraktioneller-Schlüssel-Generator **14002** kann einen fraktionellen Schlüssel beispielsweise von einem Zufallszahlengenerator, einer Blockchain oder von einem während der Herstellung in der Vorrichtung gespeicherten Schlüssel erzeugen. Beispielsweise kann der Schlüssel unter Verwendung eines digitalen Zufallszahlengenerators bzw. Digital Random Number Generator (DRNG) oder eines Pseudozufallszahlengenerators bzw. Pseudo-Random Number Generator (PRNG), der unter Verwendung eines DRNG geseedet wird, von Intel erzeugt werden. Der Fraktioneller-Schlüssel-Generator **14002** kann eine beliebige Anzahl anderer Techniken verwenden, um den fraktionellen Schlüssel zu erzeugen, beispielsweise durch Zugreifen auf einen Schlüssel von einer Blockchain, wie hier beschrieben.

[0737] Ein anderes beispielhaftes Verfahren zur Erzeugung fraktioneller Schlüssel kann einen DRNG verwenden, der einen zufälligen Seed akzeptiert, der beispielsweise vom DRNG erhalten wird, wenn er nicht im PRNG-Modus ist, wobei der Suchraum über den zirkulären Puffer im Wesentlichen unbegrenzt sein kann, wie durch die DRNG-Wortgrößenarchitektur festgelegt. Bei diesem Beispiel wird der Offset im zirkulären Puffer als Seed für den DRNG-in-PRNG-Modus von Intel genommen. Daher kann der zirkuläre Puffer im Wesentlichen von unendlicher Größe sein, wodurch gewährleistet wird, dass Kollisionen innerhalb des Puffers probabilistisch unmöglich sind.

[0738] Der Kommunikator **14004** kann Frames bilden, die fraktionelle Schlüssel im Nutzinhalt des Frames aufweisen. Bei einigen Beispielen kann ein Frame, der einen fraktionellen Schlüssel aufweist, von einer anderen IoT-Vorrichtung in den Mesh-Vorrichtungen **812** in der Art einer ferneren Vorrichtung übergeben werden.

Bei diesem Beispiel kann die IoT-Vorrichtung **14000** von anderen IoT-Vorrichtungen in den Mesh-Vorrichtungen **812** empfangene fraktionelle Schlüssel zusammenstellen, um einen endgültigen Schlüssel zu bilden.

[0739] Ein Bytevergleich **14006** kann aufgenommen sein, um überlappende Bytes von verschiedenen Vorrichtungen empfangener fraktioneller Schlüssel zu vergleichen, um zu gewährleisten, dass die überlappenden Bytes identisch sind. Der Bytevergleich **14006** kann den Prozess des Zusammenstellens eines endgültigen Schlüssels unterbrechen, falls irgendwelche der überlappenden Bytes nicht übereinstimmen, weil dies darauf hinweisen kann, dass eine IoT-Vorrichtung beschädigt wurde.

[0740] Ein Schlüsselzusammensteller **14008** kann die jeweiligen fraktionellen Schlüssel in einem zirkulären Puffer **14010** zusammenstellen, um den endgültigen Schlüssel zu bilden. Der Schlüsselbetreiber **14012** kann den endgültigen Schlüssel in einer Operation in der Art des Bereitstellens des Schlüssels für einen Gateway, um die Identität einer Mesh- oder Fog-Vorrichtung **812** zu bestätigen, verwenden.

[0741] **Fig. 141** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **14100**, das Code aufweist, um einen Prozessor **902** anzuweisen, fraktionelle Schlüssel zu empfangen, die fraktionellen Schlüssel zu einem endgültigen Schlüssel zusammenzustellen und den endgültigen Schlüssel zu verwenden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **14100** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **14100** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0742] Das nichtflüchtige maschinenlesbare Medium **14100** kann einen Code **14102** aufweisen, um den Prozessor **902** anzuweisen, einen fraktionellen Schlüssel zu einer Empfangsvorrichtung zu senden. Ein Code **14104** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen fraktionellen Schlüssel zu empfangen und diesen zu speichern. Ein Code **14106** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Bytevergleiche für überlappende Bytes auszuführen, beispielsweise um zu gewährleisten, dass die überlappenden Bytes übereinstimmen, bevor ein endgültiger Schlüssel zusammengestellt wird. Ein Code **14108** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den fraktionellen Schlüssel in den zirkulären Puffer zu schreiben und den endgültigen Schlüssel im zirkulären Puffer anhand der von Vorrichtungen empfangenen fraktionellen Schlüssel zusammenzustellen. Ein Code **14110** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den endgültigen Schlüssel zu verwenden, beispielsweise um im Auftrag der Vorrichtung oder Vorrichtungen im IoT-Netz auf eine Vorrichtung in der Cloud zuzugreifen.

[0743] Eine monetäre Befürchtung in Bezug auf die Sicherheit eines schlüsselbasierten Ansatzes für Kryptowährungen wird durch das Auftreten digitaler Wallets und anonymer schlüsselbasierter Identitäten in einem Blockchain-Zusammenhang hervorgerufen. Ein digitales Wallet ist ein System, das es einer Person ermöglicht, eine elektronische Zahlung für eine Transaktion vorzunehmen. Das digitale Wallet kann mit einem Bankkonto verknüpft sein oder einen von einem anderen Konto übertragenen Guthabenbetrag speichern. Bei einigen Beispielen kann das digitale Wallet in Software in einer elektronischen Vorrichtung in der Art eines Smartphones, einschließlich Kommunikationen, einer Verschlüsselung und anderer Systeme zur Implementation der Funktionalität, implementiert sein. Bei anderen Beispielen kann das digitale Wallet als RFID-Tag implementiert sein, wobei die Systeme auf einem zentralen Server existieren, auf die von einem Kommunikationssystem zugegriffen wird.

[0744] Eine Transaktion auf einer Blockchain wird durch die privaten Schlüssel des Eigentümers des digitalen Wallets signiert, und der Verlust oder die Exposition dieser privaten Schlüssel ermöglicht es einem Angreifer, das digitale Wallet leer zu räumen. Dies ist ein Prozess, durch den jeder nicht verbrauchte Guthabenbetrag der von diesem digitalen Wallet besessenen Währung an einen anderen Eigentümer, der beispielsweise zum Angreifer gehört, übertragen wird.

[0745] Generell haben Blockchain-Konsensmechanismen kein Verfahren, um eine solche Transaktion als betrügerisch zu identifizieren. Das Durchsuchen der Blockchain nach der Tatsache kann den Weg identifizieren, den die Währung genommen hat, die unregulierte Natur solcher Technologien bedeutet jedoch, dass die praktisch verfügbaren Verfahren zur Umkehrung der Transaktion ausgeschlossen sind und nicht skalieren. Dies kann schwieriger gemacht werden, weil die Identitäten der beteiligten Parteien ohne eine tiefere Untersuchung nicht bekannt sind. Ferner wird es problematisch, nachfolgende Transaktionen derselben Münzen zu dritten Parteien rückgängig zu machen. Dementsprechend kann es bevorzugt sein, schon von Anfang an die Situa-

tion zu verhindern und zu versuchen, die Exposition von Akteuren in einer Blockchain durch Einführen des Konzepts der bedarfsgetriebenen Schlüsselerzeugung zu verringern.

[0746] Fig. 142 ist ein schematisches Diagramm einer Prozedur **14200** zur Erzeugung von Schlüsseln nach Bedarf für Vorrichtungen auf verlustbehafteten Netzen gemäß einigen Ausführungsformen. Wie hier beschrieben, kann die bedarfsgetriebene Schlüsselerzeugung ermöglichen, dass digitale Wallets neue Schlüssel für Transaktionen unter Verwendung einer der hier beschriebenen Techniken zur Schlüsselerzeugung nach Bedarf statt regelmäßig zeitlich geplant erzeugen. Nach Bedarf würde der Ausführung einer neuen Schlüsselerzeugung für jede Transaktion und seiner nur einmaligen Verwendung gleichkommen. Der gleiche Mechanismus könnte auf den Systemzugriff und andere leere Anwendungen schlüsselbasierter Technologien angewendet werden.

[0747] Die Prozedur kann in Block **14202** eingeleitet werden, wenn eine Transaktion an ein Netz übergeben wird. Dies kann beispielsweise geschehen, wenn ein Kauf erfolgt und ein digitales Wallet verwendet wird, um für den Kauf zu bezahlen. Der Kauf kann online oder in einem Einzelhandelsgeschäft ausgeführt werden, beispielsweise wenn eine Vorrichtung, die ein digitales Wallet aufweist, an der Kommunikationsstelle angezapft wird.

[0748] In Block **14204** kann ein neuer Schlüssel erzeugt werden. Dies kann durch die in Block **14206** dargestellte Prozedur erfolgen, welche sich auf die Standard-Bitcoin-Beispiele beziehen kann. Ferner können andere hier erörterte Prozeduren verwendet werden. Bei dieser Prozedur kann ein privater Schlüssel nach dem Wallet-Importformat (WIF) verwendet werden, um einen privaten 256-Bit-Schlüssel **14210** zu importieren. Der private 256-Bit-Schlüssel **14210** kann zur Erzeugung eines öffentlichen 512-Bit-Schlüssels **14212** verwendet werden, der verwendet werden kann, um einen Hash **14214** des öffentlichen 160-Bit-Schlüssels zu erzeugen, dem die Wallet-Adresse **14216** zugeordnet werden kann. In Block **14218** kann der alte Schlüssel gelöscht werden. Die Erzeugung des neuen Schlüssels ist nicht auf die in Block **14206** dargestellte Prozedur beschränkt. Beispielsweise kann ein neuer Schlüssel durch die Verwendung der mit Bezug auf Fig. 143 beschriebenen Prozedur erzeugt werden.

[0749] Fig. 143 ist ein schematisches Diagramm eines Schlüsselerzeugungsverfahrens **14300**, dass im vorstehend beschriebenen bedarfsweisen Prozess für die Schlüsselerzeugung sowie für die Erzeugung von Schlüsseln in anderen Zusammenhängen verwendet werden kann, gemäß einigen Ausführungsformen. Das Verfahren **14300** aus Fig. 143 kann durch die mit Bezug auf Fig. 145 beschriebene IoT-Vorrichtung **14500** implementiert werden. Die schnelle Schlüsselerzeugung in verlustbehafteten Netzen hoher Latenz bleibt infolge der häufig falschen Annahmen, dass ein IoT-Netz eine Ende-zu-Ende-Verbindbarkeit, ständig sichere Verbindungen, eine zentralisierte Schlüsselstelle und einen zentralisierten Ausgebungsagenten und kostengünstige Kommunikationen sowie Netzinteraktionen zur Unterstützung von Schlüsselaustauschen aufweist, eine herausfordernde Aufgabe. Das Verfahren **14300** zur lokalen Schlüsselerzeugung kann verwendet werden, wenn befehlende Knoten Offset-Werte aussenden und vollständige oder teilweise Schlüssel nicht erforderlich sind. Ein vollständiger fraktioneller Schlüssel **14302** kann mit dem beispielsweise vom Verkäufer bereitgestellten lokalen Schlüssel **14304** verwendet werden. Der lokale Schlüssel **14304** kann in einem zirkulären Puffer gespeichert werden, und ein neuer Schlüssel kann durch eine zirkuläre Exklusiv-ODER(XOR)-Operation **14306** des vollständigen fraktionellen Schlüssels **14302** und des lokalen Schlüssels **14304** erzeugt werden.

[0750] Der neue Schlüssel **14308** kann dann nach Bedarf für den Zugriff verwendet werden. Ein Schlüssel-Offset kann verwendet werden, um mehrere neue Schlüssel zu erzeugen, indem der Offset zwischen dem vollständigen fraktionellen Schlüssel **14302** und dem lokalen Schlüssel **14304** geändert wird. Bei diesem Beispiel kann ein ferner Steuerknoten nur einen Offset-Wert zur Erzeugung des neuen Schlüssels senden.

[0751] Fig. 144 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **14400** zur Erzeugung von Schlüsseln gemäß einigen Ausführungsformen. Das Verfahren **14400** aus Fig. 144 kann durch die mit Bezug auf Fig. 145 beschriebene IoT-Vorrichtung **14500** implementiert werden. Im Allgemeinen ist die Schlüsselbehandlung verhältnismäßig statisch. Schlüssel werden, sobald sie erzeugt wurden, verwendet, bis eine Beeinträchtigungssituation erkannt wurde, ein gelegentliches Wiederauffrischen erforderlich ist und dergleichen. In IoT-Netzen können eine Störung und das Fehlen einer Ende-zu-Ende-Verbindbarkeit jedoch übliche Geschehnisse sein. Dementsprechend können eine Schlüsselwiederauffrischung und ein sicheres Senden von Schlüsseln zu einem großen Netz von Vorrichtungen Herausforderungen darstellen. Die hier beschriebenen Techniken können ein ständiges Ändern von Schlüsseln ohne direkten menschlichen Eingriff ermöglichen. Das Verfahren **14400** kann in Block **14402** beginnen, beispielsweise wenn ein Betriebssystem feststellt, dass es Zeit ist, einen Schlüssel zu ändern, oder wenn eine Anforderung zur Änderung des Schlüssels empfangen wird.

[0752] In Block **14404** wird festgestellt, ob ein Schlüssel-Offset-Wert empfangen wurde. Falls dies nicht der Fall ist, kann in Block **14406** ein Offset-Wert für den Schlüssel in einer IoT-Vorrichtung erzeugt werden. In Block **14408** kann ein fraktioneller Schlüssel von der IoT-Vorrichtung empfangen werden. Dies kann beispielsweise nicht erforderlich sein, falls ein fraktioneller Schlüssel bereits von der IoT-Vorrichtung empfangen wurde. Der fraktionelle Schlüssel kann zusammen mit anderen von anderen IoT-Vorrichtungen empfangenen fraktionellen Schlüsseln verwendet werden, um einen vollständigen fraktionellen Schlüssel zu erzeugen, beispielsweise wie mit Bezug auf die **Fig. 137** bis **Fig. 141** beschrieben wurde.

[0753] In Block **14410** kann ein neuer Schlüssel erzeugt werden, beispielsweise wie mit Bezug auf die **Fig. 140** oder **Fig. 143** beschrieben. In Block **14412** kann der neue Schlüssel überprüft werden. Die Überprüfung kann durch Entschlüsseln einer Standardnachricht von einem anderen Knoten erfolgen.

[0754] In Block **14414** kann festgestellt werden, ob der Schlüssel abgelaufen ist. Falls dies der Fall ist, kann das Verfahren **14400** zu Block **14404** zurückkehren, um einen neuen Schlüssel zu erzeugen.

[0755] Falls der Schlüssel in Block **14414** nicht abgelaufen ist, kann in Block **14416** die Verschlüsselung oder Entschlüsselung einer Datendatei stattfinden. In Block **14418** endet das Verfahren **14400** beispielsweise mit der Übertragung einer verschlüsselten Datei oder der Verwendung einer entschlüsselten Datei.

[0756] Beim Verfahren können Offset-Werte für die internen zirkulären Schlüsselgeneratoren zu Knoten übertragen werden. Ferner können, wenngleich fraktionelle Schlüssel zu Knoten übertragen werden können, die Knoten ihre eigenen Schlüssel erzeugen, wodurch der Bedarf verringert wird, neue Schlüssel zu Knoten zu senden. Die Schlüsselneuerzeugung kann auf einer regelmäßigen zeitlich geplanten Basis geschehen.

[0757] **Fig. 145** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **14500** zur Erzeugung von Schlüsseln nach Bedarf vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **14500** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0758] Der Massenspeicher **800** kann eine Anzahl von Modulen zur Implementation des hier beschriebenen Schlüsselerzeugungsprozesses aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0759] Der Massenspeicher **808** kann einen Kommunikator **14502** aufweisen, der Pakete über eine oder mehrere Kommunikationsstrecken, unter anderem beispielsweise durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** sendet und von diesen empfängt. Zusätzlich zu den mit Bezug auf **Fig. 145** beschriebenen Funktionen, wie mit Bezug auf **Fig. 45** beschrieben, kann der Kommunikator **14504** andere Funktionen ausführen, wie eine Übersetzung von Paketen zwischen Protokollen, eine Ausführung von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **14504** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems sein.

[0760] Ein Transaktor **14504** kann eine Transaktion an ein Netz übergeben, beispielsweise um einen Gegenstand zu kaufen oder zu mieten, beispielsweise von einer Vorrichtung in der Cloud **302** oder der Fog **812**. Der Transaktor **14504** kann einen zuvor erzeugten Schlüssel verwenden und die Erzeugung eines neuen Schlüssels nach Abschluss der Transaktion auslösen. Bei einem anderen Beispiel kann der Transaktor **14504** einen neuen Schlüssel erzeugen, um die Transaktion an das Netz zu übergeben.

[0761] Bei anderen Beispielen kann der Transaktor **14504** einen Schlüssel für einen bestimmten Zeitraum verwenden. Ein Schlüssellebensdauer-Zeitgeber **14506** kann den Zeitraum steuern, während dessen der Schlüssel verwendet werden kann, bevor ein neuer Schlüssel erzeugt wird. Beispielsweise kann der Schlüssellebensdauer-Zeitgeber **14506** ermöglichen, dass ein Schlüssel eine Minute, 5 Minuten, 30 Minuten, eine Stunde oder länger dauert.

[0762] Ein Schlüsselgenerator **14508** kann einen neuen Schlüssel erzeugen, beispielsweise unter Verwendung eines zirkulären Puffers **14510** zur Ausführung einer Exklusiv-ODER-Operation eines vollständigen frak-

tionellen Schlüssels **14302** mit dem lokalen Schlüssel **14304**, wie mit Bezug auf **Fig. 143** beschrieben. Der vollständige fraktionelle Schlüssel **14302** kann aus von anderen IoT-Vorrichtungen empfangenen fraktionellen Schlüsseln zusammengestellt werden, wie weiter mit Bezug auf die **Fig. 137** bis **Fig. 141** beschrieben. Beispielsweise kann der Kommunikator **14502** Frames empfangen, die fraktionelle Schlüssel in ihrem Nutzinhalt aufweisen. Bei diesem Beispiel kann die IoT-Vorrichtung **14000** von anderen IoT-Vorrichtungen in den Mesh-Vorrichtungen **812** empfangene fraktionelle Schlüssel zusammenstellen, um den vollständigen fraktionellen Schlüssel **14302** zu bilden.

[0763] **Fig. 146** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **14600**, das einen Code aufweist, um einen Prozessor **902** anzuweisen, Schlüssel nach Bedarf zu erzeugen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **14600** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **14600** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0764] Das nichtflüchtige maschinenlesbare Medium **14600** kann einen Code **14602** aufweisen, um den Prozessor **902** anzuweisen, einen fraktionellen Schlüssel von einer sendenden Vorrichtung zu empfangen. Der Code **14602** kann einen vollständigen fraktionellen Schlüssel aus einer Anzahl von verschiedenen sendenden Vorrichtungen empfangener fraktioneller Schlüssel zusammenstellen. Ein Code **14604** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Offset-Wert für die Erzeugung eines Schlüssels anhand des vollständigen fraktionellen Schlüssels und eines in der Vorrichtung gespeicherten Schlüssels zu empfangen. Ein Code **14606** kann aufgenommen sein, um eine logische Operation mit dem vollständigen fraktionellen Schlüssel und dem Vorrichtungsschlüssel für die Erzeugung eines neuen Schlüssels auszuführen, beispielsweise unter Verwendung des Offset-Werts. Ein Code **14608** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen neuen Schlüssel unter Verwendung anderer Techniken zu erzeugen, beispielsweise durch Zugreifen auf eine Blockchain, um einen neuen Schlüssel zu erhalten, durch zufälliges Erzeugen eines neuen Schlüssels oder unter Verwendung einer Entropiemultiplexiertechnik, wie mit Bezug auf die **Fig. 147** bis **Fig. 151** beschrieben. Ein Code **14610** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Schlüssel auslaufen zu lassen, beispielsweise wenn ein Zeitgeber einen bestimmten Wert erreicht. Ein Code **14612** kann aufgenommen sein, um den Prozessor anzuweisen, unter Verwendung des Schlüssels Daten zu verschlüsseln oder zu entschlüsseln.

[0765] In einigen Situationen kann eine verteilte Zusammenarbeit durch Fehler bei der Signalisierung und Synchronisation zwischen Knoten kompliziert gemacht werden. Beispielsweise kann eine Peer-IoT-Vorrichtung schlafen oder kann die Netzverbindbarkeit nicht zuverlässig sein. In diesem Fall können zusammenarbeitende Peers ein Entropiemultiplexierkonzept verwenden, um einen temporären symmetrischen Schlüssel für die Verschlüsselung, Nachrichtenintegritätscodes und dergleichen zu vereinbaren.

[0766] **Fig. 147** ist ein schematisches Diagramm eines Entropiemultiplexierprozesses **14700** zur Erzeugung einer Anzahl von Seeds, die für die Erzeugung neuer Schlüssel verwendet werden können, gemäß einigen Ausführungsformen. Der Entropiemultiplexierprozess **14700** bildet einen Seed-Baum **14702** aus Seed-Werten, die zum Seeden eines Zufallszahlengenerators verwendet werden. Der Aufbau des Seed-Baums **14702** kann mit einem Zusammenhangsattribut in der Art der Zeit, des Orts, der Nähe oder einer anderen Attributklasse, die unter Verwendung eines taxonomischen oder ontologischen Zerlegungsverfahrens beschrieben werden kann, korreliert werden. Bei diesem Beispiel beruht der Entropiemultiplexierprozess **14700** zumindest teilweise auf der Zeit.

[0767] Der Seed-Baum kann auch einen PRNG verwenden, der als zirkularer Puffer unendlicher Größe betrachtet werden kann, wie mit Bezug auf **Fig. 140** beschrieben. Der Baumzusammenhang legt die Offsets im Puffer auf der Grundlage einer wiederholbaren Vereinbarung für die Baumkonstruktion fest.

[0768] Die zusammenarbeitenden Knoten können eine Zeit-Root **14704** auswählen und einen ersten Seed-Wert **14706** erzeugen. Der erste Seed-Wert **14706** kann als Anfangspunkt in einer Ontologie verwendet werden, um den Seed-Baum **14702** zu erzeugen. Eine erste untere Ebene von Seeds **14708** kann beispielsweise unter Verwendung eines Jahreswerts **14710** des ersten Seed-Werts **14706** erzeugt werden. Ein Monatswert **14712** kann dann beispielsweise verwendet werden, um eine zweite niedrigere Ebene von Seeds **14714** zu erzeugen. Ein Tageswert **14716** kann dann beispielsweise verwendet werden, um eine dritte Ebene von Seeds **14718** zu erzeugen. Weitere Ebenen im Seed-Baum **14702** können unter Verwendung schrittweise feinerer Inkremente, wie Minuten oder sogar Sekunden, erzeugt werden.

[0769] Die zusammenarbeitenden Knoten können über den ersten Seed-Wert **14706** und den Anfangspunkt in einer Ontologie übereinstimmen. Die zusammenarbeitenden Knoten können dann getrennt eine individuelle Kopie des Seed-Baums **14702** erzeugen und speichern. Wenn ein geteiltes Geheimnis benötigt wird, das sich beispielsweise auf den ontologischen Zusammenhang bezieht, können die zusammenarbeitenden Knoten unabhängig diesen Zusammenhang verwenden, um nach der lokalen Kopie des Seed-Baums **14702** zu suchen und dadurch das gemeinsame Geheimnis zu lokalisieren. Dieses kann dann verwendet werden, um einen symmetrischen Schlüssel für die Verschlüsselung von Kommunikationen und Daten zwischen den zusammenarbeitenden Knoten zu erzeugen.

[0770] Eine beliebige Anzahl anderer ontologischer Parameter kann verwendet werden, um einen Seed-Baum zu erzeugen, einschließlich beispielsweise von Ortsinformationen in der Art von Adressinformationen, GPS-Koordinaten, IP-Adresse und dergleichen.

[0771] **Fig. 148** ist ein schematisches Diagramm, das einen Prozess **14800** zur Erzeugung eines Orts-Seed-Baums **14802** zeigt, gemäß einigen Ausführungsformen. Ebenso wie bei der Erzeugung des Seed-Baums **14702**, wie mit Bezug auf **Fig. 147** erörtert, kann der Orts-Seed-Baum **14802** unabhängig durch eine Anzahl zusammenarbeitender Knoten erzeugt werden, sobald eine Orts-Root **14804**, ein anfänglicher Seed **14808** und eine Baumontologie vereinbart wurden. Beispielsweise kann ein Adress-Seed-Baum **14810** anhand des anfänglichen Seeds **14808** erzeugt werden, indem zuerst ein Seed **14812** anhand eines Kontinents des Orts **14814** erzeugt wird. Eine niedrigere Ebene von Seeds kann dann anhand Länderangaben **14816** erzeugt werden. Eine noch niedrigere Ebene von Seeds kann dann anhand einer Städteangabe **14818** erzeugt werden. Weitere Ebenen können anhand Straßenangaben oder Adressgenerationen erzeugt werden, falls dies erforderlich ist.

[0772] Andere Typen des Orts-Seed-Baums **14802** können anhand anderer Ortsparameter erzeugt werden. Beispielsweise kann eine GPS-Koordinate **14820** zur Erzeugung eines Sehnen- und Seed-Baums **14822** im Koordinaten-Seed-Baum **14822** verwendet werden und können Seeds niedrigerer Ebene unter anderem anhand einer Breitenangabe **14824**, einer Längenangabe **14826** oder einer Höhenangabe **14828** erzeugt werden. Andere Typen des Orts-Seed-Baums **14802** können anhand einer IP-Adressangabe **14830** erzeugt werden. Teilabschnitte der IP-Adresse **14832** können zur Erzeugung von Seeds niedrigerer Ebene verwendet werden.

[0773] Mehrere Zusammenhänge können durch Kombinieren mehrerer Werte unter Verwendung einer Pseudozufallsfunktion (PRF) in der Art von HMAC kombiniert werden, um ein zusammengesetztes geteiltes Geheimnis zu erzeugen. Dies kann das Kombinieren anhand Zeitangaben erzeugter Seeds mit anhand Ortsangaben erzeugten Seeds einschließen.

[0774] **Fig. 149** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **14900** zur Erzeugung von Seeds unter Verwendung einer Entropiemultiplexierung und zur Verwendung dieser Seeds, um Schlüssel für verschlüsselte Kommunikationen zu erzeugen, gemäß einigen Ausführungsformen. Das Verfahren **14900** aus **Fig. 149** kann durch die mit Bezug auf **Fig. 150** beschriebene IoT-Vorrichtung **15000** implementiert werden. Block **14902** repräsentiert beispielsweise den Fall, in dem sich eine IoT-Vorrichtung einem Netz anschließt und einen gemeinsamen Schlüssel für verschlüsselte Kommunikationen benötigt.

[0775] In Block **14904** werden Zusammenhangsattribute, die den IoT-Vorrichtungen gemeinsam sind, identifiziert. Die Zusammenhangsattribute können beispielsweise Zeit, Ort, Aktivität, Interesse und dergleichen einschließen. In Block **14906** kann jedes der Zusammenhangsattribute zerlegt werden, um einen Satz von Subattributen zu bilden. Die Subattribute können zur Erzeugung eines Seed-Baums für die Zusammenhangsattribute verwendet werden. In Block **14908** kann ein zufälliger Seed-Wert für die Root jedes Seed-Baums erzeugt werden.

[0776] In Block **14910** kann festgestellt werden, ob der Seed für jede Root verwendet wird, um vor physischen Bedrohungen in der Art von Diebstahl oder Verlust zu schützen. Falls dies der Fall ist, wird der Prozessfluss in Block **14912** fortgesetzt. In Block **14912** kann eine Teilung eines kryptographischen Geheimnisses verwendet werden, um den Root-Seed in M von N Teilen zu zerlegen. In Block **14914** werden die M Teile über N Vorrichtungen bereitgestellt. In Block **14916** werden die Vorrichtungen physisch verteilt, beispielsweise während der Implementation des Netzes. Falls in Block **14910** ein verteilter Root-Seed nicht benötigt wird, um vor physischen Bedrohungen zu schützen, kann der Seed in Block **14918** jeder Teilnehmervorrichtung bereitgestellt werden.

[0777] Sobald die Blöcke **14902** bis **14918** abgeschlossen sind, können die IoT-Vorrichtungen in einem Netz gemeinsame Geheimnisse erzeugen, um symmetrische Schlüssel für die Verschlüsselung von Daten und Kommunikationen zu erzeugen. In Block **14920** kann festgestellt werden, ob der Root-Seed verteilt ist. Falls dies der Fall ist, kann in Block **14922** ein Netz verwendet werden, um jeden Teil des Root-Seeds von den N Vorrichtungen zu erhalten. Dies kann unter Verwendung eines persönlichen Netzes, das eine QR-Codeanzeige aufweist, und eines Lesers zum Erhalten jedes Teils geschehen.

[0778] In Block **14924** kann der Root-Seed verwendet werden, um zufällige Werte für jeden Knoten in einem Seed-Baum zu erzeugen. Dies kann für jedes Zusammenhangsattribut und jede hierarchische Zerlegung geschehen.

[0779] In Block **14926** wird festgestellt, ob ein Zusammenhangsattribut wahr ist. Dieses identifiziert, welcher Seed-Baum verwendet werden sollte, um einen kryptographischen Schlüssel zu erzeugen, falls überhaupt. In Block **14928** entspricht der Seed dem für die Erzeugung eines kryptographischen Schlüssels verwendeten Zusammenhangsattribut.

[0780] Falls in Block **14926** kein Zusammenhangsattribut wahr ist, wird in Block **14930** festgestellt, ob ein zirkularer fraktioneller Schlüssel unterstützt wird. Falls dies der Fall ist, wird in Block **14932** ein fraktioneller kryptographischer Schlüssel erzeugt oder anhand von anderen IoT-Vorrichtungen im Netz übermittelter fraktioneller Schlüssel zusammengestellt.

[0781] In Block **14934** wird der kryptographische Schlüssel verwendet, um Daten zu schützen. Beispielsweise können von einer ersten IoT-Vorrichtung zu einer anderen IoT-Vorrichtung zu sendende Daten verschlüsselt werden, bevor sie gesendet werden. Ähnlich kann der kryptographische Schlüssel verwendet werden, um von der anderen IoT-Vorrichtung gesendete Daten zu entschlüsseln.

[0782] Der Prozess endet in Block **14936**, sobald die Daten entschlüsselt oder verschlüsselt wurden. Falls in Block **14930** festgestellt wird, dass kein zirkularer fraktioneller Schlüssel unterstützt wird, endet der Prozess auch in Block **14936**.

[0783] Fig. 150 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **15000** zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz **812** zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **15000** andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0784] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Code dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0785] Der Massenspeicher **808** kann einen Zusammenhangsidentifizierer **15002** zum Feststellen eines Zusammenhangs für die Erzeugung des Seed-Baums aufweisen. Wie hier beschrieben, kann der Zusammenhang beispielsweise auf Zeit, Ort, IP-Adresse oder einer Anzahl anderer Parameter beruhen.

[0786] Ein Seed-Baum-Generator **15004** kann einen Seed-Baum für den Zusammenhang erzeugen. Hierbei kann der Zusammenhang in Teile zerlegt werden, wobei die Zeit beispielsweise in Jahr, Monat, Tag, Minute und dergleichen zerlegt wird. Der Seed-Baum-Generator **15004** kann Seeds auf verschiedenen hierarchischen Ebenen durch Auswählen von Zeitinkrementen dieses Typs um den zerlegten Wert erzeugen, beispielsweise durch Festlegen von Seeds für deine Werte minus eins oder minus zwei und dergleichen anhand des Jahr-Werts in der Zeit.

[0787] Ein Seed-Generator **15006** kann dann verwendet werden, um einen Root-Seed und einen Seed-Wert für einen Knoten im hierarchischen Seed-Baum zu erzeugen. Der Seed-Wert kann eine unter Verwendung der zerlegten Ebenen des Zusammenhangs für diesen Knoten erzeugte Zufallszahl sein.

[0788] Ein Kommunikator **15008** kann aufgenommen sein, um Pakete über eine oder mehrere Kommunikationsstrecken, beispielsweise unter anderem durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** zu senden

und davon zu empfangen. Die Pakete können Informationen aufweisen, die von anderen Knoten verwendet werden, um ein gemeinsames Geheimnis zu erzeugen. Beispielsweise können die Pakete den Zusammenhang, die hierarchische Ebene, den Root-Seed und dergleichen aufweisen.

[0789] Wie mit Bezug auf **Fig. 45** beschrieben, kann der Kommunikator **15008** andere Funktionen ausführen, wie die Übersetzung von Paketen zwischen Protokollen, das Ausführen von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **15008** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitssystems sein. Ein Fraktioneller-Schlüssel-Zusammensteller **15010** kann von anderen Mesh-Vorrichtungen **812** empfangene fraktionelle Schlüssel zusammenstellen, um einen Schlüssel zu bilden oder um einen Wert für einen Root-Seed wiederherzustellen.

[0790] Der Fraktioneller-Schlüssel-Zusammensteller **15010** kann die jeweiligen fraktionellen Schlüssel in einem zirkulären Puffer zusammenstellen, um den endgültigen Schlüssel zu bilden. Eine Verschlüsselungs-/Entschlüsselungseinrichtung **15012** kann den endgültigen Schlüssel in einer Operation in der Art des Verschlüssels von Daten, die zu einer anderen Mesh- oder Fog-Vorrichtung **812** zu senden sind, oder des Entschlüssels von Daten, die von einer anderen Mesh- oder Fog-Vorrichtung **812** empfangen wurden, verwenden.

[0791] **Fig. 151** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **15100**, das Code aufweist, um einen Prozessor **902** anzuweisen, eine Entropiemultiplexierung zur Erzeugung eines gemeinsamen Geheimnisses zwischen Vorrichtungen zu verwenden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **15100** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **15100** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0792] Das nichtflüchtige maschinenlesbare Medium **15100** kann einen Code **15102** aufweisen, um den Prozessor **902** anzuweisen, einen Seed-Baum für einen Zusammenhang zu erzeugen. Wie vorstehend erwähnt wurde, kann der Zusammenhang beispielsweise auf Zeit, Ort, IP-Adresse oder einer Anzahl anderer Parameter beruhen. Ein Code **15104** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Root-Seed für den Zusammenhang zu erzeugen. Ein Code **15106** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Zusammenhang anderen Vorrichtungen bereitzustellen. Ein Code **15108** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Root-Seed anderen Vorrichtungen bereitzustellen. Ein Code **15110** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Seeds für jeden Knoten oder jede Vorrichtung in einem hierarchischen Seed-Baum zu erzeugen. Ein Code **15112** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Seed für die Erzeugung eines kryptographischen Schlüssels zu verwenden. Ein Code **15114** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den kryptographischen Schlüssel zur Verschlüsselung zu anderen IoT-Vorrichtungen gesendeter Daten oder zur Entschlüsselung von anderen IoT-Vorrichtungen empfangener Daten zu verwenden.

[0793] Die hier beschriebenen Schlüsselbehandlungs- und -erzeugungsprozesse sehen eine Anzahl von Techniken zur Behandlung der Sicherheit in einer Umgebung vor, die IoT-Vorrichtungen aufweist. In einigen Fällen kann die Behandlung der Erzeugung, der Lebensdauer, der Beendigung und des Wiederausgebens von Schlüsseln in einer IoT-Netzumgebung jedoch komplex sein.

[0794] **Fig. 152** ist ein Leiterdiagramm eines beispielhaften Verfahrens **15200** zur vereinheitlichten Schlüsselbehandlung in einer IoT-Netzumgebung gemäß einigen Ausführungsformen. Das Verfahren **15200** aus **Fig. 152** kann durch die mit Bezug auf **Fig. 153** beschriebene IoT-Vorrichtung **15300** implementiert werden. Bei diesem Beispiel kann ein Dienstanbieter (SP) **15202** verwendet werden, um die Gesamt-Vertrauens-Roots bereitzustellen. Dieser Dienstanbieter **15202** kann eine von einer Gruppe von IoT-Vorrichtungen im IoT-Netz behandelte Blockchain sein. Bei einem anderen Beispiel kann der Dienstanbieter **15202** eine externe Vorrichtung sein, die dem IoT-Netz Sicherheitsdienste bereitstellt.

[0795] Ein IoT-Server **15204** kann die lokale Sicherheit für ein IoT-Netz behandeln, beispielsweise sichere Informationen in einem vom IoT-Server **15204** zugänglichen sicheren Speicherort **15206** speichern. Der sichere Speicherort **15206** kann eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execute Environment (TEE) sein, die beispielsweise von einem vertrauenswürdigen Plattformmodul bzw. Trusted Platform Module (TPM) behandelt wird.

[0796] Ein IoT-Client **15208** kann sowohl mit dem Dienstanbieter **15202** als auch mit dem IoT-Server **15204** interagieren, um Schlüssel für die Verschlüsselung und Entschlüsselung von Daten und Kommunikationen

zu erhalten. Eine andere Entität **15210** kann an den Kommunikationen teilnehmen, beispielsweise feststellen, dass ein Schlüssel beschädigt wurde, und den Rückruf der Schlüssel und die Erzeugung neuer Schlüssel auslösen. Die Entität **15210** kann unter anderem eine andere IoT-Vorrichtung im IoT-Netz sein, ein Benutzer an der Verwaltungskonsole sein oder ein Hersteller von IoT-Vorrichtungen im IoT-Netz sein.

[0797] Das Verfahren **15200** kann verwendet werden, um sowohl symmetrische Schlüssel als auch asymmetrische Schlüssel zu behandeln. Für bestimmte Kommunikationen können alle Daten unter Verwendung symmetrischer Schlüssel geschützt werden. Das Verfahren **15200** kann beginnen, wenn der IoT-Server **15204** in ein IoT-Netz aufgenommen wurde und einen Dienstanbieter-Berechtigungsnachweis **15212** empfängt. Der Dienstanbieter-Berechtigungsnachweis **15212** kann verwendet werden, um den IoT-Server **15204** in einer Authentifizierungsnachricht **15214** beim Dienstanbieter **15202** zu validieren. Die Authentifizierungsnachricht **15214** kann fordern, dass der Dienstanbieter **15202** Berechtigungsnachweise für sichere Kommunikationen bereitstellt. Der Dienstanbieter **15202** kann mit einer Vertrauensnachricht **15216** antworten, die einen Vertrauensanker **15218** aufweist. Der Vertrauensanker **15218** kann einen Hash eines öffentlichen Schlüssels oder einen zertifizierten Weg oder eine Kette zu einer vertrauenswürdigen Autoritäts-Root aufweisen.

[0798] Ein IoT-Client **15208** kann eine Symmetrische-Schlüssel-Nachricht **15220** zum Dienstanbieter **15202** senden, wodurch gefordert wird, dass symmetrische Schlüssel für Kommunikationen bereitgestellt werden. Die Symmetrische-Schlüssel-Nachricht **15220** kann durch einen öffentlichen oder privaten Schlüssel vom IoT-Client **15208** signiert werden.

[0799] Falls die Symmetrische-Schlüssel-Nachricht **15220** vom Dienstanbieter **15220** validiert wird, kann der Dienstanbieter **15202** mit einer Nachricht **15222** antworten, die einen symmetrischen Schlüssel oder ein Ticket aufweist. Die Nachricht **15222** kann vom Dienstanbieter **15202** unter Verwendung desselben vom IoT-Client **15208** bereitgestellten Schlüssels signiert werden. Der IoT-Client **15208** kann den symmetrischen Schlüssel dann in einer Nachricht **15224** dem IoT-Server **15204** bereitstellen. Der IoT-Server **15204** kann den symmetrischen Schlüssel **15226** im sicheren Speicher **15206** speichern. Der IoT-Server kann auch durch Vergleichen eines Zeitstempels mit einer sicheren Zeit **15229** im sicheren Speicher **15206** feststellen, ob der sichere Schlüssel abgelaufen ist. Das Ergebnis dieses Vergleichs kann in einem Sicherer-Schlüssel-Status **15230** gespeichert werden.

[0800] Die Entität **15210** kann feststellen, dass ein Schlüssel **15232** beschädigt wurde. Ro Beispiel kann die Entität **15210** bei Netzsuchen Daten, die durch den Schlüssel geschützt wurden, oder den Schlüssel selbst finden. Für den sicheren Schlüssel **15226** kann dies zu einer Nachricht **15234** an den Dienstanbieter **15202**, den sicheren Schlüssel **15226** zu widerrufen, führen. Ansprechend auf die Nachricht **15234** kann der Dienstanbieter **15202** eine Widerrufsnachricht **15236** zum IoT-Server senden. Eine andere Nachricht **15238** kann zum IoT-Client **15208** gesendet werden, wodurch der IoT-Client **15208** benachrichtigt wird. Der IoT-Server **15204** kann dann durch Senden einer Authentifizierungsnachricht **15214** zur Wiederholung des Prozesses mit dem Dienstanbieter **15202** reauthentifizieren.

[0801] Der IoT-Client **15208** ist nicht auf die Verwendung symmetrischer Schlüssel beschränkt, sondern kann unter Verwendung eines privaten Schlüssels eine Authentifizierungsnachricht **15240** zum Dienstanbieter **15202** senden. Der Dienstanbieter **15202** kann dann unter Verwendung eines zum IoT-Client **15208** gehörenden öffentlichen Schlüssels die Authentifizierungsnachricht **15240** entschlüsseln, wodurch die Identität des IoT-Clients **15208** bestätigt wird.

[0802] Falls die Authentifizierung der Authentifizierungsnachricht **15240** angibt, dass der IoT-Client **15208** gültig ist, kann der Dienstanbieter **15202** eine Nachricht **15242**, die ein Zertifikat aufweist, senden. Die Nachricht **15242** kann mit dem öffentlichen Schlüssel für den Dienstanbieter **15202** signiert werden. Der IoT-Client **1528** kann dann eine das Zertifikat aufweisende Nachricht **15244** zum IoT-Server **15204** senden. Die Nachricht **15244** kann mit einem öffentlichen Schlüssel für den IoT-Client **15208** signiert werden. Der öffentliche Schlüssel **15246** kann vom IoT-Server im sicheren Speicher **15206** gespeichert werden. Der IoT-Server **15204** kann auch feststellen **15248**, ob das Zertifikat abgelaufen ist, beispielsweise durch Vergleichen eines Zeitstempels im Zertifikat mit einem im sicheren Speicher **15206** gespeicherten sicheren Zeitwert **15250**. Der Status des privaten Schlüssels **15252** kann auch im sicheren Speicher **15206** gespeichert werden.

[0803] Der IoT-Server **15204** kann dann einen temporären symmetrischen Schlüssel bzw. Temporal Symmetric Key (TSK) **15254** für Kommunikationen erzeugen. Der IoT-Server **15204** kann eine Schlüsselaustauschnachricht **15208**, welche den TSK **15254** aufweist, zum IoT-Client **15208** senden. Der IoT-Client **15208**

kann dann unter Verwendung des TSK **15254** mit dem IoT-Server **15204** kommunizieren, beispielsweise um eine Nachricht **15258** zu verschlüsseln.

[0804] Falls die Entität **15210** erkennt **15232**, dass der öffentliche Schlüssel **15226** beschädigt wurde, kann sie eine Widerrufsnotice **15260** zum Dienstanbieter **15202** senden. Der Dienstanbieter **15202** kann dann eine Widerrufsnotice **15262** zum IoT-Server **15204** senden, wodurch der Widerruf des öffentlichen Schlüssels **15246** vorgeschrieben wird. Der Dienstanbieter **15202** kann auch eine Nachricht **15264** zum IoT-Client **15208** senden, wodurch ihm vorgeschrieben wird, den öffentlichen Schlüssel zu löschen, der für den zum IoT-Server **15204** gesendeten öffentlichen Schlüssel **15246** erzeugt wurde.

[0805] Der TSK **15254** hält nicht für immer und kann eine kürzere Lebensdauer als die öffentlichen Schlüssel aufweisen. Beispielsweise kann eine Nachricht **15266** durch den IoT-Client **15208** zum IoT-Server **15204** gesendet werden, nachdem sie unter Verwendung des TSK **15254** verschlüsselt wurde. Ein sicherer Zeitwert **15268** im sicheren Speicher **15206** kann vom IoT-Server **15204** verwendet werden, um festzustellen **15270**, ob der TSK **15254** abgelaufen ist. Der TSK-Status **15272** kann dann im sicheren Speicher **15206** gespeichert werden, und falls er abgelaufen ist, kann der TSK **15254** mit dem neuen Wert, der mit dem IoT-Client **15208** ausgetauscht wird **15256**, wiederaufgefrischt werden.

[0806] Falls die Entität **15210** ferner feststellt, dass der TSK **15254** beschädigt wurde, kann sie eine Widerrufsnotice **15274** zum Dienstanbieter **15202** senden. Der Dienstanbieter **15202** kann dann eine Widerrufsnotice **15276** zum IoT-Server **15204** senden, wodurch er angewiesen wird, den TSK-Status **15272** zu ungültig zu ändern. Der Dienstanbieter **15202** kann auch eine Nachricht **15278** zum IoT-Client **15208** senden, wodurch er angewiesen wird, den TSK **15254** zu löschen. An diesem Punkt kann der IoT-Server **15204** versuchen, sich durch Senden der Authentifizierungsnachricht **15214** am Dienstanbieter **15202** zu reauthentifizieren und den Prozess neu zu starten.

[0807] Der symmetrische Schlüssel **15226** kann eine Lebensdauer aufweisen, wie durch einen im sicheren Speicher **15206** gespeicherten sicheren Zeitwert **15282** festgelegt ist. Der IoT-Server **15204** kann durch Vergleichen der Verwendungszeit mit der sicheren Zeit **15250** feststellen **15284**, dass der sichere Schlüssel oder das Ticket abgelaufen ist. Der IoT-Server **15204** kann dann einen wiederaufgefrischten Schlüssel SK' ausgeben. Der wiederaufgefrischte Schlüssel SK' kann dann verwendet werden, bis die sichere Zeit **15250** überschritten wurde. Die Entität **15210** kann auch überwachen, um festzustellen, ob der Schlüssel SK' beschädigt wurde, und eine Widerrufsnotice **15234** aussenden, falls dies erforderlich ist.

[0808] Wie hier beschrieben, kann ein Schlüsselaustausch- oder Schlüsselbehandlungsprotokoll zu temporären oder vorübergehenden symmetrischen Schlüsseln führen, die zum Schützen von Daten verwendet werden, einschließlich Vertraulichkeit, Integrität oder beider. Die temporären Schlüssel setzen die vom Authentifizierungs-/Schlüsselaustauscheignis festgelegten Authentifizierungs- und Vertrauenseigenschaften auf der Grundlage einer Annahme voraus, dass die temporären Schlüssel nicht beschädigt wurden, weil sie anfänglich eingerichtet wurden.

[0809] Temporäre Schlüssel können jedoch veränderliche Lebenszeiten haben. Die Lebenszeit kann auf der Grundlage des Zusammenhangs und der Situation dynamisch angepasst werden. Beispielsweise kann ein Knoten, der in einen Schlafmodus eintritt und austritt, die Schlüssellebensdauer nicht verringern, während er schläft.

[0810] Ferner kann ein Widerruf jeglicher Schlüssel, ob symmetrisch oder asymmetrisch, durch Senden einer Widerrufsnotice sowohl zum Client als auch zum Server geschehen. Falls ein Schlüssel widerrufen wird, kann der Berechtigungsnachweis (Zertifikat oder Ticket) durch Senden einer Schlüssellöschnotice, welche die Clients und Server, die das Zertifikat oder das Ticket besitzen, anweist, sie zu löschen, gelöscht werden. Eine Löschung kann sich von einem Widerruf dadurch unterscheiden, dass der Widerruf den Clients oder Servern lediglich vorschreiben kann, die Überprüfung widerrufenen Schlüssel zu verweigern, während die Löschung vorschreiben kann, dass die Schlüssel physikalisch aus dem System entfernt werden. Sowohl Widerrufs- als auch Löschnotice können sofort wirksam werden, während das Auslaufen des Zertifikats oder des Tickets ermöglichen kann, dass der Schlüssel bis zum Auslaufdatum und anschließend an ein Schlüsselbeeinträchtigungseignis verwendet wird.

[0811] Die Schlüssellebenszyklusbehandlung gilt ferner für Cache-Systeme symmetrischer Schlüssel, bei denen ein temporärer Schlüssel für eine zweite oder eine dritte Nachricht wiederverwendet werden kann, auch wenn der Schlüssel als temporär angesehen wird. Die Temporalität gecachter Schlüssel wird durch die Ca-

che-Ablaufrichtlinie festgelegt. Daher erfüllt eine Schlüssel-Cache-Richtlinie einen doppelten Zweck als Ticketstruktur, wobei die Cache-Richtlinien-Konfigurationsnachricht unter Verwendung einer „Ticket“-Struktur spezialisiert werden kann, die keinen symmetrischen Schlüssel enthält.

[0812] Die vereinheitlichte Schlüsselbehandlung nutzt Schlüsselbehandlungsnachrichten aus und läuft sowohl für symmetrische als auch asymmetrische Schlüssel ab, wodurch Wiederverwendungseffizienzen bei der Implementation ermöglicht werden, welche beschränkte IoT-Umgebungen begünstigt.

[0813] Fig. 153 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 15300 zur Behandlung von Schlüsseln in einem Netz von IoT-Mesh-Vorrichtungen 812 vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Die IoT-Vorrichtung 15300 kann der IoT-Server 15204 oder der IoT-Client 15208 sein, wie mit Bezug auf Fig. 152 beschrieben. Es sei bemerkt, dass für die IoT-Vorrichtung 15300 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0814] Bei diesem Beispiel kann die IoT-Vorrichtung 15300 entweder als IoT-Server 15204 oder als IoT-Client 15208 wirken, wie mit Bezug auf Fig. 152 beschrieben. Bei anderen Beispielen kann die IoT-Vorrichtung 15300 nur als ein IoT-Client 15208 wirken, beispielsweise falls die IoT-Vorrichtung 15300 beschränkter ist. Bei weiteren Beispielen kann die IoT-Vorrichtung 15300 nur als IoT-Server 15204 wirken.

[0815] Die IoT-Vorrichtung 15300 kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) 15302 aufweisen, das beispielsweise mit der von der Trusted Computing Group 2009 als ISO/IEC 11889 herausgegebenen Spezifikation kompatibel ist. Das TPM 15302 kann einen kryptographischen Prozessor (CP) 15304, einen nichtflüchtigen Speicher (NVM) 15306 und einen sicheren Speicher (SM) 15308 aufweisen. Der CP 15304 kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM 15306 kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM 15308 kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datensegment, das im permanenten Speicher 808 oder im flüchtigen Speicher 804 vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM 15308 kann den mit Bezug auf Fig. 152 beschriebenen sicheren Speicher 15206 bereitstellen.

[0816] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Schlüsselbehandlungsfunktionen aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0817] Der Massenspeicher 808 kann eine sichere Boot-/Messeinrichtung 15310 aufweisen, die Messungen an Code oder Daten ausführt. Eine anfängliche Boot-Messung kann vom Prozessor 802 oder vom TPM 15308 ausgeführt werden, um die sichere Boot-/Messeinrichtung 15310 einzurichten, zusätzliche Messungen auszuführen. Dies kann eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execute Environment (TEE) erzeugen, um der IoT-Vorrichtung 15300 Sicherheit bereitzustellen. Aufeinander folgende Messungen in der TEE können durch das TPM 15308 ausgeführt werden, wenn Codesegmente für den Betrieb vorbereitet werden.

[0818] Ein Authentifizierer 15312 kann verwendet werden, um Kommunikationen mit anderen Mesh-Vorrichtungen 812 oder Vorrichtungen in der Cloud 302 zu authentifizieren. Der Authentifizierer 15312 kann einen Paketkommunikator 15314 zum Senden verschlüsselter Pakete zu anderen Mesh-Vorrichtungen 812 oder Vorrichtungen in der Cloud 302 und zum Empfangen von diesen verwenden. Der Authentifizierer 15312 kann die Kommunikationen unter Verwendung eines von einem Dienstanbieter 15202 bereitgestellten symmetrischen Schlüssels oder eines in der IoT-Vorrichtung 15300 erzeugten temporären symmetrischen Schlüssels (TSK) authentifizieren.

[0819] Ein Schlüsselgenerator 15316 kann verwendet werden, um die temporären symmetrischen Schlüssel (TSK) für Kommunikationen mit anderen Vorrichtungen zu erzeugen. Der Authentifizierer 15312 kann die TSK

mit anderen Vorrichtungen austauschen. Der Schlüsselgenerator **15316** kann auch neue TSK oder neue symmetrische Schlüssel (SK) erzeugen, nachdem die Schlüssel abgelaufen sind, beispielsweise wenn eine sichere Zeit für die Lebensdauer des Schlüssels überschritten wurde. Eine Verschlüsselungs-/Entschlüsselungseinrichtung **15318** kann Kommunikationen unter Verwendung der TSK oder SK verschlüsseln oder entschlüsseln.

[0820] Ein Schlüsselmanager **15320** kann aufgenommen sein, um die Schlüssel zu überwachen und zu behandeln. Hierbei kann festgestellt werden, ob ein Schlüssel abgelaufen ist, und der Schlüsselgenerator **15316** verwendet werden, um einen neuen Schlüssel zur Wiederausgabe zu erzeugen. Der Schlüsselmanager **15320** kann durch den Kommunikator **15314** empfangene Kommunikationen auf eine Widerrufsnachricht von einer anderen Mesh-Vorrichtung **812** oder einer Vorrichtung in der Cloud **302** überwachen, wodurch angegeben wird, dass ein Schlüssel beschädigt wurde. Nach dem Empfang der Widerrufsnachricht kann der Schlüsselmanager **15320** den Status des Schlüssels ändern, um anzugeben, dass der Schlüssel nicht mehr gültig ist. Der Schlüsselmanager **15320** kann dann die Authentifizierung durch den Authentifizierer **15312** wiederauslösen, um die Schlüssel wieder zu erzeugen.

[0821] Fig. 154 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **15400**, das Code aufweist, um einen Prozessor **902** anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **15400** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf Fig. 9 beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **15400** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0822] Das nichtflüchtige maschinenlesbare Medium **15400** kann einen Code **15402** aufweisen, um den Prozessor **902** anzuweisen, sich bei einem Dienstanbieter zu authentifizieren. Ein Code **15404** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Schlüssel für eine sichere Kommunikation oder Speicherung zu erhalten. Der Code **15404** kann den Prozessor **902** anweisen, einen symmetrischen Schlüssel in der Art eines Tickets oder einen asymmetrischen Schlüssel in der Art eines Zertifikats vom Dienstanbieter anzufordern.

[0823] Ein Code **15406** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen symmetrischen Schlüssel für die Kommunikationen zu erzeugen. Der symmetrische Schlüssel kann ein TSK sein, der mit einer anderen Vorrichtung nach der Authentifizierung durch Austausch eines Paares aus einem öffentlichen-/privaten Schlüssel ausgetauscht wird. Der durch den Code **15406** erzeugte symmetrische Schlüssel kann auch ein zur Wiederauffrischung eines abgelaufenen Schlüssels erzeugter neuer Schlüssel sein.

[0824] Ein Code **15408** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob der Schlüssel eine vorgegebene Lebensdauer erreicht hat. Ein Code **15410** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen abgelaufenen Schlüssel wiederaufzufrischen. Ein Code **15412** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Kommunikationen von anderen Vorrichtungen zu verschlüsseln und zu entschlüsseln. Ein Code **15414** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Schlüssel zu widerrufen und die Authentifizierung beim Dienstanbieter zu wiederholen, beispielsweise falls eine Widerrufsnachricht empfangen wird.

[0825] Die hier beschriebenen Schlüsselbehandlungstechniken können in einer Anzahl von Zusammenhängen verwendet werden. Wenn beispielsweise ein Objekt aktiviert wird und sich verbinden muss, kann es Informationen von einem Registrar über andere Dienste oder Agenten, die im Netz laufen, darüber, wie es sich registrieren muss, und darüber, wie andere Dienste und Agenten zu finden sind, verwenden. Öffentliche Registerare sind jedoch für verteilte Denial-of-Service- bzw. Distributed Denial-of-Service (DDoS)-Angriffe anfällig. Falls es möglich ist, kann die Implementation eines Registrars auf der Grundlage eines dezentralisierten Protokolls nützlich sein. Bei einem dezentralisierten Protokoll kann eine Blockchain oder ein Ledger als Ersatz für eine öffentliche Schlüsselinfrastruktur bzw. Public Key Infrastructure (PKI) zum Zugriff auf Vorrichtungs- oder Agentenidentitäten durch ihre Blockchain-Adressen wirken. Die Blockchain kann als Namespace verwendet werden, der sicher, einprägsam und dezentralisiert ist. Namen in einen Namespace sind eine begrenzte Ressource, die in einer dezentralisierten Weise behandelt werden kann. Ferner können Adressen niedrigerer Ebene, die gewöhnlich durch Leihverträge reguliert sind, wie Internetprotokoll (IP) in einem dynamischen Hostkonfigurationsprotokoll bzw. Dynamic Host Configuration Protocol (DHCP), durch Mikrozahlungen oder ein anderes Guthaben oder eine andere Währung abgerechnet und reguliert werden.

[0826] Fig. 155 ist ein schematisches Diagramm eines Prozesses **15500** zum Bootstrappen und zur Entdeckung einer Vorrichtung gemäß einigen Ausführungsformen. Wie hier verwendet, ist das Bootstrappen das

anfängliche Hochfahren einer Vorrichtung, während dessen die Vorrichtung ein Betriebssystem und anderen Code zur Ausführung von Funktionen aus einer Speichervorrichtung laden kann. Der Prozess **15500** kann in einer IoT-Netzumgebung stattfinden. Block **15502** repräsentiert beispielsweise den Fall, in dem eine Vorrichtung bootet und Code ausführt, beispielsweise in einer sicheren Enklave oder einer vertrauenswürdigen Ausführungsumgebung (TEE), wie durch ein vertrauenswürdiges Plattformmodul (TPM) oder andere Technologien eingerichtet.

[0827] In Block **15504** werden die Schlüssel für die Vorrichtung für den Betrieb als Blockchain-Client erzeugt. Dies kann beispielsweise durch den in Block **14206** dargestellten und mit Bezug auf **Fig. 142** beschriebenen Prozess geschehen. Es kann jedoch eine beliebige Anzahl von Schlüsselerzeugungsprozessen verwendet werden, wie unter anderem die mit Bezug auf die **Fig. 137** bis **Fig. 141**, die **Fig. 142** bis **Fig. 146** oder die **Fig. 147** bis **Fig. 151** beschriebenen Schlüsselerzeugungsprozesse.

[0828] In Block **15506** erzeugt die Vorrichtung eine spezielle Kommissionierungstransaktion auf der Blockchain. Die Kommissionierungstransaktion kann das Kaufen eines Domainnamens oder eines anderen eindeutigen Attributs, das Teil eines Gesamtpakets von Attributen, welche die Vorrichtungsidentität ausmachen, sein kann, aufweisen. In Block **15508** wird der Vorrichtung eine Identität zugewiesen, die entweder durch das gekaufte Attribut in der Art eines Domainnamens oder einer universell eindeutigen Kennung bzw. Universally Unique Identifier (UUID) oder durch einen Eigentümer bereitgestellt wird.

[0829] **Fig. 156** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **15600** zum Bootstrappen und zur Entdeckung von Vorrichtungen gemäß einigen Ausführungsformen. Das Verfahren **15600** aus **Fig. 156** kann durch die mit Bezug auf **Fig. 159** beschriebene IoT-Vorrichtung **15900** implementiert werden. Das Verfahren **15600** kann einen modifizierten Boot-Prozess beschreiben, der dazu führt, dass eine Vorrichtung eine Identität erhält. Die Identität kann für die Entdeckung von Diensten und die Bezahlung für die Dienste verwendet werden.

[0830] Block **15602** repräsentiert beispielsweise den Fall, in dem die Vorrichtung einen Boot-Prozess einleitet. Dies kann geschehen, nachdem die Vorrichtung zuerst hochgefahren wird, oder nach einem Reboot geschehen. In Block **15604** initialisiert das BIOS, wobei normale POST-Prüfungen ausgeführt werden. Der Boot-Prozess kann ein sicherer Boot-Prozess sein, um zu gewährleisten, dass nur vertrauenswürdige SW ausgeführt werden. Dies geschieht gewöhnlich durch von einem Hersteller dafür eingerichtete Hardware unter Verwendung von Befehlen von einem Firmwarelieferanten zur Speicherung von Schlüsseln in der Vorrichtung vor der Installation.

[0831] In Block **15606** kann der sichere Boot-Prozess zu einer sicheren Enklave oder einer vertrauenswürdigen Ausführungsumgebung (TEE) booten. Die sichere Enklave kann einen Identitäts-Client ausführen, der beispielsweise ein von Intel als modulare Open-Source-Plattform zum Bilden, Installieren und Ausführen verteilter Ledger herausgegebener Sawtooth Lake Client sein könnte. Sobald der Identitäts-Client initialisiert wurde, kann die Vorrichtung wie im Normalfall weiter booten. In Block **15608** bootet das Betriebssystem (OS) zu einer geeigneten Ausführungsebene. Bei einigen Beispielen ist kein Betriebssystem vorhanden, sondern wird die Vorrichtung vielmehr durch ein höher entwickeltes BIOS betrieben.

[0832] In Block **15610** wird festgestellt, ob der Boot-Prozess korrekt ausgeführt wird. Falls dies nicht der Fall ist, wird in Block **15612** festgestellt, ob die Vorrichtung zurückgesetzt werden sollte. Das Rücksetzen kann ein in der Fabrik ausgeführtes Rücksetzen der Vorrichtung sein, das alle Daten aus der Vorrichtung löschen kann und sie zurücksetzen können, um aus einem an Bord befindlichen Nullease-ROM-Abbild oder dergleichen zu booten. Falls dies geschieht, kehrt der Prozessfluss zu Block **15604** zurück, um den Boot-Prozess zu wiederholen. Falls festgestellt wird, dass die Vorrichtung nicht zurückgesetzt werden sollte, wird in Block **15614** eine Warnnachricht ausgesendet. Der Prozess endet dann in Block **15616**.

[0833] Falls in Block **15610** festgestellt wird, dass alles während des Boot-Prozesses korrekt funktioniert hat, wird der Prozessfluss in Block **15618** fortgesetzt, um eine Identität zu erhalten. Vorrichtungen können mehrere Identitäten zugewiesen werden, wobei Vorrichtungen beispielsweise DNS-Namen, IP-Adressen, MAC-Adressen, UUID oder andere Verfahren zum Festlegen ihrer Identität aufweisen können. Ferner können Vorrichtungsidentifikationen unter anderem durch die Verwendung von Blockchain-Techniken, wie mit Bezug auf die **Fig. 5** bis **Fig. 9** beschrieben, zugewiesen werden. Beim vorliegenden Beispiel kann eine global eindeutige Identität erhalten werden, um an einem durch einen Smart Contract oder ein ähnliches Konstrukt beherrschten Prozess teilzunehmen. Wie hier verwendet, kann ein Smart Contract ein automatisch verhandelter Vertrag

zwischen zwei Vorrichtungen sein, wobei eine erste Vorrichtung für eine zweite Vorrichtung im Austausch für eine Bezahlung von der zweiten Vorrichtung einen Dienst ausführt oder ihr Daten bereitstellt.

[0834] In Block **15620** sind mögliche Dienste, von denen eine Identität erhalten oder entdeckt werden kann, spezifiziert. Die Vorrichtung kann diese Funktion unter Verwendung dynamischer oder statischer Prozesse ausführen, einschließlich Verfahren in der Art neuer DHCP-Optionen, welche den Ort auf Smart Contracts oder Konsens beruhender Netze spezifizieren, jedoch ohne Einschränkungen darauf. Ferner können die möglichen Dienste vorab in die Vorrichtung geladen werden, wie im Fall von einigen Kryptowährungs-Netz-Clients. Die möglichen Dienste können in internetbasierten Dienstregistern angekündigt werden, welche die Vorrichtung entdeckt oder für deren Verwendung sie hartcodiert ist. Die möglichen Dienste können unter anderem in einem zentralisierten Namensdienst in der Art von Namecoin angekündigt werden. Dementsprechend kann der Client Kenntnis von einem oder mehreren solcher Netze erhalten, welche eine Netzidentität verwenden können und damit beginnen können, mit einem von einem Smart-Contract-Prozess bereitgestellten Dienst zu interagieren. Verschiedene Dienste oder Netze können gewählt worden sein, um Identitätsmechanismen zu teilen, oder sie können völlig inkompatible Ansätze für die Identität haben.

[0835] Die Vorrichtung kann Dienste, an denen sie teilzunehmen versucht, auf der Grundlage ihrer Fähigkeit auswählen, eine Identität des vom Dienst spezifizierten Typs oder auf der Grundlage ihres vorprogrammierten Zwecks zu erzeugen. Die Dienste können während des Bootens statisch in der sicheren Enklave zugewiesen werden oder durch ein Richtlinienystem dynamisch festgelegt werden. Die Dienste können jedoch zuerst durch Prozesse überprüft werden, die innerhalb der sicheren Enklave laufen, bevor ihnen vertraut wird.

[0836] In Block **15622** stellt die Vorrichtung fest, ob ein Verfahren, durch das sie Kennungen erhalten wird, ausgewählt wurde. Wie erwähnt, können mehrere Verfahren ausgewählt werden, falls mehrere Netze verfügbar sind, für welche Kennungen verwendet werden können. Falls in Block **15622** kein Verfahren ausgewählt wird, kann in Block **15614** eine Warnnachricht gesendet werden und endet das Verfahren **15600** in Block **15616**. Weil die Vorrichtung eine Vielzahl von Identitäten aufweisen kann, wie einen DNS-Namen, einen NetBIOS-Namen, eine IP-Adresse, eine UUID und dergleichen, kann die Warnung viele Formen annehmen. Beispielsweise kann die Warnung unter anderem eine E-Mail an einen Administrator, eine SMTP-Falle, ein Eintrag in einer lokalen oder fernen Protokolldatei, eine SMS-Nachricht, eine blinkende LED-Sequenz im Außenbereich der Vorrichtung sein.

[0837] Falls in Block **15622** ein Verfahren ausgewählt wurde, kann die Vorrichtung in Block **15624** eine Identität für den gewählten Dienst erzeugen. Der Vorrichtungseigentümer kann eine Option festlegen, beispielsweise durch eine Konfiguration in der sicheren Enklave, um von der Vorrichtung zu fordern, Identitätsverfahren zu verwenden, die hardwaregestützt sind. Bei anderen Beispielen kann der Eigentümer die Auswahl eines hardwaregestützten Identitätsverfahrens optional oder bevorzugt ausführen, wodurch es der Vorrichtung ermöglicht werden kann, ein weniger sicheres Verfahren zu verwenden, um Schlüssel oder andere eindeutige Kennungen zu erzeugen, wie es vom Dienst gefordert wird. Diese Einstellungen oder andere unerwartete Fehler oder Ausnahmen können dazu führen, dass die Vorrichtung keine Identität für einen bestimmten Dienst erzeugt.

[0838] In Block **15626** wird festgestellt, ob eine Identität für die Vorrichtung erfolgreich erzeugt wurde. Falls die Identität nicht erfolgreich erzeugt wurde oder eine Anzahl von Identitäten zu erzeugen ist, kann das Verfahren **15600** zu Block **15622** zurückkehren, um festzustellen, ob ein anderes Verfahren für die Erzeugung der Identifikation ausgewählt werden kann. Die Vorrichtung kann eine Liste möglicher Verfahren oder Dienste durchlaufen, bis ihre Richtlinienereinstellungen erfüllt wurden. Beispielsweise kann eine Richtlinie fordern, dass die Vorrichtung angehalten werden sollte, nachdem sie eine Identität erfolgreich erzeugt hat. Bei anderen Beispielen kann die Vorrichtung alle verfügbaren Dienste ausprobieren und dabei viele Mechanismen der Identitätserzeugung probieren, bis sie erfolgreich ist oder bis alle Optionen durchlaufen wurden. Der Identitätserzeugungsprozess kann auch Ressourcen erhalten, welche die Vorrichtung verwenden kann, um Transaktionen auszuführen, wobei der Vorrichtung beispielsweise im Fall eines Kryptowährungsnetzes ein anfänglicher Guthabenbetrag zugewiesen werden kann, wenn die Identität zugewiesen wird.

[0839] In Block **15628** kann eine Kommissionierungstransaktion erzeugt werden. Die Kommissionierungstransaktion kann ein hardwaregestützter Prozess sein, der zur sicheren und vertrauenswürdigen Erzeugung eines Guthabenbetrags für die Vorrichtung führt. Dies kann die Erzeugung neuer Münzen am Netz einschließen.

[0840] Die Kommissionierungstransaktion kann für das jeweilige Konsensnetz spezifisch sein. Sie kann die Identität der Vorrichtung im Netz validieren und die vom Konsensnetz geforderten öffentlichen Identitätsinformationen aufweisen. Beispielsweise kann eine vom privaten Schlüssel der Vorrichtung signierte Transakti-

on den öffentlichen Schlüssel und die Wallet-Kennung in der Transaktion aufweisen, so dass die Quelle der Transaktion leicht überprüft werden kann. Die Kommissionierungstransaktion kann zu einer beliebigen Zeit nach der Identitätserzeugung stattfinden. Ferner kann sie bedarfsgetrieben sein, wobei sie beispielsweise nur beim ersten Mal, wenn die Vorrichtung an einer Transaktion teilnehmen möchte, geschehen kann. Nach der ersten Transaktion ist die Identität der Vorrichtung im Netz öffentlich bekannt und können Nachrichten von ihr unter Verwendung des vom Konsensnetz bereitgestellten Mechanismus verifiziert werden.

[0841] In Block **15630** wird festgestellt, ob die Kommissionierungstransaktion abgeschlossen wurde. Falls die Kommissionierungstransaktion fehlgeschlagen ist, das Netz die Transaktion beispielsweise als ungültig abgelehnt hat, erzeugt die Vorrichtung in Block **15632** eine Warnung. Abhängig vom Fehler kann die Vorrichtung einige Parameter der Transaktion ändern und die Transaktion in Block **15634** erneut versuchen. Die Vorrichtung kann versuchen, eine neue Identität für diesen Dienst zu erzeugen oder andere Dienste auszuwählen, für welche Identitäten zu erzeugen sind.

[0842] Ein Beispiel eines Fehlers, bei dem erneut versucht werden kann, wäre der Kauf eines Domainnamens. Der Domainname kann verfügbar sein, wenn er geprüft wird, und die Transaktion wird erzeugt. Bevor sie verarbeitet ist, erhält jedoch eine andere Entität den Domainnamen. Bei diesem Beispiel kann die Vorrichtung den Domainnamensparameter aktualisieren und die Transaktion erneut versuchen. Einige Transaktionen können fehlschlagen, jedoch möglicherweise nicht erneut versucht werden. Beispielsweise kann eine doppelte Zahlung nicht erneut durchführbar sein.

[0843] Falls in Block **15630** festgestellt wird, dass die Transaktion erfolgreich abgeschlossen wurde, kann in Block **15636** bestätigt werden, dass die Vorrichtung eine Identität hat. In Block **15614** kann eine Warnung erzeugt werden, um anzugeben, dass der Prozess ganz abgeschlossen ist. Der Prozess würde dann in Block **15616** enden.

[0844] Falls die Vorrichtung an einem Punkt in der Zukunft dekommissioniert wird, kann das Blockchain-Protokoll die Beseitigung der Guthabenbeträge in der Art gemeinter oder zugewiesener Münzen bestimmen. Die Münzen können zerstört werden oder auf andere Weise aus dem Umlauf entfernt werden. Die Münzen oder der Guthabenbetrag können auf andere vom Vorrichtungseigentümer spezifizierte Vorrichtungen umverteilt werden. Bei einigen Beispielen kann der Guthabenbetrag oder können die Münzen an einer Börse verkauft und in eine Währung umgewandelt werden, um sie einem Vorrichtungseigentümer zu erstatten.

[0845] Der Prozess ist nicht auf den in **Fig. 155** und **Fig. 156** dargestellten Block beschränkt. Ein merkmalsreicher Mechanismus, der das Konzept eines Blockchain-Smart-Contracts verwendet, kann implementiert werden.

[0846] **Fig. 157** ist ein schematisches Diagramm eines Prozesses **15700** zum Bootstrappen, zur Entdeckung und für den Lebenszyklus von Vorrichtungen unter Verwendung von Smart-Contract-Funktionen gemäß einigen Ausführungsformen. Block **15702** repräsentiert beispielsweise den Fall, in dem eine Vorrichtung bootet. Dies kann nach dem Einschalten der Vorrichtung oder nach dem Reboot der Vorrichtung geschehen. Wie mit Bezug auf Block **15502** aus **Fig. 155** beschrieben wurde, bootet die Vorrichtung in einer sicheren Enklave in der Art einer TEE und führt darin Code aus.

[0847] In Block **15704** kann die Vorrichtung einen als Blockchain-Client verwendeten Schlüssel erzeugen. Dies kann beispielsweise wie mit Bezug auf Block **14206** aus **Fig. 142** beschrieben geschehen.

[0848] In Block **15706** kann die Vorrichtung mit einem Smart Contract **15708** auf der Blockchain, beispielsweise durch Erzeugen einer Kommissionierungstransaktion, interagieren. Eine Contract-Beitrittsfunktion **15710** kann ausgeführt werden, wenn eine neue Vorrichtung beim ersten Mal mit dem Smart Contract **15708** interagiert. Der Smart Contract **15708** kann Vorrichtungsattestierungsmerkmale unterstützen und entscheiden, ob eine bestimmte Vorrichtung im Smart Contract **15708** zu akzeptieren ist. Die Inhalte der Kommissionierungstransaktion können verwendet werden, um die Akzeptanz zu bestimmen. Die Contract-Beitrittsfunktion **15710** kann Richtlinien auf einer Vorrichtung erzwingen, bevor ihr erlaubt wird, dem Smart Contract **15708** beizutreten. Beispielsweise kann die Contract-Beitrittsfunktion **15710** fordern, dass die Vorrichtung ihre Festplatte oder ihren Speicher unter Verwendung eines spezifizierten minimalen Standards verschlüsselt, bevor sie beitrifft. Die Contract-Beitrittsfunktion **15710** kann andere Merkmale oder zusätzliche Interaktionen mit der Vorrichtung fordern, um sie zu präparieren, bevor sie in den Smart Contract **15708** eingelassen wird.

[0849] Ähnlich können der Vorrichtung Bedingungen oder Funktionen auferlegt werden, bevor sie den Smart Contract **15708** verlässt. Diese können Teil einer Contract-Verlassensfunktion **15712** sein. Beispielsweise kann die Contract-Verlassensfunktion **15712** fordern, dass die Vorrichtung ihren Speicher löscht, beispielsweise durch Ausführen eines Rücksetzens in der Fabrik. Andere Forderungen der Contract-Verlassensfunktion **15712** können das Senden einer Lebensendenachricht zu einem Wartungsdienstleister in der Art einer Dienstleistungsorganisation und das Senden einer Drohne oder eines Roboters an den aktuellen Vorrichtungs-ort, so dass die Vorrichtung aufgesammelt werden kann und sich dann selbst abschaltet, einschließen. Die Contract-Verlassensfunktion **15712** kann eine beliebige Anzahl vom Contract-Eigentümer spezifizierter Bedingungen aufweisen.

[0850] Falls der Vorrichtung erlaubt wird, dem Smart Contract **15708** beizutreten, wird sie einer Liste erzeugter Vorrichtungen **15714**, beispielsweise in der Blockchain, hinzugefügt. Im Allgemeinen kann nur die Steuerfunktion in der Blockchain gespeichert werden. Variablen können außerhalb der Kette in einer Anzahl verschiedener sicherer Speichermechanismen gespeichert werden. Diese Mechanismen können in der Blockchain eine Referenz aufweisen. Dies kann für Variablen, die erhebliche Speicheranforderungen haben, nützlich sein.

[0851] Eine Vorrichtungsattributliste **15716** kann in Block **15714** mit der Liste erzeugter Vorrichtungen assoziiert werden. Ferner können Vorrichtungen selbst Attribute beschreiben und die Attribute entweder in der Blockchain oder außerhalb der Chain in einem sicheren Speichermechanismus speichern. Die Attribute können Zusammenhangseigenschaften für eine einfache Vorrichtung in der Art des Typs der Vorrichtung, des Orts, von Vorrichtungsfähigkeiten und -merkmalen einschließen. Die Attribute können auch eine Liste angekündigter Dienste, welche die Vorrichtung bietet, einschließen. Dies kann als ein Dienstentdeckungsmechanismus wirken.

[0852] Der Smart Contract **15708** kann während des Kommissionierungsprozesses oder zu einer späteren Zeit Tokens **15718** an Vorrichtungen ausgeben. Die Tokens können eine Anzahl abstrakter Bedeutungen haben und für verschiedene Zwecke ausgegeben werden. Falls eine Vorrichtung beispielsweise innerhalb des Smart Contracts **15708** festgelegte Kriterien erfüllt, beispielsweise einen gewissen Grad an Verschlüsselungsfähigkeiten hat, kann ein spezieller Typ eines Vertrauens-Tokens an sie ausgegeben werden. Wenn auf einen Dienst zugegriffen wird, kann dem Dienst das Token präsentiert werden, um zu fordern, dass eine Datensenske für die von der Vorrichtung kommenden Daten diese Verschlüsselungsmerkmale aufweist. Ferner können Tokens verwendet werden, um zu ermöglichen, dass eine Vorrichtung auf andere Dienste zugreift oder die Identität überprüft.

[0853] Der Smart Contract **15708** kann Tokens **15720** widerrufen, wenn eine Vorrichtung bereit ist, aus dem Contract auszusteigen. Sobald das Token widerrufen wurde, ist der Zugriff unter diesem Token nicht mehr gültig. Die widerrufene Token-Funktion **15720** kann durch die Contract-Verlassensfunktion **15712** als Teil der Bedingungen zum Verlassen des Contracts ausgelöst werden.

[0854] Sobald die Vorrichtung auf dem Netz kommissioniert wurde, kann sie in Block **15722** mit Operationen unter dem Smart Contract **15708** beginnen. Die Vorrichtung kann zu einer beliebigen Zeit während ihres Betriebs mit dem Smart Contract **15708** interagieren, um neue Tokens anzufordern, falls neue Merkmale an der Vorrichtung verfügbar werden oder falls sich ihre Attribute ändern.

[0855] Die Beziehung von Vorrichtungen mit dem Smart Contract **15708** kann viele: 1, viele:vielen oder 1:vielen sein. Tokens und Attribute können zu jeder Zeit während der Lebensdauer der Vorrichtung in Zusammenwirken mit dem Contract geändert werden. Der Smart Contract **15708** kann ein Teil der Vorrichtung sein, beispielsweise eine geteilte Blockchain aufweisen, die auf anderen Vorrichtungen gespiegelt ist. Bei diesem Beispiel können die Funktionen des Smart Contracts **15708** Teil der zur Aufrechterhaltung der Blockchain verwendeten Blockchain-Logik sein. Bei anderen Beispielen kann sich der Smart Contract **15708** an einer anderen Vorrichtung in einem IoT-Netz oder in der Cloud befinden.

[0856] In Block **15724** kann die Vorrichtung beispielsweise durch Ausgeben einer Dekommissionierungstransaktion an die Blockchain des Smart Contracts **15708** dekommissioniert werden. Jegliche ausgegebene Tokens werden widerrufen **15720**, und die Vorrichtung wird aus der Liste erzeugter Vorrichtungen **15714** entfernt. Ferner kann die Contract-Verlassensfunktion **15712** implementiert werden.

[0857] Fig. 158 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **15800** zum Bootstrappen, zur Entdeckung und für den Lebenszyklus von Vorrichtungen unter Verwendung eines Smart Contracts gemäß einigen Ausführungsformen. Das Verfahren **15800** aus Fig. 158 kann durch die mit Bezug auf Fig. 159 be-

schriebene IoT-Vorrichtung **15900** implementiert werden. Block **15802** repräsentiert beispielsweise das Booten der Vorrichtung. Dies kann wie mit Bezug auf die Blöcke **15602** bis **15608** aus **Fig. 156** beschrieben ausgeführt werden.

[0858] In Block **15804** können Schlüssel für die Vorrichtung erzeugt werden, um an einer Blockchain oder einem Smart Contract teilzunehmen. Der Schlüsselerzeugungsschritt kann wie hier beschrieben ausgeführt werden, beispielsweise wie mit Bezug auf Block **14206** aus **Fig. 142** beschrieben.

[0859] In Block **15806** kann eine Kommissionierungstransaktion erzeugt und implementiert werden. Die Kommissionierungstransaktion kann wie mit Bezug auf Block **15628** aus **Fig. 156** beschrieben sein. In Block **15808** wird festgestellt, ob die Kommissionierungstransaktion erfolgreich war. Falls dies nicht der Fall ist, kann die Vorrichtung neu gebootet werden, wie in Block **15802** beschrieben.

[0860] Falls die Kommissionierungstransaktion erfolgreich war, wie in Block **15808** festgestellt, können die Contracts in Block **15810** spezifiziert werden. Weil die Vorrichtungen in der Lage sein können, auf verschiedene Arten zu interagieren, können die verschiedenen Optionen durch Spezifizieren der Contracts aufgelistet werden. Die Spezifizierung kann in einer statischen oder dynamischen Art erfolgen, beispielsweise an einem im Internet gehosteten Register von Contracts. Ferner kann sie unter Verwendung eines in Abschnitt **3.4.3** beschriebenen Nachschlageverfahrens ausgeführt werden.

[0861] In Block **15812** tritt die Vorrichtung einem Smart Contract durch Interaktion mit ihm bei, wobei eine Gebühr zur Wallet-Adresse des Eigentümers des Smart Contracts gesendet werden kann. Mit der Gebühr kann eine Verhandlung einhergehen, wobei der Contract beispielsweise Optionen anbieten kann, bei denen die Vorrichtung weniger bezahlen kann, falls sie einigen Bedingungen und Konditionen zustimmt, wie das Bereitstellen vertrauenswürdiger Daten oder attestierter Attribute. Es können auch andere Verhandlungsmechanismen verwendet werden, einschließlich der hier detailliert dargelegten.

[0862] In Block **15814** wird festgestellt, ob die Verhandlung erfolgreich war, und falls dies nicht der Fall ist, wird die Verhandlung in Block **15812** fortgesetzt. Falls die Verhandlung in Block **15814** erfolgreich war, wird die Vorrichtung in Block **15816** zu einer Liste erzeugter Vorrichtungen hinzugefügt, beispielsweise durch Ausführen einer Blockchain-Transaktion. Dies kann so geschehen wie mit Bezug auf die Liste erzeugter Vorrichtungen **15714** mit Bezug auf Block **15708** von **Fig. 157** beschrieben wurde.

[0863] In Block **15818** werden die Attribute der Vorrichtung veröffentlicht. Für jedes Attribut kann möglicherweise identifiziert werden, ob es eine Hardwareumgebung in der Art einer von einem vertrauenswürdigen Plattformmodul (TPM) unterstützten vertrauenswürdigen Ausführungsumgebung (TEE) oder einen anderen vertrauenswürdigen Mechanismus gibt, der verwendet werden kann, um zu attestieren oder zu verifizieren, dass die Vorrichtung tatsächlich dieses Attribut besitzt.

[0864] In Block **15820** kann die Vorrichtung Tokens zur Funktionsweise unter dem Smart Contract anfordern. Die Tokens können Eigentümern von Diensten durch die Vorrichtung präsentiert werden, wenn sie versucht, auf Dienste oder Ressourcen zuzugreifen oder diese anzubieten, sobald die Vorrichtung ganz funktionsfähig ist. Die Kriterien für die Ausgabe von Tokens können Merkmale in der Art der Attributattestierung berücksichtigen. Falls in Block **15822** ein bestimmtes Attribut attestiert wird, kann der Vorrichtung in Block **15824** ein höherwertiges Token zugewiesen werden. Falls dies nicht der Fall ist, kann beispielsweise in Block **15826** ein Token geringeren Werts zugewiesen werden. Der Vorrichtung können mehrere Token-Typen und Token-Volumina zugewiesen werden. Dies unterliegt jedoch dem Ermessen des Eigentümers des Smart Contracts, wenn er den Smart Contract entwickelt. Einige Tokens können verbrauchbar sein, wobei sie beispielsweise, wenn sie während des Vorrichtungsbetriebs einem Prozess-, Dienst- oder Systemeigentümer präsentiert werden, in einem Bezahlung-für-Benutzung-Modell verbraucht werden, wobei die Tokens vom Wallet der Vorrichtung zum Wallet des Eigentümers übertragen werden. Andere Tokens können beständig sein, wobei sie beispielsweise lediglich präsentiert werden können, um zu verifizieren, dass die Vorrichtung ein Mitglied eines bestimmten Smart Contracts oder einer Gruppe von Vorrichtungen ist, oder um zu attestieren, dass die Vorrichtung spezifische Attribute, Fähigkeiten oder Merkmale aufweist.

[0865] In Block **15828** wird die Vorrichtung kommissioniert und nimmt den Betrieb in Block **15830** auf. Dies kann wie mit Bezug auf Block **15722** aus **Fig. 157** beschrieben geschehen.

[0866] In Block **15832** wird die Vorrichtung dekommissioniert. Falls die Vorrichtung nicht verwendete Tokens aufweist, kann dies möglicherweise zu einer Erstattung von Währung zwischen am Smart Contract teilnehmenden Parteien führen. Der Prozess endet dann in Block **15834**.

[0867] **Fig. 159** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 15900 zum Bootstrappen, zur Entdeckung und zur Lebenszyklusbehandlung vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 15900 andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0868] Die IoT-Vorrichtung **15900** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) **15902** aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel ist. Das TPM **15902** kann einen kryptographischen Prozessor (CP) **15904**, einen nichtflüchtigen Speicher (NVM) **15906** und einen sicheren Speicher (SM) **15908** aufweisen. Der CP **15904** kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM **15906** kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM **15908** kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, kann eine Messung ein Hash-Code sein, der an einem im permanenten Speicher **808** oder im flüchtigen Speicher **804** gespeicherten Code- oder Datensegment berechnet wird. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM **15908** kann einen sicheren Speicher bereitstellen. Das TPM **15902** kann verwendet werden, um eine TEE oder eine sichere Enklave für das Ausführen von Programmen festzulegen.

[0869] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Schlüsselbehandlungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0870] Der Massenspeicher **808** kann eine sichere Boot-/Messeinrichtung **15910** aufweisen, die Messungen an Code oder Daten ausführt. Eine anfängliche Boot-Messung kann vom Prozessor **802** oder vom CP **15904** ausgeführt werden, um die sichere Boot-/Messeinrichtung **15910** einzurichten, zusätzliche Messungen auszuführen.

[0871] Ein Schlüsselgenerator **15912** kann verwendet werden, um Schlüssel für Kommunikationen mit anderen Vorrichtungen zu erzeugen. Dies kann beispielsweise durch den in Block **14206** dargestellten und mit Bezug auf **Fig. 142** beschriebenen Prozess geschehen. Es kann jedoch eine beliebige Anzahl von Schlüsselerzeugungsprozessen verwendet werden, wie unter anderem die mit Bezug auf die **Fig. 137** bis **Fig. 141**, die **Fig. 142** bis **Fig. 146** oder die **Fig. 147** bis **Fig. 151** beschriebenen Schlüsselerzeugungsprozesse.

[0872] Ein Dienstspezifizierer **15914** kann aufgenommen sein, um für die IoT-Vorrichtung **15900** verfügbare Dienste oder Dienste, die von der IoT-Vorrichtung **15900** bereitgestellt werden können, zu spezifizieren. Für den Betrieb in Smart-Contract-Umgebungen kann ein Contract-Spezifizierer **15916** Contracts entdecken, denen die IoT-Vorrichtung **15900** beitreten kann. Der Contract-Spezifizierer **15916** kann eine Anzahl von Entdeckungstechnologien verwenden, um Contracts zu entdecken, wie die Funktionen, die als Teil der Spezifikationen bereitgestellt werden, die unter anderem von der Open Connectivity Foundation, der Allseen Alliance oder dem Open Fog Consortium bereitgestellt werden.

[0873] Smart-Contract-Funktionen **15918**, wie beispielsweise mit Bezug auf Block **15708** aus **Fig. 157** beschrieben, können aufgenommen werden, um die Verwendung der IoT-Vorrichtung **15900** als Host für einen Smart Contract zu unterstützen.

[0874] Eine Blockchain-Logik **15920** kann aufgenommen sein, um eine Blockchain **15922** zu unterhalten, die Dienste, Attribute, Identitäten von Vorrichtungen, Contracts, Münzguthaben und dergleichen hält. Die Blockchain-Logik **15920** kann verwendet werden, um die Blockchain-Transaktionen zu anderen IoT-Vorrichtungen zu übertragen.

[0875] Fig. 160 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **16000**, das Code aufweist, um einen Prozessor **902** anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **16000** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf Fig. 9 beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **16000** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0876] Das nichtflüchtige maschinenlesbare Medium **16000** kann einen Code **16002** aufweisen, um den Prozessor **902** anzuweisen, in eine sichere Enklave zu booten. Ein Code **16004** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Identität zu erhalten. Ein Code **16006** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Schlüssel für Kommunikationen zu erzeugen.

[0877] Ein Code **16008** kann aufgenommen sein, um den Prozessor **902** anzuweisen, verfügbare Dienste oder Smart Contracts zu spezifizieren. Ein Code **16010** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einem Smart Contract beizutreten. Ein Code **16012** kann aufgenommen sein, um den Prozessor **902** anzuweisen, von der IoT-Vorrichtung verfügbare Attribute oder Dienste zu veröffentlichen. Ein Code **16014** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Tokens für den Betrieb unter einem Smart Contract anzufordern.

[0878] Zur Teilnahme an einem Netz kann eine Vorrichtung oder ein Agent, die oder der Daten oder Ressourcen benötigt, das Netz und andere miteinander verbundenen Netze durchsuchen, um die Daten oder Ressourcen zu erhalten. Wie hier verwendet, können die Daten jegliche für die Ausführung einer Funktion in der vorliegenden Vorrichtung benötigte Daten sein, wie der Abstandsverkehrsfluss für eine Kreuzungssteuereinrichtung. Ressourcen umfassen jede Funktion, die zur Ausführung einer Aufgabe verwendet werden kann, wie unter anderem ein Vorhersagemodell, das auf einem vorgeschalteten System ausgeführt wird, oder einen Code, der verwendet wird, um eine lokale Funktion auszuführen. Durch Fluten des Netzes mit Anfragen können die Netzkommunikationen jedoch überlastet werden, und es können dadurch Probleme für energetisch beschränkte Vorrichtungen hervorgerufen werden. Ferner können zentralisierte Netze für verteilte Denial-of-Service(DDoS)-Angriffe anfällig sein. Die Verwendung eines Ledger- oder Blockchain-zertifizierten Guthabens kann dabei helfen, die Netzlast zu verringern und es Objekten zu ermöglichen, ihre Ressourcen besser zu verwalten sowie die Anfälligkeit des Netzes für DDoS-Angriffe zu verringern.

[0879] Um Ressourcen für die Verfolgung besser zu organisieren, können die Ressourcen in einem auf einer verteilten Hash-Tabelle (DHT) beruhenden Netz in der Art von Kademia verteilt werden. In einem aus n Knoten bestehenden Kademia-Netz nimmt das Finden eines Knotens im Netz maximal $O(\log(n))$ Hops in Anspruch. Zusätzlich verwenden solche Netze das Konzept von k -Buckets, was im Wesentlichen bedeutet, dass Knoten in einem Netz ihre eigene Nachbarschaft gut kennen und somit ihr lokaler k -Bucket eine große Anzahl von Knoten aufweist. In einigen Fällen sind jedoch, je weiter entfernt Knoten von einem Knoten sind, desto mehr Knoten vorhanden, was darauf hinweist, dass k -Buckets mit niedrigeren k -Werten weniger Knoten haben.

[0880] Wie erwähnt, können aktuelle Blockchain-Techniken einen Merckle-Hash-Baum als Weg zum Indexieren eines bestimmten Blocks in der Blockchain bilden. Falls ein Block-Hash bekannt ist, kann der Block wirksam in einem Ablagespeicher von Blöcken lokalisiert werden. Dies kann als eine Form einer DHT angesehen werden. Eine DHT kann auch verwendet werden, um spezifische Daten zu identifizieren, die in einer Blockchain enthalten sind. Bei diesem Ansatz kann ein Datenwert zu einer DHT gehasht werden, wobei der Ort in der DHT-Datenbank den Blockchain-Block-Hash offenbart, wo die Daten gefunden werden können.

[0881] Ein System, das die Vertrauenswürdigkeit der Daten überprüfen möchte, kann einen zwei Schritte aufweisenden Nachschlageprozess ausführen, wobei die interessierenden Daten zu einer DHT-Stelle gehasht werden. Diese Stelle offenbart die Block-Hash-Werte. Die Block-Hash-Werte werden zum Merckle-Baum gehasht, der den eigentlichen Block in der Blockchain offenbart. Eine Berechnung des Block-Hashes und eine Überprüfung des nächstvorhergehenden Blocks verifiziert die Blockintegrität innerhalb der Chain. Auf diese Weise kann die Integrität aller Daten, die in einer DHT erkennbar sind, gemäß einem Infrastrukturvertrauensmechanismus validiert werden.

[0882] Ein Bloomfiltermechanismus, wie hier beschrieben, kann unter Verwendung einer DHT implementiert werden. Wenn ein DHT-Wert zur Bildung eines Bloomfilters verwendet wird, kann er angeben, dass es ein Thema für dieses Datenelement gibt, das zur Subskription durch eine Gemeinschaft von Teilnehmern verfügbar

ist. Die Gemeinschaft kann am Bloomfilterwert interessiert sein und benachrichtigt werden, wannimmer eine den Datenwert betreffende Transaktion in einer Blockchain gefunden wird.

[0883] Datenanalytiken sollen Korrelationen zwischen anscheinend unkorrelierten Daten finden. Daher könnte eine Analytikmaschine eine zuvor nicht angenommene Korrelation hypothesieren und diese Themen subskribieren. Falls die DHT für die hypothetisch korrelierten Werte innerhalb eines Zeitrahmens auslösen, der statistisch interessiert, kann ein Datenanalyst seine Hypothese testen. Weil ein erheblicher Korpus von auf die Blockchain abgebildeten Transaktionen vorhanden ist, kann dies eine wirksame Benachrichtigung über die Hypothesentests des Datenanalysten ermöglichen.

[0884] Dieser Ansatz für eine Netzstruktur bedeutet, dass Anfragen an ferne Knoten detaillierte Informationen über die ferne Nachbarschaft zurückgeben können, ohne dass eine vollständige Netzkarte für jeden teilnehmenden Knoten repliziert werden müsste. Hierdurch kann das Netz viel dynamischer gehalten werden. Rundsendevorgänge zur Entdeckung von Ressourcen im lokalen Netz sind verhältnismäßig kostengünstig, und die föderative Natur eines Gesamtnetzes bedeutet, dass das Niveau des Ressourcenentdeckungs-Rundsendeverkehrs über das gesamte Netz verringert werden kann.

[0885] Frühere Konsensnetze wiesen dieses Konzept jedoch nicht auf, weil die Verfahren zur Verwendung einer Blockchain als Steuerebene mit einer komplementären Datenspeicherebene außerhalb der Chain nicht entwickelt waren. Daher stellen hier offenbarte Aspekte ein Verfahren bereit, das verwendet werden kann, um dies zu ermöglichen, und sie adressieren demgemäß Skalierbarkeitsprobleme, die auftreten, wenn mehr Daten im Laufe der Zeit in der Chain gespeichert werden.

[0886] Wie hier beschrieben, kann eine Blockchain, die so ausgelegt ist, dass die Konsensknoten in einer k-Bucket-Weise verteilt sind, die Wirksamkeit der Blockchain für die Lokalisierung von Ressourcen verbessern. Die k-Buckets können lokale, segmentierte Netze einführen, die halbautonom sind und bei denen lokal verfügbare Dienste und Contracts gespeichert werden können, ohne sie über das gesamte Netz zu verteilen. Dieser Speicher kann außerhalb der Chain oder in der Chain vorhanden sein.

[0887] Wie hier beschrieben, können Vorrichtungen Dienst-, Smart-Contract- und andere Informationen innerhalb des Netzes lokalisieren möchten. Eine Speicherung solcher Informationen in der Chain kann Skalierbarkeits- und Leistungsfähigkeitsprobleme erzeugen, weil die Blockchain als eine Steuerebene und nicht als eine Datenebene angesehen werden kann. Durch die Verwendung dieses Konzepts des Ledger-zertifizierten Guthabens können dynamische Kosten jedem Hop zugeordnet werden, der benötigt wird, um einen Dienst oder einen Smart Contract zu erhalten. Wenngleich eine globale Suche zu den besten verfügbaren Übereinstimmungen führen kann, kann dies in Bezug auf die Zeit und die Abrechnung für die Ausführung kostspieliger sein. Eine Such-Entität muss daher eine Kompromissentscheidung zwischen dem Zahlen der Kosten für einen Hop und der Zufriedenheit mit dem aktuellen Suchergebnis, das eine leere Menge sein könnte, treffen. Die gesuchten Ressourcen müssen in einem entdeckbaren Format vorhanden sein, und der Gedanke eines Bloomfilters könnte als Technik angewendet werden, um die Wirksamkeit von Suchen über das Netz weiter zu erhöhen.

[0888] Fig. 161 ist ein schematisches Diagramm eines Prozesses 16100, bei dem Bloomfilter-Hops zur Entdeckung von Ressourcen verwendet werden. Beim Prozess 16100 überträgt ein Client-c-Knoten 16102 gemäß einigen Ausführungsformen eine Suche 16104 nach einer Ressource zu einem nächstgelegenen Mining-Knoten 16106. Der m-Knoten 16106 unterhält sein eigenes Bloomfilter 16108, das mit den Inhalten von Diensten, Smart Contracts und Dateien, die innerhalb Knoten innerhalb seines lokalen k-Buckets 16110 gehalten werden, belegt ist, oder wie durch den Vorgang des zugrunde liegenden DHT-Protokolls vorgeschrieben wird. Falls das Ergebnis negativ ist, ist die Suche fehlgeschlagen und kann der c-Knoten 16102 unter Verwendung jeglicher von einer Anzahl von Kriterien entscheiden, ob die Suche zu beenden oder fortzusetzen ist. Falls das Ergebnis in der Fortsetzung besteht, kann eine umfangreichere Suche nach den Inhalten des Netzes ausgeführt werden.

[0889] Wie hier verwendet, ist ein Bloomfilter eine probabilistische Datenstruktur in der Art einer Speicherstruktur, die eine Anzahl von Bits aufweist, welche verwendet werden kann, um zu testen, ob ein Element ein Mitglied einer Menge ist. Eine Anfrage kann eines von zwei verschiedenen Ergebnissen, möglicherweise in einer Menge oder nicht in der Menge, zurückgeben. Jedes Element oder Ergebnis im Bloomfilter ist eine zur Belegung einer Bitmenge im Filter verwendete Hash-Funktion. Falls ein für die Suche verwendeter Hash mit allen Bits in einem Bloomfilter übereinstimmt, kann das gewünschte Ergebnis in den zugeordneten K-Bucket aufgenommen werden. Falls dagegen irgendwelche der Bits nicht übereinstimmen, ist das gewünschte Ergeb-

nis nicht in diesem k-Bucket. Falls ein mögliches positives Ergebnis zurückgegeben wird, kann eine weitere Suche nach Hash-Codes in der DHT von diesem K-Bucket zugeordneten Knoten ausgeführt werden, um festzustellen, ob das gewünschte Ergebnis vorhanden ist.

[0890] Um fortzusetzen, sei bemerkt, dass der m-Knoten **16106** die Suche **16104** zu einem anderen Knoten **16112** überträgt. Dieser Knoten **16112** kann ein mit den Inhalten von Knoten in seinem lokalen K-Bucket **16116** belegtes Bloomfilter **16114** aufweisen. Falls diese Suche nicht erfolgreich ist, hat der c-Knoten **16102** die Wahl, die Suche fortzusetzen. Falls der c-Knoten **16102** wählt, die Suche fortzusetzen **16104**, kann sie zu einem anderen Knoten **16118** übertragen werden. Auch dieser Knoten **16118** weist ein Bloomfilter **16120**, das die Inhalte der Knoten in seinem K-Bucket **16122** auflistet, auf. In diesem Fall lokalisiert die Suche **16104** einen Zieldienst erfolgreich in drei Hops. Beispielhafte Kriterien für die Fortsetzung einer Suche weisen einen Ausgleich zwischen der Kritikalität, eine Übereinstimmung zu finden, und den zusätzlichen Kosten einer weiteren Suche für das Netz auf.

[0891] Fig. 162 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **16200** zur Ressourcenentdeckung unter Verwendung verteilter Hashtags (DHT) gemäß einigen Ausführungsformen. Das Verfahren **16200** aus Fig. 162 kann durch die mit Bezug auf Fig. 163 beschriebene IoT-Vorrichtung **16300** implementiert werden. Wie hier beschrieben, kann eine dezentralisierte Blockchain die meisten ihrer Daten in einer zugeordneten DHT speichern und dadurch die Größe der Blockchain-Datenbank (DB) verringern. Ressourcen in der Art von Smart Contracts oder sogar Dateien können sich in der DHT befinden. Die Vorsehung der Ressourcen kann durch das Vorhandensein des Blockchain-Datensatzes und seines zugeordneten Hashes gestützt werden.

[0892] Eine Zollzahlung kann Nachschlagevorgängen zugeordnet werden, um Clients zu ermutigen, Suchergebnisse zu akzeptieren, die ein nicht ganz optimales Ergebnis aufweisen können, solange ein Ergebnis die Anforderungen erfüllen kann. Die Zollzahlung ist eine Gebühr für das Erweitern der Suche über Vorrichtungen durch weitere Hops zwischen Vorrichtungen und Netzen. Sie kann abgerechnet werden, falls eine Suche in einem ersten k-Bucket nicht erfolgreich ist und der Client fordert, dass die Suche erweitert wird. Dies kann gegenüber der Ausführung einer umfangreichen Suche über das Netz, um ein etwas besseres Ergebnis zu erhalten, Kosten einsparen.

[0893] Block **16202** repräsentiert beispielsweise den Fall, in dem das Netz eingeschaltet wird oder neue Vorrichtungen zu ihm hinzugefügt werden. In Block **16204** wird das Netz konfiguriert. Sowohl die Blockchain als auch die DHT können konfiguriert werden müssen. Blockchain-Einstellungen können eine Auswahl des Konsensalgorithmus, ein Belohnungsniveau für Miner oder Validierer, die erfolgreiche Blöcke vorschlagen, das Schwierigkeitsniveau des Algorithmus, wie häufig die Belohnungsniveaus angepasst werden, und dergleichen einschließen. Wie hier verwendet, sind die Miner oder Validierer Vorrichtungen, die Vorrichtungen identifizieren, die in der Lage sein können, einen Dienst oder eine Funktion durch Zugreifen auf Blockchains und DHT, um ein wahrscheinliches Ziel zu lokalisieren, bereitzustellen. Das Schwierigkeitsniveau des Algorithmus kann die Anzahl der für eine erfolgreiche Suche in Übereinstimmung zu bringenden Suchbegriffe angeben. Das Belohnungsniveau kann als die an einen Miner oder Validierer zur Ausführung einer erfolgreichen Suche gemachte Zahlung betrachtet werden. Es kann auf der Komplexität der Suche, der Anzahl der Hops zum Erreichen eines Ergebnisses und dergleichen beruhen.

[0894] In Block **16206** wird die DHT initialisiert. Die DHT wird instanziiert und beginnt ihren Betrieb. Dem DHT-Eigentümer steht es frei, jegliche existierende DHT zu verwenden oder sein eigenes spezialisiertes Protokoll zu spezifizieren oder zu implementieren, welches ferner mit der Blockchain integriert werden kann oder seine eigenen differenzierenden Merkmale ermöglichen kann. Die DHT kann nicht typische Einstellungen aufweisen, beispielsweise wie viele Repliken eines Datenbestands innerhalb des Netzes gehalten werden sollten. In einer DHT können Dateien ablaufen, beispielsweise wenn der letzte jeglicher der Peers in einem Schwarm oder der Tracker-Knoten nicht mehr verfügbar ist. Wie hier beschrieben, kann eine Blockchain-bewusste DHT Repliken von Dateien automatisch innerhalb des Netzes halten. Daten können unbegrenzt bewahrt werden, falls der Eigentümer dieser Daten keine Bedingungen in Bezug darauf spezifiziert hat, wie die Daten entfernt oder gelöscht werden können. Andernfalls können Daten eine feste Lebensdauer (TTL) aufweisen oder kann der Eigentümer und spezifizierte Befugte sie entfernen.

[0895] In Block **16208** werden die zunächst leere Blockchain-Datenbank (DB) und der Geneseblock erzeugt. Nicht alle Daten müssen in der Blockchain gespeichert werden, weil in der Blockchain gespeicherte Daten auf andere Orte zeigen können. Der Blockchain-Entwickler kann durch die spezifischen implementierten Protokolle spezifizieren, welche Felder oder Parameter in Datensätze aufgenommen werden, die zur Blockchain hinzugefügt werden. Die Parteien, welche die Blockchain erzeugen oder aufrechterhalten, können diese Ent-

scheidung zu Anwendungsentwicklern delegieren, welche wählen können, bestimmte Daten abhängig von den Regeln, welche ihnen von den Blockchain-Entwicklern erlaubt wurden, in der Blockchain zu speichern. Bei diesem Beispiel können außerhalb der Blockchain gespeicherte Daten in der DHT gespeichert werden. Zu jeder beliebigen Zeit im Prozess können sich andere Einheiten dem Netz als Miner oder Validierer anschließen. Die Miner können Daten, die gespeicherte Ressourcen betreffen, zur DHT und zur Blockchain hinzufügen. Die Validierer können bestätigen, dass diese Daten korrekt sind, die DHT speichern und Suchfunktionen ausführen, um Daten zu lokalisieren, die gespeicherte Ressourcen betreffen.

[0896] In Block **16210** wird festgestellt, ob es neue Teilnehmer in der Art von Minern oder Validierern gibt, die dem Netz beitreten. Falls dies der Fall ist, kann in Block **16212** ein neu beigetretener Miner oder Validierer die Blockchain-Datenbank und die Partitions-DHT kopieren. Der Prozessfluss kehrt dann zu Block **16210** zurück, um festzustellen, ob weitere Miner oder Validierer dem Netz beitreten möchten. Jeder Knoten kann sowohl ein Miner als auch ein Validierer sein, falls dies durch das eingesetzte Blockchain-Protokoll erlaubt wird. Ferner kann jeder Knoten ein Blockchain-Speicher- oder DHT-Knoten sein. Falls neue Knoten, die dem Netz beitreten, am DHT-Netz teilnehmen, kann dies zur Neupartitionierung der DHT gemäß den Protokollregeln der DHT führen.

[0897] In Block **16214** kann der Inhalt der Blockchain-Datenbank und der Partitions-DHT erzeugt werden. Die Inhaltserzeugung kann ein zwei Schritte aufweisender Prozess sein, bei dem die Daten in der DHT gespeichert, repliziert und verteilt werden und ein auf den Datenort zeigender Datensatz in der Blockchain gespeichert wird. Die Inhaltserzeugung kann zusätzliche Schritte für spezifische Inhaltstypen aufweisen, wie unter anderem das Bestimmen von Inhaltsquellen und das Zugreifen auf diese. Ein Aspekt dieses Ansatzes gegenüber traditionellen Blockchains besteht darin, dass nicht alle Miner oder Validierer im Netz Datensätze all derselben Daten halten müssen. Bei Ethereum werden Smart Contracts beispielsweise in der Chain gespeichert, was bedeutet, dass jeder Validierer im Netz eine Duplikatkopie hat. Bei diesem Beispiel werden, falls die DHT spezifiziert, drei Repliken jedes Datenobjekts zu halten, Dateifragmente für drei Kopien über an der DHT teilnehmende Knoten verteilt und braucht nicht jeder Knoten eine vollständige Kopie jeder Datei zu haben.

[0898] in Block **16216** wird eine einheitliche Ressourcenkennung bzw. Uniform Resource Identifier (URI) für den erzeugten Inhalt erzeugt. Die URI kann traditionelle Ansätze verwenden, wobei ein Tracker beteiligt ist. Dies würde jedoch eine zentralisierte Komponente einführen. Dementsprechend kann die URI, um die Daten über das Netz verteilt zu halten, eine Magnetverbindung oder einen ähnlichen inhaltsbasierten Netzansatz an Stelle ortsbasierter Kennungen für Daten verwenden.

[0899] In Block **16218** speichert der Inhaltserzeuger die URI, jegliche zusätzliche vom Blockchain-Protokoll vorgeschriebene Metadaten und einen Hash der Inhalte des Objekts, wie in der DHT gespeichert. Der in der Blockchain gespeicherte Hash gewährleistet die Vorsehung des Datenobjekts und kann verwendet werden, um zu beweisen, dass seine Inhalte nicht geändert wurden. Ferner kann das Speichern des Hashes in einer Blockchain verwendet werden, um zu bestätigen, dass er an einem bestimmten Datum existiert hat, von einer spezifischen Identität erzeugt oder besessen wurde und dergleichen.

[0900] Die Metadaten können verwendet werden, um zu steuern, was Inhaltserzeuger machen dürfen. Beispielsweise kann ein Eigentümer eines Smart Contracts ein Computerprogramm erzeugen und es in der Kette „verwaisen“, so dass seine Ausführung nicht nachfolgend beendet werden kann. Dementsprechend kann der Blockchain-Protokolleigentümer sorgfältig überlegen müssen, welche Funktionen in der Blockchain er aktiviert. Falls aktiviert, können Daten für immer in der DHT verbleiben, ohne während der Lebensdauer der Blockchain gelöscht zu werden, und können Rechte auf die Daten nicht delegiert werden. Die Daten in der DHT können unter Verwendung eines beliebigen Verfahrens, das dem Datenerzeuger zur Verwendung zur Verfügung steht, verschlüsselt werden.

[0901] In Block **16220** wird festgestellt, ob mehr Inhalt zu erzeugen ist. Die Inhaltserzeugung kann eine kontinuierliche Schleife im Verfahren **16200** sein, die parallel mit anderen Aktivitäten abläuft, solange sowohl die DHT als auch die Blockchain weiter existieren.

[0902] In Block **16222** kann Inhalt in der Blockchain oder der DHT oder in beiden identifiziert werden. Das Nachschlagen des Inhalts kann beginnen, indem ein in der Blockchain gespeichertes Bloomfilter geprüft wird, um festzustellen, ob es eine Bitübereinstimmung zwischen dem Hash des Suchziels und dem Bloomfilter gibt. Falls dies der Fall ist, kann dies darauf hinweisen, dass der Inhalt im dem Bloomfilter zugeordneten K-Bucket vorhanden sein kann. Das Nachschlagen des Inhalts kann dann die DHT untersuchen, um festzustellen, ob der Inhalt vorhanden ist.

[0903] Das Nachschlagen des Inhalts wird ausgeführt, um Daten zu finden, die nicht mehr in der Chain gespeichert sind und für jeden Knoten im Netz repliziert sind. Die Daten können in einem globalen Netz oder einer Datenebene gespeichert sein, wodurch eine umfangreiche Durchsuchung des gesamten Netzes möglicherweise problematisch wird. Die DHT implementiert ihren eigenen Abstandsmetrikalgorithmus zur Bildung der k-Buckets. Der Abstand kann nicht notwendigerweise repräsentativ für geographische Abstände sein, sondern beispielsweise auf der Anzahl der Hops zwischen teilnehmenden Knoten, der Latenz über diese Hops und dergleichen beruhen.

[0904] Hier offenbarte Aspekte können es ermöglichen, „ausreichend gute“ Suchergebnisse zu finden, ohne notwendigerweise das beste globale Ergebnis finden zu müssen. Wie nachstehend erörtert wird, kann auf der Grundlage eines alternativen Konzepts eines „Zolls“ eine Abschreckung gegen eine umfangreiche Suche eingeführt werden. Wie zuvor erklärt wurde, kann ein Client innerhalb des Netzes nach Knoten suchen oder diese entdecken möchten, welche bestimmte Dienste oder Daten bieten oder verbrauchen. Ein typisches Szenario könnte ein dezentralisierter globaler Marktplatz für IoT-Daten sein, worin Datenlieferanten und Datenverbraucher einander finden und kontaktieren möchten, ohne dass jedoch zentralisierte Marktplatzträger (beispielsweise eBay oder Amazon) erforderlich wären.

[0905] In Block **16222** überträgt ein Client den Hash-Code des Suchziels zu seinen n nächsten Mining- oder Validierungsknoten. Die Knoten werden durch den „DHT“-Abstandsalgorithmus als dem anderen Knoten „nahe liegend“ definiert. Die Knoten, die als nahe liegend definiert werden, können als die Knoten innerhalb des K-Buckets angesehen werden. Ferner weisen die nächstgelegenen Mining- oder Validierungsknoten ein erhebliches Maß an Informationen über die innerhalb ihres Gebiets gespeicherten Ressourcen auf. Sie implementieren Bloomfilter als Optimierung, so dass auf die Übertragung des Clients beispielsweise schnell geantwortet werden kann, falls ein negatives Ergebnis vorliegt, oder eine umfangreichere Suche verwendet werden kann, falls ein positives Ergebnis vorliegt.

[0906] In Block **16226** wird festgestellt, ob die Suche erfolgreich war. Die Suche ist erfolgreich, falls sie ein oder mehrere Ergebnisse zurückgibt. Die „Zoll“-Kosten für eine lokale Übertragung sind entweder minimal oder 0, wobei dies von der Präferenz der Blockchain- oder DHT-Protokollentwickler abhängt, Clients zu ermutigen, Ergebnisse zu akzeptieren, wenn dies möglich ist. Falls die Suche einen Datenlieferanten identifiziert, muss der Client entscheiden, ob die Ergebnisse zufrieden stellend sind, wobei der Suchprozess in diesem Fall endet, oder ob er weiter suchen möchte. Das Netz wendet Algorithmen zur Berechnung von „Zoll“-Kosten für breitere Suchen an, falls der Client fortfahren möchten sollte.

[0907] In Block **16230** wird festgestellt, ob der Client den Zoll bezahlen kann. Bei einigen Beispielen kann der Client wählen, die Kriterien der Suche zu ändern und eine andere lokale Suche auszuführen, statt mit der kostspieligeren Suche fortzufahren. Die Zolllasten können auf viele Arten bezahlt werden, beispielsweise durch eine Gebühr in der von der Blockchain verwendeten Kryptowährung. Die Zolllasten können als Dienstgebühr bezahlt werden, die zu einer Contract-Ausführung hinzugefügt wird, wobei Lieferanten und Erzeuger festgestellt haben, dass eine erfolgreiche Contract-Ausführung vorgelegen hat. Die Zolllasten können Teil der Gebührenberechnung des Internetdienstanbieters sein.

[0908] Falls die Gesamtkosten in Block **16230** bezahlt werden, wird in Block **16232** die Übertragung auf benachbarte K-Buckets ausgedehnt. Der Prozessfluss kehrt dann zu Block **16226** zurück, um festzustellen, ob die Suche erfolgreich war. Falls dies der Fall ist oder falls der Zoll in Block **16230** nicht bezahlt wurde, endet die Suche in Block **16228**.

[0909] **Fig. 163** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **16300** zur Zusammensetzung mehrerer fraktioneller Schlüssel aus verschiedenen Knoten in einem IP-Mesh-Netz **812** zu einem einzigen vollständigen Schlüssel vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **16300** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[0910] Der Massenspeicher **800** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Koalitionsgruppenbildung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[0911] Der Massenspeicher **808** kann einen Kommunikator **16302** aufweisen, der Pakete über eine oder mehrere Kommunikationsstrecken, unter anderem beispielsweise durch einen Mesh-Sendeempfänger **810**, einen Uplink-Sendeempfänger **814** und eine NIC **816**, zu Mesh-Vorrichtungen **812** oder Vorrichtungen in der Cloud **302** sendet und von diesen empfängt. Wie mit Bezug auf **Fig. 45** beschrieben, kann der Kommunikator **16304** zusätzlich zu den mit Bezug auf **Fig. 162** beschriebenen auch andere Funktionen ausführen, wie die Übersetzung von Paketen zwischen Protokollen, das Ausführen von Herkunftsnachweishinzufügungen und dergleichen. Ferner kann der Kommunikator **16304** Teil eines mit Bezug auf **Fig. 33** beschriebenen Dienstbarkeitsystems sein.

[0912] Ein Bloomfilter **16304** kann in den Massenspeicher **808** aufgenommen sein, um eine Bitsequenz, die überlappende Hash-Werte für Elemente in der Art von Ressourcen oder Smart Contracts enthält, in einem zugeordneten K-Bucket zu halten. Der K-Bucket kann Informationen für eine Anzahl verschiedener IoT-Vorrichtungen aufweisen, wobei eine IoT-Vorrichtung in der Lage ist, Ressourcen, Dienste oder Smart Contracts bereitzustellen. Das Bloomfilter **16304** kann auch mit Einträgen in einer DHT-Datenbank assoziiert sein.

[0913] Eine Blockchain-Logik **16306** kann verwendet werden, um Einträge in einer Blockchain in der Art der URI, zusätzlicher vom Blockchain-Protokoll vorgeschriebener Metadaten und eines Hashes der Inhalte des Objekts, wie in der DHT gespeichert, zu erzeugen. Der Inhaltserzeuger **16308** kann aufgenommen sein, um den Inhalt für das Bloomfilter und für die Blockchain in der Art der URI, der Metadaten und der Hash-Codes zu erzeugen. Ein Suchmanager **16312** kann die Suche nach Werten steuern, beispielsweise Ergebnisse von Suchen akzeptieren, die zu möglicherweise positiven Ergebnissen geführt haben, oder feststellen, ob eine weitere Suche erforderlich ist. Der Suchmanager **16312** kann jegliche für weitere Hops bei der Suche erforderliche Zölle bezahlen.

[0914] **Fig. 164** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **16400**, das Code aufweist, um einen Prozessor **902** anzuweisen, ein Bloomfilter-Hop-Verfahren zur Ressourcenentdeckung zu verwenden, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **16400** zugreifen. Der Prozessor **902** und der Bus **904** können wie mit Bezug auf **Fig. 9** beschrieben sein. Das nichtflüchtige maschinenlesbare Medium **16400** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[0915] Das nichtflüchtige maschinenlesbare Medium **16400** kann einen Code **16402** aufweisen, um den Prozessor **902** anzuweisen, eine Blockchain-Datenbank zu erzeugen. Ein Code **16404** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Blockchain-Inhalt zu erzeugen. Ein Code **16406** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Blockchain-Inhalt in der Blockchain-DB zu speichern. Ein Code **16408** kann aufgenommen sein, um den Prozessor **902** anzuweisen, nach Inhalt zu suchen. Ein Code **16410** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Anfrage zu Vorrichtungen innerhalb eines Hops zu übertragen. Ein Code **16412** kann aufgenommen sein, um den Prozessor **902** anzuweisen, festzustellen, ob die Suche erfolgreich war. Ein Code **16414** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Zoll für weitere Such-Hops zu zahlen.

[0916] Vorrichtungen können Peer-Vorrichtungen verwenden, um eine komplexe Aufgabe, einschließlich beispielsweise eines Datenaustausches, eines Zugriffs auf Instrumente über mehrere Architekturen und einer Parallelverarbeitung zusammenarbeitend zusammenzustellen. Bei einem Beispiel kann eine Vorrichtung zur Zusammenstellung einer komplexen Vorrichtung über mehrere Vorrichtungen mögliche Peers identifizieren. Sobald die möglichen Peers identifiziert wurden, kann eine Vorrichtung eine digitale Erlaubnisanleitung zur Verwendung zwischen den Peers codieren. Die Erlaubnisanleitung kann ein Satz von Richtlinien oder Regeln sein, welche festlegen, welche Dienste oder Funktionen eine Peer-Vorrichtung verwenden darf, auf welche sie zugreifen darf oder welche sie anderen Peers bereitstellen darf. Als Teil der Erlaubnisanleitung kann die Vorrichtung die Peers auffordern, sich automatisch in Dienst zu stellen, um Teilaufgaben der komplexen Aufgabe auszuführen und eine Signatur von einem oder mehreren Peers und jeglichen Benutzern in Zusammenhang mit Peer-Vorrichtungen zu erhalten, wie in der Erlaubnisanleitung oder der Aufgabe umrissen sein kann. Bei einem Beispiel kann die Vorrichtung ansprechend darauf, dass sie erkennt, dass alle Parteien die Erlaubnisanleitung signiert haben, dann ein Signal für den zu aktivierenden Gegenstand der Erlaubnisanleitung bereitstellen. Die in der Erlaubnisanleitung umrissenen Aktionen können durch eine Blockchain ausgeführt werden. Bei einem Beispiel kann ein Wert oder Guthaben zu festgelegten Parteien übertragen werden, wie in der Erlaubnisanleitung der Vorrichtung umrissen und vereinbart ist.

[0917] Die Verwendung der Erlaubnisanleitung und die Verwendung zusammenarbeitender Vorrichtungen können auch bei der Bildung und Steuerung von Ad-hoc-Netzen verwendet werden. Die Steuerung eines Ad-hoc-Netzes durch diese Erlaubnisanleitungen kann zeitlich begrenzt sein oder auf in der Erlaubnisanleitung umrissenen Zeitangaben beruhen. Bei diesem Konzept können Erlaubnisanleitungen entweder von Personen oder von autonom arbeitenden Maschinen erzeugt werden.

[0918] **Fig. 165** ist ein schematisches Diagramm eines beispielhaften Verfahrens **16500** zur Aufgabendefinition und -kommissionierung gemäß einigen Ausführungsformen. Das Verfahren **16500** aus **Fig. 165** kann durch die mit Bezug auf **Fig. 167** beschriebene IoT-Vorrichtung **16700** implementiert werden. Das dargestellte Schema kann eine Aufgabendefinition und -kommissionierung für die Ad-hoc-Erlaubnisanleitung und Erlaubnisanleitungsfunktionen **16502** repräsentieren. Ein Prozess zur Interaktion kann jedoch bei **16504** beginnen.

[0919] In Block **16504** kann eine Vorrichtung die Peers identifizieren, die sie verwendet, um eine Aufgabe auszuführen. Wenngleich Vorrichtungen diese Entdeckung ausführen können, kann sich der Begriff Vorrichtung in diesem Zusammenhang auch auf Agenten oder Dienste beziehen, die durch eine einzige Vorrichtung oder eine Anzahl von Vorrichtungen wirken. Die Entdeckung von Peers und ihrer Fähigkeiten in Block **16504** kann durch eine Entdeckungsprozedur der Vorrichtung, das System der Anforderung, ein definiertes Protokoll oder durch ein Bloomfilter-Hop-Verfahren zur Ressourcenentdeckung, wie vorstehend beschrieben, erfolgen.

[0920] In Block **16506** kann eine Vorrichtung eine Erlaubnisanleitung und Erlaubnisanleitungsfunktionen **16502** erzeugen. Die Erlaubnisanleitung und die Funktionen können maschinenlesbar sein. Die Erlaubnisanleitung kann in einer Blockchain oder außerhalb einer Blockchain gespeichert sein. Bei einem Beispiel kann die Erlaubnisanleitung entdeckbar sein und den von der Vorrichtung entdeckten Peers angekündigt werden. In Block **16506** kann die Vorrichtung eine auszuführende Funktion zu diskreten Funktionen zusammensetzen, die in eine Erlaubnisanleitung zu schreiben sind. Bei einem Beispiel kann die Funktion eine feste Funktion, eine Funktion für allgemeine Zwecke oder eine Funktion aus spezialisierten Codesegmenten sein. Die Funktionen können von menschlichen Entwicklern, Verfahren der künstlichen Intelligenz (AI) zur Erzeugung von Code oder durch eine Kombination von diesen erzeugt werden. Bei einem Beispiel können die Funktionen durch genetische Algorithmen erzeugt werden.

[0921] In Block **16508** kann eine Erlaubnisanleitung durch die Vorrichtung, Peers oder eine andere Partei in einem Ad-hoc-Netz der Vorrichtungen und Peers verhandelt oder editiert werden. Viele verschiedene Aspekte der Erlaubnisanleitung können editiert werden. Beispielsweise kann die Erlaubnisanleitung ein vorstehend beschriebenes Format aufweisen, das Verfahren zum Annehmen und Verlassen der Erlaubnisanleitung enthält. Als Teil der Verhandlung der Erlaubnisanleitung können Editierungen vorgenommen werden, nachdem die Erlaubnisanleitung Attribute und Funktionen der Erlaubnisanleitung angekündigt hat. Ansprechend auf die Ankündigung von Attributen oder Funktionen können die Peers der Vorrichtung vereinbaren, diese Attribute oder Funktionen zuzuführen, indem sie der Erlaubnisanleitung zustimmen oder in sie einfügen oder sie editieren. Bei einem Beispiel kann die Vorrichtung durch die Erlaubnisanleitung die Erzeugung von Tokens fordern, falls eine Autorisierung durch die Vorrichtung oder einen Peer in einem Versuch bereitgestellt wird, auf Dienste unter den Peer-Ressourcen und anderen Funktionen zuzugreifen. Bei einem Beispiel kann die Erlaubnisanleitung Funktionen mit Grenzen aufweisen, die zusätzliche Informationen, einschließlich Zeitbegrenzungen, Dienstqualität oder Datenqualität, aufweisen. Bei einem Beispiel kann die Erlaubnisanleitung andere Bedingungen aufweisen, die ein Eigentümer der Erlaubnisanleitung von teilnehmenden Peers fordern kann. Die Erlaubnisanleitung kann eine begrenzte Verwendung von Quell-Peers umreißen. Bei einem Beispiel kann sich die Erlaubnisanleitung bewegen, um eine Mehrfachverpachtung zu ermöglichen.

[0922] Wie vorstehend erörtert wurde, können Begriffe durch Peers verhandelt werden. Beispielsweise können ein Datenverbraucher und ein Datenanbieter einen Mechanismus zur Verhandlung von Bedingungen vor der Eingabe in die Erlaubnisanleitung aufweisen. Bei einem Beispiel können die Parteien Bedingungen und Raten ankündigen. Bei einem Beispiel können die Bedingungen und die Rate verhandelbar sein. Auf diese Weise können die Entitäten, die an der Erlaubnisanleitung teilnehmen, eine Position behalten, um zu gewährleisten, dass sie nicht an eine nicht profitable Erlaubnisanleitung gebunden werden. Beispiele dieser Bedingungen können minimale Subskriptionsraten und -zeiträume, die Datenlieferanten möglicherweise auferlegen möchten, einschließen.

[0923] In Block **16510** kann die Erlaubnisanleitung ausgeführt werden. Die Ausführung einer Erlaubnisanleitung kann unbegrenzt ablaufen. Bei einem Beispiel kann die Ausführung der Erlaubnisanleitung während einer festen und spezifizierten Zeit stattfinden. Ansprechend auf das Fehlen von Kommunikationen mit Diensteanbietern oder Daten, die Peers mit einer Erlaubnisanleitung versehen, kann die Erlaubnisanleitung beendet wer-

den. Ähnlich können neue Peers Funktionen der Erlaubnisanleitung übernehmen, falls sie die Funktionsweise der Vorrichtung oder des Dienstes verbessern. Die Verbesserung der Erlaubnisanleitungsfunktion kann die Ausführung in der Erlaubnisanleitung verwendeter Dienste mit niedrigeren Raten, einer höheren Datenqualität oder anderen messbaren Metriken einschließen. Bei einem Beispiel kann eine Liste von Mechanismen für die Ausführung während der Erlaubnisanleitungsausführung in einer Erlaubnisanleitung aufgezeichnet werden, bevor die Erlaubnisanleitung beginnt.

[0924] In Block **16512** kann die Ausführung der Erlaubnisanleitung überwacht werden. Bei der Überwachung der Ausführung der Erlaubnisanleitung kann nach neuen Peers und neuen Knoten gesucht werden. In Block **16514** kann eine Zahlung zwischen teilnehmenden Parteien ansprechend darauf, dass eine vereinbarte Bedingung der Erlaubnisanleitung erfüllt ist, geschehen. Bei einem Beispiel kann die Bezahlung in der Erlaubnisanleitung spezifiziert werden. In Block **16516** kann die Erlaubnisanleitung beendet werden, sobald ihr Zeitraum abläuft. Bei einem Beispiel kann die Erlaubnisanleitung ansprechend darauf beendet werden, dass festgestellt wird, dass irgendwelche der teilnehmenden Parteien die Erlaubnisanleitung verlassen und keine Ersatzparteien lokalisiert werden können. Bei einem Beispiel kann die Erlaubnisanleitung ansprechend darauf beendet werden, dass erkannt wird, dass der Zweck, für den die Erlaubnisanleitung erzeugt wurde, erfüllt ist.

[0925] Innerhalb der Ad-hoc-Erlaubnisanleitung **16502** können die Erlaubnisanleitungsfunktionen beschrieben sein. Beispielsweise kann eine Funktion innerhalb der Ad-hoc-Erlaubnisanleitung **16502** eine Erlaubnisanleitungs-Beitrittsfunktion **16518** aufweisen. Die Erlaubnisanleitungs-Beitrittsfunktion kann so implementiert sein, wie vorstehend beschrieben wurde. Die Ad-hoc-Erlaubnisanleitung **16502** kann auch eine Erlaubnisanleitungs-Verlassensfunktion **16520** aufweisen, wie vorstehend beschrieben. Die Ad-hoc-Erlaubnisanleitung **16502** kann eine Funktion zum Auflisten teilnehmender Vorrichtungen **16522** aufweisen, die anderen vorstehend beschriebenen Vorrichtungsauflistungsfunktionen ähneln kann. Die Ad-hoc-Erlaubnisanleitung **16502** kann eine Vorrichtungsattributlistenfunktion **16524** aufweisen, wie vorstehend beschrieben.

[0926] Bei einem Beispiel kann die Ad-hoc-Erlaubnisanleitung **16502** eine Funktion zum Berücksichtigen von Vereinbarungen und Bedingungen zur Ad-hoc-Erlaubnisanleitung **16502** hinzugefügter Vorrichtungen aufweisen. Die Vorrichtungsvereinbarungs- und -bedingungsauflistungsfunktion **16526** kann es sich der Erlaubnisanleitung anschließenden Vorrichtungen ermöglichen, zu veranlassen, dass ihre Dienstbedingungen als Parameter oder Funktionen in die Ad-hoc-Erlaubnisanleitung **16502** aufgenommen werden. Bei einem Beispiel kann die Vorrichtungsvereinbarungs- und -bedingungsauflistungsfunktion auch eine Funktion zum Erzwingen von Strafen aufweisen, die als Teil der Erlaubnisanleitung vereinbart werden können und die teilnehmenden Parteien der Erlaubnisanleitung auferlegt werden oder von ihnen vereinbart werden.

[0927] Bei einem Beispiel kann die Ad-hoc-Erlaubnisanleitung **16502** eine Funktion zum Berücksichtigen der allgemeinen Dienstqualität(QoS)-Geschäftsbedingungen(T&C)-Liste **16528** aufweisen. In die QoS-T&C-Liste **16528** können Verbraucher- und Dienstdaten von einer Erlaubnisanleitung aufgenommen werden dürfen, um QoS-Regeln über die Anlieferung des Dienstes und der Daten zu fordern. Diese Regeln können beispielsweise die Spezifikation der Datenverfügbarkeit, der Dienstverfügbarkeit, der Häufigkeit ausgelieferter Daten, der Genauigkeit der ausgelieferten Daten und der Granularität der Daten einschließen. Die QoS-T&C-Liste **16528** kann auch eine Regel aufweisen, falls die Daten von einem vertrauenswürdigen Sensor stammen, wobei die Daten von einem vertrauenswürdigen Sensor stammen können, wenn nachgewiesen werden kann, dass die Daten beispielsweise von einer Messung durch einen Sensor stammen, statt ein durch einen Codebestandteil in einem Prozessor erzeugter Wert zu sein. Die Ad-hoc-Erlaubnisanleitung **16502** kann eine Token-Anforderungsfunktion **16530** und die Token-Widerrufsfunktion **16532** aufweisen, wie vorstehend beschrieben.

[0928] Bei einem Beispiel kann die Ad-hoc-Erlaubnisanleitung **16502** eine Funktion zum Berücksichtigen der allgemeinen Zahlungsbedingungen aufweisen. Dementsprechend kann die Ad-hoc-Erlaubnisanleitung **16502** eine Zahlungs-T&C-Funktion **16534** aufweisen, um Ereignisse zu zeigen, die Zahlungen zwischen den Parteien auslösen. Bei einem Beispiel können diese Ereignisse, die eine Zahlung zwischen Parteien auslösen, die Erfüllung der Anlieferung eines Dienstes einer Subskription oder die Erfüllung der Anlieferung von Daten bei einer Subskription einschließen. Die T&C-Funktionen **16534** können geschrieben werden, so dass sie innerhalb des Rahmens eines Bezahlung-für-Benutzung-Modells oder eines anderen Modells funktionieren, wobei auch eine Funktion vorhanden sein kann, um einer Partei der Erlaubnisanleitung eine Strafe aufzuerlegen, wenn sie eine zuvor vereinbarte Bedingung nicht erfüllt.

[0929] Bei einem Beispiel kann die Ad-hoc-Erlaubnisanleitung **16502** eine Datenebenenfunktion **16536** aufweisen. Die Datenebenenfunktion **16536** kann es Parteien der Erlaubnisanleitung ermöglichen, zu vereinbaren, wie die Daten oder der Dienst angeliefert und verbraucht werden. Die Datenebenenfunktion **16536** kann

spezifizieren, dass Daten in einem Mechanismus außerhalb der Chain geteilt werden können, und spezifische Endpunkte und Endpunkttechnologien spezifizieren, denen Daten verfügbar gemacht werden können. Bei einem Beispiel können die Daten durch eine Funktion, welche den Endpunkt für eine Quelle subskribiert, oder durch eine Funktion, welche Daten für den Verbrauch veröffentlicht, verfügbar gemacht werden. Bei einem Beispiel können die Mittel des Datenverbrauchs und des Dienstverbrauchs durch an der Erlaubnisanleitung **16502** teilnehmende Parteien Authentifizierungs- und Autorisierungsinformationen aufweisen. Parteien, die an der Ad-hoc-Erlaubnisanleitung **16502** teilnehmen, können einen Dienst oder Daten anliefern und spezifizieren, wie die Parteien Verbrauchspräferenzen verfügbar machen können. Parteien, die Daten und Dienste verbrauchen, können auch Präferenzen darüber spezifizieren, wie die verbrauchenden Parteien die Authentifizierung und Autorisierung verbrauchen können.

[0930] Die für die Anlieferungs- und Verbrauchstechnologien gezeigte Überlappung kann es den Parteien ermöglichen, Verfahren zum Teilen von Diensten und Daten zu vereinbaren, ohne dass eine Person beteiligt wäre. Bei einem Beispiel kann ein Protokollumwandlungsvermittler als Partei eingeführt werden, welche sich der Erlaubnisanleitung **16502** anschließen kann, um eine automatisierte Umwandlung oder eine automatisierte Proxy-Verarbeitung des Dienstes und der Daten zum Endpunkttyp oder zum Datenformat, der oder das von den Verbrauchern und verbrauchenden Parteien gewünscht wird, anzubieten.

[0931] Fig. 166 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **16600** zur Protokollumwandlungsvermittlung durch einen Protokollumwandlungsvermittler gemäß einigen Ausführungsformen. Das Verfahren **16600** aus Fig. 166 kann durch die mit Bezug auf Fig. 167 beschriebene IoT-Vorrichtung **16700** implementiert werden. Das Konzept eines Protokollumwandlungsvermittlers kann beispielsweise eine Partei sein, die sich der Erlaubnisanleitung anschließen kann, um eine automatisierte Umwandlung oder eine automatisierte Proxy-Verarbeitung des Dienstes/der Daten zum von den Verbrauchern gewünschten Endpunkttyp oder Datenformat anzubieten. Der Prozessfluss kann in Block **16602** beginnen.

[0932] In Block **16602** können Peers entdeckt werden. Dies kann durch den Protokollumwandlungsvermittler, durch eine Partei oder durch eine Berechnung der Erlaubnisanleitung **16502** geschehen. Bei einem Beispiel kann die Entdeckung von Peers eine Anfangsphase sein oder über den Prozess zur Gewährleistung, dass Peers bekannt sind, wiederholt werden.

[0933] In Block **16604** kann eine Erlaubnisanleitung **16502** zwischen möglichen Teilnehmern entworfen werden. Der Entwurf einer Ad-hoc-Erlaubnisanleitung **16502** kann die Definition einer Aufgabe oder von Aufgaben aufweisen, die während der Phase des Entwurfs der Ad-hoc-Erlaubnisanleitung **16502** auszuführen sind. Bei einem Beispiel kann sich eine Aufgabe auf die Anlieferung eines Dienstes beziehen. Bei einem Beispiel können bei der Anlieferung eines Dienstes Informationen verwendet werden, die von Lieferanten in Bezug auf den Dienst bereitgestellt werden. Lieferanten von Diensten können ihre Dienste durch einen Nachschlagedienst ankündigen. Ein Nachschlagedienst kann zentralisiert oder dezentralisiert sein. Ein Verfahren zum Nachschlagen von Diensten wird hier beschrieben. Bei einem Beispiel kann dieser Entwurf der Ad-hoc-Erlaubnisanleitung **16502** eine Phase von Austausch aufweisen, bei der Peers in der Erlaubnisanleitung **16502** Bereiche für bestimmte Parameter spezifiziert haben können. Parameter können von einer Partei als bevorzugt markiert werden. Parameter können eine geordnete Gewichtung der Präferenz im Vergleich mit Präferenzen anderer Parteien bereitstellen.

[0934] In Block **16604** kann der Erlaubnisanleitung **16502** beigetreten werden. Der Protokollumwandlungsvermittler kann der Erlaubnisanleitung **16502** beitreten. Der Protokollumwandlungsvermittler kann das Beitreten zur Erlaubnisanleitung **16502** durch eine Partei oder mehrere Parteien überwachen. Bei einem Beispiel kann die Erlaubnisanleitung **16502** einen Lebenszeit(TTL)-Parameter aufweisen, der später verwendet werden kann, um festzustellen, ob die Erlaubnisanleitung **16502** endet oder ob die Verbraucher des Dienstes weitermachen wollen und versuchen wollen, alternative Lieferanten zu finden. Von der Erlaubnisanleitung **16502** offengelegte Vorrichtungen können auch eine minimale Anzahl von Parteien aufweisen, um Parameter der Erlaubnisanleitung **16502** zu erfüllen. Bei einem Beispiel können diese aufgelisteten Parameter in Dienstbedingungen, Attributen der teilnehmenden Vorrichtungen, T&C und QoS-Parametern dargelegt sein. Während einer Phase des Beitretens zur Erlaubnisanleitung können Parteien dem Prozess ansprechend auf die Identifikation einer Entität geringerer Kosten zur Ausführung einer Protokollaufgabe beitreten, ihn verlassen oder daraus ausgestoßen werden. Ähnlich können Parteien ansprechend auf die Identifikation einer Entität zur Ausführung einer Aufgabe oder eines Protokolls mit einer Entität mit einem höheren Nettowert beitreten, austreten oder ausgestoßen werden.

[0935] Bei einem Beispiel könnten, falls es drei bestimmte Merkmale und Attribute gibt, deren Vorhandensein von den Aufgabenverbrauchern bevorzugt wird, diese Merkmale und Attribute zu verschiedenen Kosten anfänglich von drei verschiedenen Parteien angeliefert werden. Während dieser Phase kann bei diesem Beispiel, wenn eine einzige Partei identifiziert wird, welche den Dienst zu einem besseren Preis anliefern kann, die Verwendung dieser gefundenen einzelnen Partei die bessere Lösung sein.

[0936] In Block **16608** kann ein Protokollumwandlungsvermittler eine Autokommissionierung des Knoten bereitstellenden Dienstes fordern. Der Knoten bereitstellende Dienst kann sich auf Knoten beziehen, die in der Ad-hoc-Erlaubnisanleitung **16502** dargelegte Dienste bereitstellen. Die Autokommissionierung kann die Installation von Mikrodiensten an IoT-Vorrichtungen im Gebiet einschließen, welche die Funktionalität zur Verarbeitung von Daten und Diensten auf eine durch Aufgabenverbraucher spezifizierte Art enthalten. Bei einem Beispiel kann die Autokommissionierung Aufgaben betreffen, die in einem vernünftigen Zeitraum ohne manuellen Eingriff automatisch oder aus der Ferne ausgeführt werden können. Die Autokommissionierung kann, falls spezifiziert, auch die manuelle Installation von Vorrichtungen im Gebiet einschließen. Die manuelle Installation kann die Installation durch Personen, trainierte Tiere, Drohnen oder Roboter einschließen. Bei einem Beispiel kann eine manuelle Installation in einer Version dieses Prozesses verwendet werden, falls die QoS-Einstellungen, welche die Zeit der Installation durch Lieferanten aufweisen, die Anforderungen der Erlaubnisanleitung **16502** durch die Parteien erfüllen.

[0937] Bei einem Beispiel können Tokens oder Objekte zur Beschreibung von Funktionen, einschließlich Konstanten, Identifizierer, Operatoren, reservierter Wörter und Trenner sowie Präambeln, den Parteien innerhalb der Erlaubnisanleitung **16502** bereitgestellt werden. Eine Präambel kann, wie zuvor beschrieben, eine Konfiguration, Initialisierung und einen Austausch jeglicher Informationen zwischen Peers aufweisen, die verwendet werden können, um fortzufahren. Eine Präambel kann den Ort von Diensten, maschinenlesbare Anwendungsprotokollschnittstellen(API)-Deskriptoren, Zugangsberechtigungsnachweise sowie den Zugang zu Schlüsseln aufweisen. Bei einem Beispiel kann eine nicht erfolgreiche Präambel den Verlust einer kritischen Masse von Lieferanten, den Verlust des Verbrauchers oder einen Ausfall des Prozesses aufweisen. Falls eine Partei ausfällt, kann der Prozess zu einer Skizzierung der Ad-hoc-Erlaubnisanleitung **16502** zurückkehren.

[0938] In Block **16610** beginnt die Ausführung der Erlaubnisanleitung **16502**, falls eine Präambel und Vorgehensschritte vorhanden und erfolgreich sind. Auf der Grundlage der Bedingungen und Parameter der Präambel und der Erlaubnisanleitung **16502** sowie der vereinbarten Bedingungen der Parteien können Zahlungen freigegeben werden, falls Bedingungen erfüllt sind. Bei einem Beispiel können die Bedingungen im Entwurf der Erlaubnisanleitung **16502** ausgetauscht und vereinbart werden.

[0939] In Block **16612** können durch den Protokollumwandlungsvermittler ansprechend auf eine Erkennung, dass ein Peer seine Teilnahme an der Erlaubnisanleitung **16502** beendet, abschließende Zahlungen vorgenommen werden. Falls die Erlaubnisanleitung **16502** weiter mit den existierenden Mitgliedern funktionieren kann, kann sie weiter funktionieren, falls festgestellt wird, dass die TTL nicht verstrichen ist. Falls die TTL jedoch verstreicht, bevor der Prozess abgeschlossen wird, kann die Erlaubnisanleitung **16502** enden. Bei einem Beispiel kann der Prozess zur Phase **16602** der Entdeckung von Peers zurückkehren, falls die Erlaubnisanleitung **16502** nicht in der Lage sein kann, fortzufahren, ohne alternative Lieferanten oder Verbraucher zu finden.

[0940] **Fig. 167** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **16700** vorhanden sein können, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[0941] Wie vorstehend auch mit Bezug auf **Fig. 8** dargestellt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Erlaubnisanleitungsverfasser **16702** zur Skizzierung einer Erlaubnisanleitung **16502** für eine Anzahl entdeckter Peers aufweisen, wobei die Anzahl der entdeckten Peers jeweils einen Parameter hat und wobei eine Bedingung der Erlaubnisanleitung **16502** ansprechend darauf erzeugt werden kann, dass die Bedingung von wenigstens zwei aus der Anzahl entdeckter Peers zugelassen wird. Der Parameter jedes entdeckbaren Peers aus der Anzahl entdeckter Peers kann einen Bereich zulässiger Bedingungen für einen zugeordneten Peer aufweisen. Der Erlaubnisanleitungsverfasser **16702** kann eine Funktion zum Auflisten der Bedingungen und Konditionen der Anzahl der entdeckten Peers aufweisen. Der Erlaubnisanleitungsverfasser **16702** kann beispielsweise eine Auflistung von Dienstqualitätsbedingungen und -konditionen für die Anzahl der entdeckten

Peers aufnehmen. Der Erlaubnisanleitungsverfasser **16702** nimmt eine Auflistung von Datenebenenbedingungen und -konditionen für die Anzahl der entdeckten Peers auf. Bei einem Beispiel kann die Datenebene einen Prozess dafür angeben, wie die Daten von den Peers anzuliefern und zu verbrauchen sind. Die Erlaubnisanleitung **16502** kann auch eine Lebenszeit aufweisen, wie vorstehend beschrieben. Bei einem Beispiel kann die Erlaubnisanleitung **16502** einen Protokollumwandlungsvermittler zur Behandlung des Beitretens eines Peers zur Erlaubnisanleitung **16502** und des Verlassens der Erlaubnisanleitung **16502** durch den Peer aufweisen. Die Erlaubnisanleitung **16502** kann eine Präambel zum Behandeln des Austausches einer Konfiguration zwischen der Anzahl der entdeckten Peers aufweisen.

[0942] Der Massenspeicher **808** kann einen Aktionsausführer **16704** zur Ausführung einer Aktion der Erlaubnisanleitung **16502** ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt wurde, aufweisen. Der Aktionsausführer **16704** kann eine Funktion zur Autokommissionierung eines Dienstes an einen Peer, wodurch der Peer angewiesen wird, Daten zu verarbeiten, aufweisen. Bei einem Beispiel bezieht sich die Bedingung auf den Betrag der zwischen der Anzahl entdeckter Peers auszuführenden Zahlung, und es kann eine abschließende Zahlung zwischen Peers ausgeführt werden, nachdem erkannt wurde, dass einer von der Anzahl entdeckter Peers die Teilnahme an der Erlaubnisanleitung **16502** beendet.

[0943] Fig. 168 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **16800**, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 9 beschrieben wurden.

[0944] Das nichtflüchtige maschinenlesbare Medium **16800** kann einen Code **16802** aufweisen, um den Prozessor **902** anzuweisen, eine Erlaubnisanleitung **16502** für eine Anzahl entdeckter Peers zu verfassen, wobei jeder von der Anzahl der entdeckten Peers einen Parameter aufweisen kann und wobei eine Bedingung der Erlaubnisanleitung **16502** ansprechend darauf erzeugt wird, dass die Bedingung durch wenigstens zwei von der Anzahl entdeckter Peers zugelassen wird. Die Verfassung der Erlaubnisanleitung **16502** kann eine Funktion zum Auflisten der Bedingungen und Konditionen der Anzahl entdeckter Peers aufweisen. Beim Verfassen der Erlaubnisanleitung **16502** kann eine Auflistung der Dienstqualitätsbedingungen und -konditionen der Anzahl der entdeckten Peers aufgenommen werden. Beim Verfassen der Erlaubnisanleitung **16502** kann eine Auflistung von Datenebenenbedingungen und -konditionen für die Anzahl der entdeckten Peers aufgenommen werden. Die Datenebene kann einen Prozess dafür angeben, wie die Daten von den Peers anzuliefern und zu verbrauchen sind. Die Erlaubnisanleitung **16502** kann eine Lebenszeit aufweisen. Die Erlaubnisanleitung **16502** kann einen Protokollumwandlungsvermittler zur Behandlung des Beitretens eines Peers zur Erlaubnisanleitung **16502** und des Verlassens der Erlaubnisanleitung **16502** durch den Peer aufweisen. Die Erlaubnisanleitung **16502** kann eine Präambel zum Behandeln des Austausches einer Konfiguration zwischen der Anzahl der entdeckten Peers aufweisen.

[0945] Das nichtflüchtige maschinenlesbare Medium **16800** kann einen Code **16804** aufweisen, um den Prozessor **902** anzuweisen, eine Aktion der Erlaubnisanleitung **16502** ansprechend darauf auszuführen, dass erkannt wurde, dass Kondition der Bedingung erfüllt wurde. Bei der Ausführung einer Aktion der Erlaubnisanleitung **16502** kann beispielsweise ein Dienst für einen Peer autokommissioniert werden, wodurch der Peer angewiesen wird, Daten zu verarbeiten. Wie hier verwendet, bezieht sich die Bedingung auf den Betrag einer zwischen der Anzahl der entdeckten Peers vorzunehmenden Bezahlung. Bei einem Beispiel kann eine abschließende Zahlung zwischen Peers vorgenommen werden, nachdem erkannt wurde, dass ein Peer von der Anzahl entdeckter Peers die Teilnahme an der Erlaubnisanleitung **16502** beendet.

[0946] Ein schwebender Dienst kann eine Webseite oder ein virtueller Dienst sein, der im Internet herum-schwebt und ein digitales Wallet in Zusammenhang mit dem schwebenden Dienst behandelt und für ein Hosting sowie Aufträge, welche die Software des schwebenden Dienstes verwenden könnten, verhandelt. Der schwebende Dienst kann Software zur Ausführung auf einem Hardwarebereich aufweisen, wobei die Ausführung der Software mit veränderlicher Effizienz teilweise auf der Grundlage der Software des Dienstes und der verwendeten Hardware geschehen kann. Die Ausführung von Aufträgen unter Verwendung der vom Dienst ausgewählten Software und der vom Dienst ausgewählten Hardware kann zu einer Zahlung für den abgeschlossenen Auftrag führen.

[0947] Wie hier verwendet, kann die Zahlung durch eine Provision für Verkäufer, die ein schwebender Dienst verarbeitet, ausgeführt werden. Die Zahlung kann als Kompensation für die auf dem schwebenden Dienst oder durch den Dienst bereitgestellte Werbung ausgeführt werden. Bei einem Beispiel können mehrere Dienste für eine Verwendung bei der Verarbeitung eines Auftrags verglichen werden. Einer Anzahl von Diensten kann jeweils ihr eigenes digitales Wallet zugeordnet werden. Während ein schwebender Dienst für die von ihm ausge-

fürte Arbeit bezahlt werden kann, kann der schwebende Dienst zusätzlich für Zugriffe auf Ressourcen, Software oder Teildienste zahlen, um beispielsweise einen vereinbarten Auftrag auszuführen. Ein schwebender Dienst kann auch eine Funktion abrechnen, wenn ein Wert im digitalen Wallet null ist. Durch das Abrechnen von Funktionen ohne einen Wert kann ein Manager oder Eigentümer schwebender Dienste einen Wert zwischen digitalen Wallets für eine Anzahl von Diensten zuweisen. Ein Manager schwebender Dienste kann die digitalen Wallets so einstellen, dass ein Wert automatisch wiederhergestellt oder abgezogen wird, nachdem erkannt wurde, dass das digitale Wallet einen in einem zugeordneten Wallet festgelegten Wert erreicht. Bei einem Beispiel kann ein schwebender Dienst einen Dienst für das Mining von Bitcoins, Litecoin, Dogecoin, anderen Kryptowährungen, Proteinfaltungsprojektionen und andere Prozessor- und Software-basierte Aufgaben oder dienstzentrische Aufgaben aufweisen, die ein schwebender Dienst ausführen kann, um einen Wert an ein digitales Wallet zurückzugeben. Bei einem Beispiel könnte ein zweckgebundener Computer als Host oder gemieteter Host für einen schwebenden Dienst dienen.

[0948] Fig. 169 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 16900 zur Behandlung eines schwebenden Dienstes und Werts in einem digitalen Wallet gemäß einigen Ausführungsformen. Das Verfahren 16900 aus Fig. 169 kann durch die mit Bezug auf Fig. 172 beschriebene IoT-Vorrichtung 17200 implementiert werden. Das dargestellte Schema kann den Prozess eines Lebenszyklus des schwebenden Dienstes und der verfassten Schwebender-Dienst-Erlaubnisanleitung 16902 repräsentieren. Ein Prozess des Lebenszyklus des schwebenden Dienstes kann in Block 16904 beginnen. Gleich nummerierte Elemente entsprechen jenen, die in Fig. 165 beschrieben sind.

[0949] In Block 16904 kann ein schwebender Dienst Hosts identifizieren, die der Dienst verwenden kann, um eine Aufgabe auszuführen. Die Entdeckung von Hosts und Hostfähigkeiten kann unter Verwendung eines Bloomfilter-Hops ausgeführt werden, wie vorstehend offenbart. In Block 16906 kann der schwebende Dienst eine maschinenlesbare Erlaubnisanleitung erzeugen, die in oder außerhalb einer Blockchain gespeichert werden kann. Bei einem Beispiel kann die Erlaubnisanleitung für identifizierte Peers und Hosts entdeckbar sein. Die Erlaubnisanleitung kann identifizierten Peers und Hosts angekündigt werden oder von Vorrichtungen, die im Netz nicht identifiziert wurden, entdeckbar sein. In Block 16906 kann der schwebende Dienst eine auszuführende Aufgabe zu Funktionen zusammensetzen. Die Funktionen können in eine Erlaubnisanleitung geschrieben werden. Die Aufgabe und die zusammengesetzten Funktionen können in kleinere feste Funktionen mit einem allgemeinen Zweck zerlegt werden. Die Aufgabe und die zusammengesetzten Funktionen können auch in spezialisierte Codesegmente zerlegt werden. Die Aufgaben- und Funktionscodes können beispielsweise durch einen künstliche Intelligenz aufweisenden genetischen Algorithmus erzeugt werden.

[0950] In Block 16908 kann die Erlaubnisanleitung modifiziert werden, so dass sie zu einem vordefinierten Format passt. Ein Beispiel eines Formats für eine Erlaubnisanleitung kann ein Format sein, das es Peers und Hosts erlaubt, der Anleitung und Erzwingung der Erlaubnisanleitung beizutreten und diese zu verlassen. Die Erlaubnisanleitung kann auch eine Liste von Attributen und Funktionen aufweisen, deren Anlieferung die Hosts vereinbaren. Die von den Hosts vereinbarten Funktionen können beispielsweise Netzdienste, einen Lastausgleich, eine Verwendung voll qualifizierter Domainnamen (FQDN), die Verwendung eines Domainnamensystems (DNS) und von Firewalldiensten einschließen. Die Erlaubnisanleitung kann eine Auflistung von Zeitbegrenzungen und Dienstqualitätsbedingungen aufweisen, denen der Eigentümer der Erlaubnisanleitung sowie jegliche beitretende Peers und Hosts folgen sollen. Bei einem Beispiel kann die Erlaubnisanleitung exklusive Hardware eines Hosts durch erlaubte Mehrfachverpachtung oder durch Teilen des direkten Zugriffs auf die Host-Hardware verwenden. Die vorstehend aufgelisteten Parameter und andere Parameter, die von einem schwebenden Dienst verwendet werden können, können dazu führen, dass vom anfordernden schwebenden Dienst eine höhere oder eine niedrigere Gebühr an den oder die Hostanbieter gezahlt wird.

[0951] In Block 16910 kann die Erlaubnisanleitung mit der Ausführung beginnen. Die Ausführung kann auf der Grundlage von Bedingungen, Funktionen und einer Eingabe, die an Vorrichtungen empfangen wird, die von der Erlaubnisanleitung gesteuert werden, beruhen. Wie vorstehend erwähnt, kann die Erlaubnisanleitung eine Ausführung auf der Grundlage einer eingestellten festen Zeit, keiner festen Zeit oder von Bedingungen aufweisen. Bei einem Beispiel der Ausführung der Erlaubnisanleitung kann die Erlaubnisanleitung ansprechend darauf beendet werden, dass erkannt wird, dass ein einen Dienst anbietender Peer oder ein Daten anbietender Peer verschwindet. Bei einem Beispiel kann eine Peer-Vorrichtung oder können Host-Vorrichtungen ersetzt, ausgetauscht oder außer Dienst gestellt werden, falls erkannt wird, dass Peers und Hosts Dienste mit niedrigeren Raten bereitstellen als in der Erlaubnisanleitung vereinbart wurde. Eine Peer-Vorrichtung oder eine Host-Vorrichtung kann auch ansprechend darauf ersetzt, ausgetauscht oder außer Dienst gestellt werden, dass erkannt wird, dass die Datenqualität nicht in der Erlaubnisanleitung vereinbarten Metriken entsprechen kann.

[0952] In Block **16912** können die Dienst-Entität und die hostende Entität eine Funktion zum Austauschen von Bedingungen zwischen Hosts und Peers aufweisen, um wechselseitig vereinbarte Bedingungen für die Auflistung in der Erlaubnisanleitung zu identifizieren. Bedingungen in der Erlaubnisanleitung können die Ausführungspriorität, die Kommunikationsbandbreite, Zugangserlaubnisse und dergleichen aufweisen. In Block **16914** kann eine Zahlung zwischen Peers und Hosts, die der Anweisung durch die Erlaubnisanleitung des schwebenden Dienstes **16902** beigetreten sind, ausgetauscht werden. Die Zahlung kann ausgetauscht werden, nachdem von der Schwebender-Dienst-Erlaubnisanleitung **16902** dargelegte Bedingungen erfüllt wurden. Bei einem Beispiel kann beim Austausch einer Zahlung eine Zahlung vorbereitet werden und die Zahlungsdaten einem Dienst-Wallet **16916** bereitgestellt werden. Die Zahlung kann durch einen existierenden Wert oder durch Gutschrift an einem Dienst-Wallet von einem Peer, Host oder einer anderen Partei, die der Schwebender-Dienst-Erlaubnisanleitung **16902** beigetreten ist, erfolgen. Bei einem Beispiel kann der Guthabenaustausch zwischen zwei Wallets von einem Dienst-Wallet **16916** zu einem Host-Wallet **16918** geschehen. Die Wallets einer Entität können ein logischer Speicher einer numerischen Repräsentation des Werts, Guthabens oder einer Lastschrift sein. Bei einem Beispiel kann ein Peer oder Host durch den Wert in seinem Wallet begrenzt sein. Falls ein Peer, Host oder andere Anbieter Verpflichtungen der Schwebender-Dienst-Erlaubnisanleitung **16902** nicht erfüllt, kann ein Austausch von Wert zwischen dem Dienst-Wallet **16916** und einem Wallet geschädigter Parteien oder einer allgemeinen Werthaltestelle erlauben, dass Strafzahlungen und Werte aus dem Dienst-Wallet **16916** entfernt werden. Ein Beispiel einer Verletzung einer Verpflichtung kann darin bestehen, dass ein Peer oder Host ein vereinbartes Verfügbarkeitsniveau nicht erfüllt. Bei einem Beispiel kann die Funktion eines Hosts, eines Peers oder eines schwebenden Dienstes auf der Grundlage eines im Wallet, das diesem Dienst, diesem Peer oder diesem Host zugeordnet ist, gespeicherten Werts reguliert, gesteuert oder begrenzt werden. Bei einem Beispiel kann, sobald das Guthaben in einem Dienst-Wallet **16918** aufgebraucht sein kann, der Zugriff durch den Peer oder Host in Zusammenhang mit diesem Wallet aus der Erlaubnisanleitung **16902** entfernt werden. Warnschwellen können bereitgestellt werden, um den Eigentümer eines schwebenden Dienstes zu benachrichtigen, wenn der Wert in einem zugeordneten Wallet niedriger oder höher als eine festgelegte Schwelle ist. Warnschwellenwerte können einem automatischen Abschneiden oder Drosseln eines Dienstes auf der Grundlage davon, dass der Wert in einem Wallet eine festgelegte Schwelle erreicht oder überschreitet, zugeordnet werden.

[0953] In Block **16920** kann die Erlaubnisanleitung **16902** beendet werden. Die Beendigung kann ansprechend darauf erfolgen, dass eine Bedingung von einem Peer oder Host erfüllt wird. Die Beendigung der Erlaubnisanleitung **16902** kann ansprechend auf das Verstreichen eines Zeitraums, auf die Anzahl der verlassenden Peers oder die Anzahl der verlassenden Hosts oder den Prozentsatz der verlassenden Peers, den Prozentsatz der verlassenden Hosts, das Fehlen eingehender Peers und Hosts oder auf eine andere manuell festgelegte Richtlinie, die in der Erlaubnisanleitung **16902** vereinbart wurde, geschehen.

[0954] Als eine der Funktionen der Erlaubnisanleitung **16902** stellt eine Hostattributfunktion **16922** eine Auflistung der Fähigkeiten bereit, die ein Host, welcher der Erlaubnisanleitung beigetreten ist, anbieten kann. Bei einem Beispiel können die Fähigkeiten, die ein Host anbieten kann, attestierte Merkmale, vertrauensbasierte Merkmale und Merkmale, die beim Empfang eines Autorisierungsbeweises für den Zugriff auf den Host und auf die Funktion durch die Erlaubnisanleitung **16902** wirken können, aufweisen. Die Verfügbarkeit der Hostattributfunktion **16922** kann begrenzt werden, um die Anlieferung oder den Zugriff auf solche Merkmale zu verringern, um einen Wert der Dienste der Hostattributfunktion zu erhalten. Die Hostattributfunktion **16922** kann mit einer Auflistung von Hostfunktionsbedingungen für den Dienst um die Hostfunktionsaktivitäten und das Hostfunktionsverhalten assoziiert sein. Die Hostattributfunktion **16922** kann den Zugriff auf eine Hostfunktion verweigern oder eine Strafe auferlegen, wenn erkannt wird, dass der schwebende Dienst Bedingungen der Hostattributfunktion **16922** verletzt.

[0955] Eine Liste gehosteter Dienste **16924** und eine entsprechende Liste **16926** über vereinbarte Dienstbedingungen (T&C-Liste) werden kombiniert, um es der Erlaubnisanleitung beitretenden Diensten zu ermöglichen, Bedingungen zu ihren Dienstniveaus anzugeben, die als Parameter oder Funktionen in die Erlaubnisanleitung **16902** aufgenommen sind. Bei einem Beispiel können in der Erlaubnisanleitung **16902** aufgelistete Parameter auf einer Skala bewertet werden, die ihren Prioritätsgrad oder ihr Fehlen von Priorität in Bezug auf den schwebenden Dienst und den Betrieb des schwebenden Dienstes angibt. Die Dienst-T&C-Liste **16926** kann Strafen angeben, die von den Peers und Hosts vereinbart werden können. Diese Strafen können auf einen Peer oder Host angewendet werden, der vereinbarte Bedingungen der Schwebender-Dienst-Erlaubnisanleitung **16902** erreicht.

[0956] Fig. 170 ist ein schematisches Diagramm einer beispielhaften Schwebender-Dienst-Datenstruktur **17000** zur Behandlung eines schwebenden Dienstes **17002** und der Optionen, Konditionen und Bedingungen

gemäß einigen Ausführungsformen. Bei einem Beispiel kann die Schwebender-Dienst-Datenstruktur **17000** Konditionen, Bedingungen und Merkmale des schwebenden Dienstes auf der Grundlage der Priorität der Kondition, der Bedingung und des Merkmals zeigen. Die aufgelisteten Optionen, Konditionen, Bedingungen, Merkmale, Werte und ihre entsprechenden Prioritäten, die in der beispielhaften Schwebender-Dienst-Datenstruktur **17000** dargestellt sind, sind beispielhaft und können in einer Auflistung von Bedingungen und Konditionen einer Schwebender-Dienst-Erlaubnisanleitung **16902** aufgenommen werden.

[0957] Die Schwebender-Dienst-Datenstruktur **17000** kann die berechneten Kosten, bekannten Kosten und unbekannten Kosten bewerten, wenn ein Host gewählt wird. Bei einem Beispiel kann ein schwebender Dienst **17002** die Datenstruktur **17000** verwenden, um eine Kombination identifizierter Kosten mit einer Auflistung von Merkmalen und den identifizierten Merkmalsanforderungen des schwebenden Dienstes und Auftrags zu vergleichen. Bei einem Beispiel kann eine Liste von Merkmalen für einen schwebenden Dienst in eine Entscheidungsmatrix der Datenstruktur **17000** eingefügt werden.

[0958] Eine Entscheidungsmatrix einer Datenstruktur **17000** kann einen Vergleich identifizierter Hosts, Peers und anderer Vorrichtungen oder Ressourcen, die einem schwebenden Dienst **17002** zur Verfügung stehen, aufweisen. Beim bereitgestellten Beispiel zeigt die Datenstruktur **17000** Beispieldaten, die von drei Hosts, nämlich Host 1, Host 2 und Host 3, gesammelt wurden. Bei der beispielhaften Datenstruktur **17000** kann ein schwebender Dienst **17002** auf der Grundlage der Priorität von den Hosts gesammelter Merkmale und Daten feststellen, dass die Hosts 2 und 3 mögliche Hosts für die Ausführung des schwebenden Dienstes sind, während Host 3 zumindest teilweise infolge eines erhöhten Vorhandenseins von Merkmalen mit Priorität in von Host 3 empfangenen Daten einen höheren Rang aufweisen kann. Bei diesem Beispiel zeigt Host 3 höhere nominelle Kosten und empfängt wie dargestellt eine höhere Entscheidungsbewertung oder einen höheren Entscheidungswert, wie in der beispielhaften Schwebender-Dienst-Datenstruktur **17000** dargestellt ist. Der höhere Wert kann das Ergebnis davon sein, dass Host 3 Merkmale mit höherer Wichtigkeitspriorität gegenüber anderen Merkmalen, Optionen, Konditionen und Bedingungen, die betrachtet werden, erfüllt. Die Formel, die diese Entscheidungsbewertung und diesen Entscheidungswert berechnet, kann in einer Anzahl von Weisen berechnet werden, einschließlich eines Berechnungsverfahrens, bei dem die Summe der Kosten eines Hosts pro Stunde durch die Summe der Bewertungen für jedes Merkmal, jede Option, jede Kondition oder jede Bedingung, die für den Vergleich in der Schwebender-Dienst-Datenstruktur **17000** des schwebenden Dienstes **17002** aufgelistet ist, dividiert wird.

[0959] Fig. 171 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **17100** zur Behandlung schwebender Dienste gemäß einigen Ausführungsformen. Das Verfahren **17100** aus Fig. 171 kann durch die mit Bezug auf Fig. 172 beschriebene IoT-Vorrichtung **17200** implementiert werden. Der Prozessfluss kann in Block **17102** beginnen.

[0960] In Block **17102** kann ein schwebender Dienst erzeugt werden. Der schwebende Dienst kann in einem Einkapselungsmodul erzeugt werden, das auf einem breiten Bereich von Hardwaresystemen ausgeführt werden kann. Bei einem Beispiel können die Einkapselungsmodule Container in der Art von Docker-Containern und Virtualisierungskonstrukte einschließlich virtueller Maschinen sein. Bei einem Beispiel kann ein Einkapselungsmodul ein Framework sein, das zur Verpackung und Verteilung von Softwarebinärdateien verwendet werden kann. Der schwebende Dienst kann dann Anforderungen zuweisen, um es dem Eigentümer des schwebenden Dienstes zu erlauben, Prioritäten für den schwebenden Dienst zu spezifizieren. Bei einem Beispiel kann eine Priorität Merkmale oder spezifische Fähigkeiten, einschließlich Hardwareoptionen, einschließen. Hardwaremerkmale können CPU-Kapazitäten und -Fähigkeiten, Kapazitäten und Fähigkeiten des permanenten Speichers und Kapazitäten und Fähigkeiten des flüchtigen Speichers einschließen. Bei einem Beispiel können diese Kapazitäten und Fähigkeiten eine Beurteilung einschließen, ob Hardwarebeschleuniger vorhanden sind. Bei einem Beispiel können, falls Hardwarebeschleuniger vorhanden sind, Hardwareaktivierungsmerkmale bewertet werden, einschließlich Advanced Encryption Standard (AES), SGX, Virtualisierung (VTx) oder Dienste hoher Verfügbarkeit. Ein Eigentümer eines schwebenden Dienstes kann auch Softwareabhängigkeiten als zu bewertende Merkmale spezifizieren. Zu bewertende Softwaremerkmale können beispielsweise den Betriebssystemtyp, die Betriebssystemversion, eine Softwareversion, Patch-Niveaus und das Vorhandensein geschichteter Anwendungen für die Nachrichtenübertragung und Kommunikation einschließen. Während der schwebende Dienst in Block **17102** erzeugt wird, können die Dienstqualität und die Bedingungen und Konditionen des schwebenden Dienstes angehängt werden. Bei einem Beispiel können der Diensteanbieter oder eine angeschlossene Datenquelle den geographischen Ort des schwebenden Dienstes oder einen Exklusivitätsstatus der Hardware angeben. Bei der Erzeugung des schwebenden Dienstes in Block **17102** kann ein Dienst-Wallet angehängt werden. Bei einem Beispiel kann der Eigentümer des schwebenden Dienstes ein neues Wallet erzeugen, das dem schwebenden Dienst zuzuordnen ist. Bei einem Beispiel kann der schwebende

Dienst ein existierendes Wallet zuordnen oder dieses teilen. Wie hier verwendet, können sich Wallets auf einen beliebigen Wertspeicher beziehen und Bitcoin-Wallets, Ethereum-Wallets und Google-Wallets einschließen. Ein schwebender Dienst kann auch spezifische Finanzierungsformen außer einem Wallet in der Art von Zahlungsdiensten, ähnlich und einschließlich PayPal und Visa-Onlinediensten, aufweisen. Bei der Erzeugung des schwebenden Dienstes in Block **17102** können Finanzierungsregeln für den schwebenden Dienst zugewiesen werden. Bei einem Beispiel können die Regeln für den schwebenden Dienst Finanzierungsauslöser aufweisen, die bewirken würden, dass ein Wallet wiederaufgefüllt wird oder nicht wiederaufgefüllt wird. Bei einem Beispiel könnte eine Einstellung ein automatisches Wiederauffüllen oder Aufladen des Wallets mit einem vorgewählten Betrag durch einen Benutzer ansprechend darauf, dass erkannt wird, dass der Guthabenbetrag des Wallets unter eine Schwelle abgefallen ist, einschließen. Der Eigentümer des schwebenden Dienstes kann wählen, eine Regel für den schwebenden Dienst anzugeben, die angibt, dass der schwebende Dienst die Ausführung unterbrechen kann, falls der schwebende Dienst einen Nullwertpunkt in einem zugeordneten Wallet erreicht oder falls eine negative Werterzeugungsrate erkannt wird. Zusätzliche Regeln, die während der Erzeugung eines schwebenden Dienstes in Block **17102** eingeleitet werden, können eine Kombination von Datumsauslösern, Ereignisauslösern und Guthabenbetragsauslösern aufweisen. Der schwebende Dienst kann diese Auslöser als Angaben verwenden, dass eine bestimmte Walletfüllaktion auftreten kann. Bei einem Beispiel kann ein Wallet Guthaben zu einem getrennten Wallet, Konto oder Finanzdienst übertragen, wenn erkannt wird, dass der Guthabenbetrag eine bestimmte Schwelle überschreitet oder einen identifizierten Datumsauslöser oder Ereignisauslöser durchläuft. Eine Übertragung von Guthaben kann eine Übertragung eines spezifizierten Guthabenbetrags, der identifizierten Überschussbeträge oder der Summe der Guthaben im Wallet einschließen. Bei einem Beispiel kann das Wallet ein TTL-Kriterium aufweisen. Bei einem Beispiel kann der Eigentümer des schwebenden Dienstes einen Wert für eine TTL spezifizieren. Eine TTL kann eine Grenze für die Anzahl der auszuführenden Operationen, die Anzahl der Guthabenübertragungen oder die Anzahl der Transaktionen zu einem Wallet einschließen. Bei einem Beispiel kann eine TTL für einen schwebenden Dienst auch automatisch erweitert werden, falls bestimmte Kriterien für Datumsangaben, Aktivitätsniveaus beim Dienst und Kriterien für die Bewegung des schwebenden Dienstes.

[0961] In Block **17104** kann der schwebende Dienst abgesendet werden. Die Absendung des schwebenden Dienstes kann ansprechend auf eine Angabe beginnen, dass die vollständige Konfiguration des schwebenden Dienstes abgeschlossen ist. Die Konfiguration des schwebenden Dienstes wurde vorstehend teilweise mit Bezug auf Block **17102** offenbart. Bei einem Beispiel kann ein Absendemechanismus durch das verwendete Einkapselungsmodul vorgeschrieben werden, wie vorstehend beschrieben. Bei einem Beispiel können, falls der Dienst ein Container ist, existierende Verfahren zur Installation des Containers verwendet werden, sobald ein geeigneter Ziel-Home dafür gefunden wurde. Ansprechend auf die Absendung des schwebenden Dienstes können Hosts entdeckt werden. Bei einem Beispiel kann beim Finden eines Ziel-Hosts zuerst nach Systemen gesucht werden, die Hosting-Dienste anbieten. Ansprechend auf die Absendung des schwebenden Dienstes von Block **17104** können die Contracts spezifiziert werden. Bei einem Beispiel können Systeme, die Dienste anbieten, mehrere Erlaubnisanleitungen anbieten, wobei eine Erlaubnisanleitung verschiedene Kriterien aufweisen kann. Die Erlaubnisanleitungen können spezifiziert werden. Ansprechend auf die Absendung des schwebenden Dienstes von Block **17104** können ein Host und eine Erlaubnisanleitung gewählt werden. Bei einem Beispiel kann das Verfahren zur Auswahl eines bestimmten Hosts und zur Auswahl einer bestimmten Erlaubnisanleitung wie vorstehend erörtert stattfinden.

[0962] Ansprechend auf die Absendung des schwebenden Dienstes von Block **17104** können Bedingungen und Konditionen verhandelt oder ausgetauscht werden, wie nachstehend beschrieben. Bei einem Beispiel können, falls ein Peer, ein Host oder eine andere Partei einen Teil der Erlaubnisanleitung als verhandelbar markiert hat, Bereiche um diese Parameter spezifiziert werden. Andere Richtlinien können implementiert werden, um zu ermöglichen, dass Teile der Erlaubnisanleitung verhandelbar sind, wie unter anderem das Bezahlen einer Gebühr für das Recht. Bei einem Beispiel kann das Hosting zu bestimmten Kosten geteilt werden, und dieses Angebot kann mit einem anderen Angebot in Kontrast stehen, bei dem ein begrenzter Zugriff auf Hardware zu höheren Kosten verfügbar sein kann. Bei einem Beispiel kann ein bestimmter schwebender Dienst Bereiche aufweisen, die der schwebende Dienst für verschiedene Dienstqualitäten zu zahlen berechtigt sein kann. Ansprechend auf eine Erkennung, dass eine begrenzte Hardwareverwendung in einen akzeptierbaren Zahlungsbereich passt, kann der schwebende Dienst wählen, das Angebot für einen begrenzten Zugriff auf die Hardware zu akzeptieren. Ein schwebender Dienst kann die begrenzte Hardwarekonfiguration stattdessen nicht als bevorzugt etikettieren, und der schwebende Dienst kann ansprechend auf diese Etikettierung standardmäßig auf eine Option im Markt zurückfallen, welche die minimale Schwelle des schwebenden Dienstes erfüllt.

[0963] Ansprechend auf die Absendung des schwebenden Dienstes von Block **17104** kann eine Präambel bereitgestellt werden. Wie vorstehend beschrieben, kann die Präambel einen Informationsaustausch aufwei-

sen, der für die Erlaubnisanleitung verwendet werden kann, um mit der Ausführung zu beginnen. Die Präambel kann Wallet-Kennungen, Identitätsinformationen, Zugriffsinformationen, Schlüsselaustausche für den Dienst und die Hardware, Host-Orte, die Host-IP-Adresse oder den Ort, wo der schwebende Dienst verfügbar ist, aufweisen. Ansprechend auf eine Erkennung, dass die Präambel fehlschlägt, kann ein anderer Host mit dem Prozess ausgewählt werden, der mit der Betrachtung und Auswahl des Hosts als Teil von Block **17102** wiederaufgenommen wird. Ansprechend auf die Erkennung eines Präambelfehlers kann eine Benachrichtigung zum Eigentümer des schwebenden Dienstes gesendet werden. Die Benachrichtigung kann eine Anforderung einer Eingabe in Bezug darauf aufweisen, ob der Eigentümer des schwebenden Dienstes das Niveau der Hardware, der Software, der Bedingungen und Konditionen oder der Dienstqualität verringern kann, die angestrebt werden, um auf der Grundlage der Versorgung mit fähigen Hosts auf dem Markt mehr Optionen für den schwebenden Dienst zu öffnen.

[0964] In Block **17106** kann die Erlaubnisanleitung mit der Ausführung beginnen. Bei einem Beispiel kann die Ausführung der Erlaubnisanleitung ansprechend auf den Abschluss der Präambelphase beginnen. Ansprechend auf den Beginn der Ausführung der Erlaubnisanleitung können die Ausführungsbedingungen gemessen werden. Während der Ausführung der Erlaubnisanleitung können Zahlungen freigegeben werden, wenn Ereignisse oder Bedingungen der Erlaubnisanleitung erfüllt sind. Wenngleich eine Partei, die der Erlaubnisanleitung beigetreten ist und ihr zugestimmt hat, sie verlassen kann, kann sie mit einer Strafe belegt werden, bei der ein der Partei zugeordnetes Wallet belastet wird. Bei einem Beispiel kann die Erlaubnisanleitung zumindest teilweise auf der Natur des schwebenden Dienstes beruhen und auf dem Konzept einer Erlaubnisanleitung basieren.

[0965] Bei einem Beispiel könnte der Abrechnungszeitraum der Erlaubnisanleitung so klein wie gewünscht sein, möglicherweise Sekunden oder Mikrosekunden dauern. Bei einem Beispiel kann der Prozess, falls während der Ausführung einer Erlaubnisanleitung ein Host oder ein Peer eine QoS-Bedingung erfüllt, fortgesetzt werden und kann auf andere Bedingungen zugegriffen werden. Ansprechend auf eine Erkennung, dass eine QoS-Bedingung als unzufrieden stellend eingestuft wird, kann die Erlaubnisanleitung beendet werden oder können einem verletzenden Host Strafen auferlegt werden. Bei einem Beispiel kann die Beendigung einer Erlaubnisanleitung eine Entscheidung sein, die von der Erlaubnisanleitung automatisch auf der Grundlage einer durch AI behandelten Implementation getroffen wird. Die Beendigung einer Erlaubnisanleitung kann eine manuell getroffene Entscheidung sein, bei einem Beispiel im Ermessen sowohl des Diensteanbieters als auch des Dienstverbrauchers.

[0966] Ansprechend auf die Ausführung der Erlaubnisanleitung in Block **17106** kann eine Zahlung erreicht werden, wenn Bedingungen und Konditionen der Erlaubnisanleitung Auslöseschwellen erreichen. Die bewerteten Zahlungen und Strafen können mehrdirektional sein, so dass Zahlungen zwischen mehreren Parteien, Peers und Hosts übertragen oder verschoben werden können. Wie vorstehend erwähnt, kann die Erlaubnisanleitung beendet werden, falls eine Partei beendet wird oder austritt.

[0967] In Block **17108** können abschließende Zahlungen ausgetauscht werden. Bei einem Beispiel kann der Prozess beendet oder zurückgesetzt werden, wenn eine Erlaubnisanleitung ein natürliches Ende erreicht. Bei einem Beispiel kann sich ein natürliches Ende auf das Verstreichen einer TTL beziehen. Ansprechend auf eine Erkennung, dass die TTL eines schwebenden Dienstes nicht verstrichen ist, kann der schwebende Dienst einen neuen Zyklus der Entdeckung eines anderen Hosts einleiten.

[0968] **Fig. 172** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 17200 zur Behandlung schwebender Dienste vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[0969] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Schwebender-Dienst-Erlaubnisanleitungsverfasser **17202** aufweisen. Bei einem Beispiel kann der Schwebender-Dienst-Erlaubnisanleitungsverfasser **17202** eine Schwebender-Dienst-Erlaubnisanleitung für eine Anzahl entdeckter Hosts zur Ausführung der Aufgaben eines schwebenden Dienstes verfassen, wobei die Anzahl der entdeckten Hosts auf eine Erfüllung eines in der Schwebender-Dienst-Erlaubnisanleitung spezifizierten Parameters durch die Hosts bewertet werden kann.

[0970] Bei einem Beispiel kann die Schwebender-Dienst-Erlaubnisanleitung Strafen angeben, die ansprechend auf eine erkannte Verletzung der Diensturlaubnisanleitung gegen einen Host zu erwägen sind, wobei die Strafen von einem Host-Wallet einzuziehen sind.

[0971] Der Massenspeicher **808** kann einen Host-Hardwarewähler **17204** aufweisen. Bei einem Beispiel kann der Host-Hardwarewähler **17204** eine Host-Hardware für den schwebenden Dienst auf der Grundlage einer Datenstruktur des schwebenden Dienstes auswählen.

[0972] Bei einem Beispiel ist die Datenstruktur eine Entscheidungsmatrix. Die Entscheidungsmatrix kann ein vom schwebenden Dienst angestrebtes Merkmal, die Anzahl der verfügbaren Hosts und einen Bewertungswert der Hosts in Bezug auf die in der Entscheidungsmatrix aufgelisteten Merkmale auflisten. Der schwebende Dienst kann einen Host auf der Grundlage des besten Werts auswählen, der anhand der Kosten pro Stunde, dividiert durch die Anzahl der Merkmale, berechnet wird, wobei die Qualitätsmetrik eine zufrieden stellende Verwendung des schwebenden Dienstes angibt und die Kosten pro Stunde projizierte Kosten pro Stunde des Betriebs des schwebenden Dienstes unter Verwendung eines bewerteten Hosts sind. Die Merkmale des schwebenden Dienstes können die Merkmale bei einer Wertberechnung unter Verwendung der Entscheidungsmatrix auf verschiedene Weise gewichten.

[0973] Der Massenspeicher **808** kann einen Schwebender-Dienst-Erlaubnisanleitungsausführer **17206** zur Implementation der schwebenden Erlaubnisanleitung für die IoT-Vorrichtung 17200 aufweisen. Bei einem Beispiel kann die Schwebender-Dienst-Erlaubnisanleitung die Host-Hardware verwenden.

[0974] Der Massenspeicher **808** kann einen Wertübertrager **17208** aufweisen. Bei einem Beispiel kann der Wertübertrager **17208** einen Wert zu einem dem schwebenden Dienst zugeordneten Dienst-Wallet ansprechend auf eine Erkennung übertragen, dass eine Bedingung der schwebenden Erlaubnisanleitung erfüllt wurde. Bei einem Beispiel kann das Dienst-Wallet einen Blockchain-codierten Wert enthalten. Der schwebende Dienst kann die Funktion einstellen, wenn das Dienst-Wallet einen Wert von Null aufweist. Bei einem Beispiel kann die Erlaubnisanleitung angeben, dass ein Dienst-Wallet einen Wert ansprechend auf eine Erkennung übertragen kann, dass das Dienst-Wallet einen Auslöseschwellenwert erreicht hat. Der schwebende Dienst kann eine Werttransaktion zwischen dem Dienst-Wallet und einem Host-Wallet einleiten.

[0975] **Fig. 173** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **17300**, das Code aufweist, um schwebende Dienste zu behandeln, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[0976] Das nichtflüchtige maschinenlesbare Medium **17300** kann einen Code **17302** aufweisen, um eine Schwebender-Dienst-Erlaubnisanleitung für eine Anzahl entdeckter Hosts zu verfassen, wobei die Anzahl der entdeckten Hosts auf die Erfüllung eines Parameters durch einen Host beurteilt wird. Bei einem Beispiel kann die Schwebender-Dienst-Erlaubnisanleitung Strafen angeben, die ansprechend auf eine erkannte Verletzung der Diensturlaubnisanleitung gegen einen Host zu erwägen sind, wobei die Strafen von einem Host-Wallet einzuziehen sind.

[0977] Das nichtflüchtige maschinenlesbare Medium **17300** kann einen Code **17304** zur Auswahl einer Host-Hardware für den schwebenden Dienst auf der Grundlage einer Datenstruktur des schwebenden Dienstes aufweisen. Bei einem Beispiel ist die Datenstruktur eine Entscheidungsmatrix. Die Entscheidungsmatrix kann beispielsweise ein vom schwebenden Dienst angestrebtes Merkmal, die Anzahl der verfügbaren Hosts und einen Bewertungswert der Hosts in Bezug auf die in der Entscheidungsmatrix aufgelisteten Merkmale auflisten. Der schwebende Dienst kann einen Host auf der Grundlage des besten Werts auswählen, der anhand der Kosten pro Stunde, dividiert durch die Anzahl der Merkmale, berechnet wird, wobei die Qualitätsmetrik eine zufrieden stellende Verwendung des schwebenden Dienstes angibt und die Kosten pro Stunde projizierte Kosten pro Stunde des Betriebs des schwebenden Dienstes unter Verwendung eines bewerteten Hosts sind. Die Merkmale des schwebenden Dienstes können die Merkmale bei einer Wertberechnung unter Verwendung der Entscheidungsmatrix auf verschiedene Weise gewichten.

[0978] Das nichtflüchtige maschinenlesbare Medium **17300** kann einen Code **17306** zur Ausführung der Schwebender-Dienst-Erlaubnisanleitung unter Verwendung der Host-Hardware aufweisen. Das nichtflüchtige maschinenlesbare Medium **17300** kann einen Code **17308** aufweisen, um ansprechend auf die Erkennung, dass eine Bedingung der schwebenden Erlaubnisanleitung erreicht wurde, einen Wert zu einem dem schwebenden Dienst zugeordneten Dienst-Wallet zu übertragen. Bei einem Beispiel kann das Dienst-Wallet einen Blockchain-codierten Wert enthalten. Der schwebende Dienst kann die Funktion einstellen, wenn das Dienst-

Wallet einen Wert von Null aufweist. Bei einem Beispiel kann die Erlaubnisanleitung angeben, dass ein Dienst-Wallet einen Wert ansprechend auf eine Erkennung übertragen kann, dass das Dienst-Wallet einen Auslösschwellenwert erreicht hat. Der schwebende Dienst kann eine Werttransaktion zwischen dem Dienst-Wallet und einem Host-Wallet einleiten.

[0979] Erlaubnisanleitungen können eine Laufzeitberechnung für die Kosten der Dienstbereitstellung sowie den historischen Ruf eines Hosts oder eines Dienstes aufnehmen. Die Kosten können sich auf Energiekosten, Gerätekapitalkosten, Wertverlustkosten, Zeitpunktkapazitätskosten, Datenschutzkosten und Datenentropiekosten beziehen. Wie hier offenbart, kann ein Erlaubnisanleitungs-Verhandlungsprozess zeitbasiert sein. Die Erlaubnisanleitung kann in der Lage sein, zwischen Anbietern zu wechseln, selbst wenn Aufgaben zugewiesen wurden, und mitten in der Ausführung. Bei einem Beispiel kann der Wechsel zwischen Anbietern ansprechend auf die Änderung von Bedingungen geschehen, die den Verbraucher oder Anbieter des Dienstes beeinflussen können.

[0980] Fig. 174 ist ein schematisches Diagramm eines beispielhaften Erlaubnisanleitungs-Verhandlungsprozesses 17400 gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in Fig. 165 beschrieben sind.

[0981] Bei einem Beispiel kann eine Verhandlung für eine Erlaubnisanleitung nicht existieren oder eine Erlaubnisanleitungsschablone sein. Eine Erlaubnisanleitungsschablone kann eine unvollständige Version einer erzwingbaren Vereinbarung sein, die als eine Reihe über ein Speichermedium verstreuter Erlaubnisse oder als Einzeldokument, das Erlaubnisse, Rechte und Verpflichtungen der Parteien, die vereinbaren, die Erlaubnisanleitung anzunehmen, angibt, gespeichert ist. Eine Erlaubnisanleitungsschablone kann einer interessierten Partei Zugriff für das Lesen und Ausführen von Änderungen erlauben.

[0982] Der Erlaubnisanleitungs-Verhandlungsprozess 17400 kann ansprechend auf die Entdeckung von Peers und das anfängliche Verfassen einer Erlaubnisanleitung beginnen. Bei einem Beispiel kann eine anfängliche Erlaubnisanleitung mit QoS-T&C belegt werden, wie vom Dienst oder vom Datenverbraucher oder von Datenverbrauchern gefordert.

[0983] Der Erlaubnisanleitungs-Verhandlungsprozess 17400 kann Angaben zum Interesse von Peers, Hosts und anderen Diensten zum Beitreten empfangen. Dementsprechend kann ein Kandidatendienstanbieter oder -verbraucher, der möchte, sich den von der Erlaubnisanleitung festgelegten Erlaubnissen anzuschließen und sich an diese zu halten, den Prozess zum Beitritt durch Bewerben für den Beitritt 17402 einleiten. Ein Kandidatendienstanbieter oder -verbraucher, der sich für den Beitritt bewirbt, kann Informationen über Anbieterattribute bzw. Verbraucherattribute bereitstellen. Das Anbieterattribut und die Verbraucherattribute können sich auf bewertete Fähigkeiten oder Merkmale der Vorrichtungen beziehen oder die Fähigkeiten und Merkmale validieren, bevor damit fortgefahren wird, diese Fähigkeiten und Merkmale in eine Vorrichtungsattributliste 16524 aufzunehmen.

[0984] Eine Angebotsfunktion, eine Anforderungsfunktion oder eine Zuweisungsfunktion 17404 kann verwendet werden, um einen verwendbaren Satz von Diensteanbietern, Datenanbietern und Verbrauchern zu identifizieren. Der Satz von Diensteanbietern, Datenanbietern und Verbrauchern kann verwendbar sein, falls Attribute und Fähigkeiten überlappend sind, so dass die Attribute und Fähigkeiten die Bedingungen der Erlaubnisanleitung erfüllen können. Die Erfüllung der Bedingungen der Erlaubnisanleitung kann sich beispielsweise darauf beziehen, dass ein vollständiger Satz der Anforderungen der Parteien erfüllt wird. Die Erfüllung der Bedingungen der Erlaubnisanleitung kann sich beispielsweise darauf beziehen, dass so viele Anforderungen der Parteien wie praktisch möglich erfüllt werden.

[0985] Bei einem Beispiel können einem höchstrangigen Diensteanbieter oder Datenanbieter Angebote von einem Kandidatendienstverbraucher gemacht werden. Anbieter, die ein Angebot empfangen, können eine Anforderung zur Bestätigung ihrer Annahme des Angebots senden. Ansprechend auf den Empfang eines Angebots kann der angenommene Anbieter an die Erlaubnisse der Erlaubnisanleitung gehalten werden und Teil der Liste bestätigter Vorrichtungen 17406 werden. Während des Beitrittsprozesses kann eine Verhandlung geschehen. Während der Verhandlung können Kandidaten vereinbaren, wie auf den Dienst oder die Daten zugegriffen werden kann. Falls kein überlappende Satz von Technologien vereinbart werden kann, kann ein Protokoll- und Datenschemavermittler in der Art eines Erlaubnisvermittlers einer dritten Partei eingeladen werden, sich der Erlaubnisanleitung als Zwischenvermittler anzuschließen.

[0986] Bestätigte Anbieter und Verbraucher können optional wählen, die Erlaubnisanleitung zu verlassen. Das optionale Verlassen kann mit keinen Kosten verbunden sein, oder es kann Bedingungen geben, bei denen eine Strafe anzuwenden ist. Bei einem Beispiel kann eine Strafe erwogen werden, falls eine Vorrichtung ihre Verpflichtungen nicht erfüllt und keine Ersatzvorrichtung identifiziert werden kann.

[0987] Während der Ausführung der Erlaubnisanleitung **16510** können sich andere Anbieter und Verbraucher für den Beitritt bewerben und beitreten. Wenn die Erlaubnisanleitung ausgeführt wird **16510**, können Anbieter und Verbraucher ersetzt werden.

[0988] **Fig. 175** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **17500** zur Erlaubnisanleitungsverhandlung gemäß einigen Ausführungsformen. Das Verfahren **17500** aus **Fig. 175** kann durch die mit Bezug auf **Fig. 177** beschriebene IoT-Vorrichtung **17700** implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 166** beschrieben wurden. Der Prozessfluss kann in Block **16602** beginnen. In Block **17502** können sich Knoten für den Beitritt bewerben. Die Knoten können Anbieter, Beitragende und andere Vorrichtungen und Dienste einschließen, die von der Erlaubnisanleitung gesteuert werden möchten.

[0989] In Block **17504** können die Knoten ihre Angebote, Attribute und jegliche Bedingungen oder Konditionen, die ein Knoten haben kann, auflisten. Während des Knotenbewerbungsprozesses kann eine Kostenfunktion auf die von den Knoten empfangenen Eingaben angewendet werden. Bei einem Beispiel kann die Kostenfunktion ein Infocoin-Algorithmus sein, wie nachstehend offenbart. Die Kostenfunktion kann auf Knoten in einem IoT-Marktplatz angewendet werden, weil bei einem Beispiel eine Kostenbewertung die Kosten der Installation und Bereitstellung von IoT-Vorrichtungen auf dem Gebiet einschließen kann. Kostenbewertungen können beispielsweise die Energiekosten, laufenden Kosten und Wartungskosten für den Betrieb der Vorrichtung, den Datentransport und Speichervorrichtungen einschließen. Kostenbewertungen können die Kosten dieser über eine weitverbreitete Infrastruktur installierten Vorrichtungen zusätzlich zu den Kosten für eine Betriebsmarge einschließen. Bei einem Beispiel kann sich die Marge auf einen Bereich beziehen, in dem eine Verhandlung durch die Verwendung unterer und oberer Bereiche von verschiedenen Parteien stattfinden kann.

[0990] In Block **17506** kann eine Datenebene aktualisieren. Die Datenebene kann einen Mechanismus in der Blockchain oder außerhalb der Blockchain repräsentieren. Wie vorstehend erörtert, können die in einer Blockchain verwendeten und referenzierten Daten durch Integration mit einer verteilten Hash-Tabelle (DHT) ausgeführt werden.

[0991] In Block **17508** können Vorrichtungen hinzugefügt werden, die bestätigt wurden. Bei einem Beispiel können bestätigte Vorrichtungen durch ein Vorrichtungskriterium, durch eine Parameterauswahl oder auf der Grundlage einer Kostenfunktion identifiziert werden. Beispielsweise kann eine spezifizierte Kriterien erfüllende Vorrichtung standardmäßig akzeptiert werden. Eine Vorrichtung mit einem bestimmten Eignungsparameter kann akzeptiert werden. Eine Vorrichtung, welche die Ausgabe einer Kostenfunktion erfüllt, kann akzeptiert werden. Eine Kostenfunktion kann das Ordnen von Knoten und das Akzeptieren der obersten N geeignetsten Knoten in Bezug auf die Kosten pro Liefereinheit priorisieren. Wie bei anderen hier beschriebenen Verfahren kann eine Präambel im Protokoll-Frame verwendet werden. Die Präambel kann es Teilnehmern ermöglichen, Daten zu verhandeln, die erforderlich sind, um die Fortsetzung des Prozesses zu ermöglichen, bevor Tokens zwischen der Erlaubnisanleitung und ihren Teilnehmern verhandelt werden. Parteien, welche die korrekten Tokens besitzen, kann anschließend vertraut werden, um auf spezifische Dienste zuzugreifen oder diese bereitzustellen.

[0992] Wie vorstehend erörtert wurde, kann eine Knotenverhandlung von einer Erlaubnisanleitung eine Kostenfunktion in der Art eines Infocoin-Algorithmus verwenden. Ein Infocoin-Algorithmus kann annehmen, dass der Sensor Daten kontinuierlich mit einer vordefinierten Rate sendet. Ein Infocoin-Algorithmus kann annehmen, dass der Lebensdauer- und Wartungsplan des Sensors vorhersehbar ist. Ein Infocoin-Algorithmus kann annehmen, dass Außerbandanforderungen für Daten nicht erlaubt sind. Ein Infocoin-Algorithmus kann annehmen, dass der Sensor, der Gateway und der Server weniger Ressourcenrandbedingungen haben, wie beispielsweise Leistungsrandbedingungen, Verarbeitungsrandbedingungen, Kommunikationsrandbedingungen oder Speicherrandbedingungen.

[0993] Wie in der nachstehenden Gleichung verwendet, bezieht sich D auf eine Dateneinheit. Diese Dateneinheit kann ein primärer Datenbestandteil sein. Bei einem Beispiel kann ein primärer Datenbestandteil eine direkt beobachtete Messung durch einen Sensor in einem IoT-Netz sein. Ein primärer Datenbestandteil kann

sich auf einen auf der Grundlage von Eingaben von einer oder mehreren primären Datenquellen berechneten abgeleiteten Datenbestandteil beziehen.

[0994] Wie in der nachstehenden Gleichung verwendet, bezieht sich C_t auf die Kosten des Transports der Dateneinheit. Bei einem Beispiel kann eine Dateneinheit als Infocoin bezeichnet werden. Die Kosten des Transports der Dateneinheit können von Netztransportkosten oder von der Größe der zu transportierenden Daten abhängen. Die Kosten für den Transport der Dateneinheit können davon abhängen, ob die Daten über das Netz zu einer neuen Speicherstelle kopiert werden oder ob eine URI für ein Datenheim verwendet wird. Bei einem Beispiel kann ein Datenheim ein interplanetarisches Dateisystem bzw. Inter Planetary File System (IPFS) oder ein leichtgewichtiges Fog-Dateisystem sein. Wie in der nachstehenden Gleichung verwendet, bezieht sich C_{Speicher} auf die Kosten für das Speichern der Dateneinheit, wobei die Speicherkosten eine Funktion der Größe der Daten sein können. Die Kosten für das Speichern von Daten können sich darauf, ob eine Replikation von Daten für eine Redundanz verwendet wird, und auf die Kosten des spezifischen Speichermediums beziehen.

[0995] Wie in der nachstehenden Gleichung verwendet, kann der Begriff Marge den von Daten bereitgestellten Wert widerspiegeln. Bei einem Beispiel nimmt der Wert von Daten zu, wenn Daten mit anderen Datenquellen kombiniert werden können. Wie in der nachstehenden Gleichung verwendet, kann sich C_{roh} auf die Kosten des Erhaltens oder die Kosten der Erzeugung einer Einheit primärer Daten plus einer Betriebsmarge beziehen. Die Kosten für das Erhalten einer Dateneinheit oder die Kosten für das Erzeugen einer Dateneinheit können beide Festkosten für den Sensor (C_s) einschließen und Kosten für die Wartung über die Lebensdauer des Sensors (C_m) und laufende Energiekosten (C_e) für den Sensorknoten einschließen. Bei einem Beispiel können die Kosten für das Erhalten einer Dateneinheit oder die Kosten für das Erzeugen einer Dateneinheit beide die Abtastrate pro Tag (Rate) und die Anzahl der Tage (t), während derer der Sensor verwendet wird, berücksichtigen. C_{roh} kann von einer Erlaubnisanleitung als Angabe eines verhandelten Werts für Parteien, welche die Erlaubnisanleitung übernommen haben, verwendet werden.

$$C_{\text{roh}} = [C_s + (C_e * t) + C_m] / [Rate * t] * Margendaten * C_{\text{abgeleitet}}$$

[0996] Bei einem anderen Beispiel können die Kosten für das Erhalten abgeleiteter Daten oder virtueller Daten durch Verarbeiten oder Analysieren eines oder mehrerer Sätze von Primärdaten, um neue Einsichten und neuen Wert zu erhalten, erzeugt werden. Wie hier verwendet, kann es wenigstens drei Typen abgeleiteter Daten geben. Ein Typ abgeleiteter Daten kann innerhalb eines Sensorknotens abgeleitete Daten einschließen. Ein anderer Typ abgeleiteter Daten kann innerhalb eines Netzes abgeleitete Daten einschließen. Ein weiterer Typ abgeleiteter Daten kann von historischen Daten abgeleitete Daten einschließen.

[0997] Bei einem Beispiel können Rohkosten auf der Grundlage der Anzahl von Datenquellen variieren. Falls abgeleitete Daten beispielsweise anhand mehrerer Eingaben am selben Sensorknoten berechnet werden können, sind die Kosten für das Erhalten der Daten gleich oder ähnlich jenen für das Erhalten von Rohdaten. Die Festkosten für den Sensorknoten und die laufenden Kosten können gleich sein, unabhängig davon, ob alle Sensoren am Knoten verwendet werden. Dementsprechend können bei einem Beispiel keine zusätzlichen Kosten für das Berechnen eines abgeleiteten Werts am selben Knoten auftreten. Beispielsweise kann das Berechnen eines abgeleiteten Werts für einen Komfortindex anhand Eingaben der Temperatur und der Feuchtigkeit Daten vom selben Knoten einschließen, so dass Rohkosten für den Datentransport nicht erhöht werden können.

[0998] Abgeleitete Daten können mehr Wert bereitstellen als Rohdaten, und es kann eine berechnete „Marge für abgeleiteten Wert“ geben, wie die nachstehende Gleichung zeigt.

$$C_{\text{abgeleitet\_lokal}} = C_{\text{roh}} * Margeninformationen$$

[0999] Daten können von einer Anzahl verschiedener Quellen abgeleitet werden. Bei einem Beispiel können Daten an einem Gateway, einem Server, einem Instrument, einem Zentralprozessor oder anderen Vorrichtungen abgeleitet werden. Wenn Rohdaten zu einem Ort für die Erzeugung abgeleiteter Daten zu transportieren sind, können bei einer Kostenberechnung Kosten für den Transport von Daten addiert werden. Beispielsweise können sich die Kosten für den Transport von Daten auf die Kosten von einem Knoten zu einem Gateway oder Server laufender Daten sowie auf die Kosten für das Speichern der Daten an diesem Ort beziehen. Bei einem Beispiel kann eine Rohdateneinheit mehrere Stufen für den Transport zu einem Endziel für die Daten aufweisen. Während des Transports kann eine Dateneinheit an einer auf mittlerem Wege liegenden oder Zwi-

schonstufe zwischen den Beförderungen zu einem Endziel für die Daten lokal gespeichert werden. Ein Kostenbetrag kann als Summe der Kosten für das Erreichen des Endziels durch Rohdatenbestandteile plus einer „Marge für den abgeleiteten Wert“ erzeugt werden. In der nachstehenden Formel könnte die variable C_{roh} durch $C_{abgeleitet\_lokal}$ ersetzt werden, falls die Daten an einem Punkt auf ihrem Weg zum Endziel abgeleitet werden, um die mit $C_{abgeleitet\_fern}$ bezeichneten Daten zu erzeugen.

$$C_{abgeleitet\_fern} = \sum_0^n \left[C_{roh} + \sum_0^n (C_t + C_{Speicher}) \right] * \text{Margenkenntnis}$$

[1000] Falls Daten anhand historischer Daten abgeleitet werden, können die Kosten für das Speichern der Daten zu den Kosten für das Erzeugen der Daten hinzugefügt werden. Die Kosten können infolge des erhöhten Werts von Daten, wenn zusätzliche Datenquellen hinzugefügt werden, im Wesentlichen proportional zur Anzahl für die Erzeugung dieser Daten verwendeter historischer Proben sein.

[1001] In der nachstehenden Beispielgleichungen repräsentiert C_{acq} Kosten, die für das Erhalten von Daten D berechnet werden können. Daten können einen monetären Wert, beispielsweise US Dollars, aufweisen. Daten können Wert auch in Bezug auf ein anderes natives oder überlagertes Asset ausdrücken. Die Kosten von C_{acq} können gleich C_{roh} , $C_{abgeleitet\_lokal}$ oder $C_{abgeleitet\_fern}$ sein. In der nachstehenden Beispielgleichung kann Div den Informationswert der Dateneinheit repräsentieren. Div kann von Dateneinheit zu Dateneinheit variieren, weil nicht alle Dateneinheiten den gleichen Wert haben können.

$$C_{abgeleitet\_historisch} = \sum_0^n (C_{acq} + C_S) * \text{Margenweisheit}$$

[1002] Um einen Wert einer Dateneinheit oder von Daten allgemein zu identifizieren, kann ein Beweiskraftmodell über einen Informationsbewertungswert informieren, der zur Klassifizierung eines Datenwerts zur Zeit der Erzeugung der Daten verwendet wird. Ein Informationswert (IV) kann zur Auswahl von Variablen in einem Vorhersagemodell verwendet werden. Bei einem Beispiel kann eine IV-Statistik als Prediktor für die Modellierung nicht nützlich sein, falls die IV-Statistik weniger als eine Schwelle abfällt. Eine Verwendung und Variierung einer Schwelle für einen berechneten IV kann verwendet werden, um den Wert einer Dateneinheit oder eines Infocoins zu beurteilen. Dateneinheiten mit einem IV unterhalb einer Schwelle würden einen niedrigeren Wert empfangen. Dateneinheiten mit einem IV oberhalb einer Schwelle, jedoch unterhalb einer zweiten Schwelle, könnten einen mittleren zugewiesenen Wert aufweisen. Diese Beurteilung eines Bewertungswerts könnte zunehmen, wenn die Anzahl der IV-Schwellen durch die Eingaben für einen IV-Datenbewertungswert überschritten wird. Bei einem Beispiel könnten Daten hohen Werts einen größeren monetären Wert haben, wenn die Daten von Verbrauchern in einem IoT-Ökosystem stärker nachgefragt werden. Bei einem Beispiel ist der Wert der Dateneinheit umso größer, je stärker eine Dateneinheit nachgefragt wird.

[1003] Zusätzliche Verfahren zum Speichern und Beurteilen des Werts einer Dateneinheit können in einem Verhandlungssystem austauschend eingesetzt werden. Die Verwendung eines IV-Bewertungswerts für Dateneinheiten kann der Satz für einen Bewertungswert für Daten sein, wodurch ermöglicht wird, dass Informationen selbst als handelbares Asset innerhalb eines Verhandlungsrahmens oder auf andere Weise verwendet werden.

[1004] **Fig. 176** ist ein schematisches Diagramm einer beispielhaften Datenstruktur **17600** zur Beurteilung des Werts einer Dateneinheit und zum Zuweisen des Werts zu der Dateneinheit gemäß einigen Ausführungsformen. Die dargestellten Daten dienen lediglich als Beispiel und zeigen einen beispielhaften Weg zur Berechnung des Werts von Dateneinheiten sowie zur Auswahl des wertvollsten Datenbestandteils. Ferner können die Daten, denen ein Wert zugewiesen werden kann, als Verhandlungspunkt oder Zahlungsverfahren einer Erlaubnisanleitung verwendet werden. Bei der beispielhaften Datenstruktur **17600** ist die Spalte für die Beweiskraft- bzw. Weight-of-Evidence(WOE)-Berechnung **17602** als auf einem Prozentsatz von Ereignissen, für die Daten in einem bestimmten Knoten gesammelt werden, beruhend dargestellt.

[1005] Bei der beispielhaften Datenstruktur **17600** kann eine Spalte für Bin eine Identifikation von Knoten sein, die wenigstens 5 % der Beobachtungen für einen bestimmten Datentyp haben. Bei einem Beispiel kann es mehrere solcher Wertberechnungsmodelle für jeden Knoten und jeden Datentyp geben. Bei der beispielhaften Datenstruktur **17600** erscheint Bin **7** als Daten, die einen hohen Vorhersagewert haben können. Bei der beispielhaften Datenstruktur **17600** erscheint die Gesamt-Div für den Datensatz als Wert von 0,3138. Daten aus Bin **7** können relativ einen höheren Wert in einem Datenmarkt bedingen. Der C_{acq} -Wert beim dargestellten Beispiel kann als ein flacher Wert über Bins und Knoten erscheinen. Marktkräfte können den Wert von C_{acq}

jedoch ändern. Die Erzeugung eines Markts für Informationseinheiten kann Datenlieferanten ermutigen, Datentypen zu liefern, die einen Profit für ihre Investition bedingen.

[1006] Fig. 177 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 17700 für die Verhandlung mit einem Wert versehener Dateneinheiten vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in Fig. 8 beschrieben sind.

[1007] Wie vorstehend auch mit Bezug auf Fig. 8 gezeigt wurde, kann der Massenspeicher 808 eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher 808 kann einen Erlaubnisanleitungsverfasser 17702 zum Verfassen einer Erlaubnisanleitung für einen ersten entdeckten Peer mit einem ersten Parameter und einem ersten Parameterwert und einen zweiten entdeckten Peer mit einem zweiten Parameter und einem zweiten Parameterwert aufweisen. Bei einem Beispiel können sich der erste Parameter und der zweite Parameter auf akzeptable Datenwertbereiche für einen ersten bzw. einen zweiten Knoten beziehen. Die akzeptablen Datenwertbereiche können mit einer Kostenfunktion berechnet werden. Die Kostenfunktion kann Betriebskosten für einen Knoten, der die Erlaubnisanleitung implementiert, berechnen und kombinieren. Die Betriebskosten umfassen beispielsweise Energiekosten, laufende Kosten und/oder Wartungskosten für den Betrieb einer Vorrichtung, den Datentransport und Speichervorrichtungen. Bei einem Beispiel können sich die Datenwertbereiche auf eine Berechnung des Werts der Daten als Funktion der Anzahl der Datenquellen beziehen. Die Daten können von einer Anzahl von Sensoren synthetisierte abgeleitete Daten sein. Der Wert von Daten kann zunehmen, wenn die Rate der Datennachfrage zunimmt.

[1008] Der Massenspeicher 808 kann einen Parametergewichtsberechner 17704 zur Berechnung eines ersten Parametergewichts und eines zweiten Parametergewichts durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts, wie beispielsweise für das Gewicht der Ereignisspalte mit Bezug auf Fig. 176 beschrieben, aufweisen. Der Massenspeicher 808 kann einen Bedingungsgenerator 17706 zur Erzeugung einer Bedingung der Erlaubnisanleitung ansprechend darauf, dass eine vorgeschlagene Bedingung innerhalb vom ersten Parameter und vom zweiten Parameter vorgeschlagener Bereiche liegt, aufweisen, wobei der erste Parameter durch das erste Parametergewicht angepasst wird und der zweite Parameter durch das zweite Parametergewicht angepasst wird. Der Massenspeicher 808 kann einen Aktionsausführer 17706 zur Ausführung einer Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt wurde, aufweisen.

[1009] Bei einem Beispiel kann ein Prozessor 802 eine Anforderung von einem Kandidaten-Peer an die Erlaubnisanleitung zur Aufnahme eines Beitrittsparameters und eines Beitrittsparameterwerts verarbeiten. Bei einem Beispiel kann ein Prozessor 802 ein Beitrittsparametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts mit dem Beitrittsparameterwert berechnen.

[1010] Fig. 178 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums 17800, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 9 beschrieben wurden.

[1011] Das nichtflüchtige maschinenlesbare Medium 17800 kann einen Code 17802 aufweisen, um den Prozessor 902 anzuweisen, eine Erlaubnisanleitung für einen ersten entdeckten Peer, der einen ersten Parameter und einen ersten Parameterwert aufweist, und einen zweiten entdeckten Peer, der einen zweiten Parameter und einen zweiten Parameterwert aufweist, zu verfassen. Bei einem Beispiel können sich der erste Parameter und der zweite Parameter auf akzeptable Datenwertbereiche für einen ersten bzw. einen zweiten Knoten beziehen. Die akzeptablen Datenwertbereiche können mit einer Kostenfunktion berechnet werden. Die Kostenfunktion kann Betriebskosten für einen Knoten, der die Erlaubnisanleitung implementiert, berechnen und kombinieren. Die Betriebskosten umfassen Energiekosten, laufende Kosten und/oder Wartungskosten für den Betrieb einer Vorrichtung, den Datentransport und Speichervorrichtungen. Bei einem Beispiel können sich die Datenwertbereiche auf eine Berechnung des Werts der Daten als Funktion der Anzahl der Datenquellen beziehen. Die Daten können beispielsweise von einer Anzahl von Sensoren synthetisierte abgeleitete Daten sein. Der Wert von Daten kann zunehmen, wenn die Rate der Datennachfrage zunimmt.

[1012] Das nichtflüchtige maschinenlesbare Medium 17800 kann einen Code 17804 aufweisen, um den Prozessor 902 anzuweisen, ein erstes Parametergewicht und ein zweites Parametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts zu berechnen. Das nichtflüchtige maschinenles-

bare Medium **17800** kann einen Code **17806** aufweisen, um den Prozessor **902** anzuweisen, eine Bedingung der Erlaubnisanleitung ansprechend darauf zu erzeugen, dass eine vorgeschlagene Bedingung innerhalb vom ersten Parameter und vom zweiten Parameter vorgeschlagener Bereiche liegt, wobei der erste Parameter durch das erste Parametergewicht angepasst wird und der zweite Parameter durch das zweite Parametergewicht angepasst wird. Das nichtflüchtige maschinenlesbare Medium **17800** kann einen Code **17808** aufweisen, um den Prozessor **902** anzuweisen, eine Aktion der Erlaubnisanleitung ansprechend darauf auszuführen, dass erkannt wurde, dass Kondition der Bedingung erfüllt wurde.

[1013] Bei einem Beispiel kann der Prozessor **902** eine Anforderung von einem Kandidaten-Peer an die Erlaubnisanleitung zur Aufnahme eines Beitrittsparameters und eines Beitrittsparameterwerts verarbeiten. Bei einem Beispiel kann der Prozessor **902** ein Beitrittsparametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts mit dem Beitrittsparameterwert berechnen.

[1014] Nachrichtenflüsse über ein IoT-Netz können im Laufe der Zeit ein erkennbares Muster erzeugen, falls jedoch ein nicht autorisierter Agent Zugang zum Netz erhält, kann er in der Lage sein, Operationen für seine eigenen Zwecke zu ändern. Falls Transaktionen in einer Blockchain sichtbar sind, kann daher eine solche unbefugte Aktivität im Netz erkannt werden und können Aktionen getätigt werden, um im Wesentlichen nicht autorisierte Transaktionen zu lösen oder sogar ihr Auftreten zu verhindern.

[1015] Bei einem Beispiel kann eine Blockchain verwendet werden, um einen Datensatz von Transaktionen in einem Netz zu halten sowie eine Vorautorisierung für einen Netzagenten zur Ausführung einer Operation vorzunehmen. Diese Vorautorisierungsfunktion kann als Dezentralisiertes-Netzzugangs-Proxy- bzw. Decentralized-Network-Access-Proxy(DNAP)-Protokoll bezeichnet werden.

[1016] **Fig. 179** ist ein schematisches Diagramm einer beispielhaften Organisation **17900** für das dezentralisierte Netzzugangs-Proxy zur Verwendung von Funktionen gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 165** offenbart wurden. Ein Prozess für die Funktion eines DNAP-Protokolls und seine Wechselwirkungen mit einer Erlaubnisanleitung **17902** kann bei **17904** beginnen.

[1017] In Block **17904** kann eine Vorrichtung booten. Der Boot-Prozess kann die Initialisierung eines Netzstapels an einer Netzschnittstellenvorrichtung in einer Vorausführungsumgebung bzw. Pre-Execution Environment (PXE) sein und nicht das Vorhandensein eines Softwarestapels höherer Ebene oder Betriebssystems implizieren.

[1018] In Block **17906** kann ein Netzschnittstellenadapter Schlüssel für die Verwendung beim Betrieb als Blockchain-bewusste Vorrichtung erzeugen. Die Vorrichtung, welche die erzeugten Schlüssel verwendet, kann auch eine durch Hardware ermöglichte sichere Enklave verwenden oder aus dieser arbeiten. Die erzeugten Schlüssel können verwendet werden, um die Vorrichtung verlassenden Verkehr zu signieren, so dass der Ursprung des Verkehrs für jedes Paket und der Inhalt jedes Pakets bestimmt werden können. Bei einem Beispiel kann die schlüsselbasierte Verschlüsselung für diese Vorrichtung durch Hardware auf der Vorrichtung ermöglicht sein und dabei helfen, Mittelsmannangriffe bzw. Man-in-the-Middle-Angriffe zu verhindern. Ein Netz kann an der Vorrichtung ankommenden Verkehr ausschließen, falls der Verkehr nicht mit dem privaten Schlüssel von einem gültigen Agenten signiert ist. Bei einem Beispiel kann zur Verwendung von Netz-Switches und -Routern eine Modifikation an den Netz-Switches vorgenommen werden, so dass die Hardwareverschlüsselung und -entschlüsselung des Verkehrs geschehen kann.

[1019] In Block **17908** kann ein Netzschnittstellenadapter eine Zugriffsanforderungstransaktion auf der Blockchain erzeugen. Bei einem Beispiel kann ein auf dem Netz laufendes Paket zwangsweise zu einem DNAP geleitet werden. In diesem Zusammenhang kann das DNAP als eine Funktion der Schicht-2-Datenverbindungsschicht angesehen werden, weil es als ein Dienst auf den physikalischen Switches und Routern des Netzes laufen kann. Sobald eine Netzvorrichtung versucht, die Kernnetzinfrastruktur zu verwenden, kann die Netzvorrichtung nicht in der Lage sein, zu verhindern, dass der Netzvorrichtungsverkehr zum dezentralisierten Netzzugangs-Proxy weitergeleitet wird, falls sie versucht, die Kernnetzinfrastruktur oder eine Verbindung zu benutzen, die über eine private Peer-to-Peer-Verbindung über ein zweckgebundenes Medium hinausgeht. Bei einem Beispiel kann eine Peer-to-Peer-Verbindung über ein zweckgebundenes Medium eine Kommunikation durch Bluetooth oder ein überkreuztes Ethernet-Kabel aufweisen.

[1020] In Block **17910** kann das DNAP-Protokoll einer Vorrichtung bestimmte Netzzugangsfunktionen gewähren. Bei einem Beispiel kann das DNAP-Protokoll zuvor erörterte Funktionen der Erlaubnisanleitung verwenden.

den. Knoten wie Switches und Router auf einem ein DNAP-Protokoll ausführenden Netz können zu Minern einer Blockchain werden. Bei einem Beispiel können die Knoten eines Netzes einen Konsensalgorithmus ausführen, der keinen großen Rechen-Overhead verwendet, oder auf einer direkten Teilnahme an einer Transaktion beruht. Ein Nachweis-derverstrichenen-Zeit-Algorithmus kann ein Beispiel einer in diesem Protokoll verwendeten Technologie sein. Die Verwendung des DNAP-Protokolls kann auch vor dem Einbringen bössartiger Switches und Router schützen, weil ein Übeltäter in der Lage sein müsste, 51 % der Netzinfrastruktur zu installieren oder zu beeinträchtigen, beispielsweise um einen erfolgreichen Angriff auszuführen. Ein Versuch einer DNAP-Vorrichtung, die Zugangsanforderungs-Transaktionsfunktion zu verwenden, kann dazu führen, dass sich ein Netzschnittstellenadapter dem Netz selbst durch den Mechanismus einer Erlaubnisanleitung identifiziert. Der Netzschnittstellenadapter kann eine durch Hardware ermöglichte sichere Enklave ausführen, um bei diesem Prozess zu helfen.

[1021] In Block **17912** kann eine DNAP verwendende Vorrichtung zu einer Erlaubnisanleitungsliste erzeugter oder autorisierter Vorrichtungen im Netz hinzugefügt werden, falls die DNAP verwendende Vorrichtung durch die Beitrittsfunktion in der Erlaubnisanleitung akzeptiert wird. In Block **17910** kann ein Initialisierungsprozess auftreten und kann die Vorrichtung ihre Attribute und Merkmale der Erlaubnisanleitung beschreiben. Bei einem Beispiel können die durch DNAP beschriebenen Attribute durch eine durch Hardware ermöglichte sichere Enklave auf der DNAP-Vorrichtung attestiert werden, um ein Vertrauensniveau herzustellen. Bei einem Beispiel kann die Beschreibung von Attributen der DNAP-Vorrichtung in einer Erweiterung einer menschlichen Schnittstellenvorrichtung bzw. Human Interface Device (HID) definiert werden. Die Beschreibung von Attributen oder die in der Erlaubnisanleitung gespeicherten Daten können außerhalb der Chain gespeichert werden. Bei einem Beispiel kann bei einem mit dem DNAP-Protokoll versehenen Switch oder Router die Speicherung von Daten außerhalb der Chain mit der Integration eines gewissen Speichers innerhalb des Switches einhergehen. Die Switches und Router in einem DNAP-Netz können Edge- oder Fog-Knoten sein. Die Speicherung kann zu einem verteilten Speichermechanismus vom DHT-Typ auf den Routern und Switches im Netz werden.

[1022] In Block **17914** können Tokens an Vorrichtungen ausgegeben werden, um den Vorrichtungen zu ermöglichen, Aktionen in einer orchestrierten Weise auszuführen. Die Verwendung von Tokens in einer Vorrichtung kann individuelle Vorrichtungsfirewalls für die Entitäten im DNAP-Netz ermöglichen. Bei einem Beispiel kann, falls eine Vorrichtung ein Internet-Steuernachrichtenprotokoll bzw. Internet-Control-Message-Protocol (IMCP)-Token hält, die Vorrichtung Ping-Verkehr senden und empfangen. Die Verwendung von Tokens kann die Bildung virtueller lokaler Netze bzw. Virtual Local Area Networks (VLAN) ermöglichen, indem es Vorrichtungen mit den gleichen Tokens ermöglicht wird, miteinander zu sprechen, ohne durch einen Router zu gehen. Tokens können auch für die Erzeugung privater Netze, die nicht mit größeren Unternehmensnetzen verbunden sind, verwendet werden.

[1023] Tokenzuweisungen können Regeln für das Zuweisen von Standardtokentypen zu bestimmte Kriterien erfüllenden Vorrichtungen aufweisen. Diese Regeln können den Vorrichtungstyp bestimmen und bestimmen, ob die Vorrichtung minimalen Sicherheitsstandards genügt. Bei einem Beispiel kann der Vorrichtungstyp eine von einer Firma besessene und unterstützte Vorrichtung oder eine von einem Angestellten besessene Vorrichtung in einem Plan von einem „Bringe-deine-eigene“-Stil sein. In einigen Umgebungen in der Art einer Person, die auf eine Finanzdatenbank über eine persönliche Vorrichtung zugreift, können die hier beschriebenen Tokenzuweisungen außerhalb einer Firmenumgebung gelten. Bei einem Beispiel können DNAP-Vorrichtungen, die nicht autorisiert sind oder welche die Tokens für bestimmte Aktionen nicht besitzen, eine Benachrichtigung empfangen, dass eine von einer Vorrichtung angeforderte Funktion fehlgeschlagen ist, weil die Vorrichtung durch das Netz nicht autorisiert ist. Durch die Verwendung eines Token-basierten Genehmigungsansatzes kann die Erzwingung von Sicherheit in einem Netz dezentralisiert werden. Bei einem Beispiel können Netzadministratoren Tokens manuell erzeugen, um Aktionen zu repräsentieren, welche die Netzadministratoren in dem Netz erlauben oder verweigern. Bei einem Beispiel kann ein vorgelegter Satz von Tokens von den Netzgeräteherstellern bereitgestellt werden.

[1024] In Block **17916** kann eine DNAP-Vorrichtung im Netz autorisiert werden, um bestimmte Funktionen auszuführen. Der DNAP-Vorrichtung können zusätzliche Tokens gewährt werden oder Tokens von ihr können widerrufen werden. Die Steuerebene für diese Operation kann durch eine Blockchain gestützt sein. Blockchain-gestützt kann sich darauf beziehen, dass Regeln an einem Port oder Zugangspunkt, womit eine Vorrichtung verbunden ist, erzwungen werden, nachdem der Vorrichtung Tokens erteilt wurden, wobei sich die bereitgestellten Regeln für die angeschlossene Vorrichtung im Allgemeinen nicht ändern und die Regeln auf der Grundlage der bestätigten Identität der Vorrichtung erzwungen werden. Bei einem Beispiel können Switches und Router im Netz Miner sein und Transaktionen zu einem gemeinsam geteilten Ledger synchronisieren.

[1025] In Block **17918** können Funktionen, die eine Vorrichtung auszuführen versuchen kann, blockiert werden und kann die Vorrichtung eine Nachricht empfangen, die angibt, dass das Netz die Kommunikation blockiert hat.

[1026] Fig. **180** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **18000** für ein dezentralisiertes Netzzugangs-Proxy zur Verwendung von Funktionen gemäß einigen Ausführungsformen. Das Verfahren **18000** aus Fig. **180** kann durch die mit Bezug auf Fig. **181** beschriebene IoT-Vorrichtung **18100** implementiert werden. Der Prozessfluss kann in Block **18002** beginnen.

[1027] In Block **18002** kann eine Netzvorrichtung initialisiert werden. Bei einem Beispiel kann die Netzvorrichtung ein Client, ein Server, ein Teil der Netzinfrastruktur oder eine Netzschnittstelle sein. In Block **18004** erzeugen die Firmware und die Hardware an der Vorrichtung eine Identität und ermöglichen, dass die Vorrichtung in der Kapazität eines Blockchain-Clients wirkt. Bei einem Beispiel kann ein Knoten die Rolle eines Netzschalters oder eines Routers aufweisen und kann die Vorrichtung in der Kapazität eines Validierers für die DNAP-Blockchain wirken. Die DNAP-Blockchain kann über alle Netzinfrastrukturknoten verteilt sein.

[1028] In Block **18006** kann die Vorrichtung ähnlich einer Vorboot-Ausführungsumgebung bzw. einer Preboot Execution Environment (PXE) oder eines dynamischen Hostkonfigurationsprotokolls bzw. Dynamic Host Configuration Protocol (DHCP) eine Entdeckungsausstrahlungsnachricht veröffentlichen. Bei einem Beispiel könnten die Vorrichtung und das DNAP-Protokoll unter Verwendung von PXE- und DHCP-Protokollen implementiert werden. Bei einem Beispiel kann die Netzvorrichtung, falls die Entdeckungsausstrahlung nicht den Ort jeglicher DNAP-bewusster Systeme zurückgibt, verzögern und erneut versuchen. Falls die Entdeckungsausstrahlung nicht den Ort jeglicher DNAP-bewusster Systeme zurückgibt, kann die Vorrichtung eine Legacy-Operation ausführen, die es ermöglicht, dass die Vorrichtung auf Nicht-DNAP-Netzen arbeitet. Der Prozess der Verzögerung und des erneuten Versuchs oder der Prozess des Wechsels zu einem anderen Netz kann durch eine vorgegebene Richtlinie, BIOS-Einstellungen, Firmware-Einstellungen, physikalische Jumper-Einstellungen an der Vorrichtung gesteuert oder auf andere Weise manuell eingestellt werden.

[1029] In Block **18008** bewirbt sich die DNAP-Vorrichtung, dem DNAP-Netz beizutreten, ansprechend auf die Entdeckung eines sich in Betrieb befindenden DNAP-Netzes. Wie vorstehend erörtert wurde, kann beim Beitreten zum DNAP-Netz einer Erlaubnisanleitung beigetreten werden, der im Netz gefolgt wird.

[1030] In Block **18010** kann eine DNAP-Vorrichtung ihre Attribute und Merkmale veröffentlichen und Tokens anfordern, die der DNAP-Vorrichtung auf der Grundlage der Attribute oder der Identität der Vorrichtung zugewiesen werden können. Eine Entscheidung zum Zuweisen von Tokens kann durch einen Netzadministrator durch die Verwendung von Richtlinien oder beispielsweise auf der Grundlage des Netzes, der Adresse, der Identität der Vorrichtung, des Vorrichtungstyps, von Vorrichtungsfähigkeiten, Vorrichtungsmerkmalen oder auf der Grundlage eines Wirksamkeitsmaßes der Richtlinie an der Vorrichtung und der Erlaubnisanleitung gesteuert werden. Wie vorstehend erörtert, kann das Bilden einer Erlaubnisanleitung von Netzingenieuren erreicht werden, die eine Netzschnittstelle oder eine Anwendungsprogrammierschnittstelle verwenden können. Die Implementation der Erlaubnisanleitung und von Tokens kann eine detaillierte Steuerung des Netzverkehrs auf einer Pro-Vorrichtung-Basis ermöglichen. Bei einem Beispiel können Unternehmenssysteme Hypertext-Transfer-Protocol(HTTP)-Verkehr oder andere spezifische Verkehrstypen als Standard für Vorrichtungen ermöglichen. Unternehmenssysteme, die das DNAP-Protokoll verwenden, können auch Vorrichtungen mit festgelegten zusätzlichen Geschäftsfunktions-Tokens versehen, um andere Verkehrstypen zu ermöglichen, wenn diese Vorrichtungen den Wunsch haben können, andere Netzdienste zu verwenden.

[1031] In Block **18012** kann die Vorrichtung ein Paket auf dem Netz senden. Das Betriebssystem und die höheren Schichten des Open-System-Interconnection(OSI)-Stapels können sich dieses Prozesses nicht bewusst sein. Bei einem Beispiel kann das Senden des Vorrichtungspakets auf der Netzschnittstelle ausgeführt werden. Das Netz kann die Pakete auf mehrere Arten authentifizieren. Beispielsweise können die Tokens an den Header des Pakets angefügt werden, oder die Pakete können mit dem privaten Schlüssel der das Paket sendenden Identität signiert werden. Pakete, die am Netz ankommen, können zugelassen werden, falls die sie sendende Identität verifiziert werden kann und sie das Token für das Senden dieses Verkehrstyps besitzen. Falls der Verkehr nicht zulässig ist, können die Netzbetreiber entscheiden, eine negative Bestätigung (NACK) zum Client zurückzusenden, und andernfalls wird das Paket über das Netz zu seinem Ziel weitergeleitet.

[1032] Bei DNAP kann die Netzinfrastruktur selbst als die Validiererknoten in einer Blockchain als Ort, wo der Konsens über den Zustand des Systems gespeichert ist, wirken. Damit eine bösartige Entität diesen Ansatz beschädigen könnte, würde sie beispielsweise 51 % der Netzinfrastruktur beschädigen müssen. Das Be-

schädigen des größten Teils der Netzinfrastruktur kann zu einer höheren Last für bösartige Entitäten führen, weil es an Stelle eines einzelnen zentralisierten Firewall-Dienstes viele Stellen gibt, die beschädigt werden müssten. Der Konsens der Netzinfrastruktur kann eine Zugriffssteuerungsliste bzw. Access Control List (ACL)-Befehlsliste bzw. Command List (C-List) sein. Bei einem Beispiel könnten die vorstehend beschriebenen Verfahren, sobald ein Konsens der Netzinfrastruktur mit einem dezentralisierten Protokoll eingerichtet wurde, neu geschrieben oder auf die Behandlung der in der Blockchain gespeicherten ACL oder C-LIST abgebildet werden. Bei einem Beispiel könnte das DNAP-Protokoll eine Statusänderung auf der Grundlage einer Auslösung durch von Agenten mit einer gültigen Adresse im Protokoll signierte Transaktionen aktualisieren.

[1033] Wie hier in Bezug auf die Sicherheit und Kommunikationen mit DNAP verwendet, kann der Erzeuger einer Ressource Tokens ausgeben, können Tokens selbst übertragbar oder nicht übertragbar sein und können Tokens auf der Grundlage einer Anweisung von einem Netzbetreiber wie einmal verwendbare Berechtigungsnachweise verwendet werden. Bei Verwendung funktioneller DNAP-Tokens kann, sobald ein Token verwendet wurde, das Token nicht erneut verwendet werden, so dass in DNAP und ähnlichen Systemen verwendete Tokens wie Quoten für die Steuerung, wie viel Zugriff eine Vorrichtung auf das Netz erhält, verwendet werden können. Ein Token kann festgelegt werden, um für eine Anzahl X von Paketen oder ein Volumen X von Daten oder einen Zeitraum X zu funktionieren, oder es kann für einige Verkehrstypen eine unendliche Miete und für andere Quoten aufweisen.

[1034] **Fig. 181** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 18100 für die Verhandlung mit einem Wert versehener Dateneinheiten vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1035] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Vorrichtungsidentitätsgenerator **18102** zur Erzeugung einer Vorrichtungsidentität für eine Vorrichtung als Blockchain-Client aufweisen. Die Vorrichtung kann Tokens vom DNAP anfordern. Die Tokens können der Vorrichtung die Fähigkeit geben, Netzdaten anders als über Peer-to-Peer zu senden und zu empfangen. Bei einem Beispiel können die Tokens der Vorrichtung die Fähigkeit geben, Daten auf einer Schicht einer Open-System-Interconnection-Schicht eines Netzes zu senden und zu empfangen. Bei einem Beispiel kann die Vorrichtung einen Transaktionsdatensatz von der Vorrichtung empfangener und gesendeter Transaktionen speichern, wobei der Transaktionsdatensatz mit dem DNAP zu teilen ist. Die Vorrichtung kann Schlüssel zur Angabe des Ursprungs eines von der Vorrichtung gesendeten Pakets erzeugen. Die Vorrichtung kann eine Blockchain-fähige Vorrichtung sein, und sie kann von der Vorrichtung gesendete und empfangene Transaktionen in der Blockchain speichern. Die Beschreibungen der Vorrichtungsattribute können außerhalb einer Blockchain gespeichert werden.

[1036] Der Massenspeicher **808** kann einen Nachrichtenveröffentlicher **18104** zur Veröffentlichung einer Entdeckungsausstrahlungsnachricht von der Vorrichtung aufweisen. Der Massenspeicher **808** kann einen Netzbewerber **18106** aufweisen, damit sich die Vorrichtung bewerben kann, sich ansprechend darauf, dass die Vorrichtung auf der Grundlage der veröffentlichten Entdeckungsausstrahlungsnachricht eine Antwort vom DNAP empfängt, einem Dezentralisiertes-Netzzugangs-Proxy(DNAP)-Netz anzuschließen. Der Massenspeicher **808** kann einen Vorrichtungsbeschreiber **18108** zur Beschreibung der Identität und von Attributen der Vorrichtung für das DNAP aufweisen.

[1037] Der Massenspeicher **808** kann einen Paketabsender **18110** zum Senden eines Pakets von der Vorrichtung durch das Netz ansprechend darauf, dass der Vorrichtung durch das Netz auf der Grundlage der Identität und der Attribute der Vorrichtung Zugang gewährt wird, aufweisen. Bei einem Beispiel kann das Paket ein Token anhängen und kann die Kombination aus dem Paket und dem Token zur Überprüfung zum DNAP gesendet werden, wobei das DNAP sowohl das Paket als auch das Token ablehnt, wenn festgestellt wird, dass das Token nicht vom DNAP akzeptiert wird. Bei einem Beispiel kann das Token für die Verwendung mit einer Schwellenanzahl von Paketen, einem Schwellenvolumen von Daten und/oder einem Schwellenzeitraum gültig sein.

[1038] **Fig. 182** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **18200**, das Code aufweist, um Aufgaben zu definieren und Knoten zu kommissionieren, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1039] Das nichtflüchtige maschinenlesbare Medium **18200** kann einen Code **18202** aufweisen, um den Prozessor **902** anzuweisen, eine Vorrichtungsidentität für eine Vorrichtung als Blockchain-Client zu erzeugen. Die Vorrichtung kann Tokens vom DNAP anfordern. Die Tokens können der Vorrichtung die Fähigkeit geben, Netzdaten anders als über Peer-to-Peer zu senden und zu empfangen. Bei einem Beispiel können die Tokens der Vorrichtung die Fähigkeit geben, Daten auf einer Schicht einer Open-System-Interconnection-Schicht eines Netzes zu senden und zu empfangen. Bei einem Beispiel kann die Vorrichtung einen Transaktionsdatensatz von der Vorrichtung empfangener und gesendeter Transaktionen speichern, wobei der Transaktionsdatensatz mit dem DNAP zu teilen ist. Die Vorrichtung kann Schlüssel zur Angabe des Ursprungs eines von der Vorrichtung gesendeten Pakets erzeugen. Die Vorrichtung kann eine Blockchain-fähige Vorrichtung sein, und sie speichert von der Vorrichtung gesendete und empfangene Transaktionen in der Blockchain. Die Beschreibungen der Vorrichtungsattribute können außerhalb einer Blockchain gespeichert werden.

[1040] Das nichtflüchtige maschinenlesbare Medium **18200** kann einen Code **18204** aufweisen, um den Prozessor **902** anzuweisen, eine Entdeckungsausstrahlungsnachricht von der Vorrichtung zu veröffentlichen. Das nichtflüchtige maschinenlesbare Medium **18200** kann einen Code **18206** aufweisen, um den Prozessor **902** anzuweisen, dafür zu sorgen, dass sich die Vorrichtung bewerben kann, sich ansprechend darauf, dass die Vorrichtung auf der Grundlage der veröffentlichten Entdeckungsausstrahlungsnachricht eine Antwort vom DNAP empfängt, einem Dezentralisiertes-Netzzugangs-Proxy(DNAP)-Netz anzuschließen. Das nichtflüchtige maschinenlesbare Medium **18200** kann einen Code **18208** aufweisen, um den Prozessor **902** anzuweisen, die Identität und die Attribute der Vorrichtung für das DNAP zu beschreiben.

[1041] Das nichtflüchtige maschinenlesbare Medium **18200** kann einen Code **18210** aufweisen, um den Prozessor **902** anzuweisen, ansprechend darauf, dass der Vorrichtung durch das Netz auf der Grundlage der Identität und der Attribute der Vorrichtung Zugang gewährt wird, ein Paket von der Vorrichtung durch das Netz zu senden. Bei einem Beispiel kann das Paket ein Token anhängen und kann die Kombination aus dem Paket und dem Token zur Überprüfung zum DNAP gesendet werden, wobei das DNAP sowohl das Paket als auch das Token ablehnt, wenn festgestellt wird, dass das Token nicht vom DNAP akzeptiert wird. Bei einem Beispiel kann das Token für die Verwendung mit einer Schwellenanzahl von Paketen, einem Schwellenvolumen von Daten und/oder einem Schwellenzeitraum gültig sein.

[1042] Die Erlaubnisanleitung kann zur Bereitstellung einer dezentralisierten Autorisierung, Authentifizierung und Abrechnung für Vorrichtungen verwendet werden. Die vorliegende Offenbarung offenbart Baublöcke für eine Erweiterung des Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)- und des verwandten DIAMETER-Protokolls. Bei einem Beispiel adressieren die offenbarten Techniken Skalierbarkeitsprobleme, die durch zentral gesteuerte Systeme hervorgerufen werden. Diese Techniken können auf größere verteilte RADIUS-Netze angewendet werden. Bei einem Beispiel können Mitglieder großer Netze ihre eigenen RADIUS-Server auf ihrem Campus betreiben und dabei ihre eigenen Benutzerkonten aufrechterhalten. Bei einem Beispiel kann eine Authentifizierung durch RADIUS-Proxies geschehen, die eine Netzzugangsanforderung eines Mitglieds unabhängig vom Ort der Anforderung zum Netz des Mitglieds zurück leiten. Falls eine Anforderung, einem Netz beizutreten, an einem Mitgliedsnetz akzeptiert wird, akzeptiert der Rest des großen Netzes den Verkehr von diesem Ursprung als authentifiziert. Diese Technik ermöglicht es einem Netz, das Synchronisieren von Benutzerkonten über ein solches großes, verteiltes und dynamisches Netz zu vermeiden. Diese Technik kann erweitert werden, um einen Sicherheitsprüfungsprozess bereitzustellen, wenn eine neue Entität dem Netz beitrifft. Diese Technik kann erweitert werden, um eine Bestätigung bereitzustellen, dass eine Entität ihre RADIUS-Server sicher betreibt und mit den von der festgelegten Richtlinie festgelegten Kriterien übereinstimmt.

[1043] Fig. 183 ist ein schematisches Diagramm einer beispielhaften Organisation **18300** für eine dezentralisierte Version des Bereitstellens einer Authentifizierung, Autorisierung und Abrechnung mit einer Erlaubnisanleitung **18302** gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 165 offenbart wurden. Ein Prozess für die Funktion kann bei 18304 beginnen.

[1044] Die Organisation **18300** und das Verfahren können ein vollständiges System sein und auch eine Erweiterung existierender Autorisierungs-, Authentifizierungs- und Abrechnungsprotokolle sein. In Block **18304** kann sich ein Benutzer bei einer zentralisierten Stelle anmelden, wo er bereits ein Benutzer ist. Bei einem Beispiel kann der Benutzer ein Student oder ein Fachbereichsmitglied einer Universität sein und kann die zentralisierte Stelle ein Universitätsnetz sein. Während er angemeldet ist, kann ein Benutzer sein Profil erzeugen. Das Benutzerprofil, ein Passwort oder das Netz können zusammen verwendet werden, um an diesem Punkt eine Benutzeridentität beim System zu validieren. Falls der Benutzer eine Vorrichtung ist, kann die Vorrichtung,

statt sich bei einem Benutzerkonto anzumelden, booten und die Vorrichtungsauffertifizierung durch Module des Systems stellvertretend ausführen lassen.

[1045] In Block **18306** kann eine Vorrichtung eines Benutzers Zahlungen bei der Anweisung einer Verwendung austauschen. Bei einem Beispiel kann eine Vorrichtung eines Benutzers auf ein Bezahlung-für-Benutzungs-Netz zugreifen und kann eine Zahlung erforderlich sein, um auf das Netz zuzugreifen. Ein Benutzer kann durch eine Benutzervorrichtung eine Zahlung mit dem Netzbetreiber durch das Netz verhandeln. Diese Art der Zahlung kann optional sein, wobei der Netzanbieter beispielsweise freien Zugang bereitstellen kann. Ein Netzanbieter kann wählen, ob er Gebühren berechnen möchte, und beim Berechnen von Gebühren kann der Netzanbieter die Zahlungsformen, die er akzeptieren kann, spezifizieren. Bei einem Beispiel kann das Netz Kryptowährungen oder Infocoin akzeptieren. Ein Zahlungsaustausch, wie mit Bezug auf **18306** beschrieben, kann wie aufgelistet oder am Ende des Prozesses geschehen, wenn die beitretende Entität Bedingungen akzeptiert hat und der Anbieter der beitretenden Entität den Eingangszugang für den Beitritt erlaubt.

[1046] In Block **18308** können private Schlüssel für den Benutzer erzeugt werden und mit einer Adresse assoziiert werden. In Block **18310** kann eine Benutzervorrichtung fordern, einer Erlaubnisanleitung beizutreten. Wie vorstehend in Block **16518** beschrieben wurde, kann das Beitreten zur Erlaubnisanleitung ein Geschehnis sein, das einmal geschieht, wobei die Benutzervorrichtung zu einem permanenten Mitglied eines Netzes werden kann. Bei einem Beispiel kann der Beitritt zu einer Erlaubnisanleitung zeitlich oder durch andere Bedingungen begrenzt sein. Wie vorstehend beschrieben, könnte eine Erlaubnisanleitungs-Beitrittsfunktion gewährleisten, dass bestimmte Bedingungen erfüllt sind, bevor Bewerber akzeptiert werden. Bei einem Beispiel kann, falls eine Zahlung gemacht wurde, die Zahlung abgeschlossen werden oder nicht abgeschlossen werden, bevor der gesamte Prozess des Beitretens zur Erlaubnisanleitung abgeschlossen wurde.

[1047] Wie vorstehend beschrieben wurde, kann in Block **16522** die Liste teilnehmender Identitäten gespeichert werden. Die Speicherung teilnehmender Identitäten kann außerhalb der Chain erfolgen. Die Speicherung kann auch in einem Hash erfolgen. Ferner kann in Bezug auf die Speicherung teilnehmender Identitäten ein Zeiger verwendet werden, um einen Ort zu identifizieren, wo die Identitätsinformationen gespeichert werden könnten. Die gespeicherten Daten können auch verschlüsselt werden, und ihre Betrachtung kann auf autorisierte Entitäten begrenzt werden.

[1048] In Block **18312** kann ein Attributfilter verwendet werden, um Attribute zu validieren. Bei einem Beispiel kann die Validierung mit einem kenntnisfreien Beweismechanismus erfolgen. Die Validierung von Attributen kann eine Attestierung verwenden. Bei einem Beispiel kann das Attributfilter Bedingungen für den Betrieb auf dem Netz validieren, beispielsweise identifizieren, ob eine Person älter als 18 Jahre ist. Das Attributfilter kann die Attestierung eines Attributs für eine Person erlauben, ohne dass die Person ihre vollständige Identität offenbaren müsste.

[1049] In Block **16530** kann eine Bewerbervorrichtung wie zuvor Tokens anfordern. Wie zuvor können Tokens unbegrenzt oder begrenzt sein. Tokens können durch Kryptowährungs-Münzen gestützt werden oder nicht. Die Verwendung von Tokens kann eine Mischung erlauben, wobei einige Tokens eine Zahlung für den Erwerb verwenden können und andere frei sind, wie es durch einen Netzbetreiber entschieden wird. Die Anforderung von Tokens kann mit zusätzlichen Schritten verbunden sein, welche durch eine Blockchain **18314**, die Abrechnungsfunktionen ausführt, und eine Sidechain **18316** der Blockchain **18314** laufen. In Block **18318** reserviert eine Zahlung oder ein Funktionsaufruf von der Erlaubnisanleitung **18302** innerhalb der Blockchain **18314** Münzen auf der Blockchain. In Block **18320** können innerhalb der Sidechain **18316** reservierte Tokens der Sidechain **18316** zugeordnet werden, worin Tokens erzeugt werden. Die Aktion des Reservierens von Münzen oder des Erzeugens von Tokens in einer Sidechain **18316**, wobei die Tokens zur Blockchain hinzugefügt werden können, bildet eine Form der Abrechnung, wobei identifiziert und konstruiert werden kann, welche Identitäten welche Sorten von Tokens angefordert haben.

[1050] In Block **16534** können wie oben Tokens durch Verwirklichen einer Richtlinie der Erlaubnisanleitung **18302** widerrufen werden. Bei einem Beispiel kann von einer Entität die Rückerstattung von Tokens gefordert werden, falls sie die Erlaubnisanleitung **18302** verlassen möchte. Ansprechend auf eine Forderung von der Erlaubnisanleitung **18302** können die Tokens in Block **18322** von der Sidechain **18316** gelöscht werden. In Block **18324** können innerhalb der Blockchain **18314** jegliche Münzen in Zusammenhang mit den gelöschten Tokens in der Sidechain **18316** einem Dienstanbieter freigegeben werden oder der Entität rückerstattet werden, wobei dies vom Grund für die Transaktion abhängt.

[1051] In Block **16534** können wie oben Zahlungs-T&C, welche der Netzanbieter geltend macht, in die Erlaubnisanleitung **18302** codiert werden. In Block **18326** kann eine Authentifizierungsanforderung gesendet werden. Bei einem Beispiel funktioniert die Authentifizierung dadurch, dass die Vorrichtung eine Anforderung zum Netz sendet. Eine Vorrichtung kann der verifizierenden Partei einen öffentlichen Schlüssel präsentieren. Bei einem Beispiel kann die Partei, die den öffentlichen Schlüssel gesendet hat, in der Blockchain **18314** prüfen, um festzustellen, ob Tokens einem solchen öffentlichen Schlüssel gutgeschrieben werden. Für den Zugang zu verschiedenen Diensten auf dem Netz kann der Halter verschiedene Token-Typen besitzen müssen.

[1052] In Block **18328** können Tokens verbraucht werden. Bei einem Beispiel können Tokens auf einer Verwendungsbasis verbraucht werden. Die Verwendung von Tokens auf einer Verwendungsbasis kann eine Autorisierungsform sein, die dem Netzanbieter ein Verfahren bereitstellt, Entitäten auf ihrem Netz auf einer Pro-Dienst-Basis Budgets zuzuordnen. Der Anbieter kann stattdessen angeben, dass Tokens nicht pro Verwendung gelten und ohne Einschränkung durch die Verwendung eingesetzt werden können. In Block **18330** kann der Verbrauch oder die Präsentation von Tokens durch die Sidechain **18316** als Transaktionen auf den Sidechains aufgezeichnet werden. Diese Aufzeichnung kann als ein anderer Abrechnungsdienst angesehen werden. In Block **18316** kann die Sidechain angeben, ob Tokens verbraucht werden. Falls Tokens verbraucht werden, kann es eine Aufzeichnung dieses Verbrauchs in der Sidechain **18316** geben. Bei einem Beispiel können auf der Sidechain **18316** verbrauchte Tokens durch Münzen auf der Haupt-Blockchain **18314** gestützt werden. In Block **18332** können auf der Blockchain **18314** Münzen wieder freigegeben werden oder zu einem Netzbetreiber und zum Wallet eines Anbieters zurückgezahlt werden.

[1053] Fig. **184** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **18400** für eine dezentralisierte Version zur Bereitstellung von Authentifizierung, Autorisierung und Abrechnung mit einer Erlaubnisanleitung gemäß einigen Ausführungsformen. Das Verfahren **18400** aus Fig. **184** kann durch die mit Bezug auf Fig. **185** beschriebene IoT-Vorrichtung **18500** implementiert werden. Der Prozessfluss kann in Block **18402** beginnen.

[1054] In Block **18402** können Entitäten, welche die Verwendung des Netzes fordern, sich beispielsweise durch ein Portal oder eine API registrieren. Bei einem Beispiel kann ein Portal durch einzelne Universitäten eingeschriebenen Studenten für die Registrierung und Gebührenzahlung bereitgestellt werden. Bei einem Beispiel könnten für Entitäten, die einem maschinenorientierten Netz beitreten möchten, diese Entitäten automatisch unter Verwendung von Guthaben aus einem Wallet oder Kreditkartenanbieter beitreten.

[1055] In Block **18404** kann ein Zahlungsaustausch verwendet werden, falls die beitretende Entität auf dem Netz, dem sie beitreten möchte, kein Guthaben hat. In Block **18406** können beitretende Entitäten in einen Smart Contract aufgenommen werden, indem sie an einem Zahlungsaustausch teilnehmen. Bei einem Beispiel können die Attribute der beitretenden Entitäten registriert werden. Bei einem Beispiel können Attribute für eine Verwendung das Geburtsdatum und andere persönliche Daten einschließen. Bei einem Beispiel können Attribute für Maschinen den Vorrichtungstyp oder den Typ und die Version von Software einschließen. Attributdaten können attestiert werden, falls die Attributdaten mit einer unterstützenden Dokumentation eingereicht werden. Im Fall von Attributen für Maschinen können diese Maschinenattribute durch technische Verfahren, einschließlich einer vertrauenswürdigen Mess- oder Hardware-Vertrauens-Root bzw. Sensing or Hardware Root of Trust (HWROT), attestiert werden. Ansprechend auf diese Attestierung kann eine Liste der teilnehmenden Entitäten in der Erlaubnisanleitung gehalten werden und können die Entitäten nun Tokens von der Erlaubnisanleitung anfordern.

[1056] In Block **18408** können Tokens einer Entität, ansprechend auf eine Bestätigung einer gültigen Identitätsanforderung, wie durch ein Konsensnetz festgestellt, erteilt werden. Tokens können einer Entität ansprechend darauf erteilt werden, dass das Guthaben einer Identität auf dem Netz größer als null ist. Bei einem Beispiel kann eine TTL für die Attestierung für die Entität festgelegt werden. Bei einem Beispiel kann die Attestierung für die Entität durch die Zeit, die Verwendung und geographisch begrenzt werden. Bei einem Beispiel können die Grenzen durch Tokens erzwungen werden, weil Tokens in einigen Gebieten funktionieren können und in anderen nicht, falls die Entität mobil ist.

[1057] In Block **18410** können Münzen gegen Tokens reserviert werden. Bei einem Beispiel können Münzen in einer Sidechain reserviert werden. Der Prozess des Reservierens von Münzen kann ansprechend auf nicht erfolgreiche Versuche erneut versucht werden. Bei einem Beispiel kann der Prozess auch das Zurücktreten von der Transaktion aufweisen, wobei beim Prozess ausgetauschtes Guthaben erstattet wird. Falls ein Versuch zum Reservieren von Münzen erfolgreich ist, können Tokens erzeugt und an die Entität ausgegeben werden, welche dann Authentifizierungsanforderungen senden kann. Authentifizierungsanforderungen können wie zuvor beschrieben attributgefiltert werden. Wenn Tokens verbraucht werden, können die ihnen in der Sidechain

zugeordneten Münzen entsperrt oder freigegeben werden, und diese Tokens können an einen Netzanbieter weitergeleitet werden.

[1058] In Block **18412** kann eine Entität eine Erlaubnisanleitung verlassen. Um das Verlassen einer Erlaubnisanleitung zu vermeiden, kann eine Entität zusätzliche Tokens anfordern, falls die Tokens der Entität bereits verbraucht sind. Bei einem Beispiel kann die Erlaubnisanleitung enden, falls die Identität der Entitäten auf dem Netz nicht mehr gültig ist. Bei einem Beispiel kann die Netzentität oder der Netzanbieter auch diesen Prozess einleiten, um die Entität auszuschließen. Nicht verbrauchte Tokens können widerrufen oder zerstört werden, und verbleibende Guthabenbeträge für eine Entität können gemäß den Bedingungen der Erlaubnisanleitung erstattet werden.

[1059] **Fig. 185** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **18500** für eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1060] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Vorrichtungsregistrator **18502** zur Registrierung einer Vorrichtung in einem ersten Netz durch ein Portal zu einem zweiten Netz aufweisen, wobei das zweite Netz autorisiert ist, auf das erste Netz zuzugreifen. Die Vorrichtung kann einen Zahlungsaustausch zu einem Wallet im zweiten Netz ausführen.

[1061] Der Massenspeicher **808** kann einen Vorrichtungsaufnehmer **18504** zum Aufnehmen einer Vorrichtung in eine Erlaubnisanleitung durch Zustimmung zu Verpflichtungen der Erlaubnisanleitung aufweisen. Der Massenspeicher **808** kann einen Token-Antragsteller **18506** zur Anforderung eines Tokens unter Verwendung einer Funktion der Erlaubnisanleitung aufweisen, wobei das Token die Vorrichtung als für den Zugang zum zweiten Netz authentifiziert identifiziert. Bei einem Beispiel kann die Anforderung für das Token dazu führen, dass eine Reservierung einer Münze auf einer abrechnenden Blockchain einem auf einer Sidechain erzeugten Token entspricht. Eine Münze der Blockchain kann ansprechend darauf freigegeben werden, dass erkannt wurde, dass ein Token von einer Sidechain widerrufen und/oder verbraucht wurde. Bei einem Beispiel können beim Beitreten zur Erlaubnisanleitung von der Vorrichtung Attribute der Vorrichtung der Erlaubnisanleitung bereitgestellt werden, damit ein Attributfilter validiert, dass die Attribute der Vorrichtung im ersten Netz erlaubt sind. Die Attribute können ein Attribut eines aktiven Benutzerprofils aufweisen, während die Vorrichtung der Erlaubnisanleitung beitrifft. Das Token kann sich ansprechend auf seine Verwendung als Form der Autorisierung für die Vorrichtung zerstören.

[1062] Der Massenspeicher **808** kann eine Anforderungsabsender **18508** zum Senden einer Authentifizierungsanforderung von der Vorrichtung zum ersten Netz aufweisen, wobei das erste Netz die Authentifizierung ansprechend auf die Erkennung des Tokens bestätigt. Das Token kann ansprechend auf die Authentifizierung der Vorrichtung durch Präsentieren des Tokens durch die Vorrichtung dem ersten Netz auf einer Sidechain verbraucht werden. Die Vorrichtung kann autorisiert werden, auf der Grundlage der Authentifizierung beim ersten Netz, dass die Vorrichtung Berechtigungsnachweise hat, auf das zweite Netz zuzugreifen, auf das erste Netz zuzugreifen. Bei einem Beispiel kann die Autorisierung der Vorrichtung zur Verwendung des ersten Netzes auf der Grundlage der Anzahl der Zugriffe, des Datenvolumens, worauf durch das erste Netz zugegriffen wird, und/oder der Zeit des gewährten Zugriffs verstreichen.

[1063] **Fig. 186** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **18600**, das einen Code zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung aufweist, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1064] Das nichtflüchtige maschinenlesbare Medium **18600** kann einen Code **18602** aufweisen, um den Prozessor **902** anzuweisen, eine Vorrichtung durch ein Portal zu einem zweiten Netz bei einem ersten Netz zu registrieren, wobei das zweite Netz berechtigt ist, auf das erste Netz zuzugreifen. Die Vorrichtung kann einen Zahlungsaustausch zu einem Wallet im zweiten Netz ausführen.

[1065] Das nichtflüchtige maschinenlesbare Medium **18600** kann einen Code **18604** aufweisen, um den Prozessor **902** anzuweisen, eine Vorrichtung einer Erlaubnisanleitung durch Zustimmung zu Verpflichtungen der Erlaubnisanleitung anzuschließen. Das nichtflüchtige maschinenlesbare Medium **18600** kann einen Code **18606** aufweisen, um den Prozessor **902** anzuweisen, ein Token unter Verwendung einer Funktion der Erlaubnisanleitung anzufordern, wobei das Token die Vorrichtung als für den Zugriff auf das zweite Netz authentifiziert identifiziert. Bei einem Beispiel kann die Anforderung für das Token dazu führen, dass eine Reservierung einer Münze auf einer abrechnenden Blockchain einem auf einer Sidechain erzeugten Token entspricht. Eine Münze der Blockchain kann ansprechend darauf freigegeben werden, dass erkannt wurde, dass ein Token von einer Sidechain widerrufen und/oder verbraucht wurde. Bei einem Beispiel können beim Beitreten zur Erlaubnisanleitung von der Vorrichtung Attribute der Vorrichtung der Erlaubnisanleitung bereitgestellt werden, damit ein Attributfilter validiert, dass die Attribute der Vorrichtung im ersten Netz erlaubt sind. Die Attribute können ein Attribut eines aktiven Benutzerprofils aufweisen, während die Vorrichtung der Erlaubnisanleitung beitrifft. Das Token kann sich ansprechend auf seine Verwendung als Form der Autorisierung für die Vorrichtung zerstören.

[1066] Das nichtflüchtige maschinenlesbare Medium **18600** kann einen Code **18608** aufweisen, um den Prozessor **902** anzuweisen, eine Authentifizierungsanforderung von der Vorrichtung zum ersten Netz zu senden, wobei das erste Netz die Authentifizierung ansprechend auf die Erkennung des Tokens bestätigt. Das Token kann ansprechend auf die Authentifizierung der Vorrichtung durch Präsentieren des Tokens durch die Vorrichtung dem ersten Netz auf einer Sidechain verbraucht werden. Die Vorrichtung kann autorisiert werden, auf der Grundlage der Authentifizierung beim ersten Netz, dass die Vorrichtung Berechtigungsnachweise hat, auf das zweite Netz zuzugreifen, auf das erste Netz zuzugreifen. Bei einem Beispiel kann die Autorisierung der Vorrichtung zur Verwendung des ersten Netzes auf der Grundlage der Anzahl der Zugriffe, des Datenvolumens, worauf durch das erste Netz zugegriffen wird, und/oder der Zeit des gewährten Zugriffs verstreichen.

[1067] Einige Ausführungsformen der vorliegenden Techniken offenbaren eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung an einer IoT-Vorrichtung beispielsweise unter anderem unter Verwendung eines Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)- und/oder eines DIAMETER-Protokolls. Ein dezentralisiertes Proxy kann vor einem RADIUS-Server, einem DIAMETER-Server oder einem ein DIAMETER-Protokoll ausführenden RADIUS-Server sitzen. Eine dezentralisierte API kann in einen RADIUS-Dienst und/oder einen DIAMETER-Dienst eingebaut sein. Existierende Aufrufe können in einem Blockchain-Verschlüsselungsmechanismus in einen RADIUS-Dienst und/oder einen DIAMETER-Dienst eingepackt sein. Der Blockchain-Verschlüsselungsmechanismus kann als Beweisschicht der Quelle einer Anforderung verwendet werden, um beispielsweise zu ermöglichen, dass die Anforderung durch die Verarbeitung durch den RADIUS-Server und/oder den DIAMETER-Server läuft.

[1068] Fig. 187 ist ein schematisches Diagramm einer Technik **18700** zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung unter Verwendung eines Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)-Protokolls und/oder eines DIAMETER-Protokolls gemäß einigen Ausführungsformen. Der RADIUS-Server **18702** kann gegen Modifikationen gesperrt sein, während die Funktionalität des dezentralisierten RADIUS-Proxy **18704** erweitert werden kann. Das dezentralisierte RADIUS-Proxy **18704** kann Maßnahmen ergreifen, bevor eine Nachricht an einen traditionellen RADIUS-Server ankommt. Eine dezentralisierte API **18706** kann zwischen dem RADIUS-Server **18702** und einer Backend-Datenbank **18708** eingefügt werden und Modifikationen für den Betrieb des RADIUS-Dienstes aufweisen.

[1069] Das dezentralisierte RADIUS-Proxy **18704** kann wirken, wenn der RADIUS-Server **18702** eine Backend-Datenbank **18708** implementiert. Bei einem Beispiel kann die Datenbank **18708** eine Datei sein oder jegliche einer Anzahl unterstützter Datenspeicher verwenden. Beim dezentralisierten RADIUS-Proxy **18704** kann ein Dienst vor dem RADIUS-Server **18702** sitzen und als ein dezentralisiertes Filter wirken. Das dezentralisierte RADIUS-Proxy **18704** kann eine Sicherheitsprüfung bereitstellen, indem die Identität eines Antragstellers unter Verwendung dezentralisierter Mechanismen bestätigt wird.

[1070] Die Aufrufe, die der RADIUS-Server **18702** verwendet, können modifiziert werden, um sie durch eine dezentralisierte API **18706** zu leiten. Die dezentralisierte API **18706** kann als Satz von Klassen, welche die Weiterleitung von RADIUS-Funktionen zu einer Blockchain unterstützen, in die Codebasis des RADIUS-Servers aufgenommen sein. Der RADIUS-Server **18706** kann zu einem Blockchain-Client werden und Identitäts- und Transaktionsgültigkeitsprüfungen ausführen. Alternativ oder zusätzlich könnten Identitäts- und Gültigkeitsprüfungen als externer Dienst implementiert werden, für deren Unterstützung der RADIUS-Server modifiziert ist. Mit der dezentralisierten API kann der Code des RADIUS-Servers modifiziert werden, so dass die Opera-

tion eine Identitäts- und Gültigkeitsprüfungsfunktionalität ermöglichen kann. Beispielhafte Mechanismen zur Ausführung der Gültigkeitsprüfungen werden nachstehend beschrieben.

[1071] Fig. 188 ist ein Leiterdiagramm eines beispielhaften Verfahrens 18800 für die Komponenten aus Fig. 187, um durch ein dezentralisiertes RADIUS-Proxy 18704 zur Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung zu wirken, gemäß einigen Ausführungsformen. Das Verfahren 18800 aus Fig. 188 kann durch die mit Bezug auf Fig. 191 beschriebene IoT-Vorrichtung 19100 implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 187 offenbart wurden.

[1072] Das dezentralisierte RADIUS-Proxy 18704 kann eine RADIUS-Authentifizierungsanforderung 18802 von einem RADIUS-Client 18804 verarbeiten. Der RADIUS-Client 18804 kann modifiziert werden, um die RADIUS-Anforderung mit einem privaten Schlüssel von der Blockchain des RADIUS-Clients oder von einer Identität eines verteilten Ledgers 18806 zu signieren. Eine Identität kann verwendet werden, um die Identität 18808 der Quelle der Anforderung zu verifizieren und zu verifizieren, ob die Anforderung tatsächlich der Halter des privaten Schlüssels entsprechend der Identität auf der Blockchain oder dem verteilten Ledger 18806 sein kann. Die Identität auf der Blockchain oder dem verteilten Ledger 18806 kann zuvor unter Verwendung einer Erlaubnisanleitung festgelegt worden sein, wie in vorhergehenden Abschnitten beschrieben. Beispielsweise kann sich die Identität für einen Dienst registriert haben, einer Erlaubnisanleitung beigetreten sein und als eine teilnehmende Entität innerhalb dieses Vertrags aufgelistet sein, oder die Identität kann auch ein Token-Halter sein. Die Identitätsverifizierung könnte bei der Laufzeit erfolgen, wobei eine Blockchain oder ein verteilter Ledger 18806 die Identität der Anforderung akzeptieren können, wenn zum ersten Mal eine durch eine neue Identität signierte Authentifizierungsanforderung gesehen wird. Eine Identitätsverifizierungsantwort 18810 kann zum dezentralisierten Proxy 18704 zurückgegeben werden. Ansprechend darauf, dass eine Identität als akzeptabel verifiziert wurde, kann das dezentralisierte Proxy 18704 einen geeigneten RADIUS-Server anfordern 18812. Ansprechend darauf kann der RADIUS-Server 18702 antworten 18814, dass die Anforderung als Erfolg genehmigt oder als Nichterfolg abgelehnt wurde.

[1073] Eine erfolgreiche Verifizierung der Identität kann mehrere Identitäten verknüpfen, so dass künftige RADIUS-Anforderungen vom selben Benutzer mit dem korrekten privaten Schlüssel signiert werden können, während den privaten Schlüssel nicht aufweisende Anforderungen abgelehnt werden können. Die Identität kann ein Token mit einer RADIUS-Anforderung präsentieren und durch Vergleichen der Verifizierung mit der Blockchain oder dem Ledger antworten, um die Identität zu validieren. Wie zuvor kann eine Validierung einen Anfordernden als Halter eines gültigen Tokens angeben und kann bei einer nicht erfolgreichen Validierung die Identität noch verifiziert werden, indem sie als Mitglied einer bestimmten Erlaubnisanleitung aufgelistet wird. Die Validierung kann angeben, wann eine Währung auf der Blockchain gegenüber der RADIUS-Anforderung ausgegeben wird. Beispielsweise kann die Identität, um eine RADIUS-Anforderung zu machen, einige Guthaben und Münzen auf der Blockchain ausgeben müssen.

[1074] Fig. 189 ist ein Leiterdiagramm eines beispielhaften Verfahrens 18900 für die Komponenten aus Fig. 187, um durch eine dezentralisierte API 18706 zur Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung zu wirken, gemäß einigen Ausführungsformen. Das Verfahren 18900 aus Fig. 189 kann durch die mit Bezug auf Fig. 191 beschriebene IoT-Vorrichtung 19100 implementiert werden. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 187 und Fig. 188 offenbart wurden.

[1075] Die Aufrufsequenz unterscheidet sich von der Aufrufsequenz mit Bezug auf Fig. 188, weil die Aufrufe, wenngleich sie in der Substanz ähnlich sind, andere Akteuren adressieren können. Beispielsweise kann in Fig. 189 die signierte Autorisierungsanforderung 18902 vom RADIUS-Client 18804 zum RADIUS-Server 18702 gerichtet sein. Eine Verifizierung der Identität 18904 kann vom RADIUS-Server 18702 auf die dezentralisierte API 18706 gerichtet sein. Eine zweite Anforderung 18906 zur Verifizierung der Identität kann von der dezentralisierten API 18706 und zum verteilten Ledger 18806 gesendet werden. Ansprechend darauf kann der verteilte Ledger 18806 eine Identitätsantwort 18908 zur dezentralisierten API 18706 zurückgeben, wodurch entweder der Erfolg oder der Nichterfolg der Identitätsverifizierung angegeben wird. Ansprechend darauf kann die dezentralisierte API 18706 eine zweite Identitätsantwort 18910 zum RADIUS-Server 18702 zurückgeben, wodurch entweder der Erfolg oder der Nichterfolg der Identitätsverifizierung angegeben wird. Der RADIUS-Server 18702 kann eine Anforderung bzw. Antwort des RADIUS-Servers zum RADIUS-Client 18804 zurückgeben. Ein Ergebnis dieser Aktionen kann die Validierung der Identität der Quelle der RADIUS-Anforderung sein, wobei die Validierung beispielsweise durch eine Blockchain oder einen dezentralisierten Ledger geführt werden kann, bevor die Anforderung zur Validierung verarbeitet werden kann.

[1076] Fig. 190 ist ein schematisches Aktionsdiagramm **19000** zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung auf einer IoT-Vorrichtung gemäß einigen Ausführungsformen. Eine Autorisierungsanforderung **19002** interagiert mit einem Transaktionsvalidierungsprüfer **19004**, der eine Blockchain **19006** verwendet.

[1077] Innerhalb der Autorisierungsanforderung **19002** kann in Block **19008** der Transaktionsinhalt zu einer Nachricht hinzugefügt werden. Beim in Fig. 190 dargestellten Beispiel kann der Transaktionsinhalt ein Benutzername und ein Passwort sein, beispielsweise der Benutzername und Berechtigungsnachweise für „Mike“. Die sensiblen Informationen werden bei diesem Verfahren, wie nachstehend beschrieben, dritten Parteien nicht offengelegt. Die Transaktion kann Metadaten aufweisen. Die Metadaten können in einem öffentlichen Ledger gespeichert werden. Falls Geld- oder Kryptowährungs-Nennwerte Teil der Transaktion sind, kann der Transaktionsinhalt die Einzelheiten darüber aufweisen, wie viel Wert ausgetauscht wird. Die Gültigkeit der Transaktion kann davon abhängen, ob die Bedingungen der Transaktion erfüllt werden. Beispielsweise können beim vorstehend beschriebenen Beispiel erfüllte Bedingungen der Transaktion Zahlungsaktionen und Authentifizierungsaktionen einschließen.

[1078] Innerhalb der Autorisierungsanforderung **19002** kann in Block **19010** eine angeforderte Netzadresse in den Transaktionsinhalt aufgenommen werden. An Stelle der Netzadresse kann eine angeforderte Ressource in den Transaktionsinhalt aufgenommen werden. Die Netzadresse kann beispielsweise ein voll qualifizierter Domainname (FDQN) oder eine Internetprotokoll(IP)-Adresse für einen RADIUS-Server sein. Die Netzadresse kann eine Ressource auf dem Netz sein. Die Netzadresse kann eine Wallet-Adresse auf der Grundlage des privaten Schlüssels des RADIUS-Servers oder des Netzressourcen-Eigentümers einschließen. Eine Netzadresse kann das Wallet ansprechend auf eine für die Verwendung des Dienstes geforderte Zahlung aufweisen.

[1079] Innerhalb der Autorisierungsanforderung **19002** kann der Transaktionsinhalt in Block **19012** durch den privaten Schlüssel der Partei signiert werden. Beim Prozess des Signierens des Inhalts einer Transaktion kann eine Signatur gebildet werden **19014**, kann eine Referenz auf den Ort des öffentlichen Schlüssels aufgenommen werden **19016** oder könnte die Transaktion den öffentlichen Schlüssel selbst enthalten und den öffentlichen Schlüssel der Autorisierungsanforderung **19002** selbst bereitstellen **19018**.

[1080] Innerhalb des Transaktionsvalidierungsprüfers **19004** kann eine Anforderung zum Verifizieren eines öffentlichen Schlüssels **19020** vorgenommen werden. Der Ort des öffentlichen Schlüssels kann von der Blockchain **19006** nachgeschlagen **19022** oder angefordert werden. Ein Netzeigentümer kann die Blockchain **19006** erzeugen, und Entitäten können Identitäten auf der Blockchain **19006** kaufen oder erhalten, indem sie der Blockchain **19006** einen öffentlichen Schlüssel der Entität mitteilen. Die Mitteilung eines öffentlichen Schlüssels für eine Entität für die Blockchain **19006** während der Verhandlung kann im Austausch gegen Kryptowährungen, Tokens oder eine andere Bezahlung erfolgen. Der Zahlungsbetrag kann festlegen, wie lange ein Schlüssel in der Blockchain **19006** gehalten werden kann. Ein Schlüssel kann von einer Blockchain **19006** unbegrenzt oder während eines spezifizierten Zeitraums gehalten werden. Bedingungen für die Einrichtung oder Bestätigung einer Identität können von einem Netzadministrator angepasst werden.

[1081] Die Blockchain **19006** kann eine Anzahl von Blöcken **19024** aufweisen. Die für das Speichern von Identitäten verwendete Blockchain **19006** kann eine virtuelle Blockchain sein, die auf einer größeren Blockchain sitzt, welche eine kritische Masse von Minern aufweist. Die Blockchain kann beispielsweise das Konzept des dualen Minings aufweisen, wobei die für den Beweis in einer Blockchain **19006** ausgeführte Arbeit auch als Beweis in der anderen dient. Das Nachschlagen **19022** kann beispielsweise unter Verwendung eines vorstehend offenbarten Bloomfilter-Hop-Verfahrens erfolgen. Ein Ergebnis des Nachschlagens **19022** kann darin bestehen, dass ein öffentlicher Schlüssel bekannt ist. Ein Ergebnis des Nachschlagens **19022** kann darin bestehen, dass der Schlüssel von Anfang an in die Transaktion aufgenommen wurde.

[1082] Innerhalb des Transaktionsvalidierungsprüfers **19004** kann der Schlüssel in Block **19026** die Transaktion entschlüsseln und die identifizierte Entität bestätigen. Der Schlüssel kann im Fall eines asymmetrischen Schlüssels öffentlich sein oder im Fall eines symmetrischen Schlüssels privat sein. Nachrichtenkommunikationen verwenden im Allgemeinen private symmetrische Schlüssel für die Verschlüsselung/Entschlüsselung. Eine Transaktion kann an die Blockchain **19006** übergeben werden. Eine Transaktion kann eine Referenz zu einem Speichermechanismus außerhalb der Chain sein. Der Speichermechanismus außerhalb der Chain kann in Block **19026** verwendet werden, um das Ergebnis des Identitätsverifizierungsschritts aufzuzeichnen. Die Aufzeichnung eines Ergebnisses des Identitätsverifizierungsschritts kann eine Abrechnung bereitstellen. Eine Aufzeichnung kann beispielsweise an die Blockchain **19006** von Netzanbietern und/oder an eine virtuelle Blockchain übergeben werden. Aufzeichnungen auf der Blockchain **19006** können in manchen Fällen auf

Metadaten über die Transaktion begrenzt werden. In manchen Fällen kann verhindert werden, dass Informationen in Bezug auf Benutzernamen und Passwörter in die Blockchain **19006** aufgenommen werden. Falls die Informationen in die Blockchain **19006** aufgenommen werden, können sie Teil der Transaktionen zwischen einem RADIUS-Proxy und/oder einem modifizierten RADIUS-Server sein.

[1083] Innerhalb des Transaktionsvalidierungsprüfers **19004** kann ein Ereignis außerhalb der Chain in Block **19028** auftreten, wobei Inhalte der Transaktion zur normalen Verarbeitung zum RADIUS-Server weitergeleitet werden können, falls die Transaktionsidentität gültig ist. Im Fall einer Authentifizierungsanforderung kann der Inhalt beispielsweise einen Benutzernamen und ein Passwort aufweisen. Die Übergabe des Inhalts an einen Server kann zwischen dem RADIUS-Server und seinem Proxy oder dem modifizierten dezentralisierten Code innerhalb des RADIUS-Proxys stattfinden.

[1084] Innerhalb des Transaktionsvalidierungsprüfers **19004** kann ein Ereignis außerhalb der Chain in Block **19030** stattfinden, wobei eine Antwort vom RADIUS-Server direkt zu einem Client zurückgeleitet werden kann. Die Weiterleitung einer Antwort kann durch ein Proxy und/oder durch einen RADIUS-Server erfolgen, wobei dies teilweise von einer Implementationsarchitekturwahl abhängt. Der RADIUS-Server kann eine Protokollierung und Abrechnung von Anforderungen, die er empfängt, ausführen.

[1085] Innerhalb des Transaktionsvalidierungsprüfers **19004** kann in Block **19032** eine Antwort zurückgeleitet werden. Die Antwort kann positiv oder negativ sein. Die Antwort kann als unveränderlicher Datensatz in der Blockchain **19006** gespeichert werden. Durch Speichern einer Antwort auf der Blockchain **19006** kann es für einen Übeltäter erschwert werden, seine Aktionen zu verbergen.

[1086] **Fig. 191** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **19100** für eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **19100** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1087] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation einer dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1088] Der Massenspeicher **808** kann einen Identitätsverifizierer **19102** zum Verifizieren der Identität einer Authentifizierungsanforderung mit einer dezentralisierten API aufweisen, wobei die Authentifizierungsanforderung von einem RADIUS-Client empfangen wird und die dezentralisierte API die Identität durch Senden einer Anforderung an einen verteilten Ledger und Zurückgeben einer Antwort ansprechend auf den Empfang einer Identitätsverifizierungsantwort vom verteilten Ledger an einen RADIUS-Server verifiziert. Der RADIUS-Client kann ansprechend auf eine Antwort der authentifizierten Identität eine Transaktion vornehmen. Die Transaktion kann einen Benutzernamen, ein Passwort und/oder Metadaten aufweisen. Die Transaktion kann eine Werttransaktion einschließen. Die Transaktion kann eine Kryptowährungstransaktion sein. Die Authentifizierungsanforderung kann eine Anforderung einer Netzadresse einschließen. Die Netzadresse kann einen voll qualifizierten Domainnamen für den RADIUS-Server und/oder eine Internetprotokolladresse für den RADIUS-Server aufweisen. Der RADIUS-Server kann einen öffentlichen Schlüssel durch Anfordern des Orts des öffentlichen Schlüssels von einer Blockchain verifizieren. Die Anforderung an einen RADIUS-Server kann außerhalb der Chain ansprechend darauf geschehen, dass ein RADIUS-Client eine Bestätigung der authentifizierten Identität empfängt. Der RADIUS-Server kann eine Protokollierung und Abrechnung von Anforderungen, die er empfängt, ausführen. Die Antwort auf die Authentifizierungsanforderung kann als ein unveränderlicher Datensatz in einer Blockchain gespeichert werden.

[1089] Der Massenspeicher **808** kann einen Antwortzurückgeber **19104** zum Zurückgeben einer Antwort auf die Authentifizierungsanforderung an den RADIUS-Client ansprechend auf den Empfang der Antwort von der dezentralisierten API aufweisen. Der Massenspeicher **808** kann einen Anforderungsabsender **19106** zum Senden einer Anforderung zu einem RADIUS-Server ansprechend auf den Empfang einer positiven Identitätsverifizierungsantwort vom verteilten Ledger aufweisen.

[1090] Fig. 192 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **19200**, das Code aufweist, um einen Prozessor **902** anzuweisen, eine dezentralisierte Autorisierung, Authentifizierung und Abrechnung mit einer IoT-Vorrichtung vorzunehmen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **19200** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **19200** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1091] Das nichtflüchtige maschinenlesbare Medium **19200** kann einen Code **19202** aufweisen, um den Prozessor **902** anzuweisen, die Identität einer Authentifizierungsanforderung mit einem verteilten Ledger zu verifizieren, wobei die Authentifizierungsanforderung von einem Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)-Client empfangen wird. Der RADIUS-Client kann ansprechend auf eine Antwort der authentifizierten Identität eine Transaktion vornehmen. Die Transaktion kann einen Benutzernamen, ein Passwort und/oder Metadaten aufweisen. Die Transaktion kann eine Werttransaktion einschließen. Die Transaktion kann eine Kryptowährungstransaktion sein. Die Authentifizierungsanforderung kann eine Anforderung einer Netzadresse einschließen. Die Netzadresse kann einen voll qualifizierten Domainnamen für den RADIUS-Server und/oder eine Internetprotokolladresse für den RADIUS-Server aufweisen. Der RADIUS-Server kann einen öffentlichen Schlüssel durch Anfordern des Orts des öffentlichen Schlüssels von einer Blockchain verifizieren. Die Anforderung an einen RADIUS-Server kann außerhalb der Chain ansprechend darauf geschehen, dass ein RADIUS-Client eine Bestätigung der authentifizierten Identität empfängt. Der RADIUS-Server kann eine Protokollierung und Abrechnung von Anforderungen, die er empfängt, ausführen. Die Antwort auf die Authentifizierungsanforderung kann als unveränderlicher Datensatz in einer Blockchain gespeichert werden.

[1092] Das nichtflüchtige maschinenlesbare Medium **19200** kann einen Code **19204** aufweisen, um den Prozessor **902** anzuweisen, ansprechend auf den Empfang einer positiven Identitätsverifizierungsantwort vom verteilten Ledger eine Anforderung zu einem RADIUS-Server zu senden. Das nichtflüchtige maschinenlesbare Medium **19200** kann einen Code **19206** aufweisen, um den Prozessor **902** anzuweisen, ansprechend auf den Empfang einer Antwort vom RADIUS-Server eine Antwort auf die Authentifizierungsanforderung an den RADIUS-Client zurückzugeben.

[1093] Hier offenbarte Techniken können sich auf eine native dezentralisierte Datenbank beziehen. Die native dezentralisierte Datenbank kann eine Datenbank sein, die das Konzept der Teilnahme an einem dezentralisierten Cluster im Gegensatz zu jenem eines verteilten Clusters versteht. Bei einem Beispiel kann die dezentralisierte Datenbank durch die Verwendung einer öffentlichen und privaten Partitionierung von Tabellen innerhalb einer Datenbank zur nativen Unterstützung des dezentralisierten Betriebs verteilter Datenbanken funktionieren. Hierdurch kann der Betrieb eines IoT-Systems verbessert werden, indem die verteilte Speicherung von Daten über eine Anzahl beschränkter Vorrichtungen erlaubt wird.

[1094] Fig. 193 ist ein schematisches Diagramm eines Prozesses **19300** zur Konfiguration und zum Betrieb eines Konsensnetzes **19302** unter Verwendung einer nativen dezentralisierten Datenbank gemäß einigen Ausführungsformen. Das Konsensnetz **19302** kann einen Knoten **19304** aufweisen. Das Konsensnetz kann eine Anzahl von Knoten **19304**, einschließlich eines ersten Knotens bis zu einem n-ten Knoten, aufweisen. Während ein nativ dezentralisierter Datenbank-Cluster verwendet wird, kann eine Partei, die dem Netz nicht bekannt ist, dem Netz beitreten. Die existierenden Knoten **19304** können daran gehindert werden, eine Zentralstelle zu bilden. Der Beitritt zum Netz kann eine Anforderung zum Beitritt zu einer Transaktion einschließen, die an eine öffentliche Uniform Resource Locator (URL) oder einen bekanntgemachten Namespace gestellt werden kann. Der Namespace kann in einen Speicher hartcodiert sein oder von einem Benutzer einstellbar sein. Falls ein Knoten **19304** im Konsensnetz **19302** eine Nachricht angibt, welche die Implementation einer Zentralstelle fordert, die aus den Knoten im Netz besteht, können die Knoten **19304** über den Eintritt neuer Mitglieder abstimmen. Knoten in einer Zentralstelle können Regeln festlegen, die es neuen Mitgliedern standardmäßig erlauben, dem Netz beizutreten. Sobald ein neues Mitglied beigetreten ist, kann die Datenbank **19306** des neuen Mitglieds mit den Datenbanken **19306** existierender Mitglieder synchronisiert werden. Bei der Synchronisation können die geteilten S Partitionen **19308** repliziert werden. Die Datenbanken können Blockchain-Datenbanken sein.

[1095] Geteilte Partitionen **19308** können durch die zugrunde liegende Datenbank **19306** repliziert werden. Geteilte Partitionen **19308** können verwendet werden, um eine Datenebene aufzunehmen. Geteilte Partitionen **19308** können verwendet werden, um die Konsens-Blockchain aufzunehmen. Ein Netz kann aus vielen

Diensten **19310** und Clients **19312**, die Aufgaben ausführen können, bestehen. Die Dienste **19310** und die Clients **19312** können beispielsweise IoT-Systeme sein, welche Daten sammeln und verarbeiten, um Betätigungsentscheidungen lokal zu treffen. Von den Diensten **19310** und den Clients **19312** gesammelte und berechnete Daten können zu einer privaten Partition **19314** gesendet werden. Die privaten Partitionen können zentral gesteuert werden.

[1096] Immer dann, wenn ein Netzeigentümer angibt, dass ein Dienst geteilt werden kann oder dass die von einem Dienst abgeleiteten Dienstdaten geteilt werden können, können sich die Einstellungen der privaten Partition ändern oder in eine öffentliche Partition **19308** kopiert werden. Die Verschiebung der Daten aus einer privaten Partition **19314** zu einer öffentlichen Partition **19308** kann das Hinzufügen von Daten zu einem Mechanismus außerhalb der Chain einschließen. Beim Ändern von Daten von privat zu öffentlich kann beispielsweise die Konsensnatur einer dezentralisierten Datenbank **19306** verwendet werden, um an einer Abstimmung innerhalb des dezentralisierten Netzes **19302** teilzunehmen. Andere Techniken zum Ändern der Daten von öffentlich zu privat oder umgekehrt können von zentralen Systemen empfangene Befehle, das Ablaufen des Datums an den Daten und dergleichen einschließen. Kombinationen von diesen können verwendet werden. Beispielsweise kann ein Ablaufdatum in eine Richtlinie aufgenommen werden, woraufhin ein Konsens von Vorrichtungen in einem Netz feststellt, dass der Status von privat zu öffentlich geändert werden sollte.

[1097] Private Partitionen **19314** können zu anderen vom Netzeigentümer besessenen Knoten repliziert werden. Private Partitionen **19314** können in manchen Fällen in ihrer Replikation auf andere Datenbankinstanzen beschränkt werden, die von anderen Mitgliedern des Konsensnetzes betrieben werden. Die geteilten Partitionen können permissioniert und/oder verschlüsselt sein.

[1098] Ein Netzeigentümer kann beispielsweise der Dateneigentümer sein, und bei der Erzeugung einer geteilten Partition **19308** können die Erlaubnisse und die Verschlüsselung an der Partition vom Netzeigentümer festgelegt werden. Erlaubnisse können beispielsweise unter anderem rollenbasiert oder RADIUS/DIAMETER-protokollbasiert sein. Rollenbasierte Erlaubnisse können andere Akteure im Netz aufweisen, die eine bestimmte Rolle beim Zugriff auf bestimmte Daten besitzen. RADIUS- oder DIAMETER-basiert kann sich beispielsweise auf ein Authentifizierungsverfahren beziehen, das vom Internet als Erlaubnissteuermechanismus verwendet wird. Eine Verschlüsselung kann durch das Netz verwendet werden. Beispielsweise kann die Verschlüsselung öffentliche Schlüsselverfahren, private Schlüsselverfahren, Passwörter, Passphrases, Triple Data Encryption Standard (DES), Blowfish, Twofish oder AES aufweisen. Durch Anpassen der Erlaubnisse und der Verschlüsselung mit einer geteilten Partition kann ein Dateneigentümer die Fähigkeit behalten, die Parteien im Netz zu steuern, die berechtigt sein können, auf die Daten zuzugreifen. Durch Anpassen der Erlaubnisse und der Verschlüsselung mit einer geteilten Partition kann ein Dateneigentümer in der Lage sein, Daten außerhalb der Chain zu speichern.

[1099] Kopien der Daten können zu Knoten repliziert werden, welche die identifizierten Vorrechte enthalten. Knoten, die identifizierte Vorrechte enthalten, können diese identifizierten Vorrechte zu jeder Zeit durch den Dateneigentümer widerrufen lassen. Der Widerruf identifizierter Vorrechte kann entweder zum Verlust des Zugriffs auf künftige vom Dateneigentümer geteilte Daten oder zum Widerruf von Vorrechten, die sich auf historische Daten erstrecken, führen. Das Erlaubnissystem kann erzeugt werden, um die Fähigkeit eines Datenverbrauchers, Kopien der Daten anzufertigen, zu steuern. Die Begrenzung der Fähigkeit des Datenverbrauchers, Kopien von Daten anzufertigen, kann die Fähigkeit aufweisen, den Zugriff auf zuvor geteilte Daten zu widerrufen, falls eine Rolle widerrufen wird und der Datenverbraucher keine Erlaubnisse hat, Kopien der Daten anzufertigen.

[1100] Die Fähigkeit, Rollen zu gewähren und zu widerrufen, kann von der Steuerebene behandelt werden. Die Steuerebene kann als Teil des Konsensnetzes laufen, und solche Rollen und der Zugriff auf Daten kann zwischen Parteien im Austausch gegen eine Digitalwährung gewährt werden. Die Digitalwährung kann eine Vereinbarung sein, Daten wechselseitig zwischen Peers zu teilen.

[1101] Fig. 194 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **19400** für den Beitritt und den Betrieb innerhalb eines Konsensnetzes unter Verwendung einer nativen dezentralisierten Datenbank gemäß einigen Ausführungsformen. Das Verfahren **19400** aus Fig. 194 kann durch die mit Bezug auf Fig. 195 beschriebene IoT-Vorrichtung **19500** implementiert werden. In Block **19402** kann sich eine Vorrichtung mit dem Netz einer dezentralisierten Datenbank verbinden. Das Verbinden kann bei einigen Beispielen vom Beitreten unterschieden werden, weil das Beitreten Akzeptanz und Vertrauen durch andere Knoten implizieren kann. Es kann dezentralisierte binäre Software installiert werden. Die Datenbank kann private Datenbankpartitionen oder -tabellen erzeugen, die gegen eine Replikation auf anderen Knoten abgegrenzt sein können. Die Anzahl,

die Größe und die Funktion dieser Datenbanken kann dem Ermessen des Systemeigentümers oder Systementwicklers unterliegen.

[1102] In Block **19404** kann das System Namensräume entdecken. Namensräume können sich beispielsweise auf andere Netze beziehen, und die anderen Netze können dezentralisierte Datenbankdienste anbieten. Die Entdeckung von Namensräumen kann beispielsweise durch Ortsnachsorgevorgänge, Netzentdeckung, Vorrichtungsentdeckung und dergleichen geschehen. Der Entdeckungsprozess kann automatisiert oder hartcodiert sein. Eine Anforderung zum Beitritt zum Netz kann von der Vorrichtung, die beizutreten versucht, eingeleitet werden. Die Anforderung kann durch ein Konstrukt in der Art einer Erlaubnisanleitung gesteuert werden. Die Anforderung zum Beitritt zum Netz kann durch einen bekannten Knoten auf einem existierenden Netz der Vorrichtung geschehen, wobei der beitretende Knoten den bekannten Knoten des Clusters verwenden kann, um dem Cluster beizutreten. Eine Entscheidung darüber, wie es einem neuen Knoten zu erlauben ist, einem Netz beizutreten, kann getroffen werden, wenn die Netzentwickler das Netz zum ersten Mal initialisieren, oder zu einer früheren oder späteren Zeit. Wie vorstehend erörtert, können die Netzentwickler die Bedingungen für das Zulassen von Knoten durch auf teilnehmenden Knoten in einem dezentralisierten Datenbank-Cluster implementierte Richtlinien festlegen. Die Richtlinien können Teilnehmer, die einen Beitritt anfordern, automatisch akzeptieren, sobald die Teilnehmer eine verifizierte Version der dezentralisierten Datenbanksoftware ausführen. Die Verifizierung der dezentralisierten Datenbanksoftware kann unter Verwendung einer gemessenen Umgebung ausgeführt werden, um zu bestätigen, dass sich die Software auf einer Whitelist befindet, wie hier beschrieben. Eine beliebige Anzahl anderer Techniken kann auch verwendet werden, um die Version und die Gültigkeit zu bestätigen. Die Akzeptanzrichtlinien können ein Votum verwenden, um die neue Entität zu akzeptieren oder abzulehnen.

[1103] Neue Entitäten können anfänglich mit Rollen beitreten, die im dezentralisierten Datenbank-Cluster eine begrenzte Autorität aufweisen, und die Entität kann im Laufe der Zeit autoritativer werden, wenn Vertrauensmaße für die Entität zunehmen. Die Netzentwickler können es einem designierten Knoten erlauben, ein Validierer auf dem Netz zu werden. Beispielsweise können Netzentwickler Knoten als Validierer für Blockchains in der Art von Bitcoin festlegen. Falls ein Knoten, der versucht, einem dezentralisierten Datenbank-Cluster beizutreten, abgelehnt wird, kann der Knoten weiter als eine autonome Datenbank wirken. Die autonome Datenbank kann einer zentralisierten Anwendung dienen, die in der gleichen Sicherheitsdomain und/oder im gleichen Netz wie die autonome Datenbank existiert. Ein Knoten, der versucht, einem dezentralisierten Datenbank-Cluster beizutreten, kann versuchen, einen oder mehrere Namensräume zu entdecken. Ein Knoten, der versucht, einem dezentralisierten Datenbank-Cluster beizutreten, kann mehr als einem Konsensnetz beitreten, falls von den Netzentwicklern implementierte Richtlinien dies zulassen.

[1104] In Block **19406** kann ein Knoten, dem erlaubt wurde, einem dezentralisierten Datenbank-Cluster beizutreten, eine Anzahl geteilter Partitionen und Tabellen erzeugen, wie beispielsweise von den Netzentwicklern spezifiziert wurde. In geteilten Partitionen und geteilten Tabellen gespeicherte Daten können innerhalb des Netzes repliziert werden. Das Netz kann angeben, wie viele Kopien eines Datenobjekts zur Redundanz gespeichert werden dürfen. Ein Replikationsfaktor kann global sein oder auf unterschiedlicher Basis angewendet werden, beispielsweise auf der Basis des Datenobjekttyps. Der Replikationsfaktor kann auf der Grundlage der Kritikalität der Daten ausgewählt werden oder stückweise für Partitionen und Tabellen ausgewählt werden, wobei dies von der Wichtigkeit der Daten abhängt. Gespeicherte Daten können bruchstückhaft sein. Bruchstückhafte Daten können sich auf in Teilstücken über teilnehmende Knoten im Netz gespeicherte Daten beziehen, so dass kein einzelner Knoten einen vollständigen Satz von Bruchstücken für die Rekonstruktion eines bestimmten Objekts hat.

[1105] In Block **19408** kann ein Knoten mit dem Rest des Netzes synchronisiert werden und seine Dienste ankündigen. Die Ankündigung von Diensten kann beispielsweise das Auflisten an einem bestimmten Port oder Bereich von Ports aufweisen. Clients, welche die Datenbank verwenden, können über einen Bereich von Ports auf die Datenbankdienste zugreifen. Die nativ zentralisierte Datenbank kann Daten, die sie empfängt, weiterleiten, so dass die Daten in den privaten Partitionen und privaten Tabellen oder den geteilten Partitionen und geteilten Tabellen gespeichert werden können. Die Clients können sich der dezentralisierten Natur der Datenbank bewusst sein, und der Client kann fordern, dass die dezentralisierte Datenbank die Daten privat speichert. Die Clients können sich der dezentralisierten Natur der Datenbank bewusst sein, und der Client kann fordern, dass die dezentralisierte Datenbank die Daten öffentlich speichert. Teilnehmer am dezentralisierten Netz können ihre Daten an einer Stelle halten und später wählen, welche Daten sie teilen möchten und welche Daten nicht. In privaten Partitionen oder geteilten Partitionen gespeicherte Daten können auf Anweisung des Dateneigentümers verschlüsselt werden, bevor die Daten gespeichert werden. Die Verschlüsselung kann von den Clients vorgenommen werden und/oder in der dezentralisierten Datenbank implementiert werden,

beispielsweise falls der Dateneigentümer der Datenbank vertraut. Eine dezentralisierte Datenbank kann einen geteilten Marktplatz für IoT-Daten ermöglichen.

[1106] **Fig. 195** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **19500** für den Beitritt zu einer dezentralisierten Datenbank und zum Betreiben von dieser vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **19500** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1107] Der Massenspeicher **808** kann eine Anzahl von Modulen zum Beitritt zu einer dezentralisierten Datenbank und zum Betreiben von dieser aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1108] Der Massenspeicher **808** kann einen Vorrichtungsverbinder **19502** zum Verbinden einer Vorrichtung mit einem Netz einer dezentralisierten Datenbank aufweisen. Die Vorrichtung kann dezentralisierte Datenbanksoftware ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank installieren. Die Vorrichtung kann eine geteilte Datenbankpartition ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank erzeugen.

[1109] Der Massenspeicher **808** kann einen Namespaceentdecker **19504** zum Entdecken eines Namespace eines Knotens der dezentralisierten Datenbank aufweisen. Die Vorrichtung kann ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank fordern, der dezentralisierten Datenbank beizutreten. Die Vorrichtung kann ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank in der dezentralisierten Datenbank akzeptiert werden.

[1110] Der Massenspeicher **808** kann einen Partitionserzeuger **19506** zur Erzeugung einer geteilten Datenbankpartition ansprechend auf die Akzeptanz durch den Knoten erzeugen. Die geteilte Datenbankpartition kann permissioniert und/oder verschlüsselt werden. In einer geteilten Datenbankpartition gespeicherte Kopien von Daten können ansprechend darauf zu einem zweiten Knoten repliziert werden, dass der zweite Knoten Vorrechte präsentiert, welche die Erlaubnis des zweiten Knotens, die Daten zu kopieren, angeben. Die Vorrichtung kann eine geteilte Knotenpartition für die Speicherung in der geteilten Datenbankpartition ansprechend auf die Erzeugung der geteilten Datenbankpartition replizieren.

[1111] Der Massenspeicher **808** kann einen Dienstbekanntmacher **19508** zum Bekanntmachen eines Dienstes für die dezentralisierte Datenbank aufweisen. Der Massenspeicher **808** kann einen Datenrouter **19510** zum Übertragen während der Ausführung eines Dienstes empfangener und erzeugter Daten zwischen einer privaten Datenbankpartition und einer geteilten Datenbankpartition aufweisen. Die Daten in der geteilten Partition können ansprechend darauf, dass Daten zur geteilten Datenbankpartition geleitet werden, zur Speicherung in einer geteilten Knotenpartition repliziert werden. Die Vorrichtung kann ansprechend darauf, dass der Knoten für die Akzeptanz der Vorrichtung stimmt, die Akzeptanz an der dezentralisierten Datenbank empfangen.

[1112] **Fig. 196** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **19600**, das Code aufweist, um einen Prozessor **902** anzuweisen, einer dezentralisierten Datenbank beizutreten und diese zu betreiben, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **19600** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **19600** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1113] Das nichtflüchtige maschinenlesbare Medium **19600** kann einen Code **19602** aufweisen, um den Prozessor **902** anzuweisen, eine Vorrichtung mit einem Netz einer dezentralisierten Datenbank zu verbinden. Die Vorrichtung kann dezentralisierte Datenbanksoftware ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank installieren. Die Vorrichtung kann eine geteilte Datenbankpartition ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank erzeugen.

[1114] Das nichtflüchtige maschinenlesbare Medium **19600** kann einen Code **19604** aufweisen, um den Prozessor **902** anzuweisen, einen Namespace eines Knotens der dezentralisierten Datenbank zu entdecken. Die Vorrichtung kann ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank fordern, der dezentralisierten Datenbank beizutreten. Die Vorrichtung kann ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank in der dezentralisierten Datenbank akzeptiert werden.

[1115] Das nichtflüchtige maschinenlesbare Medium **19600** kann einen Code **19606** aufweisen, um den Prozessor **902** anzuweisen, eine geteilte Datenbankpartition ansprechend auf die Akzeptanz durch den Knoten zu erzeugen. Die geteilte Datenbankpartition kann permissioniert und/oder verschlüsselt werden. In einer geteilten Datenbankpartition gespeicherte Kopien von Daten können ansprechend darauf zu einem zweiten Knoten repliziert werden, dass der zweite Knoten Vorrechte präsentiert, welche die Erlaubnis des zweiten Knotens, die Daten zu kopieren, angeben. Die Vorrichtung kann eine geteilte Knotenpartition für die Speicherung in der geteilten Datenbankpartition ansprechend auf die Erzeugung der geteilten Datenbankpartition replizieren.

[1116] Das nichtflüchtige maschinenlesbare Medium **19600** kann einen Code **19608** aufweisen, um den Prozessor **902** anzuweisen, der dezentralisierten Datenbank einen Dienst bekannt zu machen. Das nichtflüchtige maschinenlesbare Medium **19600** kann einen Code **19610** aufweisen, um den Prozessor **902** anzuweisen, ansprechend auf die Ausführung eines Dienstes empfangene und erzeugte Daten zwischen einer privaten Datenbankpartition und einer geteilten Datenbankpartition zu übertragen. Die Daten in der geteilten Partition können ansprechend darauf, dass Daten zur geteilten Datenbankpartition geleitet werden, zur Speicherung in einer geteilten Knotenpartition repliziert werden. Die Vorrichtung kann ansprechend darauf, dass der Knoten für die Akzeptanz der Vorrichtung stimmt, die Akzeptanz an der dezentralisierten Datenbank empfangen.

[1117] Gemäß einigen Ausführungsformen offenbaren die hier dargelegten Techniken die Zugriffssteuerung in einem IoT-Objekt. Bei IoT-Systemen ist die Sicherheit infolge der beschränkten Natur der beteiligten Vorrichtungen kompliziert, welche nicht in der Lage sein können, die in weniger beschränkten Vorrichtungen unter anderem in der Art von Desktops, Laptops oder Smartphones verwendeten Sicherheitssysteme zu implementieren. Die Implementation einer Zugriffssteuerung, die weniger komplizierte Parameter verwendet, kann die sichere Implementation von IoT-Anwendungen in sicheren Umgebungen erweitern und den Betrieb und die Annahme von IoT-Systemen verbessern.

[1118] Beim IoT-Systementwurf kann sich ein Objekt auf eine Datenmodellbeschreibung und eine physikalische Instanziierung einer Betriebseinheit beziehen. Ein IoT-System kann in Bezug auf mehrere Objekte beschrieben werden, die zusammenarbeiten, um ein Ziel oder ein Ergebnis zu erreichen. Objekte können in dem Sinne, dass die Definition des Objekts rekursiv sein kann, aus mehreren Betriebsschichten zusammengesetzt sein. Ein Objektzerlegungsverfahren in der Art der Introspektion kann die Rekursion zu seinen Blattknotenattributen lösen. Ein IoT-Objektzugriff kann in manchen Fällen anhand einer geschichteten Zerlegung mit wenigstens sechs Schichten verstanden werden, und in anderen Fällen können mehr oder weniger Schichten verwendet werden.

[1119] Fig. 197 ist ein schematisches Diagramm einer logischen Unterteilung **19700** für die Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen. Bei einem Beispiel kann die logische Unterteilung für die Zugriffssteuerung zeigen, dass eine Autorisierung eines Aufrufenden mit einer Anforderung für den Zugriff einhergehen kann. Die Aufruferautorisierung kann innerhalb der Zugriffssteuerungslisten- bzw. Access-Control-List(ACL)-Struktur **19702** qualifiziert werden, die ein Aufruferobjekt **19704** und ein Zielobjekt **19706** identifiziert. Die ACL-Struktur **19702** kann zeigen, dass Erzeugungs-, Lese-, Aktualisierungs-, Lösch- und Benachrichtigungs- bzw. Create, Read, Update, Delete, and Notify(CRUDN)-Erlaubnisse **19708** auf jeder beliebigen Schicht in der Schichtzerlegung angewendet werden können. Das ACL-Aufruferobjekt **19704** und das ACL-Zielobjekt **19706** können Strukturen mit dem gleichen Objektreferenztyp sein, so dass es eine volle Flexibilität geben kann, einen Bereich der Granularität des Aufruferobjekts **19704** bzw. des Zielobjekts **19706** entsprechend einem Schichtmodell zu spezifizieren. Die CRUDN-Richtlinie **19708** kann auf jeder Schicht der Granularität bedeutungsvoll sein.

[1120] Einem Aufruferobjekt **19704** kann ein Berechtigungsnachweis mit einer die Vorrechte, mit denen der Aufrufer eine Anforderung macht, definierenden Autorisierungsstruktur erteilt werden. Vorrechte können gemäß der vorstehend erwähnten Schichtstruktur definiert werden. Die Plattform, die Vorrichtung, die Sammlung, die Ressource, der Datensatz oder die Eigenschaft, welche die Anforderung eingeleitet hat, kann im Autorisierungsabschnitt spezifiziert werden.

[1121] Fig. 198 ist ein schematisches Diagramm logischer Unterteilungen **19800** zwischen einem Aufruferberechtigungs-nachweis **19802** und einer Anforderung **19804** zur Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen. Die Autorisierung **19806** eines Aufrufers kann mit einer Anforderung zum Zugriff und sich daraus ergebenden Erlaubnissen **19808** einhergehen. Ein Objekt, auf das zuzugreifen ist, kann durch die intrinsischen Beschränkungen eingeschränkt werden, die dem Objekt durch die Physikalität des Objekts auferlegt werden. Beispielsweise kann eine Nurlese-Speichervorrichtung (ROM) nicht die Physikalität haben, die Schreibvorgänge ermöglicht. Physikalität kann unter Verwendung von CRUDN ausgedrückt werden. Der erwartete Zugriff kann durch die Physikalität eines Objekts beschränkt werden, weshalb die Zugriffsanforderung erwarten kann, dass die angeforderte Erlaubnis durch die Physikalität beherrscht wird. Der erwartete Zugriff kann eine Anforderung **19804** sein, die ein Objekt **19810** und Erlaubnisse **19812** aufweist. Falls sie nicht durch die Physikalität eines Objekts begrenzt wird, kann eine Zugriffsanforderung durch ein Objekt, falls ihr entsprochen wird, in manchen Fällen bewirken, dass sich eine Vorrichtung undefiniert oder unsicher verhält.

[1122] Fig. 199 ist ein schematisches Diagramm logischer Unterteilungen **19900** zwischen einer Objektfähigkeit **19902** zur Zugriffssteuerung unter Verwendung von Schichten **19904** in einem IoT-Objekt gemäß einigen Ausführungsformen. Eine erste Schicht eines IoT-Objektzugriffs kann eine Plattformschicht **19906** sein. Eine Plattformschicht kann eine physikalische Instanz eines Computers sein, die Rechen-, Netzwerk-, Speicher-, Mess- oder Betätigungsfähigkeiten aufweist. Die Plattformzugriffssteuerung kann in Bezug auf eine Plattformkennung und einen Berechtigungsnachweis verstanden werden. Der Berechtigungsnachweis kann beispielsweise von einem Hersteller eingebettet werden oder während der Konfiguration oder Implementation in der Einheit gespeichert werden, so dass der Berechtigungsnachweis als ein Attestierungsberechtigungsnachweis dienen könnte. Der Plattformberechtigungs-nachweis kann bei einer Zugriffsanforderung ohne den Berechtigungsnachweis verifizieren, ob die Zugriffsanforderung eine Bedingung für das Erteilen eines Vorrichtungsberechtigungsnachweises sein kann. Der Plattformberechtigungs-nachweis kann verwendet werden, um Plattform-eigenschaften einschließlich ihrer Physikalität erneut zu attestieren.

[1123] Eine zweite Schicht eines IoT-Objektzugriffs kann eine Vorrichtungsschicht **19908** sein. Eine Vorrichtungsschicht kann eine logische Instanz eines Computers sein, die Rechen-, Netzwerk-, Speicher-, Mess- oder Betätigungsfähigkeiten aufweist. Die Vorrichtungszugriffssteuerung kann in Bezug auf eine Vorrichtungskennung und einen Berechtigungsnachweis verstanden werden.

[1124] Eine dritte Schicht eines IoT-Objektzugriffs kann eine Sammelschicht **19910** sein. Eine Sammelschicht kann eine logische Struktur einer oder mehrerer Ressourcen aufweisen, wie nachstehend offenbart. Die Zugriffssteuerung kann in Bezug auf eine Typkennung, eine Schnittstellendefinition und eine Autoritätskennung, wodurch die Struktur benannt wird, verstanden werden.

[1125] Eine vierte Schicht eines IoT-Objektzugriffs kann eine Ressourcenschicht **19912** sein. Eine Ressourcenschicht kann eine logische Struktur eines oder mehrerer Datensätze aufweisen, wie nachstehend offenbart. Die Zugriffssteuerung kann in Bezug auf eine Typkennung, eine Schnittstellendefinition und eine Autoritätskennung, wodurch die Struktur benannt wird, verstanden werden.

[1126] Eine fünfte Schicht eines IoT-Objektzugriffs kann eine Aufzeichnungsschicht **19914** sein. Eine Aufzeichnungsschicht kann eine logische Struktur einer oder mehrerer Eigenschaften aufweisen, wie nachstehend offenbart. Die Zugriffssteuerung kann in Bezug auf eine Ressource plus einem Datensatzindexversatz verstanden werden.

[1127] Eine sechste Schicht eines IoT-Objektzugriffs kann eine Eigenschaftsschicht **19916** sein. Eine Eigenschaftsschicht kann beispielsweise eine atomare Datenstruktur und/oder eine komplexe Datenstruktur einer beliebigen unter Verwendung einer Datenmodellierungssprache (DML) definierbaren Struktur aufweisen. Beispielsweise kann eine atomare Datenstruktur eine Zeichenkette, eine Zahl und/oder ein Datum aufweisen. Die DML kann eine Struktur von Metadaten bereitstellen, um beispielsweise Begrenzungen für eine akzeptable Datenformatierung, Struktur- und Datenwertrandbedingungen in der Art eines JSON-Schemas zu erfassen. Eine Zugriffssteuerungs-Richtlinie kann in Bezug auf einen Datenstruktur-Lebenszyklus aus Erzeugen, Lesen, Aktualisieren, Löschen und Benachrichtigen bzw. Create, Read, Update, Delete, and Notify (CRUDN) ausgedrückt werden. Die Benachrichtigung kann ferner in Beobachten und Benachrichtigen unterteilt werden, wobei Beobachten eine Leseerlaubnis bei einem Strukturänderungsereignis voraussetzt und Benachrichtigen eine Schreiberlaubnis für ein anderes Objekt voraussetzt.

[1128] Die Zugriffssteuerungslisten(ACL)-Beurteilung kann ein Prozess zur Überprüfung der Autorisierung eines Aufruferobjekts sein, wodurch ein Aufruferabschnitt beherrscht und/oder überlappt wird. Eine ACL-Beur-

teilung kann ein Prozess sein, bei dem eine Struktur, auf die zugegriffen wird, durch einen Zielabschnitt beherrscht und/oder überlappt wird. Sofern nicht ein vollständiger Satz von Schichten im Zielabschnitt mit der Struktur, auf die zuzugreifen ist, übereinstimmt, kann die ACL in ihrer Anwendung beschränkt sein. Der Zugriff kann verweigert werden, sofern nicht festgestellt wird, dass eine ACL übereinstimmt.

[1129] Fig. 200 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 20000 zur Zugriffssteuerung in einem IoT-Objekt gemäß einigen Ausführungsformen. Das Verfahren 20000 aus Fig. 200 kann durch die mit Bezug auf Fig. 201 beschriebene IoT-Vorrichtung 20100 implementiert werden. Der Prozessfluss kann in Block 20002 beginnen. In Block 20002 kann ein Berechtigungsnachweis einer Aufrufer-Entität erteilt werden. Der Berechtigungsnachweis kann beispielsweise eine sechsschichtige Autorisierungsstruktur enthalten, wenngleich auch andere Autorisierungsstrukturen in der Art vierschichtiger Autorisierungsstrukturen verwendet werden können, wobei dies von den Sicherheitsanforderungen abhängt. In Block 20004 können Objekt-Entitäten mit ACL versehen werden. Die ACL können die sechsschichtige Referenz zum Zielobjekt und eine CRUDN- oder CRUDON-Erlaubnis spezifizieren. In Block 20006 kann ein Aufrufer dem Objekt über eine geeignete Verbindungsschnittstelle einen Autorisierungsberechtigungsnachweis präsentieren. In Block 20008 kann eine Zugriffserzwingungsmaschine bzw. Access Enforcement Engine (AEE) die ACL-Richtlinie mit dem gegebenen Lieferantenberechtigungsnachweis anwenden.

[1130] In Block 20010 kann festgestellt werden, ob die Berechtigungsnachweisautorisierung mit dem Aufrufer überlappt. Falls dies nicht der Fall ist, überlappt die Berechtigungsnachweisautorisierung den Aufrufer nicht, und der Prozessfluss wird in Block 20012 fortgesetzt, wo der Zugriff verweigert werden kann.

[1131] In Block 20014 kann festgestellt werden, ob ein Ziel die Anforderung überlappt. Falls dies nicht der Fall ist, überlappt das Ziel die Anforderung nicht, und der Prozessfluss wird in Block 20012 fortgesetzt, wo der Zugriff verweigert werden kann.

[1132] In Block 20016 können Schichten der Aufruferobjekt-Schichtidentifikationen mit den Berechtigungsnachweisobjekt-Schichtidentifikationen verglichen werden, um festzustellen, ob es eine Übereinstimmung gibt. Falls dies nicht der Fall ist, stimmen die Aufruferobjekt-Schichtidentifikationen nicht mit den Berechtigungsnachweisobjekt-Schichtidentifikationen überein, und der Prozessfluss wird in Block 20012 fortgesetzt, wo der Zugriff verweigert werden kann. Aufruferobjekt-Schichtidentifikationen können eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht aufweisen.

[1133] In Block 20018 können Schichten der Zielobjekt-Schichtidentifikationen mit den Anforderungsobjekt-Schichtidentifikationen verglichen werden, um festzustellen, ob es eine Übereinstimmung gibt. Falls dies nicht der Fall ist, stimmen die Zielobjekt-Schichtidentifikationen nicht mit den Anforderungsobjekt-Schichtidentifikationen überein, und der Prozessfluss wird in Block 20012 fortgesetzt, wo der Zugriff verweigert werden kann. Zielobjekt-Schichtidentifikationen können eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht aufweisen. Falls die vorstehenden Feststellungen bejaht werden, kann in Block 20020 der Zugriff für ein IoT-Objekt erlaubt werden.

[1134] Fig. 201 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 20100 zur Zugriffssteuerung in einem IoT-Objekt vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 3 und Fig. 8 beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung 20100 andere Komponenten als für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1135] Der Massenspeicher 808 kann eine Anzahl von Modulen zur Zugriffssteuerung in einem IoT-Objekt aufweisen. Wenngleich sie im Massenspeicher 808 als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1136] Der Massenspeicher 808 kann einen Berechtigungsnachweiserteiler 20102 zur Erteilung eines Berechtigungsnachweises an eine Aufrufer-Entität aufweisen, wobei der Berechtigungsnachweis eine Anzahl von Schichten der Autorisierungsstruktur aufweist. Der Berechtigungsnachweis kann eine sechsschichtige Erlaubnis sein. Die sechsschichtige Erlaubnis kann eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht aufweisen. Die Anzahl der Schichten können eine Plattformschicht umfassen, um eine physikalische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und/oder Betätigungsfähigkeiten auf.

Die Anzahl der Schichten können eine Vorrichtungsschicht umfassen, um eine logische Instanz eines Computers widerzuspiegeln, der Rechen-, Netzwerk-, Speicher-, Mess- und/oder Betätigungsfähigkeiten aufweist. Die Anzahl der Schichten können eine Sammelschicht für eine logische Struktur einer Ressource umfassen, wobei die Ressource eine logische Struktur für einen Datensatz aufweist, wobei der Datensatz eine logische Struktur einer Eigenschaft aufweist und wobei die Eigenschaft eine atomare Datenstruktur und/oder eine komplexe Datenstruktur aufweist. Die Eigenschaft kann eine komplexe Datenstruktur sein, und die komplexe Datenstruktur ist für eine Struktur vorgesehen, die unter Verwendung einer Datenmodelliersprache definierbar ist. Die Eigenschaft kann eine atomare Datenstruktur aufweisen, und die atomare Datenstruktur kann eine Zeichenkette, eine Zahl und/oder ein Datum sein. Der Berechtigungsnachweis kann die Installation durch einen Hersteller angeben.

[1137] Der Massenspeicher **808** kann einen Objekt-Entitäts-Bereitsteller **20104** zum Versehen einer Objekt-Entität mit einer Zugriffssteuerungsliste, die eine Referenz zu einem Zielobjekt und eine Erlaubnis spezifiziert, aufweisen. Der Massenspeicher **808** kann einen Berechtigungsnachweispräsentierer **20106** aufweisen, um der Objekt-Entität einen Autorisierungsberechtigungsnachweis zu präsentieren. Der Autorisierungsberechtigungsnachweis für die Objekt-Entität kann durch Beschränkungen beschränkt werden, die dem Objekt durch die Physikalität der Objektdaten auf der Grundlage einer Erzeugen-, Lesen-, Aktualisieren-, Löschen- und Benachrichtigen(CRUDN)-Lebenszyklusbenachrichtigung auferlegt werden. Der Massenspeicher **808** kann einen Zugriffssteuerungslisten-Richtlinienanwender **20108** aufweisen, um eine Zugriffssteuerungslisten-Richtlinie anzuwenden, um auf der Grundlage eines Vergleichs, ob der Berechtigungsnachweis die Aufrufer-Entität überlappt, ob das Zielobjekt eine Anforderung überlappt, ob eine Anzahl von Vorrichtungsschichtidentifikationen mit einer Anzahl von Berechtigungsnachweisidentifikationen übereinstimmt und ob eine Anzahl von Ziel-schichtidentifikationen mit einer Anzahl von Anforderungsschichtidentifikationen übereinstimmt, festzustellen, ob ein Zugriff für eine IoT-Vorrichtung erlaubt ist.

[1138] **Fig. 202** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **19600**, das Code aufweist, um einen Prozessor **902** für die Zugriffssteuerung in einem IoT-Objekt anzuweisen, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **20200** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **19600** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1139] Das nichtflüchtige maschinenlesbare Medium **20200** kann einen Code **20202** aufweisen, um den Prozessor **902** anzuweisen, einen Berechtigungsnachweis einer Aufrufer-Entität zu erteilen, wobei der Berechtigungsnachweis eine Anzahl von Schichten der Autorisierungsstruktur aufweist. Der Berechtigungsnachweis kann eine sechsschichtige Erlaubnis sein. Die sechsschichtige Erlaubnis kann eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht aufweisen. Die Anzahl der Schichten können eine Plattformschicht umfassen, um eine physikalische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und/oder Betätigungsfähigkeiten auf. Die Anzahl der Schichten können eine Vorrichtungsschicht umfassen, um eine logische Instanz eines Computers widerzuspiegeln, der Rechen-, Netzwerk-, Speicher-, Mess- und/oder Betätigungsfähigkeiten aufweist. Die Anzahl der Schichten können eine Sammelschicht für eine logische Struktur einer Ressource umfassen, wobei die Ressource eine logische Struktur für einen Datensatz aufweist, wobei der Datensatz eine logische Struktur einer Eigenschaft aufweist und wobei die Eigenschaft eine atomare Datenstruktur und/oder eine komplexe Datenstruktur aufweist. Die Eigenschaft kann eine komplexe Datenstruktur sein, und die komplexe Datenstruktur kann für eine Struktur vorgesehen sein, die unter Verwendung einer Datenmodelliersprache definierbar ist. Die Eigenschaft kann eine atomare Datenstruktur aufweisen, und die atomare Datenstruktur kann eine Zeichenkette, eine Zahl und/oder ein Datum sein. Der Berechtigungsnachweis kann die Installation durch einen Hersteller angeben.

[1140] Das nichtflüchtige maschinenlesbare Medium **20200** kann einen Code **20204** aufweisen, um den Prozessor **902** anzuweisen, eine Objekt-Entität mit einer Zugriffssteuerungsliste zu versehen, die eine Referenz auf ein Zielobjekt und eine Erlaubnis spezifiziert. Das nichtflüchtige maschinenlesbare Medium **20200** kann einen Code **20206** aufweisen, um den Prozessor **902** anzuweisen, der Objekt-Entität einen Autorisierungsberechtigungsnachweis zu präsentieren. Der Autorisierungsberechtigungsnachweis für die Objekt-Entität kann in einigen Fällen durch Beschränkungen beschränkt werden, die dem Objekt durch die Physikalität der Objektdaten auf der Grundlage einer Erzeugen-, Lesen-, Aktualisieren-, Löschen- und Benachrichtigen(CRUDN)-Lebenszyklusbenachrichtigung auferlegt werden. Das nichtflüchtige maschinenlesbare Medium **20200** kann einen Code **20208** aufweisen, um den Prozessor **902** anzuweisen, eine Zugriffssteuerungslisten-Richtlinie an-

zuwenden, um auf der Grundlage eines Vergleichs, ob der Berechtigungsnachweis die Aufrufer-Entität überlappt, ob das Zielobjekt eine Anforderung überlappt, ob eine Anzahl von Vorrichtungsschichtidentifikationen mit einer Anzahl von Berechtigungsnachweisidentifikationen übereinstimmt und ob eine Anzahl von Zielschichtidentifikationen mit einer Anzahl von Anforderungsschichtidentifikationen übereinstimmt, festzustellen, ob ein Zugriff für eine IoT-Vorrichtung erlaubt ist.

[1141] Selbstverwaltende Vorrichtungen und Systeme gemäß einigen Ausführungsformen sind in der Lage, sich selbst und ihre Merkmale für sich selbst und andere Vorrichtungen zu beschreiben. Beispielsweise kann, wie hier beschrieben, eine Introspektion verwendet werden. Die Introspektion ist eine Form eines Bewusstseins über sich selbst, wobei eine Datenbeschreibungssprache (DDL), unter anderem beispielsweise ein JSON-Schema oder XML, welche maschinenlesbar ist, die semantische Zerlegung der abgefragten oder bekanntgemachten Vorrichtung einkapselt. Wie hier verwendet, können selbstverwaltende Vorrichtungen und Systeme sich selbst bewusst sein und in der Lage sein, die Funktionsweise der Vorrichtung zu optimieren oder es zu erkennen, wenn sie beschädigt ist oder ihre Ressourcen zur Neige gehen. Ferner können selbstbeschreibende Module menschliche Eingaben und Bemühungen verringern, indem sie die Aufgabe des Lesens eines Datenblatts und des Entwickelns eines spezifischen Codes für das Modul automatisieren. Beispielsweise kann ein selbstbeschreibender Wandler einen integrierten Speicher aufweisen, der die im Datenblatt gefundenen Daten beschreibt.

[1142] Die Datenblattinformationen können Herstellereinzelheiten, Kalibrierparameter, Signalaufbereitungs- und Signalverarbeitungsanforderungen einschließen. Ein Datenblatt kann ferner ein Knoten-Metamodell bzw. Node Meta-Model (NMM) für die Interaktion beschreiben. Beim Metamodell kann ein Knoten eine Knotenken- nung, einen Satz von Eigenschaften und einen Satz von Befehlen in der Art der Befehle, die der Knoten sendet, und der Befehle, die der Knoten empfängt, und einen Satz von Befehlsparametern aufweisen. Parameter kön- nen durch eine Kennung, einen Editor und einen Initialisierer qualifiziert werden. Editoren können auf Eigen- schaften und/oder Befehlsparameter angewendet werden. Ein Knoten kann seinen eigenen Editor aufweisen. Demgemäß können die Datenblattinformationen bei einem Knoten-Metamodell zusätzlich zu Eigenschaftsin- formationen eine Befehlsinteraktionssemantik aufweisen.

[1143] Das NMM kann unter Verwendung einer die automatisierte Introspektion erleichternden DDL ausdrück- bar sein. Daher können IoT-Vorrichtungen, die mit dem Knoten interagieren, dynamisch auf Änderungen in einem Datenblatt reagieren, wie hier weiter detailliert dargelegt. Wenn beide Seiten einer Datenblattinterakti- on das gleiche Vokabular des NMM erkennen, kann das System von IoT-Vorrichtungen Änderungen im Vor- richtungsverhalten und in der Vorrichtungsfähigkeit dynamisch ausnutzen, ohne Vorrichtungstreiber oder Sys- temsoftware zu installieren oder zu aktualisieren. Dementsprechend kann ein selbstbeschreibender Wandler in einer Plug-and-Play-Konfiguration mit einer Mikrosteuereinrichtung oder einer IoT-Vorrichtung verwendet werden, ohne dass manuell spezifischer Code zum Zugreifen auf die Informationen auf dem Datenblatt entwi- ckelt werden müsste. Selbstbeschreibende Vorrichtungen können in ein Netz, in dem sie ihre Ressourcen und Anforderungen bekannt machen, durch Plug and Play aufgenommen werden.

[1144] Ferner können selbstbeschreibende externe Module, einschließlich Wandlern, Funkeinrichtungen, En- ergiespeicherung, Energiegewinnung und Mikrosteuereinrichtungen, verwendet werden, um Abfall zu verrin- gern, indem abgelaufene oder beschädigte Komponenten entsorgt werden und die Komponenten mit einer längeren Lebensdauer umgewidmet werden. Beispielsweise kann ein externes Modul unter anderem exter- ne Sensoren oder Betätigungselemente, Kommunikationsmodule, Energiegewinnungskomponenten oder ei- ne externe Batterie oder einen externen Speicher umfassen. Die externen Module in der Art eines Sensors oder einer Funkeinrichtung können ein Ablaufdatum aufweisen, bei dem vorhergesagt werden kann, dass die Genauigkeit oder die Funktionalität abnehmen. Wenn austauschbare externe Module in einer IoT-Vorrichtung verwendet werden, können die externen Module beim Erreichen des Ablaufdatums ersetzt werden, wodurch es ermöglicht wird, dass der Rest der IoT-Vorrichtung rekonfiguriert und umgewidmet wird. Die Fähigkeit zum Ersetzen oder Entfernen alternder oder nicht funktionsfähiger externer Module und zum Rekonfigurieren der restlichen IoT-Vorrichtung und funktionierenden externen Module kann die Gesamtlebensdauer der gesamten IoT-Vorrichtung verlängern.

[1145] Bei einer Anordnung mit einer einzigen IoT-Vorrichtung kann die Lebensdauer mit der Lebenszeit der ersten ausfallenden Komponente verknüpft werden. Bei Verwendung der gegenwärtig offenbarten Techniken kann der Gesamtsensorknoten gemäß einigen Ausführungsformen jedoch über die Lebenszeit der kürzest- lebigen Komponente hinaus automatisch repariert oder für einen anderen Zweck rekonfiguriert werden. Bei- spielsweise kann die IoT-Vorrichtung das externe Modul in der Nähe des Lebenszeitendes deaktivieren und rekonfiguriert werden, um auf der Grundlage der restlichen Module eine andere Aufgabe auszuführen.

[1146] Ferner kann die Funktion der selbstbeschreibenden modularen IoT-Vorrichtung eine ganz andere sein, nachdem eine Komponente deaktiviert wurde. Beispielsweise kann ein fehlerhaftes externes Modul durch ein funktionierendes externes Modul für eine andere Funktion ersetzt werden, wodurch die Funktion der Gesamt-IoT-Vorrichtung geändert wird. Ein Funkmodul an einem Sensorknoten kann durch eine neuere, einen geringeren Stromverbrauch aufweisende oder eine längere Reichweite aufweisende Funkressource ersetzt werden. Dies kann die Nutzungsdauer des Sensorknotens verlängern, weil der Sensorknoten rekonfiguriert werden kann, falls ein System-Gateway auf ein neueres Funkprotokoll aufgerüstet wird. Ferner kann eine selbstbeschreibende IoT-Vorrichtung auf die Werte von diesen mehreren Modulen verweisen und mehr kalibrierte Daten durch die Verwendung zusätzlicher externer Module ausgeben. Dies kann erleichtert werden, wenn eine maschinenlesbare DDL ein schematisches Markup aufweist, das zu der selbstbeschreibenden Vorrichtung, worauf verwiesen wurde, übertragbar ist. Daher kann ein getrennter manueller Schritt des Anwendens des semantischen Markups vermieden werden. Die IoT-Kalibrierparameter könnten es einem Prozessor ermöglichen, diese kalibrierten Werte direkt zu lesen und anzuwenden, statt Rohdaten mit zusätzlicher Verarbeitung behandeln zu müssen.

[1147] Ein gemeinsames Protokoll kann von Vorrichtungen und Modulen verwendet werden, die in der Lage sind, ihre Ressourcen und Anforderungen selbst zu beschreiben. Bei diesen Anordnungen können die externen Module in viele Vorrichtungen integriert werden. Die Vorrichtungen können Konflikte zwischen der Vorrichtungsfähigkeit und den Anforderungen einer angeschlossenen Komponente auszeichnen.

[1148] **Fig. 203** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **20300** zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Abbildung von Ressourcen und Anforderungen selbstbeschreibender Hardware gemäß einigen Ausführungsformen. Das Verfahren **20300** aus **Fig. 203** kann durch die mit Bezug auf **Fig. 204** beschriebene IoT-Vorrichtung **20400** implementiert werden. Das Verfahren **20300** kann unter Verwendung des mit Bezug auf **Fig. 8** beschriebenen Systems **802** ausgeführt werden. Das Verfahren **20300** kann in Block **20302** beginnen, wenn eine IoT-Vorrichtung bootet.

[1149] In Block **20304** kann die IoT-Vorrichtung Ressourcen, die von der IoT-Vorrichtung gesteuert werden, spezifizieren. Bei einem Beispiel können die Ressourcen Hardwarekomponenten sein und eine Energiequelle in der Art einer Stromversorgung, einer Batterie oder eines Energiegewinnungssystems, einschließlich unter anderem Solarkollektoren, Windturbinen oder Wasserturbinen, umfassen. Die Hardwarekomponenten der IoT-Vorrichtung können beispielsweise einen Prozessor, Kontextsensoren, Kontextbetätigungselemente, eine Signalaufbereitungs-Schaltungsanordnung, einen permanenten Speicher und einen flüchtigen Speicher umfassen. Ressourcenhardwarekomponenten könnten beispielsweise integrierte Kommunikationen einschließlich einer interintegrierten Schaltung (12C), einer seriellen Peripherieschnittstelle (SPI), eines universellen asynchronen Empfängers/Senders bzw. Universal Asynchronous Receiver/Transmitter (UART) oder einer integrierten Funkeinrichtung umfassen. Die Komponenten der IoT-Vorrichtung gemäß einigen Ausführungsformen werden mit Bezug auf **Fig. 204** weiter erörtert.

[1150] In Block **20306** wird festgestellt, ob einige oder alle externen Module spezifiziert wurden, und werden Einzelheiten über die Anforderungen eines externen Moduls festgestellt. Falls nicht alle externen Module identifiziert wurden, werden in Block **20308** die Anforderungen für das externe Modul identifiziert und wird das externe Modul spezifiziert. Die Spezifikation externer Module ermöglicht es einer IoT-Vorrichtung, auf die externen Module Bezug zu nehmen und auf die Anforderungen des externen Moduls zuzugreifen. In Block **20310** wird festgestellt, ob die Ressourcen der IoT-Vorrichtung durch die Anforderungen des externen Moduls überschritten werden. Die Anforderungen können beispielsweise die Modulleistung, Kommunikationsfähigkeiten, Kommunikationsgeschwindigkeiten, Speicheranforderungen und andere Fähigkeiten der IoT-Vorrichtungen und des Moduls einschließen.

[1151] Falls die Anforderungen der externen Module die Ressourcen der IoT-Vorrichtung an sich überschreiten, sendet die IoT-Vorrichtung in Block **20312** ein Signal zum externen Modul, um es zu deaktivieren. In Block **20314** kann die IoT-Vorrichtung einen sichtbaren oder hörbaren Alarm aktivieren. Der Alarm kann die Betätigung einer Leuchtdiode (LED), ein Audioton oder beides sein. Der Alarm in der Art einer LED kann einem Benutzer signalisieren, dass die Ressourcen durch die Anforderungen eines angegebenen externen Moduls überschritten wurden. Beispielsweise kann ein Mikrofon mit einem hohen Durchsatz, das als externes Modul wirkt, die Ressourcen einer einfachen Mikrosteuereinrichtung überschreiten, weil eine Verarbeitung mit einem hohen Durchsatz in der Mikrosteuereinrichtung nicht möglich sein kann. Zusätzlich zu einem lokalen Alarm kann von der IoT-Vorrichtung eine Nachricht zur Master-Vorrichtung gesendet werden.

[1152] Falls die Ressourcen der IoT-Vorrichtung ausreichen, um die Anforderungen der externen Module zu erfüllen, kann die IoT-Vorrichtung in Block **20316** eine Auflistung von sich, um ihre restlichen Ressourcen aufzunehmen, sowie eine Auflistung der Gesamtanforderungen einiger oder aller von dieser IoT-Vorrichtung betriebenen externen Module aktualisieren.

[1153] Der Prozessfluss wird in Block **20306** wieder aufgenommen, wo festgestellt wird, ob einige oder alle der mit der IoT-Vorrichtung verbundenen externen Module identifiziert und spezifiziert wurden. Sobald die externen Module identifiziert und spezifiziert wurden, können sie auf Ressourcen abgebildet werden. Beispielsweise kann ein Gassensor, der als externes Modul verwendet wird, Temperatur- und Feuchtigkeitsmessungen benötigen, um über Daten genau zu berichten. Die IoT-Vorrichtung kann jedoch keine Temperatur- und Feuchtigkeitssensoren aufweisen. Ansprechend auf eine Erkennung, dass ein Gassensor angeschlossen ist und Temperatur- und Feuchtigkeitsmessungen verwendet, kann die IoT-Vorrichtung eine Anforderung mit diesen Anforderungen zu einer Master-Vorrichtung senden. Die Master-Vorrichtung kann dann feststellen, ob die angeforderten externen Module in der Art des Temperatursensors und des Feuchtigkeitssensors entweder direkt oder durch eine andere angeschlossene IoT-Vorrichtung für die Master-Vorrichtung zugänglich sind.

[1154] Falls von der Master-Vorrichtung ein Temperatur- oder Feuchtigkeitssensor gefunden wird, beispielsweise in einem externen Modul, kann das externe Modul rekonfiguriert werden, so dass es von der IoT-Vorrichtung gesteuert wird. Die Sensoren können sich lokal an der IoT-Vorrichtung befinden oder sich in einem Modul außerhalb der IoT-Vorrichtung befinden, solange die Messung ausreichend nahe an einer Nutzbarkeit ist. Falls eine IoT-Vorrichtung beispielsweise Feuchtigkeits- und Temperaturinformationen wünscht, kann eine Master-Vorrichtung auf einen Temperatursensor oder einen Feuchtigkeitssensor im selben Raum wie die IoT-Vorrichtung oder in einem nahe gelegenen Flur zugreifen und diesen rekonfigurieren. Diese Module außerhalb der IoT-Vorrichtung können konfiguriert werden, so dass sie von der IoT-Vorrichtung gesteuert werden. Die Ressourcen dieser Sensoren können verwendet werden, um zu ermöglichen, dass ein Gassensor an der IoT-Vorrichtung für die Variablen der Temperatur und der Feuchtigkeit kalibriert wird, statt Rohdaten zurückzugeben.

[1155] Aus einer anderen Perspektive kann, falls ein externes Modul in der Art eines Gassensors Stromversorgungs-, Kommunikations- und Speicheranforderungen erfüllt, das externe Modul selbst dann zum System hinzugefügt werden, wenn der Gassensor keinen Zugriff auf die Temperatur- oder Feuchtigkeitsdaten hat und keine durch diese Faktoren kalibrierten Daten bereitstellen kann. Das Hinzufügen der Gassensorkomponente zur IoT-Vorrichtung kann jedoch von anderen IoT-Vorrichtungen in verschiedenen Konfigurationen, die eine Gasmessung benötigen, verwendet werden.

[1156] Sobald die externen Module identifiziert und spezifiziert wurden, wird in Block **20318** festgestellt, ob die Gesamtanforderungen der Summe der kombinierten Module und der IoT-Vorrichtung die Gesamtressourcen der IoT-Vorrichtung überschreiten. Die Gesamtressourcen der IoT-Vorrichtung, wie hier verwendet, beziehen sich im Allgemeinen auf die Ressourcen der IoT-Vorrichtung zuzüglich jeglicher externer Ressourcen, auf welche die IoT-Vorrichtung zugreifen kann, ohne eine Master-Vorrichtung zu benachrichtigen. Ressourcen einer IoT-Vorrichtung können sich in IoT-Fähigkeiten widerspiegeln. Bei einem Beispiel können diese Ressourcen der IoT-Vorrichtung zugeordnet werden oder zwischen mehreren miteinander verbundenen IoT-Vorrichtungen auf der Grundlage des Bedarfs der IoT-Vorrichtung und der angeschlossenen externen Module zugeordnet werden.

[1157] Falls die Gesamtressourcen der IoT-Vorrichtung durch die Gesamtmodulanforderungen überschritten werden, können die externen Module in Block **20320** außer einem gemeinsamen Modul deaktiviert werden. In Block **20322** kann die IoT-Vorrichtung das gemeinsame Modul verwenden, um eine Master-Vorrichtung über den Mangel an Gesamtressourcen zu benachrichtigen. Ansprechend auf den Empfang dieser Benachrichtigung kann die Master-Vorrichtung bestimmen, welche Ressourcen sie durch Rekonfigurieren eines Ressourcen-Pools einer spezifischen IoT-Vorrichtung neu zuordnen kann. Alternativ kann die Master-Vorrichtung ansprechend auf den Empfang einer Benachrichtigung die externen Module der IoT-Vorrichtung rekonfigurieren, so dass eine zweite IoT-Vorrichtung sie verwenden kann, während die erste IoT-Vorrichtung für eine andere Aufgabe oder einen anderen Zweck reinstalliert werden kann.

[1158] In Block **20324** kann eine LED, ein Audiosignal oder beides von der IoT-Vorrichtung betätigt werden, um eine lokale Angabe bereitzustellen, dass externe Module deaktiviert werden. In Block **20326** kann die Master-Vorrichtung eine Konfiguration zur Erfüllung fehlender Anforderungen identifizieren, indem sie externe Module der Steuerung durch die IoT-Vorrichtung unterordnet. Die Aktualisierung in der Konfiguration kann zur IoT-Vorrichtung gesendet und darauf angewendet werden. Beim Anwenden einer neuen Konfiguration auf eine

IoT-Vorrichtung können die der IoT-Vorrichtung zur Verfügung stehenden Ressourcen geändert werden. Beim Anwenden einer neuen Konfiguration auf eine IoT-Vorrichtung kann eine Änderung vorgenommen werden, falls externe Module unter der Steuerung durch die IoT-Vorrichtung verbleiben. Falls externe Module aus einer IoT-Vorrichtung entfernt werden, kann die IoT-Vorrichtung eine andere Prüfung vornehmen, um festzustellen, ob die restlichen Anforderungen der restlichen externen Module erfüllt werden können. Ansprechend auf eine Rekonfiguration kann die IoT-Vorrichtung in der Lage sein, ihre externen Module zu unterstützen, falls sich die Ressourcen der IoT-Vorrichtung geändert haben, falls sich die Summe der externen Anforderungen geändert hat oder falls die Rekonfiguration eine Funktion geändert hat, welche die IoT-Vorrichtung ausführen möchte. In Block **20328** und nach der Rekonfiguration durch die Master-Vorrichtung können neue Gesamtanforderungen für die neue Konfiguration externer Module an der IoT-Vorrichtung berechnet werden.

[1159] Falls in Block **20318** die Gesamtressourcen der IoT-Vorrichtung nicht durch die Gesamtmodulanforderungen überschritten werden, kann in Block **20330** die erweiterte Lebensdauer der IoT-Vorrichtung unter Verwendung eines Algorithmus, der die Lebensdauer einer Komponente vergleicht, berechnet werden. Bei einem beispielhaften Algorithmus kann die verlängerte Lebensdauer einer IoT-Vorrichtung festgelegt werden, so dass sie mit der kürzesten Restlebenszeit einer Komponente übereinstimmt, die, falls sie verloren geht oder deaktiviert wird, zu einer Rekonfiguration der IoT-Vorrichtung führen könnte, damit sie wie erwartet funktioniert.

[1160] Eine modulare IoT-Vorrichtung in Zusammenhang mit einem Benutzer oder einem Benutzerkonto kann ein in einer Dienstniveauvereinbarung bzw. Service Level Agreement (SLA) spezifiziertes Dienstniveau aufweisen. Eine SLA kann vereinbarte Fähigkeiten der IoT-Vorrichtung und -Konfiguration, eine erwartete Lebensdauer und eine erwartete Funktion, eine erwartete Leistungsfähigkeit und eine erwartete Verfügbarkeit der Vorrichtung aufweisen. In Block **20332** stellt die IoT-Vorrichtung fest, ob die Vorrichtungslebenszeit kürzer als die in einer SLA für einen bestimmten Benutzer oder ein bestimmtes Konto spezifizierte Lebenszeit ist. Falls dies der Fall ist, wird der Prozessfluss in Block **20322** fortgesetzt, wo eine Master-Vorrichtung benachrichtigt wird. Falls die Restlebenszeit der Vorrichtung kürzer ist als in der SLA vorgesehen, würde die IoT-Vorrichtung in ihrer vorliegenden Konfiguration die Anforderungen der SLA nicht erfüllen. Wenn die Master-Vorrichtung in Block **20332** benachrichtigt wird, kann eine neue Konfiguration mit externen Modulen, welche die SLA erfüllen, hinzugefügt werden.

[1161] Bei einem Beispiel kann eine Konfiguration einer IoT-Vorrichtung ein Modul oder Module aufweisen, wodurch Lebenszeiten von Vorrichtungen verlängert werden, um eine in der SLA spezifizierte Sensorlebenszeit zu erfüllen. Beispielsweise können die Lebenszeiten der einer IoT-Vorrichtung zur Verfügung stehenden externen Module mit der in der SLA spezifizierten Lebenszeit verglichen werden. Falls eine Lebenszeit kürzer ist als in der SLA spezifiziert, kann die IoT eine neue Konfiguration externer Module von der Master-Vorrichtung anfordern, welche den aufgelisteten SLA-Lebenszeitwert erfüllt.

[1162] Falls die Vorrichtungslebenszeit die in der SLA erwähnte Lebenszeit jedoch übersteigt, kann in Block **20334** festgestellt werden, ob eine Dienstqualitäts(QoS)-Messung für die IoT-Vorrichtung in ihrer vorliegenden Konfiguration austritt. Falls für die IoT-Vorrichtung und ihre externen Module keine QoS existiert, können in Block **20336** QoS-Metriken für die IoT-Vorrichtung erzeugt werden. Sobald diese QoS-Metriken erzeugt wurden oder falls QoS-Metriken in der IoT-Vorrichtung bereits vorhanden waren, kann die IoT-Vorrichtung in Block **20338** feststellen, ob die QoS kleiner als eine in der SLA spezifizierte QoS-Schwelle ist.

[1163] Falls die QoS kleiner als eine in der SLA spezifizierte angeforderte Schwelle ist, kann die IoT-Vorrichtung in Block **20340** die Master-Vorrichtung darüber benachrichtigen, dass die QoS niedriger als in der SLA gefordert ist, und das externe Modul oder die externen Module identifizieren, die erforderlich sein können, um die QoS zu ändern. In Block **20342** kann ein sichtbares oder Audiosignal in der Art einer LED oder eines Tons betätigt werden, um der IoT-Vorrichtung lokal anzugeben, dass die IoT-Vorrichtung eine QoS nicht erfüllt. In Block **20344** kann die IoT-Vorrichtung eine aktualisierte Konfiguration mit zusätzlichen, Ersatz- oder weniger externen Modulen empfangen, so dass die QoS-Messungen mit den Anforderungen der SLA übereinstimmen. Der Prozessfluss wird in Block **20334** fortgesetzt, wo eine neue QoS auf der Grundlage der aktualisierten Konfiguration gefunden wird.

[1164] Bei einem Beispiel kann die QoS für eine IoT-Vorrichtung durch Hinzufügen, Entnehmen und Ersetzen externer Module geändert werden. Diese Änderungen können zu einer QoS führen, die kleiner ist als die in der SLA spezifizierte QoS. Falls es beispielsweise keine historische QoS an einer IoT-Vorrichtung für das IoT-Vorrichtungs-Kommunikationsmodul gibt, kann die QoS auf der Basis dieser Vorrichtung getestet werden. Die QoS für das Kommunikationsmodul an einer IoT-Vorrichtung kann von der QoS für das Kommunikationsmo-

dul an einer anderen gleichen IoT-Vorrichtung mit einer unterschiedlichen Konfiguration gegenüber anderen externen Modulen abweichen.

[1165] Bei diesem Beispiel kann, wenn eine Kommunikationsmodul-QoS unterhalb einer in der SLA spezifizierten Schwelle liegt, die Master-Vorrichtung durch die IoT-Vorrichtung benachrichtigt werden und kann eine Anforderung einer neuen Kommunikationskonfiguration gemacht werden. Falls von der Master-Vorrichtung eine Aktualisierung der Konfiguration gewährt wird, kann ein neuer QoS-Test ausgeführt werden, um eine Beurteilung vorzunehmen und eine neue QoS für die aktualisierte Konfiguration zu finden. Wenn die QoS größer oder gleich der in der SLA aufgelisteten Schwelle ist, endet der Prozess in Block **20334** durch Einleiten einer Anwendung auf der IoT-Vorrichtung, welche die Fähigkeiten der in der vorliegenden Konfiguration der IoT-Vorrichtung vorhandenen externen Module benutzt.

[1166] Bei einem Beispiel kann die Konfiguration der IoT-Vorrichtung nach einer Anwendung, bei der eine IoT-Vorrichtung und ein bestimmter Satz externer Module verwendet werden, aufgelöst werden und können externe Module zur Rekonfiguration mit anderen IoT-Vorrichtungen aus der IoT-Vorrichtungssteuerung entfernt werden.

[1167] Ferner kann die selbstbeschreibende Hardware das hier beschriebene Knoten-Metamodell aufnehmen und eine Dienstniveauvereinbarung (SLA) als Parameter für Befehle, die davon akzeptiert werden, erfassen. Beispielsweise kann der Parameter die erwartete Energie spezifizieren, die für die Ausführung des Befehls verwendet wird, und kann ein Editor die verwendete Energie anpassen, um an eine erwartete SLA-Schwelle für eine erwartete Lebensdauer einer Vorrichtungsstromquelle anzupassen. Unter Verwendung von NMM und diesen SLA-Konventionen können IoT-Vorrichtungen gemäß einigen Ausführungsformen die hier beschriebenen Funktionen unterstützen und ausführen, ohne einen getrennten Treiber hinzuzufügen oder eine Aktualisierung der Systemsoftware auszuführen.

[1168] **Fig. 204** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **20400** vorhanden sein können, um Ressourcen und Anforderungen selbstbeschreibender Hardware abzubilden, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1169] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Ressourcenhardwarekomponentenidentifizierer **20402** zum Identifizieren einer von der IoT-Vorrichtung gesteuerten Ressourcenhardwarekomponente aufweisen, wobei die Ressourcenhardwarekomponente eine Fähigkeitsschwelle aufweist. Bei einem Beispiel kann die Ressourcenhardwarekomponente eine Stromquelle, eine Verarbeitungsressource, eine integrierte Kommunikationskomponente, einen Kontextsensor und ein Kontextbetätigungselement, eine Signalaufbereitungsschaltung, eine flüchtige Speicherressource oder eine permanente Speicherressource umfassen. Die Fähigkeitsschwelle, wie hier verwendet, bezieht sich im Allgemeinen auf eine minimale Funktionskompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine minimale Fähigkeit, miteinander zu funktionieren, angegeben wird. Die Fähigkeitsschwelle, wie hier verwendet, kann auch eine vollständige Kompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul einschließen, wodurch eine Fähigkeit angegeben wird, bei den höchsten Fähigkeiten des externen Moduls zu funktionieren.

[1170] Ein Angabenempfänger **20404** kann eine empfangene Angabe einer Anforderung externer Modulhardware von einem externen Modul verarbeiten. Bei einem Beispiel umfasst das externe Modul eine Modulressource, die zur Verwendung auf Anleitung der IoT-Vorrichtung mit der ersten Ressourcenhardwarekomponente zu poolen ist.

[1171] Ein Externes-Modul-Vergleicher **20406** kann die Anforderung externer Modulhardware mit der Fähigkeitsschwelle der Ressourcenhardwarekomponente der IoT-Vorrichtung verwenden. Der Deaktivierungssender **20408** sendet ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, ein Deaktivierungssignal zum externen Modul.

[1172] **Fig. 205** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **20500**, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Ressourcen und Anforderungen selbst-

beschreibender Hardware abzubilden, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1173] Das nichtflüchtige maschinenlesbare Medium **20500** kann einen Code **20502** aufweisen, um den Prozessor **902** anzuweisen, eine von der IoT-Vorrichtung gesteuerte Ressourcenhardwarekomponente zu identifizieren, wobei die Ressourcenhardwarekomponente eine Fähigkeitsschwelle aufweist. Die Fähigkeitsschwelle, wie hier verwendet, bezieht sich im Allgemeinen auf eine minimale Funktionskompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine minimale Fähigkeit, miteinander zu funktionieren, angegeben wird. Die Fähigkeitsschwelle kann auch eine Kompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul aufweisen. Dies kann die Fähigkeit angeben, bei den höchsten Fähigkeiten des externen Moduls zu funktionieren.

[1174] Das nichtflüchtige maschinenlesbare Medium **20500** kann einen Code **20504** aufweisen, um den Prozessor **902** anzuweisen, eine empfangene Angabe einer Anforderung externer Modulhardware von einem externen Modul zu verarbeiten. Das nichtflüchtige maschinenlesbare Medium **20500** kann einen Code **20506** aufweisen, um den Prozessor **902** anzuweisen, die Anforderung externer Modulhardware mit der Fähigkeitsschwelle der Ressourcenhardwarekomponente der IoT-Vorrichtung zu vergleichen. Das nichtflüchtige maschinenlesbare Medium **20500** kann einen Code **20508** aufweisen, um den Prozessor **902** anzuweisen, ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, ein Deaktivierungssignal zum externen Modul zu senden.

[1175] Das nichtflüchtige maschinenlesbare Medium **20500** kann Befehle aufweisen, die, wenn sie ausgeführt werden, den Prozessor anweisen, ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, eine Anforderung zur Master-Vorrichtung zu senden, wobei die Anforderung an die Master-Vorrichtung fordert, dass eine zweite Ressourcenhardwarekomponente zugewiesen wird, die von der IoT-Vorrichtung zu steuern ist. Das nichtflüchtige maschinenlesbare Medium **20500** kann eine zweite Ressourcenhardwarekomponente aufweisen, die von der IoT-Vorrichtung gesteuert wird, wobei die erste Ressourcenhardwarekomponente und die zweite Ressourcenhardwarekomponente gepoolt werden können, so dass die Fähigkeitsschwelle die Summe der Fähigkeitsschwellen der ersten Ressourcenhardware und der zweiten Ressourcenhardware ist.

[1176] Auf der Grundlage im computerlesbaren Medium gespeicherter ausgeführter Befehle kann eine Angabe gesendet werden, um eine nicht erfüllte Fähigkeitsschwelle anzugeben und einen sichtbaren Indikator zu aktivieren. Das nichtflüchtige maschinenlesbare Medium **20500** kann Befehle aufweisen, die, wenn sie ausgeführt werden, den Prozessor anweisen, das externe Modul ansprechend darauf, dass die Fähigkeitsschwelle erfüllt wird, unter die Steuerung der IoT-Vorrichtung zu stellen.

[1177] Das nichtflüchtige maschinenlesbare Medium **20500** kann zusätzliche Codeblöcke für die Ausführung. Dieser Code kann ansprechend darauf verwendet werden, dass die Lebenszeit des externen Moduls kürzer als die Betriebsdauer der IoT-Vorrichtung ist, um eine Anforderung eines aktualisierten externen Moduls auszusenden. Dieser Code kann ansprechend darauf verwendet werden, dass die Lebenszeit einer Ressourcenhardwarekomponente kürzer als die Betriebsdauer der IoT-Vorrichtung ist, und dem Prozessor können Anweisungen gesendet werden, eine Anforderung einer aktualisierten Ressourcenhardwarekomponente zu senden.

[1178] Wie vorstehend beschrieben wurde, kann sich das Internet der Dinge (IoT) zu einem Zustand fortentwickeln, bei dem jegliche Module über eine gemeinsame physikalische Schnittstelle in einer Plug-and-Play-Konfiguration mit jeder anderen Vorrichtung verwendet werden kann. Bei dieser Konfiguration kann unter Verwendung eines gemeinsamen Zusammenarbeitsfähigkeitsprotokolls Modul mit jeder Vorrichtung zusammenarbeiten können. IoT-Vorrichtungen können jedoch nicht die Energiekosten der Implementation einer Anwendung zwischen Modulen oder Vorrichtungen kennen, bevor die Module oder Vorrichtungen miteinander verbunden wurden. Wie hier verwendet, kann eine Anwendung einen Algorithmus, eine Operation, eine Subroutine, Software oder eine andere Aktion, die Rechenhardware ausführen kann, welche Energie verbrauchen könnte, einschließen.

[1179] Beispielsweise kann ein Modul, das eine Bluetooth-Low-Energy(BLE)-Funkeinrichtung aufweist, durch ein WiFi-Funkmodul in einer mit einer Knopfzelle betriebenen IoT-Vorrichtung ersetzt werden. Die WiFi-Funkeinrichtung kann in der Lage sein, sich erfolgreich mit der IoT-Vorrichtung zu verbinden, sie kann jedoch die Batterie der IoT-Vorrichtung innerhalb einer Stunde entleeren, so dass die IoT-Vorrichtung nutzlos wird. Wie hier beschrieben, kann die IoT-Vorrichtung in der Lage sein, die zur Verarbeitung von Anwendungen benötigte Energie zu berechnen und einen Alarm auszugeben, wenn ein Modul zu viel Strom verbrauchen kann. En-

ergieineffiziente Anwendungen können durch solche Merkmale identifiziert werden, die beispielsweise darin bestehen können, ob die Anwendung zu oft mit einem viel Strom entnehmenden Modul kommuniziert oder darauf beruht oder ob die Anwendung keinen Schlafmodus mit geringem Stromverbrauch implementiert.

[1180] Die Energieanforderungen für eine Anwendung können entsprechend der angeschlossenen Hardware unter Verwendung eines Berechnungswerkzeugs berechnet werden. Das Berechnungswerkzeug kann an einer IoT-Vorrichtung verwendet werden, wenn die IoT-Vorrichtung hochgefahren wird, wenn Module hinzugefügt oder entfernt werden oder wenn eine Anwendung durch eine interne Anforderung aktualisiert wird, falls die IoT-Vorrichtung selbstverwaltend ist, oder durch eine externe Anforderung aktualisiert wird, falls die Master-Vorrichtung die Vorrichtung steuert. Das Berechnungswerkzeug kann an einer Master-Vorrichtung in der Art eines Gateways nach einer Anforderung einer neuen Konfiguration von einer Slave-Vorrichtung in der Art eines Endknotens verwendet werden. Das Berechnungswerkzeug kann auch an einer Master-Vorrichtung als Vorverifizierungsverfahren, bevor Softwareaktualisierungen implementiert werden, verwendet werden. Das Berechnungswerkzeug kann auch Teil eines Vorinstallations-Simulationswerkzeugs sein, das Verletzungen minimaler in einer Dienstniveauevereinbarung (SLA) spezifizierter Parameter identifizieren kann.

[1181] Ein Konfigurationsoptimierungsprozess kann durch die Verwendung des Berechnungswerkzeugs unterstützt werden. Beispielsweise kann das Berechnungswerkzeug als Teil des Vorinstallations-Simulationswerkzeugs verwendet werden, wobei Anwendungseinstellungen oder Vorrichtungseinstellungen manipuliert werden können, um eine Konfiguration zu identifizieren, welche die Energieeffizienz der Vorrichtung verbessert. Ferner kann die Feststellung, was ein Modul oder eine Vorrichtung energieeffizienter macht, einen Vorschlag von Anwendungen bereitstellen, wodurch die Lebenszeit des Moduls oder der Vorrichtung verlängert werden könnte.

[1182] Das Berechnungswerkzeug kann Vorrichtungen und Module adressieren, die in der Lage sind, Ressourcen, Anforderungen und Anwendungen entsprechend einem gemeinsamen Protokoll über Vorrichtungen und Module selbst zu beschreiben. Wie hier offenbart, kann das Berechnungswerkzeug beispielsweise die Standardenergieverwendung eines Moduls identifizieren, indem es die historische Energieverwendung eines Moduls identifiziert, um Messungen zur Verwendung bei der Vornahme von Schätzungen, des spezifizierten Energieverbrauchs und dergleichen bereitzustellen. Beispielsweise kann der Nennenergieverbrauch eines 3G-Moduls von der Nähe des Moduls zu einer Basisstation abhängen.

[1183] Das Berechnungswerkzeug kann den veränderlichen Energieverwendungen zwischen verschiedenen getesteten Hardwarekonfigurationen Rechnung tragen. Bei dem Beispiel beruht das Berechnungswerkzeug auf mit zerlegbaren Aufgaben geschriebenen Anwendungen. Das Werkzeug kann auf dem Messen der Energieverwendung einer Anwendung auf einem Modul beruhen, wobei das Modul seinen Energieverbrauch für eine Aufgabe, die auf dem Modul ausgeführt werden könnte, selbst beschreiben kann. Beispiele von Aufgaben, über deren Energieverbrauch ein Modul berichten kann, umfassen unter anderem das Senden einer Nachricht, das Ablesen eines Sensors und das Versorgen einer LED mit Strom.

[1184] **Fig. 206** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **20600** zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Abbildung von Ressourcen und Anforderungen selbstbeschreibender Hardware gemäß einigen Ausführungsformen. Das Verfahren **20600** aus **Fig. 206** kann durch die mit Bezug auf **Fig. 207** beschriebene IoT-Vorrichtung **20700** implementiert werden. Das Verfahren **20600** kann unter Verwendung des mit Bezug auf **Fig. 8** beschriebenen Systems **802** ausgeführt werden. Das Verfahren **20600** kann in Block **20602** beginnen, wenn eine IoT-Vorrichtung bootet.

[1185] In Block **20604** stellt ein Berechnungswerkzeug fest, ob eine Vorrichtung durch ein festverdrahtetes Netz oder ein Drahtlosnetz mit einem Netz verbunden ist. Falls es durch ein festverdrahtetes Netz verbunden ist, kann es nicht erforderlich sein, den Strombedarf für die Energiebehandlung der lokalen Vorrichtung zu berechnen. Falls es festverdrahtet ist, kann das Berechnungswerkzeug in Block **20606** ein Signal senden, um die Anwendung an der Vorrichtung einzuleiten. Bei einigen Beispielen kann das Berechnungswerkzeug jedoch noch verwendet werden, falls die Vorrichtung ein festverdrahtetes Netz verwendet. Dies kann wichtig sein, falls die Vorrichtung andere Modultypen aufweist, die viel Energie entnehmen, wie eine Graphikverarbeitungseinheit (GPU), selbst wenn diese Module nicht drahtlos sind.

[1186] Falls das Berechnungswerkzeug feststellt, dass die Vorrichtung nicht festverdrahtet ist, werden in Block **20608** die Stromversorgungsressourcen, welche den Kern der Vorrichtung versorgen, spezifiziert. Hierbei können beispielsweise die Anzahl der Stromquellen, der Typ der Stromquellen, die Menge des in den Stromquel-

len enthaltenen Stroms, die Energiequellen für die Kernvorrichtung, die Batteriekapazität und die Energiegewinnungsfähigkeit sowie der Energieverbrauch durch die Kernvorrichtung aufgenommen werden.

[1187] In Block **20610** spezifiziert das Berechnungswerkzeug den Strombedarf für ein externes Modul. Wie hier erörtert, kann die Spezifikation des Strombedarfs für ein externes Modul beispielsweise die Angabe eines Grundenergieverbrauchs für das externe Modul, einen Energieverbrauch bei voller Stromentnahme für das externe Modul und dergleichen aufweisen. Der angegebene Grundenergieverbrauch kann für jede Peripherievorrichtung oder jedes hinzugefügte Modul, die oder das eine Vorrichtung während der Ausführung der Anwendung mit Strom versorgen kann, bestimmt werden. Das Berechnungswerkzeug kann einen Bereich für den Energieverbrauch, den durchschnittlichen Energieverbrauch und den maximalen Energieverbrauch für ein externes Modul bestimmen und identifizieren. Das Berechnungswerkzeug kann in einem externen Modul oder in einem zentralen Knoten gespeicherte und dem externen Modul zugeordnete Energieverwendungsspezifikationen identifizieren. Wie vorstehend erörtert, kann das Berechnungswerkzeug auf einem System arbeiten, bei dem ein externes Modul selbstbeschreibend ist und den für bestimmte Aktionen an diesem Modul erwarteten Energieverbrauch selbst beschreiben kann.

[1188] In Block **20612** zerlegt das Berechnungswerkzeug die analysierte Anwendung in kleinere Aufgaben. Die Aufgaben können diskrete Aktionen oder eine Reihe von Aktionen sein und werden in immer kleinere Aufgaben zerlegt, bis die Aufgaben mit bekannten Energieverwendungswerten auf der analysierten Hardware assoziiert sind. Diese Zerlegungsaktionen können zur späteren Verwendung beim Tabellieren der Gesamtenergiekosten über eine Vielzahl von Hardwarekonfigurationen gespeichert werden. Bei einem Beispiel kann die Spezifikation der Energieanforderungen für ein Modul vorgenommen werden, wobei der Ruhestromverbrauch und der aktive Stromverbrauch berücksichtigt werden. Der aktive Stromverbrauch kann der Strom sein, der beispielsweise verwendet wird, wenn eine Nachricht gesendet wird oder eine Messung ausgeführt wird.

[1189] In Block **20614** kann ein Berechnungswerkzeug feststellen, ob alle Aufgaben in einer Anwendung auf ihre Energieverwendung beurteilt wurden. Falls dies nicht der Fall ist, wird in Block **20616** festgestellt, ob die Aufgabe in Zusammenhang mit einem Wandlermodul steht. Wenngleich dieser Prozess eine Verfahrensunterteilung zwischen Aufgaben für ein Wandlermodul und Aufgaben für ein Kommunikationsmodul zeigt, können durch diesen Prozess andere Modultypen unter anderem in der Art einer GPU beurteilt werden. Die in Block **20614** und im in **Fig. 206** dargestellten Verfahren dargestellte Entscheidung erwägt die Tabellierung der Energie pro Aufgabe und pro Aufgabentyp für eine bestimmte Hardware und Konfiguration. Die Energieverwendung wird sowohl für die auf der Hardware der Vorrichtung, welche die Module hostet, implementierten Aufgaben als auch für die Module tabelliert. Dementsprechend sind die in **Fig. 206** dargestellten Entscheidungsblöcke und die Tabellierung zur Vereinfachung der Darstellung eines Tabellierungsprozesses aufgenommen, wodurch nicht nur Aufgaben für die Energieverwendung, sondern auch spezifische Module und Modulgruppen analysiert werden können.

[1190] Falls in Block **20616** das Berechnungswerkzeug feststellt, dass sich die Aufgabe auf ein Wandlermodul bezieht, wird in Block **20618** die Anzahl der Wandleranforderungen pro Stunde spezifiziert. Die Spezifikation der Anzahl der Wandleranforderungen kann beispielsweise einen Zahlenwert aufweisen, der über den Verlauf einer Stunde während der Ausführung der Anwendung gemessen wird. Die Spezifikation der Anzahl der Wandleranforderungen kann auch einen historischen Durchschnitt für die Anzahl der Anforderungen aufweisen, die bei Analysen für die fragliche Aufgabe an den Wandler gestellt werden.

[1191] In Block **20620** kann das Berechnungswerkzeug die Energie, die erforderlich ist, um die fragliche Aufgabe auszuführen, mit der Anzahl der Male multiplizieren, bei denen durch die Anwendung in einer Stunde eine Wandleranforderung gemacht wird. Der Zeitraum ist ein Beispiel und kann bei Bedarf länger oder kürzer sein. Beispielsweise kann der Zeitraum durch einen Benutzer oder einen Installateur des Werkzeugs eingestellt werden, oder der Zeitraum kann automatisch an eine geschätzte Ausführungszeit für die Anwendung angepasst werden. Der Zeitraum kann verlängert oder verkürzt werden. Gemäß einigen Ausführungsformen ist der gemessene Zeitraum für alle Anwendungen, deren Energieverbrauch gemessen wird, gleich. Bei dem Beispiel kann ein Modul seinen Energieverbrauch für eine spezifische Aufgabe selbst speichern. Falls beispielsweise eine Aufgabe oder eine Anzahl verwandter Aufgaben häufig an einem Modul ausgeführt wird, kann ein Modul den Energieanalyseschritt durch Speichern einer Anzahl von Energieverbrauchswerten verbessern. Falls ein Berechnungswerkzeug beispielsweise den Energieverbrauch für eine Aufgabe an einem Modul fordert, bei dem sich der Energieverbrauch der Aufgabe bereits im Modulspeicher befindet, ist keine weitere Analyse erforderlich, weil der Wert sofort zurückgegeben werden kann.

[1192] In Block **20622** kann das Berechnungswerkzeug seinen berechneten Wert dafür aktualisieren, wie viel Energie die Anwendung in der aktuellen Hardwarekonfiguration verwenden würde. Der Prozess kann dann zu Block **20614** zurückkehren, um festzustellen, ob alle Aufgaben für die Anwendung beurteilt wurden.

[1193] Falls in Block **20616** das Berechnungswerkzeug feststellt, dass sich die Aufgabe nicht auf ein Wandlerrmodul bezieht, wird in Block **20624** die Anzahl der Kommunikationsanforderungen in einer Stunde spezifiziert. Wie hier erwähnt, werden, während der Prozess eine Unterteilung des Verfahrens zwischen Aufgaben für ein Wandlermodul und Aufgaben für ein Kommunikationsmodul zeigt, andere Modultypen bei diesem Prozess und dieser Dichotomie erwogen. Die Darlegung des Messens des Energieverbrauchs für die Aufgaben eines Kommunikatormoduls ist lediglich ein Beispiel von Modulen, die gemessen werden können.

[1194] In Block **20626** kann das Berechnungswerkzeug die Energie, die erforderlich ist, um die Kommunikationsmodulaufgabe auszuführen, mit der Anzahl der Male multiplizieren, bei denen von der Anwendung in einer Stunde eine Kommunikationsanforderung gemacht wird. Wie vorstehend erörtert, ist der in der Figur dargestellte und hier offenbarte Zeitraum lediglich ein Beispiel und kann angepasst werden. In Block **20622** kann das Berechnungswerkzeug seinen berechneten Wert dafür, wie viel Energie die Anwendung in der aktuellen Hardwarekonfiguration verbrauchen würde, auf der Grundlage dessen aktualisieren, was es in Bezug auf den Energieverbrauch von Aufgaben für ein Kommunikatormodul gelernt hat. Der Prozess kann dann zu Block **20614** zurückkehren, um festzustellen, ob alle Aufgaben für die Anwendung beurteilt wurden.

[1195] Sobald das Berechnungswerkzeug in Block **20614** festgestellt hat, dass eine ausreichende Anzahl von Aufgaben in einer Anwendung beurteilt wurde, kann der Prozess in Block **20628** fortgesetzt werden. Beispielsweise kann der Stromversorgungsplan implementiert werden, während noch eine Berechnung des Stromverbrauchs anderer Aufgaben ausgeführt wird, wobei beispielsweise, falls die beurteilten Aufgaben im Verhältnis zu den Stromreserven einer Einheit einen hohen Stromverbrauchswert aufweisen, der Prozessfluss die folgenden Blöcke parallel implementieren kann, während noch der Stromverbrauch anderer Aufgaben beurteilt wird.

[1196] In Block **20628** stellt das Berechnungswerkzeug fest, ob eine Schlafmodusfunktion beurteilt wurde. Falls dies nicht der Fall ist, wird in Block **20630** die Zeit spezifiziert, die eine Anwendung im Schlafmodus verbringt. Falls es keine Schlafmodusfunktion gibt, kann die spezifizierte Zeit null sein. Das Spezifizieren der im Schlafmodus verbrachten Zeit kann in Sekunden, in Taktzyklen der Hardware oder in einer Unterteilung der objektiv messbaren Zeit gemessen werden. Die Spezifikation der Zeit kann das Messen der Schlafzeitdauer, die im Laufe einer Stunde während der Ausführung der Anwendung auftritt, einschließen. Bei einem Beispiel kann die Spezifikation der Zeit im Schlafmodus auch einen historischen Durchschnitt für die von einem Prozessor, der die fragliche Anwendung ausführt, im Schlafmodus verbrachte Zeit einschließen.

[1197] In Block **20632** wird der von der Anwendung, bei der der Prozessor im Schlafmodus ist, gesparte Prozessorstrom berechnet. Bei einem Beispiel kann der gesparte Prozessorstrom durch Multiplizieren des pro Zeiteinheit verbrauchten Prozessorstroms mit der Anzahl der Zeiteinheiten, die er im Schlafmodus verbringt, wie in Block **20630** bestimmt, bestimmt werden. Wie vorstehend erörtert, werden, wenngleich eine Zeiteinheit von einer Stunde dargestellt ist, auch andere Zeiteinheiten erwogen. Gemäß einigen Ausführungsformen sind die Zeiteinheiten über gemessene Module und Vorrichtungen konsistent.

[1198] Wenngleich die Berechnung bei diesem Beispiel für einen Schlafmodus ausgeführt wird, können ähnliche Energieverbrauchsberechnungen durch das Berechnungswerkzeug ausgeführt werden. Beispielsweise kann die Zeit gemessen werden, während derer ein Prozessor aktiv oder verfügbar ist, um andere Aufgaben auszuführen, während er die fragliche Anwendung ausführt. Diese Zeit kann, wenngleich dabei nicht notwendigerweise Strom gespart wird, tabelliert werden und als Zeit und Energie gezeigt werden, die für das Ausführen von Aufgaben verbraucht werden könnten, die sich nicht auf die gemessene Anwendung beziehen.

[1199] Sobald der von einem Prozessor im Schlafmodus gesparte Gesamtstrom berechnet wurde, kann das Berechnungswerkzeug in Block **20622** seinen berechneten Wert darüber aktualisieren, wie viel Energie die Anwendung in der aktuellen Hardwarekonfiguration verwenden kann. Weil der durch die Verwendung dieser Anwendung gesparte Gesamtstrom Strom für den Prozessor spart, kann dieser Wert den berechneten Wert dafür verringern, wie viel Energie die Anwendung verwenden würde. Der Prozessfluss kann dann zu Block **20614** zurückkehren, um festzustellen, ob eine ausreichende Anzahl von Aufgaben für die Anwendung beurteilt wurde, wie hier beschrieben.

[1200] Sobald das Berechnungswerkzeug in Block **20614** feststellt, dass eine ausreichende Anzahl von Aufgaben in einer Anwendung beurteilt wurde und dass die Schlafmodusfunktion in Block **20628** beurteilt wurde,

kann der Prozessfluss in Block **20634** fortgesetzt werden. In Block **20634** kann das Berechnungswerkzeug die Vorrichtungsstromlebenszeit berechnen. Bei einem Beispiel kann dies durch Vergleichen der Ressourcen der Module mit den Anforderungen der Module geschehen. Beim Vergleich können die Vorrichtungsressourcen durch Modulanforderungen dividiert werden. Bei einem Beispiel kann beim Vergleich eine Division des in den Vorrichtungsressourcen vorhandenen Stroms durch die berechneten Gesamtstromanforderungen pro Zeiteinheit, wie in Block **20622** aktualisiert, vorgenommen werden.

[1201] In Block **20636** kann das Berechnungswerkzeug feststellen, dass die Vorrichtungslebenszeit größer als eine vorgegebene minimale Schwelle des Energieverbrauchs ist, so dass der verbrauchte Strom ausreichend unterhalb der Energieverbrauchsschwelle liegt. Falls dies der Fall ist, kann das Prozessberechnungswerkzeug enden, so dass die Anwendung in Block **20606** eingeleitet werden kann.

[1202] Falls die Vorrichtungslebenszeit die minimale Schwelle des Energieverbrauchs nicht überschreitet, kann das Berechnungswerkzeug in Block **20638** eine Master-Vorrichtung darüber benachrichtigen, dass die minimale Schwelle des Energieverbrauchs nicht erfüllt wurde, und eine neue Konfiguration von Modulen fordern. Die Benachrichtigung der Master-Vorrichtung über den Energieverbrauch kann einen Durchbruch der von den Modulen verwendeten Energiemenge, die Menge der von einzelnen Aufgaben verbrauchten Energie, ob die Anwendung einen Schlafmodus aufweist, und andere Informationen, die während der Analyse des Energieverbrauchs der Module und der Kernvorrichtung gesammelt wurden, einschließen. Auf der Grundlage dieser Informationen kann eine Master-Vorrichtung Module identifizieren, die nicht so effizient arbeiten wie sie es könnten, und sie in einer neuen Konfiguration ersetzen.

[1203] Die neue Konfiguration von Modulen kann dem Berechnungswerkzeug in Block **20640** bereitgestellt werden. Die neue Konfiguration kann hinzugefügte Hardware, ersetzte Hardware oder Hardware aufweisen. Das Berechnungswerkzeug kann dann den Energieverbrauch der Anwendung in dieser neuen Konfiguration prüfen. Dementsprechend kann der Prozessfluss zu **20610** zurückkehren, wo die Anforderungen der von der Anwendung verwendeten Module spezifiziert werden können, wie vorstehend beschrieben.

[1204] **Fig. 207** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung 20700 für ein Berechnungswerkzeug für selbstbeschreibende Hardware vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind. Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation des hier beschriebenen Berechnungswerkzeugs aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Energieressourcenspezifizierer **20702** zum Spezifizieren von Energieressourcen für eine Vorrichtung, die dafür ausgelegt ist, eine Anzahl von Modulen mit Strom zu versorgen und zu steuern, aufweisen.

[1205] Der Massenspeicher **808** kann einen Energiebedarfsspezifizierer **20704** zum Spezifizieren des Energiebedarfs von Modulen unter anderem in der Art eines Mesh-Sendeempfängers **810** oder des Uplink-Sendeempfängers **814** aufweisen. Der Massenspeicher **808** kann einen Anwendungszerleger **20706** zum Zerlegen einer Anwendung in Aufgaben aufweisen, die, wenn sie aktiviert werden, vollständig auf einem einzigen Modul funktionieren.

[1206] Ein Stromverbrauchsidentifizierer **20708** kann den Stromverbrauch eines Moduls für jeden für die Analyse verwendeten Zeitraum identifizieren. Ein Aufgabenaktivierungszähler **20710** kann die Anzahl der Male zählen, mit denen die Aufgabe das Modul in einer Zeiteinheit aktiviert.

[1207] Der Massenspeicher **808** kann einen Gesamtenergieberechner **20712** zum Berechnen der in der Zeiteinheit durch ein Modul in der Anzahl von Modulen auf der Grundlage der Zeit, während derer die Aufgabe aktiv ist, der Zeitdauer des Aufgabenabschlusses auf dem einen Modul und der vom Modul während des aktiven Zeitraums benötigten Energie verbrauchten Gesamtenergie aufweisen.

[1208] Bei einem Beispiel kann der Massenspeicher **808** auch einen Gesamtenergieberechner **20712** mit einem Vorrichtungslebenszeitgenerator zur Erzeugung der Vorrichtungslebenszeit auf der Grundlage der in der Zeiteinheit von der Gesamtzahl von Modulen verbrauchten Gesamtenergie und der Energieressourcen für die Vorrichtung aufweisen. Ein Vorrichtungsanweiser kann zum Gesamtenergieberechner **20712** hinzugefügt werden, um die Vorrichtung anzuweisen, die Anwendung ansprechend darauf zu starten, dass die berechnete Vorrichtungslebenszeit größer oder gleich einer vor Beginn der Anwendungsanalyse für die Vorrichtung fest-

gelegten minimalen Zeitschwelle ist. Bei diesem Beispiel kann der Vorrichtungsanweiser anweisen, dass eine Anforderung einer neuen Konfiguration ansprechend darauf zu einer Master-Vorrichtung gesendet wird, dass die berechnete Vorrichtungslebenszeit kleiner als eine vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegte minimale Zeitschwelle ist. Ferner kann die Anforderung einer neuen Konfiguration bei einem Beispiel eine detaillierte Auflistung zumindest eines der Folgenden aufweisen: des Energieverbrauchs für die Aufgabe auf dem Modul, der vom Modul über die Zeiteinheit verbrauchten Energie und des Energieverbrauchs für einen Modultyp in der Anzahl von Modulen.

[1209] Bei einem Beispiel kann der Massenspeicher **808** Codeblöcke zur Implementation der Steuerung des der Vorrichtung zugeführten Stroms, der Aktualisierung einer Hardwarekonfiguration der Vorrichtung und einer Aktualisierung des Codes der Anwendung bereitstellen. Die Anwendungsenergie-Berechnungsvorrichtung kann auch einen Vorrichtungsanweiser aufweisen, um die Vorrichtung anzuweisen, die für eine erste Konfiguration bei diesem Beispiel berechnete Gesamtenergie pro Zeiteinheit zu speichern. Bei diesem Beispiel kann der Vorrichtungsanweiser die Vorrichtung anweisen, von einer Master-Vorrichtung eine zweite Konfiguration anzufordern. Ferner kann bei diesem Beispiel ein zweiter Gesamtenergieberechner eine zweite Gesamtenergie pro Einheit für die zweite Konfiguration der Vorrichtung berechnen.

[1210] Bei einem Beispiel kann der Massenspeicher **808** auch einen Gesamtenergievergleich als Teil des Gesamtenergieberechners **20712** aufweisen, um die Gesamtenergie pro Einheit der ersten Konfiguration mit der zweiten Gesamtenergie pro Einheit für die zweite Konfiguration zu vergleichen. Bei einem Beispiel kann der Massenspeicher **808** auch den Vorrichtungsanweiser aufweisen, um die Vorrichtung anzuweisen, entweder die erste Konfiguration oder die zweite Konfiguration oder beide auf der Grundlage des Vergleichs der Gesamtenergie pro Einheit und der zweiten Gesamtenergie pro Einheit anzufordern. Bei einem Beispiel kann der Massenspeicher **808** auch einen Vorrichtungsanweiser aufweisen, um die Vorrichtung anzuweisen, die Anwendung zu starten, falls festgestellt wird, dass die Vorrichtung von einer festverdrahteten Ressource mit Strom versorgt wird. Bei einem Beispiel kann die Anwendungsenergie-Berechnungsvorrichtung auch den Stromverbrauch eines Moduls für jeden aktiven Zeitraum identifizieren, wobei das Modul historische Energieverbrauchswerte in Zusammenhang mit dem Modul für die Aufgabe abrufen kann.

[1211] Bei einem Beispiel kann der Massenspeicher **808** auch einen Anwendungsschlafmodusidentifizierer als Teil des Gesamtenergieberechners **20712** zur Identifikation, ob die Anwendung eine Schlafmodusfunktion hat, aufweisen. Bei diesem Beispiel kann der Massenspeicher **808** auch einen Zeitmesser zum Messen der Zeit, während derer sich die Anwendung in einer Zeiteinheit im Schlafmodus befindet, aufweisen. Der Massenspeicher **808** kann auch einen Gesamtenergieberechner zur Berechnung der verbrauchten Gesamtenergie auf der Grundlage des Gesamtstroms, der auf der Grundlage der Zeit gespart wird, während derer sich der Prozessor für eine Anwendung im Schlafmodus befindet, aufweisen.

[1212] **Fig. 208** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **20800**, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Ressourcen und Anforderungen selbstbeschreibender Hardware abzubilden, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1213] Bei einem Beispiel kann der Prozessor das Bereitstellen von Strom für die Vorrichtung, das Aktualisieren einer Hardwarekonfiguration der Vorrichtung und das Aktualisieren des Codes der Anwendung anweisen. Das nichtflüchtige maschinenlesbare Medium **20800** kann einen Code **20802** aufweisen, um den Prozessor **902** anzuweisen, Energieressourcen für eine Vorrichtung, die dafür ausgelegt ist, eine Anzahl von Modulen mit Strom zu versorgen und zu steuern, zu spezifizieren.

[1214] Das nichtflüchtige maschinenlesbare Medium **20800** kann einen Code **20804** aufweisen, um den Prozessor **902** anzuweisen, den Energiebedarf der Anzahl von Modulen zu spezifizieren. Das nichtflüchtige maschinenlesbare Medium **20800** kann einen Code **20806** aufweisen, um den Prozessor **902** anzuweisen, eine Anwendung in Aufgaben zu zerlegen, die, wenn sie ausgeführt werden, vollständig als einzelnes Modul in der IoT-Vorrichtung wirken. Das nichtflüchtige maschinenlesbare Medium **20800** kann einen Code **20808** aufweisen, um den Prozessor **902** anzuweisen, den Stromverbrauch des Moduls in der Anzahl von Modulen für jeden aktiven Zeitraum zu identifizieren. Bei einem Beispiel werden beim Identifizieren des Stromverbrauchs des einen Moduls in der Anzahl von Modulen für jeden aktiven Zeitraum historische Energieverbrauchswerte in Zusammenhang mit dem Modul für die Aufgabe abgerufen.

[1215] Das nichtflüchtige maschinenlesbare Medium **20800** kann Befehle aufweisen, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Anzahl der Male, bei denen die Aufgabe das Modul in einer Zeiteinheit

aktiviert, zu zählen. Das nichtflüchtige maschinenlesbare Medium **20800** kann Befehle aufweisen, die, wenn sie ausgeführt werden, den Prozessor anweisen, die in der Zeiteinheit vom Modul verbrauchte Gesamtenergie zumindest teilweise auf der Grundlage der Zeit, während derer die Aufgabe aktiv ist, der Zeitdauer, während der die Aufgabe vom Modul ausgeführt wird, und der vom Modul im aktiven Zeitraum benötigten Energie zu berechnen.

[1216] Das maschinenlesbare Medium ist nicht auf die vorstehend aufgelisteten Codeblöcke beschränkt, sondern kann Code zur Ausführung anderer Funktionen aufweisen. Beispielsweise kann der Code den Prozessor anweisen, die Vorrichtungslebenszeit auf der Grundlage der in der Zeiteinheit für die Anzahl von Modulen verbrauchten Gesamtenergie und auf der Grundlage der Energieressourcen für die Vorrichtung zu erzeugen. Bei diesem Beispiel kann der Prozessor angewiesen werden, ansprechend darauf, dass die berechnete Vorrichtungslebenszeit kürzer als eine vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegte minimale Zeitschwelle ist, eine neue Konfiguration anzufordern, die zu einer Master-Vorrichtung gesendet wird. Wie hier verwendet, weist die Anforderung einer neuen Konfiguration eine detaillierte Auflistung wenigstens eines der Folgenden auf: des Energieverbrauchs für die Aufgabe am Modul, der vom Modul über die Zeiteinheit verbrauchten Energie und des Energieverbrauchs für einen Modultyp in der Anzahl von Modulen.

[1217] Bei den gegenwärtig offenbarten Techniken für das Internet der Dinge (IoT) können gemäß einigen Ausführungsformen Sensoren in der Lage sein, Algorithmen zu implementieren, die in der Lage sind, einen vom Sensor gesammelten Rohdatenpunkt zu verarbeiten und den Datenpunkt in bedeutungsvolle Informationen zu übersetzen. Beispielsweise kann ein Datenpunkt von einem Analogwert in der Art von 3,4 V in einen Temperaturmesswert in der Art von 78 °F umgewandelt werden.

[1218] Wie hier offenbart ist, können Sensoren in der Lage sein, sich mit generischen Stiften an einer IoT-Vorrichtung zu verbinden, um ihre Signalaufbereitungsanforderungen zu beschreiben und in Zusammenwirken mit der IoT-Vorrichtung eine geeignete Signalaufbereitungs-Schaltungsanordnung zu erzeugen. Ein Beispiel einer Signalaufbereitungs-Schaltungsanordnung kann ein Verstärker sein, dem ein Butterworth-Filter, beispielsweise zum Entfernen von Rauschen, folgt.

[1219] IoT-Vorrichtungen können feldprogrammierbare Gate-Array- bzw. Field-Programmable-Gate-Array (FPGA)-Vorrichtungen einschließen, die auf allen Ebenen einer IoT-Netzhierarchie eingebettet werden können. Ein FPGA ist allgemein eine integrierte Schaltung, die dafür ausgelegt ist, nach der Herstellung konfiguriert zu werden, gewöhnlich in einem manuellen Prozess durch einen OEM. Wie hier beschrieben, können Module ihre Signalaufbereitungsanforderungen selbst dann beschreiben, wenn die Module Sensoren sind, die keine Signalaufbereitungs-Schaltungsanordnung aufweisen. Die selbstbeschreibenden Module können mit FPGA-Schaltungen in IoT-Vorrichtungen verwendet werden, um eigens ausgelegte Signalaufbereitungshardware für die Module zu erzeugen.

[1220] Die IoT-Vorrichtung, welche die Module aufweist, kann weniger beschränkt sein als die Module. Wie hier verwendet, kann sich beschränkt allgemein auf Hardware oder Software beziehen, die Beschränkungen aufweist, welche den Betrieb einer Vorrichtung einschränken können, wie Hardwarebeschränkungen, Ressourcenbeschränkungen oder Lizenzierungs- oder Erlaubnisrestriktionen. Bei einem Beispiel kann eine weniger beschränkte IoT-Vorrichtung ein Linux®-Betriebssystem oder ein ähnliches Betriebssystem (OS) und einen allgemeinen Prozessor, wie mit Bezug auf **Fig. 8** beschrieben, aufweisen, wobei die IoT-Vorrichtung von einem elektrischen Stromnetz versorgt werden kann und Zugang zu einer Netzverbindbarkeit hoher Bandbreite aufweisen kann. Bei diesem Beispiel weist die IoT-Vorrichtung ein FPGA und einen FPGA-Compiler auf. Eine IoT-Vorrichtung, die einen FPGA-Compiler aufweist, kann eine Signalverarbeitungs-Schaltungsanordnung im FPGA für neue Module erzeugen, wenn die neuen Module angeschlossen werden.

[1221] Die Erzeugung eigens eingerichteter Hardware für die Signalaufbereitung in einem FPGA kann beginnen, wenn ein neues Modul an eine weniger beschränkte IoT-Vorrichtung angeschlossen wird. Ansprechend darauf, dass die IoT-Vorrichtung den Anschluss des neuen Moduls erkennt, liest die IoT-Vorrichtung Signalaufbereitungsanforderungen und einen Signalverarbeitungsalgorithmus aus dem Modul. Wie vorstehend erwähnt, können die beschränkten Module und Sensoren, die selbstbeschreibend sind, die Signalaufbereitungsanforderungen und einen bevorzugten Signalverarbeitungsalgorithmus aus dem Speicher für das Modul bereitstellen. Die weniger beschränkte IoT-Vorrichtung mit dem FPGA-Compiler kann dann eine eigens eingerichtete Aufbereitungsschaltungsanordnung erzeugen und die Aufbereitungsschaltungsanordnung im FPGA der IoT-Vorrichtung implementieren. Sobald die eigens eingerichtete Aufbereitungsschaltungsanordnung durch einen FPGA-Compiler implementiert wurde, kann die IoT-Vorrichtung ausgegebene Daten lesen und den Signalverarbeitungsalgorithmus auf die Daten anwenden, um die Informationen zu erzeugen.

[1222] Wenn eine eigens eingerichtete Hardware für die Signalaufbereitung erzeugt wird, kann die Vorrichtung, welche die Module empfängt, eine beschränkte Vorrichtung sein. Eine beschränkte Vorrichtung kann beispielsweise ein drahtloser Knoten sein, die Vorrichtung kann ein Prozessor mit einer geringen Leistungsaufnahme aufweisen, oder die Vorrichtung kann eine Netzverbindbarkeit mit einer geringen Bandbreite aufweisen. In der beschränkten Vorrichtung kann es eine weniger wirksame Implementation eines FPGA-Compilers geben. Die weniger wirksame Implementation eines FPGA-Compilers kann auf vorgefertigten Signalaufbereitungsblöcken am FPGA-Compiler an Stelle einer für den spezifischen FPGA-Compiler optimierten Signalaufbereitung beruhen.

[1223] Bei einem Beispiel können die Parameter an Signalaufbereitungsblöcken und Verbindungen zwischen ihnen durch den Prozessor definiert werden, nachdem der Sensor angeschlossen wurde. Während die Vorrichtung beschränkt sein kann, können die Module selbstbeschreibend sein und Nachrichten durch ein gemeinsames Datenprotokoll senden und empfangen. Wie vorstehend erörtert, ist ein gemeinsames Datenprotokoll allgemein ein Protokoll, das von allen Vorrichtungen verstanden wird.

[1224] Bei einem Beispiel können die Module die Signalaufbereitungsanforderungen von sich selbst beschreiben. Anhand dieser Modulselbstbeschreibung kann ein beispielhaftes Verfahren zur Erzeugung einer kunden-spezifischen Hardware zur Signalaufbereitung erläutert werden.

[1225] Die Erzeugung einer Signalaufbereitung in einer beschränkten Vorrichtung kann beginnen, wenn das Modul an die Vorrichtung angeschlossen wird. Die beschränkte Vorrichtung kann Signalaufbereitungsanforderungen und einen Signalverarbeitungsalgorithmus lesen. Die beschränkte Vorrichtung kann dann Parameter existierender FPGA-Compilerblöcke konfigurieren und die FPGA-Logikblöcke verbinden, um eine eigens eingerichtete Signalaufbereitungs-Schaltungsanordnung zu erzeugen. Die beschränkte Vorrichtung kann dann die Ausgabe von der eigens eingerichteten Signalaufbereitungs-Schaltungsanordnung, die sie erzeugt hat, lesen und demgemäß die vorgefertigten Signalaufbereitungsblöcke und den Verarbeitungsalgorithmus anwenden, um von Personen lesbare Daten zu erzeugen.

[1226] Wenn eine eigens eingerichtete Hardware zur Signalaufbereitung erzeugt wird, kann die die Module empfangende Vorrichtung eine beschränkte Vorrichtung sein, welche die weniger beschränkten Ressourcen einer IoT-Vorrichtung ausnutzt. Bei einem Beispiel könnte ein beschränkter Knoten die Signalaufbereitungsanforderungen von einem neu angeschlossenen Modul zu einem weniger beschränkten Gateway, einer weniger beschränkten Netzvorrichtung oder einer weniger beschränkten Cloud-Vorrichtung senden. Die weniger beschränkte IoT-Vorrichtung kann FPGA-Code erzeugen, der zur beschränkten Vorrichtung übertragen wird, beispielsweise zur Implementation in einem FPGA im beschränkten Modus. Die Übertragung des optimierten Codes kann durch Datenübertragungsmittel einschließlich festverdrahteter und Über-die-Luft(OTA)-Übertragungsmittel geschehen.

[1227] Bei diesem Beispiel kann das Verfahren zur Erzeugung eigens eingerichteter Hardware oder Signalaufbereitung beginnen, wenn die beschränkte Vorrichtung erkennt, dass ein neues Modul angeschlossen wurde. Die beschränkte Vorrichtung kann die Signalaufbereitungsanforderungen und den Signalverarbeitungsalgorithmus, die vom Modul beschrieben werden, lesen und beide zu einer weniger beschränkten IoT-Vorrichtung senden. Die beschränkte Vorrichtung kann auch Parameter der beschränkten Vorrichtung und Parameter, welche das FPGA der beschränkten Vorrichtung beschreiben, zur weniger beschränkten IoT-Vorrichtung senden. Die Signalaufbereitungsanforderungen können den Signalverarbeitungsalgorithmus und die Vorrichtungsparemeter und FPGA-Parameter für die beschränkte Vorrichtung aufweisen. Ansprechend auf den Empfang der Signalaufbereitungsanforderungen kann die weniger beschränkte IoT-Vorrichtung eine eigens eingerichtete Aufbereitungsschaltungsanordnung erzeugen. Die erzeugte eigens eingerichtete Aufbereitungsschaltungsanordnung kann für die beschränkte Vorrichtung und das FPGA der beschränkten Vorrichtung eigens eingerichtet werden. Sobald die eigens eingerichtete Aufbereitungsschaltungsanordnung durch die weniger beschränkte IoT-Vorrichtung erzeugt wurde, kann die beschränkte Vorrichtung die Ausgabe von der eigens eingerichteten Signalaufbereitungs-Schaltungsanordnung lesen und einen Signalverarbeitungsalgorithmus anwenden, um die Rohdaten zu verarbeiten und Informationen zu erzeugen.

[1228] Fig. 209 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens 20900 zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Konfiguration einer Signalaufbereitungs-Schaltungsanordnung gemäß einigen Ausführungsformen. Das Verfahren 20900 aus Fig. 209 kann durch die mit Bezug auf Fig. 210 beschriebene IoT-Vorrichtung 21000 implementiert werden. Das Verfahren 20900 kann unter Verwendung des mit Bezug auf Fig. 8 beschriebenen Systems 802 ausgeführt werden. Das Verfahren 20900 kann in Block 20902 beginnen, wenn ein neues Modul an die IoT-Vorrichtung angeschlossen wird.

[1229] In Block **20904** kann die IoT-Vorrichtung Signalaufbereitungsanforderungen bestimmen. Die Signalaufbereitungsanforderungen können beispielsweise ein Format, in dem eine Vorrichtung einem Signal entsprechende Informationen ausliefern kann, in der Art des Formatierens der Temperatur in Grad statt in gemessenen Spannungen oder des Formatierens einer Lichtmessung in Lumen an Stelle von Rohspannungen, die von einer Photozelle oder einem Photowiderstand ausgegeben werden, einschließen.

[1230] In Block **20906** kann festgestellt werden, ob die IoT-Vorrichtung eine beschränkte Vorrichtung oder eine weniger beschränkte Vorrichtung ist. Wie hier verwendet, kann sich weniger beschränkt allgemein auf eine Vorrichtung beziehen, die Hardware oder Software aufweist, welche allgemeinere Verarbeitungssoftware oder Kommunikationshardware implementieren kann oder eine erweiterte Lizenzierung oder erweiterte Erlaubnisse zum Lesen, Schreiben oder Ausführen empfangener oder erzeugter Daten aufweist.

[1231] Falls die Vorrichtung weniger beschränkt ist, kann in Block **20908** FPGA-Code für die Vorrichtung auf der Grundlage des selbstbeschreibenden Moduls und der Signalaufbereitungsanforderungen der Vorrichtung erzeugt werden. In Block **20910** wird eine Entscheidung getroffen, ob das FPGA an der Vorrichtung für die Aufgabe ungeeignet ist. Falls dies der Fall ist, kann in Block **20912** eine Warnung gesendet werden und kann die Vorrichtung das Modul abtrennen.

[1232] Falls in Block **20910** festgestellt wird, dass das FPGA für die Aufgabe geeignet ist, kann der FPGA-Code in Block **20914** zur Vorrichtung heruntergeladen werden. Der FPGA-Code kann an der Vorrichtung selbst erzeugt werden. Der FPGA-Code kann fern erzeugt werden, beispielsweise durch eine Vorrichtung in der Cloud, und dann zur IoT-Vorrichtung heruntergeladen werden. Sobald der FPGA-Code erzeugt oder heruntergeladen wurde, kann ein FPGA-Compiler eine Signalaufbereitungsschaltung auf der Grundlage des FPGA-Codes erzeugen. Die Signalaufbereitungsschaltung kann dann am FPGA implementiert werden. Dementsprechend kann das Daten sammelnde Modul in Block **20916** Daten sammeln, die zur Verarbeitung durch die Signalaufbereitungsschaltung im FPGA zu führen sind, um die zugeführten endgültigen Informationen zu erzeugen.

[1233] Falls in Block **20906** festgestellt wird, dass die Vorrichtung beschränkt ist, wird in Block **20918** entschieden, ob die Vorrichtung genügend konfigurierbare Signalaufbereitungsblöcke an einem verfügbaren FPGA aufweist. Das verfügbare FPGA kann sich lokal an der Vorrichtung oder fern von der Vorrichtung befinden und von der Vorrichtung gesteuert werden. Falls dies der Fall ist, kann die Vorrichtung in Block **20920** die Konfiguration der Parameter der existierenden Blöcke am FPGA anweisen. Dies kann beispielsweise auf FPGA-Code beruhen, der an einer weniger beschränkten Vorrichtung erzeugt wird und zur beschränkten Vorrichtung heruntergeladen wird.

[1234] Sobald sie konfiguriert wurde, kann die Vorrichtung in Block **20922** die existierenden FPGA-Blöcke miteinander verbinden, um eine Signalaufbereitungsschaltung für das Modul in der beschränkten Vorrichtung zu bilden. Wie vorstehend beschrieben, können die verbundenen Schaltungen des FPGA gemäß einigen Ausführungsformen verwendet werden, um ein an einem Modul empfangenes Signal aufzubereiten. Dadurch kann das Daten sammelnde Modul Daten in Block **20916** sammeln und die Daten der Signalaufbereitungsschaltung im FPGA bereitstellen.

[1235] Falls in Block **20918** festgestellt wird, dass die Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke an einem FPGA hat, werden dann in Block **20924** die FPGA-Blöcke der Vorrichtung und die Vorrichtungseigenschaften spezifiziert. Wie hier verwendet, kann gemäß einigen Ausführungsformen die Spezifikation von FPGA-Blöcken und Vorrichtungseigenschaften verwendet werden, um Einzelheiten über die FPGA-Blöcke und die Vorrichtung zu identifizieren. Die Spezifikation von FPGA-Blöcken und Vorrichtungseigenschaften kann auch verwendet werden, um die Einzelheiten über die FPGA-Blöcke und die Vorrichtung von der aktuellen beschränkten Vorrichtung in Block **20926** zu einer Master-Vorrichtung auf einer höheren hierarchischen Ebene zu senden.

[1236] Sobald sie von der Master-Vorrichtung auf der höheren hierarchischen Ebene empfangen wurden, kann in Block **20928** festgestellt werden, ob die Master-Vorrichtung die FPGA-Kompilierfähigkeiten hat, welche für die Vorrichtungseigenschaften und das spezifizierte FPGA geeignet sind. Falls dies nicht der Fall ist, kann die ursprüngliche Vorrichtung in Block **20926** die spezifizierten FPGA-Blöcke und Vorrichtungseigenschaften zu einer zweiten Master-Vorrichtung auf einer höheren hierarchischen als die Master-Vorrichtung senden. Falls die Master-Vorrichtung die angeforderten Fähigkeiten hat, kann sie in Block **20930** den FPGA-Code zur Verarbeitung der Daten von der beschränkten Vorrichtung erzeugen.

[1237] In Block **20932** wird festgestellt, ob das FPGA an der beschränkten Vorrichtung für die Aufgabe ungeeignet ist. Falls dies der Fall ist, kann in Block **20912** eine Warnung über die Inkompatibilität ausgesendet werden und kann die Vorrichtung vom Modul getrennt werden. Falls in Block **20932** festgestellt wird, dass das FPGA an der Vorrichtung für die Aufgabe nicht ungeeignet ist, wird in Block **20934** der FPGA-Code zur beschränkten Vorrichtung heruntergeladen. Sobald der FPGA-Code zur Vorrichtung heruntergeladen wurde, kann der FPGA-Compiler eine Signalaufbereitungsschaltung auf der Grundlage des FPGA-Codes erzeugen. Daher kann das Daten sammelnde Modul in Block **20916** damit beginnen, Daten zu sammeln, die durch das Modul und die Vorrichtungssignalaufbereitungsschaltung auf der Grundlage des erzeugten FPGA-Codes zu leiten sind.

[1238] **Fig. 210** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **21000** zur Konfiguration einer Signalaufbereitungs-Schaltungsanordnung vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1239] Die IoT-Vorrichtung **21000** kann ein FPGA **21002** aufweisen. Wie hier verwendet, kann gemäß einigen Ausführungsformen ein FPGA eine integrierte Schaltung sein, die dafür ausgelegt ist, nach der Herstellung von einem Kunden oder einem Entwickler konfiguriert zu werden. Bei einem Beispiel kann ein selbstbeschreibendes Sensormodul **21004** ein Sensor-FPGA **21006** aufweisen. Das Sensor-FPGA **21006** kann ähnlich dem FPGA **21002** der IoT-Vorrichtung **21000** wirken. Die IoT-Vorrichtung **21000** kann hier als eine selbstbeschreibende IoT-Vorrichtung **21000** beschrieben werden.

[1240] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Sensormoduldetektor **21008** zur Erkennung des selbstbeschreibenden Sensormoduls **21004** ansprechend auf den Anschluss des selbstbeschreibenden Sensormoduls **21004** an die IoT-Vorrichtung **21000** aufweisen.

[1241] Der Massenspeicher **808** kann einen Prozessor **21010** für empfangene Daten zur Verarbeitung vom selbstbeschreibenden Sensormodul **21004** empfangener Daten aufweisen, wobei die empfangenen Daten Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul **21004** angeben. Der Massenspeicher **808** kann einen FPGA-Codegenerator **21012** zur Erzeugung von FPGA-Code zur Implementation auf dem FPGA **21002** aufweisen.

[1242] Der Massenspeicher **808** kann einen Rohdatenmodifizierer **21014** zum Modifizieren vom selbstbeschreibenden Sensormodul **21004** empfangener Rohdaten aufweisen, um unter Verwendung des FPGA auf der Grundlage der Signalaufbereitungsinformationen signalaufbereitete Daten zu erzeugen. Bei einem Beispiel kann der Prozessor **802** feststellen, dass das FPGA den FPGA-Code nicht unterstützen kann. Bei diesem Beispiel kann der Prozessor ansprechend auf die Erkennung, dass die FPGA-Blöcke für die Signalaufbereitung der vom selbstbeschreibenden Sensormodul **21004** empfangenen Rohdaten ausreichen, auch mit existierenden FPGA-Blöcken verbinden.

[1243] Der Prozessor **802** kann auch identifizierte FPGA-Eigenschaften und Eigenschaften selbstbeschreibender Vorrichtungen übertragen. Falls dies der Fall ist, kann der Prozessor **802** auch eine Anforderung zu einer Master-Vorrichtung in der Fog **812** oder Cloud **302** senden, die FPGA-Eigenschaften und Eigenschaften selbstbeschreibender Vorrichtungen aufweist. Die Anforderung kann ansprechend auf die Erkennung, dass die selbstbeschreibende IoT-Vorrichtung **21000** beschränkt ist, und die Erkennung, dass das FPGA der selbstbeschreibenden Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke aufweist, gesendet werden. Ferner kann die Anforderung einen von der Master-Vorrichtung erzeugten FPGA-Code erzeugen und zurückgeben. Ähnlich kann eine selbstbeschreibende IoT-Vorrichtung **21000**, die eine Batterie **824**, eine Internetverbindung auf der Grundlage einer drahtlosen Technologie, wie beispielsweise durch den Mesh-Sendeempfänger **810** oder den Uplink-Sendeempfänger **814** bereitgestellt, oder einen Prozessor **802** mit einem Modus mit geringer Leistungsaufnahme aufweist, gemäß einigen Ausführungsformen als beschränkte selbstbeschreibende Vorrichtung angesehen werden.

[1244] Bei einem anderen Beispiel kann der Prozessor **802** die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul **21004** und Vorrichtungsinformationen ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, zu einer weniger beschränkten Vorrichtung senden. Bei diesem Beispiel kann der Prozessor **802** auch zur weniger beschränkten Vorrichtung eine Anforderung

senden, einen Rückkehr-FPGA-Code zu erzeugen, der am FPGA **21006** des selbstbeschreibenden Sensormoduls **21004** zu implementieren ist. Ferner kann der Prozessor **802** die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul **21004** und Vorrichtungsinformationen ansprechend auf eine Erkennung, dass die selbstbeschreibende IoT-Vorrichtung **21000** beschränkt ist, zu einer zweiten beschränkten Vorrichtung senden, wobei die zweite beschränkte Vorrichtung auf eine weniger beschränkte Vorrichtung zugreifen kann.

[1245] Das selbstbeschreibende Sensormodul **21004** kann dem hier beschriebenen Knoten-Metamodell (NMM) folgen. Dieses kann als FPGA implementiert werden, wobei ein Verfahren zur dynamischen Aktualisierung des FPGA zu einem dynamischen Betrieb des selbstbeschreibenden Sensormoduls **21004** führen kann, wodurch auf der Grundlage eines Knoten-Metamodells eine universell zusammenarbeitsfähige Schnittstelle implementiert wird (beispielsweise in einem Plug-and-Play-Modus).

[1246] **Fig. 211** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **21100**, das Befehle aufweist, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Signalaufbereitungs-Schaltungsanordnung zu konfigurieren, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1247] Das nichtflüchtige maschinenlesbare Medium **21100** kann einen Code **21102** aufweisen, um den Prozessor **902** anzuweisen, ein selbstbeschreibendes Sensormodul zu erkennen, nachdem das selbstbeschreibende Sensormodul mit der selbstbeschreibenden Vorrichtung verbunden wurde, wobei die selbstbeschreibende Vorrichtung ein FPGA aufweist. Bei einem Beispiel kann die selbstbeschreibende Vorrichtung eine Batteriestromquelle und/oder eine Internetverbindung auf der Grundlage einer drahtlosen Technologie aufweisen, oder ein Prozessor mit einem Modus mit geringer Leistungsaufnahme wird als beschränkte selbstbeschreibende Vorrichtung angesehen.

[1248] Das nichtflüchtige maschinenlesbare Medium **21100** kann einen Code **21104** aufweisen, um den Prozessor **902** anzuweisen, vom selbstbeschreibenden Sensormodul empfangene Daten zu verarbeiten, wobei die empfangenen Daten Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul angeben. Das nichtflüchtige maschinenlesbare Medium **21100** kann einen Code **21106** aufweisen, um den Prozessor **902** anzuweisen, einen FPGA-Code zu erzeugen, der am FPGA der selbstbeschreibenden Vorrichtung zu implementieren ist. Schließlich kann das nichtflüchtige maschinenlesbare Medium **21100** einen Code **21108** zum Modifizieren vom selbstbeschreibenden Sensormodul empfangener Rohdaten zur Erzeugung signalaufbereiteter Daten unter Verwendung des FPGA auf der Grundlage der Signalaufbereitungsinformationen aufweisen.

[1249] Bei einem Beispiel kann das maschinenlesbare Medium Befehle aufweisen, die, wenn sie ausgeführt werden, den Prozessor anweisen, zu erkennen, dass das FPGA den FPGA-Code nicht unterstützen kann. Bei diesem Beispiel können die Befehle auch einen Prozessor anweisen, existierende FPGA-Blöcke ansprechend auf die Erkennung, dass die FPGA-Blöcke für die Signalaufbereitung der vom selbstbeschreibenden Sensormodul empfangenen Rohdaten ausreichen, zu verbinden. Ferner können Befehle auch einen Prozessor anweisen, identifizierte FPGA-Eigenschaften und Eigenschaften einer selbstbeschreibenden Vorrichtung zu übertragen. Der Prozessor kann dann eine Anforderung zu einer Master-Vorrichtung senden, wobei die Anforderung FPGA-Eigenschaften und Eigenschaften einer selbstbeschreibenden Vorrichtung aufweist, wobei die Anforderung ansprechend auf die Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, und auf die Erkennung, dass das FPGA der selbstbeschreibenden Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke aufweist, zu senden ist. Bei einem Beispiel dient die Anforderung der Erzeugung und Rückgabe eines von der Master-Vorrichtung erzeugten FPGA-Codes.

[1250] Das maschinenlesbare Medium ist nicht auf die vorstehend aufgelisteten Codeblöcke beschränkt, sondern kann Code zur Ausführung anderer Funktionen aufweisen. Beispielsweise kann der Code die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen zu einer weniger beschränkten Vorrichtung senden, nachdem erkannt wurde, dass die selbstbeschreibende Vorrichtung beschränkt ist. Der Code kann zur weniger beschränkten Vorrichtung eine Anforderung zur Erzeugung eines Rückkehr-FPGA-Codes, der am FPGA der selbstbeschreibenden Vorrichtung zu implementieren ist, senden. Alternativ kann der Prozessor die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, zu einer zweiten beschränkten Vorrichtung senden, wobei die zweite beschränkte Vorrichtung auf eine weniger beschränkte Vorrichtung zugreifen kann.

[1251] IoT-Netze können eine Vielzahl von Vorrichtungstypen aufweisen. Einige IoT-Vorrichtungen können von einer Batterie versorgt werden, und andere IoT-Vorrichtungen können Strom von einem elektrischen Stromnetz verwenden. Ferner sind einige IoT-Vorrichtungen stationär, während andere IoT-Vorrichtungen mobil sind. Die Berücksichtigung des Verhaltens mobiler Vorrichtungen und von Vorrichtungen, die häufig in Schlafzustände eintreten, kann die Art beeinflussen, in der eine Netzfunktionstüchtigkeits-Überwachungseinrichtung skaliert, um einer großen Anzahl von Vorrichtungen Rechnung zu tragen. Wie hier beschrieben, kann die Erzeugung von Funktionstüchtigkeitsberichten, beispielsweise unter Verwendung von Bloomfiltern, Blockchains, DHT und dergleichen, skaliert werden, um eine Netzfunktionstüchtigkeits-Überwachungseinrichtung zu implementieren.

[1252] Fig. 212 ist ein schematisches Diagramm einer hierarchischen Vorrichtungs- und Netz-Funktionstüchtigkeitsberichterstattung **21200** gemäß einigen Ausführungsformen. Wie gegenwärtig organisiert, können Netztopologien hierarchisch sein. Die beschriebenen Techniken zur Netz-Funktionstüchtigkeitsberichterstattung können auf den hierarchischen Topologien aufbauen. Beispielsweise kann ein IoT-Framework-Akteur **21202** eine Hierarchie vernetzen und Vorrichtungen überblicken möchten. Der IoT-Framework-Akteur **21202** kann Parteien aufweisen, die sich auf das Netz verlassen, wie unter anderem Netzadministratoren, vernetzte Konsolen und Geschäftsdienstebenen-Vereinbarungsanbieter.

[1253] Eine einzige Vorrichtung in der Art einer Vorrichtung-A oder viele Vorrichtungen innerhalb eines Subnetzes können einen Vorrichtungs-Funktionstüchtigkeitsbericht **21204** erzeugen. Der Vorrichtungs-Funktionstüchtigkeitsbericht **21204** kann Vorrichtungs-Funktionstüchtigkeitsberichtsdaten **21206** aufweisen, die eine Zeitvariable, die einen Wert aufweist, welcher den Zeitraum angibt, den der Bericht abdeckt, und eine Spärliches-Feld-Variable mit einer Shadow-Filter-Eingabe aufweisen. Ein Shadow-Filter kann Daten sammeln, wenn ein Bericht erwartet wird, jedoch nicht vorhanden ist.

[1254] Ein Überwachungsagent kann einen Representational-State-Transfer-(REST)-Aufruf **21208** von einem Subnetz vornehmen, um Vorrichtungs-Funktionstüchtigkeitsberichte für alle Vorrichtungen innerhalb des Subnetzes abzurufen, wie den Vorrichtung-A-Funktionstüchtigkeitsbericht **21204** und andere, und sie als Subnetz-Funktionstüchtigkeitsbericht **21210** zusammenzustellen. Wie hier verwendet, ist ein REST-Aufruf ein Aufruf, der es anfordernden Systemen ermöglicht, auf Textrepräsentationen von Webressourcen unter Verwendung eines einheitlichen und vordefinierten Satzes zustandsloser Operationen zuzugreifen und diese zu manipulieren. Bei einem Beispiel kann ein Subnetz-Funktionstüchtigkeitsbericht **21210** einer von vielen Subnetz-Funktionstüchtigkeitsberichten in der Art von Subnetz-A-, Subnetz-B- und Subnetz-C-Funktionstüchtigkeitsberichten sein. Jedes Subnetz kann eine Anzahl von Vorrichtungen überwachen. Der Subnetz-Funktionstüchtigkeitsbericht **21210** kann Subnetz-Funktionstüchtigkeitsberichtsdaten **21212** aufweisen. Subnetz-Funktionstüchtigkeitsberichtsdaten **21210** können unter anderem den Zeitraum, den die Vorrichtungsberichte abdecken, und eine Spärliches-Feld-Variable mit den netzgefilterten Shadow-Filter-Daten in der Art von Netzfilterdaten aufweisen. Die Subnetz-Funktionstüchtigkeitsberichtsdaten **21210** können eine Auflistung der Sammlung von Vorrichtungen, die innerhalb des Subnetzes abgedeckt werden, mit Verbindungen, die verwendet werden können, um auf diese Vorrichtungen zuzugreifen oder sie zu adressieren, aufweisen.

[1255] Ein Überwachungsagent kann einen Representational-State-Transfer-(REST)-Aufruf **21208** von einem Netz zum Abrufen von Subnetz-Funktionstüchtigkeitsberichten für alle Subnetze innerhalb des Netzes in der Art des Subnetz-A-Funktionstüchtigkeitsberichts **21210** und anderer vornehmen und sie zu einem Netz-Funktionstüchtigkeitsbericht **21214** zusammenstellen. Der Netz-Funktionstüchtigkeitsbericht **21214** kann Netz-Funktionstüchtigkeitsberichtsdaten **21216** aufweisen. Netz-Funktionstüchtigkeitsberichtsdaten **21216** können den Zeitraum, den die Subnetzberichte abdecken, und eine Spärliches-Feld-Variable mit den netzgefilterten Shadow-Filter-Daten in der Art der Netzfilterdaten aufweisen. Die Netz-Funktionstüchtigkeitsberichtsdaten **21216** können eine Auflistung der Sammlung innerhalb des Netzes abgedeckter Subnetze mit Verbindungen, die für den Zugriff auf diese Subnetze und das Adressieren von ihnen verwendet werden können, aufweisen.

[1256] Unter Verwendung dieser Netztopologie und -hierarchie können die Funktionstüchtigkeitsberichte durch den IoT-Framework-Akteur **21202** zusammengestellt werden, um die Funktionstüchtigkeit des IoT-Netzes zu überwachen. Die Überwachung der Netzfunktionstüchtigkeit auf einer Topologiebasis ermöglicht es, ein hohes Niveau an Informationen zu erhalten und sie nach Anforderung sowohl breit als auch schmal zu repräsentieren.

[1257] Fig. 213 ist ein schematisches Diagramm einer Vorrichtungsniveau-Bloomfilter- und einer Shadow-Filter-Funktionstüchtigkeitsberichterstattung **21300** gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 212 beschrieben wurden. Eine Vorrichtung **21302** kann dem

Vorrichtung-A-Funktionstüchtigkeitsbericht **21204** eine Funktionstüchtigkeitsnachricht **21304** bereitstellen. Die Funktionstüchtigkeitsnachricht **21304** kann geplant werden, so dass sie mit einem bestimmten Zeitraum wiederholt wird, beispielsweise einmal pro Sekunde. Der Bericht kann Funktionstüchtigkeitsinformationen aufweisen. Es kann jedoch ein einfacher Ping, der die Funktionstüchtigkeit und die Funktion angibt, oder eine andere ähnliche Warnung zu Operationen angemessen sein.

[1258] Bei einem Beispiel kann die Funktionstüchtigkeitsberichterstattung auf der Vorrichtungsebene auf einem Watchdog-Agenten beruhen. Der Watchdog-Agent kann erkennen, dass eine erwartete Funktionstüchtigkeit nicht empfangen wurde. Die Erkennung, dass eine erwartete Funktionstüchtigkeit nicht empfangen wurde, kann im Allgemeinen von der Erkennung eines Fehlerszenarios und vom aktiven Berichten über den Fehler verschieden sein. Bei einem Beispiel kann der Watchdog-Agent ein Bloomfilter **21306** verwenden, um den Empfang einer Watchdog-Nachricht gemäß einem erwarteten Plan zu bestätigen.

[1259] Ein Bloomfilter ist allgemein ein Filter, das gemäß dem erwarteten Plan gebildet werden kann. Beispielsweise kann, falls die erwartete Rate von Watchdog-Nachrichten innerhalb eines Zeitraums, beispielsweise von Sekunden, empfangen wird, ein zweites Bloomfilter aktualisiert werden, beispielsweise durch eine durch die Pfeile in **Fig. 42B** angegebene Rollover-Funktion, um anzugeben, dass eine gute Funktionstüchtigkeit über einen längeren Zeitraum, beispielsweise von Minuten, Stunden, Tagen, Wochen und dergleichen, gegeben ist. Nach diesem Muster kann eine Aufwärtskaskade von Bloomfiltern zunehmend längere Zeiträume eines gesunden Betriebs speichern. Wie hier offenbart, kann gemäß einigen Ausführungsformen ein Bericht zur Funktionstüchtigkeit einer Vorrichtung untersucht werden, wenn es eine erwartete Watchdog-Nachricht gibt, die nicht innerhalb eines erwarteten Zeitfensters empfangen wird.

[1260] Jedes Bloomfilter **21306** deckt wie dargestellt einen bestimmten Zeitraum mit einem entsprechenden Bit für jedes Inkrement dieses Zeitintervalls ab. Beispielsweise codiert das Bloomfilter für Sekunden zu 60 Bits, wodurch ein Bit für jede Sekunde in einer Minute ermöglicht wird. Das Bloomfilter für Sekunden kann an das Bloomfilter für Minuten übergeben, das für 60 Bits codiert, wodurch ein Bit für jede Minute in einer Stunde ermöglicht wird. Das Bloomfilter für Minuten kann an das Bloomfilter für Stunden übergeben, das für 24 Bits codiert, wodurch ein Bit für jede Stunde an einem Tag ermöglicht wird. Dieses Muster kann fortgesetzt werden, um zu erweitern, wie vorstehend erörtert.

[1261] Nach dem Endzeitinkrement kann ein Bloomfilter auf seine entsprechenden Bits angewendet werden und kann ein anderes als Schlafbit **21308** bezeichnetes Bit angeben, ob die Vorrichtung eine Nachricht gesendet hat, die angibt, dass sie im Schlafmodus oder heruntergefahren ist. Bei einem Beispiel kann das System, falls das Schlafbit **21308** gesetzt ist, wählen, das Verfolgen von Systemfunktionstüchtigkeitsinformationen zu vermeiden, bis die Vorrichtung wieder hochgefahren ist und das Setzen des Schlafbits **21308** deaktiviert ist.

[1262] Wie in **Fig. 213** dargestellt ist, kann die intermittierende fehlende Watchdog-Nachricht für die historische Berichterstattung unter Verwendung von Bloomfiltern in Zusammenhang mit gepaarten Shadow-Filtern **21310**, die im Allgemeinen nur aktualisiert werden, wenn ein erwarteter, jedoch nicht vorhandener Funktionstüchtigkeitsbericht erkannt wird, erkannt und aufgezeichnet werden. Die Shadow-Filter **21310** sind als mit dem Bloomfilter eines ähnlichen Zeitinkrements gepaart dargestellt. Der Wert des Shadow-Filters **21310** kann durch Anwenden der logischen UND-Funktion auf das gepaarte Bloomfilter und ein entsprechendes Rollover-Filter berechnet werden. Bei einem Beispiel gibt die logische UND-Funktion einen Wert WAHR oder 1 zurück, falls das Bloomfilter keine fehlenden Berichte zeigt und das Rollover-Filter über die innerhalb dieses Bloomfilters aufgezeichnete Zeitspanne keine fehlenden Berichte zeigt. Dagegen kann die logische UND-Funktion einen Wert FALSCH oder 0 zurückgeben, falls das Bloomfilter einen fehlenden Bericht über die aufgezeichnete Zeitspanne zeigt oder das Rollover-Filter einen fehlenden Bericht über die aufgezeichnete Zeitspanne zeigt oder beides der Fall ist. Dieser Ausgangswert kann zur weiteren Zusammenstellung der Netzfunktionstüchtigkeitsstatistik im Vorrichtung-Shadow-Filter gespeichert werden.

[1263] Bei einem Beispiel könnten Blockchains von der Vorrichtung, vom Subnetz und vom Netz beim Auflisten und Austauschen von Funktionstüchtigkeitsberichtsdaten verwendet werden. Blockchains können sich allgemein auf eine verteilte Datenbank beziehen, die eine kontinuierlich wachsende Liste als Blöcke bezeichnete Datensätze unterhält. Bei einem Beispiel enthält ein Block einen Zeitstempel und eine Verbindung zu einem vorhergehenden Block, wodurch eine Blockchain gebildet ist. Die Blockchain-Attestierung ist ein breites Konzept, das in vertrauenswürdigen Bootmechanismen verwendet werden kann, die Vertrauens-Roots für die Verkettung und Archivierung verwenden. Beispielsweise kann eine Blockchain-Attestierung nicht nur gegenüber den Blockchain-Vertrauens-Roots auf der lokalen Plattform attestieren, sondern auch die Messungen teilnehmender „Miner“-Plattformen sammeln. Die erste Transaktion, die über die Blockchain läuft, kann einer

Attestierung der Vertrauens-Roots der einzelnen Plattformen für die Verkettung sein. Durch Vergleichen der Messungen jeder der Roots der Miner kann der Verifizierer folgern, dass die Blockchain auch für die Verarbeitung nachfolgender Transaktionen vertrauenswürdig und zuverlässig ist.

[1264] Auf die Blockchain kann an einem vertrauenswürdigen Plattformmodul bzw. Trusted Platform Module (TPM) zugegriffen werden, wie vorstehend erörtert. Bei einem Beispiel kann das TPM mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** veröffentlichten Spezifikation kompatibel sein. Falls eine Vorrichtung von einem Bootcode-Segment ausgeht, kann die Blockchain zuerst für die Einrichtung einer vertrauenswürdigen Ausführungsumgebung bzw. Trusted Execute Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten verwendet werden. An Stelle des TPM **13128** kann die TEE auf anderen Technologien in der Art von EPID, SGX oder ähnlichen Technologien beruhen.

[1265] Wie hier verwendet, können Blockchains von Funktionstüchtigkeitsinformationen vertikal von der Vorrichtungsschicht zur Subnetzschicht bis zur Netzschicht in der Gesamttopologie reichen. Ähnlich können Blockchains die horizontale Erfassung mehrerer Instanzen von Vorrichtungsfunktionstüchtigkeitsinformationen auf der Grundlage des Nichtempfangs einer geplanten Funktionstüchtigkeitsnachricht widerspiegeln.

[1266] Bei einem Beispiel können die Bloomfilter in eine Blockchain-Transaktion aufgenommen werden, um einen historischen Datensatz der Netzfunktionstüchtigkeit auf der untersten Schicht der Netztopographie einzurichten. Bei einem Beispiel kann sich die unterste Schicht auf einen Endpunkt in der Art einer Vorrichtung beziehen. Eine Vorrichtung der untersten Schicht kann als Mining-Knoten bezeichnet werden, weil der Mining-Knoten die Funktionstüchtigkeitsinformationen in den Bloomfiltern sammeln kann, bevor er sie mit anderen Vorrichtungen der untersten Schicht unter Verwendung eines Shadow-Filters kombiniert.

[1267] Bei einem Beispiel könnte, falls ein Router, ein Konzentrator oder ein Gateway als ein Mining-Knoten angesehen werden sollte, der Router, der Konzentrator oder der Gateway in eine Blockchain-Repräsentation eines verbundenen Datensatzes der Netzfunktionstüchtigkeit aufgenommen werden. Die Verwendung einer Blockchain für die Speicherung von Funktionstüchtigkeitsinformationen kann es anderen Blockchain-Minern infolge der verbundenen Natur der Mining-Knoten ermöglichen, über die Funktionstüchtigkeit der Population von Minern informiert zu werden. Durch die Verwendung dieser Informationen können bössartige und nicht bössartige Ausfälle vorhergesagt werden. Beispielsweise kann ein Ausfall von wenigstens 50 % der eine Blockchain verwendenden Mining-Knoten die Integrität der Blockchain beeinträchtigen, weil kein Konsens gebildet werden kann.

[1268] **Fig. 214** ist ein schematisches Diagramm eines Netzebenen-Bloomfilters, wodurch über einen historischen intermittierenden Verlust der Watchdog-Berichterstattung **21400** berichtet wird. Gleich nummerierte Figuren entsprechen jenen, die in **Fig. 213** beschrieben wurden. Wie dargestellt, kann eine Vorrichtung oder ein Satz von Vorrichtungen **21402** gepaart werden und Daten für Shadow-Filter oder einen Satz von Shadow-Filtern **21404** bereitstellen. Bei einem Beispiel kann jedes der Shadow-Filter im Satz von Shadow-Filtern **21404** einer einzigen Vorrichtung oder einem Satz von Vorrichtungen **21402** auf der Grundlage der Rücksetzrate und der Rollover-Zeit der Vorrichtungen und gepaarten Shadow-Filter **21404** entsprechen.

[1269] Wie hier verwendet, können die Shadow-Filter aus dem Satz von Shadow-Filtern **21404** nur dadurch aufeinander bezogen sein, dass alle Vorrichtungen zu einem bestimmten Subnetz gehören, beispielsweise Subnetz-A **21406**, Subnetz-B **21408**, Subnetz-C **21410**, Subnetz-D **21412** oder Subnetz-E **21414**. Jedes dieser Subnetze kann sein eigenes Subnetz-Shadow-Filter aufweisen, wobei beispielsweise Subnetz-A **21406** das Shadow-Filter F_A aufweisen kann, Subnetz-B **21408** das Shadow-Filter F_B aufweisen kann, Subnetz-C **21410** das Shadow-Filter F_C **21416** aufweisen kann, Subnetz-D **21412** das Shadow-Filter F_D aufweisen kann und Subnetz-E **21414** das Shadow-Filter F_E aufweisen kann.

[1270] Es ist ein beispielhaftes Subnetz-Shadow-Filter dargestellt, nämlich dass Shadow-Filter F_C **21416** für Subnetz-C **21410**. Wie dargestellt, kann der Wert für die Aufnahme in das Shadow-Filter F_C **21416** durch Anwenden der logischen UND-Operation auf mehrere Vorrichtungs-Shadow-Filter erzeugt werden. Wie vorstehend erörtert, kann die logische UND-Funktion über eine Anzahl von Quellen WAHR oder 1 angeben, falls alle Quellen funktionieren, und FALSCH oder 0 angeben, falls auch nur eines der Quellenvorrichtungs-Shadow-Filter eine nicht funktionierende Vorrichtung angibt. Dementsprechend kann das Shadow-Filter F_C **21416** verwendet werden, um intermittierende Shadow-Ereignisse für alle Vorrichtungen innerhalb des Subnetzes-C **21410** zu erfassen und dadurch die Funktionstüchtigkeitsaktivität des gesamten Subnetzes auf einmal zu zeigen. Ähnlich können bei den Subnetz-Shadow-Filtern eine logische UND-Funktion auf die in jedem der Subnetz-Shadow-Filter des Netzes erzeugten und gespeicherten Werte angewendet werden. Hierdurch kann

die Funktionstüchtigkeitsberichterstattung von jedem Subnetz-Shadow-Filter das Netz-Shadow-Filter in einem Netz-Funktionstüchtigkeitsbericht **21214** informieren, um die intermittierende Funktionstüchtigkeit des Netzes widerzuspiegeln. Anhand dieser Informationen kann ein Netz-Funktionstüchtigkeitsüberwachungssystem Daten des Netz-Shadow-Filters analysieren, um einen Zeitraum zu beobachten, in dem ein Netz oder mehrere Netze größere oder kleinere Vorfälle unerwarteter Ausfälle ansprechend auf Watchdog-Berichte erfahren. Bei einem Beispiel kann eine Feststellung eine Angabe sein, dass Last-, Bandbreiten-, Stau- oder andere Verarbeitungsbedingungen physikalische Kapazitäten überschreiten, und diese Angaben könnten die Aufmerksamkeit auf weniger sichere Betriebsbedingungen für ein IoT-Netz richten.

[1271] Fig. 215 zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens **21500** zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Berichterstattung über die Funktionstüchtigkeit unter Verwendung von Shadow- und Bloomfiltern gemäß einigen Ausführungsformen. Das Verfahren **21500** aus Fig. 215 kann durch die mit Bezug auf Fig. 216 beschriebene IoT-Vorrichtung **21600** implementiert werden. Das Verfahren **21500** kann unter Verwendung des mit Bezug auf Fig. 8 beschriebenen Systems **802** ausgeführt werden. Der Prozessfluss kann in Block **21502** beginnen.

[1272] In Block **21504** kann ein Funktionstüchtigkeits-Überwachungsagent auf IoT-Vorrichtungen und Netzgeräten in einem Netz installiert werden. In Block **21506** kann die Software des Funktionstüchtigkeitsagenten nach dem Booten oder Rücksetzen einer IoT-Vorrichtung oder Teile der Netzgeräte ausgeführt werden.

[1273] In Block **21508** kann der Funktionstüchtigkeits-Überwachungsagent auf der Vorrichtungs- oder Geräteebene eine Watchdog-Ereignisnachricht mitteilen, beispielsweise jedes Mal dann, wenn ein Funktionstüchtigkeitsbericht nicht nach einem regelmäßigen Plan empfangen wird, indem das Bloomfilter entsprechend der Berichterstattungsfrequenz aktualisiert wird. In Block **21510** wird festgestellt, ob das Bloomfilter eine Rollover-Bedingung erfüllt hat. Falls dies nicht der Fall ist, kann der Prozessfluss zu Block **21508** zurückkehren, wo zusätzliche Watchdog-Berichte gesendet werden können. Falls das Bloomfilter eine Rollover-Bedingung erfüllt hat, kann es in Block **21512** durch das Rollover aktualisiert werden. Die Aktualisierung des Bloomfilters gibt an, dass die Vorrichtung während des gesamten Berichterstattungszeitraums des Bloomfilters funktionstüchtig war.

[1274] In Block **21514** wird festgestellt, ob eine Rücksetzschwelle erreicht wurde. Falls dies nicht der Fall ist, kann der Prozessfluss zu Block **21508** zurückkehren, wo zusätzliche Watchdog-Berichte gesendet werden können. Falls in Block **21514** eine Rücksetzschwelle erreicht wurde, kann in Block **21516** die logische UND-Operation für ein Bloomfilter und ein Rollover-Filter berechnet werden, wobei das Ergebnis in einem Shadow-Filter gespeichert wird. In Block **21518** wird festgestellt, ob das Schlafbit gesetzt ist. Falls dies der Fall ist, wird in Block **21520** die Verarbeitung des Bloom- und des Shadow-Filters ausgesetzt, bis eine Schlafzeit verstrichen ist und das Setzen des Schlafbits aufgehoben ist. Der Prozessfluss kehrt dann zu Block **21518** zurück.

[1275] Falls in Block **21518** festgestellt wird, dass das Schlafbit nicht gesetzt ist, kann der Prozess in Block **21522** warten, bis das System eine Netzfunktionstüchtigkeitsstatistik-Sammelanforderung empfängt und verarbeitet. Sobald die Sammlung verarbeitet ist, kann in Block **21524** eine Subnetz-Überwachungseinrichtung das Shadow-Filter einer Vorrichtung im Subnetz erhalten. Die Ergebnisse dieses Vorrichtungs-Shadow-Filters können auf das Subnetz-Shadow-Filter angewendet werden. In Block **21526** wird festgestellt, ob ein anderes Vorrichtungs-Shadow-Filter im Subnetz zu sammeln ist. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **21526** zurück, wo das restliche Vorrichtungs-Shadow-Filter gesammelt und auf das Subnetz-Shadow-Filter angewendet wird.

[1276] Falls kein anderes Vorrichtungs-Shadow-Filter im Subnetz zu sammeln ist, kann in Block **21528** eine Netzüberwachungseinrichtung das Subnetz-Shadow-Filter von einem Subnetz im Netz erhalten. Die Ergebnisse dieses Subnetz-Shadow-Filters können auf das Netz-Shadow-Filter angewendet werden. In Block **21530** wird festgestellt, ob ein anderes Subnetz-Shadow-Filter im Netz zu sammeln ist. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **21528** zurück, wo das restliche Subnetz-Shadow-Filter gesammelt und auf das Netz-Shadow-Filter angewendet wird. Falls kein weiteres Subnetz-Shadow-Filter im Netz zu sammeln ist, kann in Block **21532** das Netzfunktionstüchtigkeits-Shadow-Filter den Subskribenten mitgeteilt werden. Der Prozess endet dann in Block **21534**.

[1277] Fig. 216 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **21600** zum Berichten über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in Fig. 8 beschrieben sind.

[1278] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Subnetz-Funktionstüchtigkeitsdatenanforderer **21602** aufweisen. Bei einem Beispiel kann der Subnetz-Funktionstüchtigkeitsdatenanforderer **21602** Subnetz-Funktionstüchtigkeitsdaten ansprechend auf eine empfangene Netz-Funktionstüchtigkeitsanforderung für ein Netz von einem Subnetz anfordern. Bei einem Beispiel kann das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von einer Vorrichtung ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern. Das Subnetz kann auch angewiesen werden, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren eines Subnetz-Shadow-Filters unter Verwendung anhand empfangener Vorrichtungsfunktionstüchtigkeitsdaten erzeugter Blockchain-Daten zu erzeugen.

[1279] Das Subnetz kann auch Vorrichtungsfunktionstüchtigkeitsdaten von einer Anzahl von Vorrichtungen ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern sollen. Ferner kann das Subnetz angewiesen werden, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren des Subnetz-Shadow-Filters unter Verwendung einer Anzahl der empfangenen Vorrichtungsfunktionstüchtigkeitsdaten durch die Verwendung eines logischen Operators zum Vergleich der Anzahl der empfangenen Funktionstüchtigkeitsdaten zu erzeugen. Bei einem Beispiel kann die Vorrichtung Vorrichtungsfunktionstüchtigkeitsdaten auf der Grundlage eines Vorrichtungs-Shadow-Filters zurückgeben, wobei der Vorrichtungs-Shadow-Filter auf der Grundlage eines Vorrichtungs-Bloomfilters erzeugt werden kann, der das Nichtvorhandensein geplanter Funktionstüchtigkeitsnachrichten an der Vorrichtung verfolgt. Andere Techniken zur Implementation des Vorrichtungs-Shadow-Filters können DHT, Blockchains und dergleichen aufweisen. Das Vorrichtungs-Shadow-Filter kann eine Anzahl von Shadow-Filtern umfassen, die jeweils einem Zeitintervall einer Anzahl von Bloomfiltern entsprechen.

[1280] Der Massenspeicher **808** kann einen Netz-Shadow-Filter-Modifizierer **21604** aufweisen, um ein Netz-Shadow-Filter auf der Grundlage anhand Blockchain-Daten erzeugter empfangener Subnetz-Funktionstüchtigkeitsdaten zu modifizieren. Der Massenspeicher **808** kann einen Berichtsbereitsteller **21606** zum Bereitstellen eines Berichts über die Netzfunktionstüchtigkeit auf der Grundlage des Netz-Shadow-Filters aufweisen. Bei einem Beispiel wirkt das Netz-Shadow-Filter durch Anwenden eines logischen Operators.

[1281] Bei einem Beispiel kann ein Subnetz-Funktionstüchtigkeitsdatenanforderer Subnetz-Funktionstüchtigkeitsdaten von einer Anzahl im Netz logisch organisierter Subnetze anfordern und kann ein Netz-Shadow-Filter-Modifizierer ein Netz-Shadow-Filter auf der Grundlage einer Anzahl empfangener Subnetz-Funktionstüchtigkeitsdaten modifizieren. Bei diesem Beispiel und anderen vorstehend erörterten Beispielen können die Blockchain-Daten auf Bloomfiltern beruhen, die in eine Blockchain-Transaktion aufgenommen werden können, um an einem Endpunkt einer Netztopographie einen historischen Datensatz der Netzfunktionstüchtigkeit zu erzeugen.

[1282] **Fig. 217** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **21700**, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1283] Das nichtflüchtige maschinenlesbare Medium **21700** kann einen Code **21702** aufweisen, um den Prozessor **902** anzuweisen, Subnetz-Funktionstüchtigkeitsdaten von einem Subnetz ansprechend auf eine empfangene Netzfunktionstüchtigkeitsanforderung für ein Netz anzufordern. Bei einem Beispiel kann das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von einer Vorrichtung ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern. Das Subnetz kann auch angewiesen werden, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren eines Subnetz-Shadow-Filters unter Verwendung anhand empfangener Vorrichtungsfunktionstüchtigkeitsdaten erzeugter Blockchain-Daten zu erzeugen. Das Subnetz kann auch Vorrichtungsfunktionstüchtigkeitsdaten von einer Anzahl von Vorrichtungen ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern sollen. Ferner kann das Subnetz angewiesen werden, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren des Subnetz-Shadow-Filters unter Verwendung einer Anzahl der empfangenen Vorrichtungsfunktionstüchtigkeitsdaten durch die Verwendung eines logischen Operators zum Vergleich der Anzahl der empfangenen Funktionstüchtigkeitsdaten zu erzeugen. Bei einem Beispiel kann die Vorrichtung Vorrichtungsfunktionstüchtigkeitsdaten auf der Grundlage eines Vorrichtungs-Shadow-Filters zurückgeben, wobei der Vorrichtungs-Shadow-Filter auf der Grundlage eines Vorrichtungs-Bloomfilters erzeugt werden kann, der das Nichtvorhandensein geplanter Funktionstüchtigkeitsnachrichten

ten an der Vorrichtung verfolgt. Das Vorrichtungs-Shadow-Filter kann eine Anzahl von Shadow-Filtern aufweisen, die jeweils einem Zeitintervall einer Anzahl von Bloomfiltern entsprechen.

[1284] Bei einem Beispiel kann die selbstbeschreibende Vorrichtung eine Batteriestromquelle und/oder eine Internetverbindung auf der Grundlage einer drahtlosen Technologie aufweisen, oder ein Prozessor mit einem Modus mit geringer Leistungsaufnahme wird als beschränkte selbstbeschreibende Vorrichtung angesehen.

[1285] Das nichtflüchtige maschinenlesbare Medium **21100** kann einen Code **21704** zum Modifizieren eines Netz-Shadow-Filters auf der Grundlage anhand Blockchain-Daten erzeugter empfangener Subnetz-Funktions-tüchtigkeitsdaten aufweisen. Das nichtflüchtige maschinenlesbare Medium **21100** kann einen Code **21704** aufweisen, um einen Bericht über die Netzfunktionstüchtigkeit auf der Grundlage des Netz-Shadow-Filters bereitzustellen. Bei einem Beispiel kann ein Subnetz-Funktionstüchtigkeitsdatenanbieter Subnetz-Funktionstüchtigkeitsdaten von einer Anzahl im Netz logisch organisierter Subnetze anfordern und kann ein Netz-Shadow-Filter-Modifizierer ein Netz-Shadow-Filter auf der Grundlage einer Anzahl empfangener Subnetz-Funktions-tüchtigkeitsdaten modifizieren. Bei diesem Beispiel und anderen vorstehend erörterten Beispielen können die Blockchain-Daten auf Bloomfiltern beruhen, die in eine Blockchain-Transaktion aufgenommen werden können, um an einem Endpunkt einer Netztopographie einen historischen Datensatz der Netzfunktionstüchtigkeit zu erzeugen.

[1286] IoT-Vorrichtungen können ein Weitbereichsnetz mit geringer Leistungsaufnahme bzw. Low Power Wide Area Network (LPWAN) zur Ausführung von Steuerkanalfunktionen für ein Netz verschiedener verbundener Vorrichtungen und Infrastruktur verwenden. Andere Netze können abhängig von Strom- und Kommunikationsfähigkeiten der IoT-Vorrichtung in der Art mobiler Netze, festverdrahteter Netze und dergleichen verwendet werden, um die beschriebenen Techniken zu implementieren. Ein Steuerkanal-LPWAN kann bei der auf dem geografischen Ort basierenden Aktivierung und Deaktivierung von Vorrichtungen verwendet werden. Ein Steuerkanal-LPWAN kann auch für netzweite Dienstwarnungen und Ausstrahlungen verwendet werden. Sicherheitsaktionen in der Art des Benachrichtigens von Vorrichtungen, dass Sicherheitsschlüssel neu erzeugt werden müssen, können auch unter Verwendung von LPWAN ausgeführt werden. Ferner kann LPWAN für Auditierungszwecke in der Art des Ausführens eines Infrastrukturinventars oder des Auslösens von Ortsnachrichten, die von den Vorrichtungen abzusenden sind, verwendet werden.

[1287] **Fig. 218** ist ein schematisches Diagramm eines drahtlosen Weitbereichsnetzes bzw. Wireless Wide Area Network (WWAN) **21800**, wobei ein Steuerkanal **21802** über jede Verbindung verwendet werden kann, gemäß einigen Ausführungsformen. Bei einem Beispiel kann der Steuerkanal durch Weitbereichstechnologien mit geringer Leistungsaufnahme bzw. Low-Power-Wide-Area(LPWA)-Technologien verwirklicht werden.

[1288] Das WWAN **21800** kann eine Anzahl von Technologien und Netztypen, die miteinander arbeiten, aufweisen. Das WWAN **21800** kann einen Faser-Backbone **21804** zwischen Kern-Routern aufweisen. Das WWAN **21800** kann auch eine Mobilfunknetzverbindung **21806** aufweisen. Bei einem Beispiel kann die Mobilfunknetzverbindung **21806** eine Verbindung zwischen einem Kern-Router und einem Mobilfunksenderknoten sein. Das WWAN **21800** kann auch eine Netzverbindung zu einem drahtlosen lokalen Netz bzw. Wireless Local Area Network (WLAN) **21808** aufweisen. Das WWAN **21800** kann auch eine Netzverbindung zu einem Mesh-Netz **21810** aufweisen. Bei einem Beispiel kann das Mesh-Netz irgendeine Art eines drahtlosen persönlichen Netzes bzw. Wireless Personal Area Network (WPAN) in der Art eines 802.15.4x-Mesh aufweisen. Das WWAN **21800** kann auch eine Netzverbindung zwischen Endpunkten unter Verwendung von Kurzstrecken-Drahtloskommunikationsnetzen **21812** aufweisen. Bei einem Beispiel können die Kurzstrecken-Drahtloskommunikationsnetze unter Verwendung von Protokollen einschließlich Bluetooth, Bluetooth Low Energy (BLE), Zigbee und anderer ähnlicher Verbindungsprotokolle, gebildet werden. Beim WWAN **21800** können auch LPWA-Verbindungen **21814** zwischen Vorrichtungen verwendet werden.

[1289] Wie vorstehend erwähnt, kann das WWAN **21800** Kommunikationen aufweisen, die durch einen Steuerkanal **21802** zwischen Vorrichtungen behandelt werden. Wie vorstehend erörtert, kann der Steuerkanal eine beliebige Anzahl von Protokollen, einschließlich unter anderem LWPA-Protokollen in der Art von LoRa-WAN, verwenden. Die Verwendung eines Steuerkanals über ein Netz unter Verwendung von LWPA, wie hier beschrieben, kann Geolokationsmerkmale, andere Sensormerkmale und dergleichen ermöglichen. Die hier dargestellten Geolokationsmerkmale können durch ein Verfahren zum Codieren globaler Geolokationsdatenpunkte sowie ein Verfahren zum Erzeugen von Zonenkennungen auf der Grundlage eines Gitteransatzes arbeiten. Bei einem Beispiel einer Geolokationstechnik werden Zonenkennungen für die Verwendung mit Transaktionen und Nachrichten zwischen Netzinfrastrukturkomponenten erzeugt, wobei die Zonenkennungen auch in von Menschen lesbare Breiten-, Längen- und/oder Höhendaten zurück transformiert werden können. Um

diese Informationen zu visualisieren und aufzutragen, kann ein Zonenkennungsgitter verwendet werden, wie nachstehend in **Fig. 219** gezeigt ist.

[1290] **Fig. 219** ist ein schematisches Diagramm einer Karte **21900** eines in Zonen unterteilten physikalischen Bereichs gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 218** beschrieben wurden.

[1291] Die dargestellte Karte **21900** weist Breitenlinien **21902** und Längelinien **21904** auf. Bei einem Beispiel können die von diesen Linien gebildeten Gitterquadrate zur Angabe verschiedener Zonen verwendet werden, wobei jeder Zone eine Zonenkennung gegeben ist. Die Karte **21900** zeigt die Zonencodierüberlagerung von Gitterquadraten, die eine Unterteilung des physikalischen Bereichs in Zonen, die jeweils ihre eigene Zonenkennung aufweisen, angeben. Bei einem Beispiel kann die Zonenkennungs-Gittergröße durch eine Parametermodifikation in der Art einer Modifikation der Ortsmetriken oder Zonenmarkiereinheiten konfiguriert werden. Wie in diesem Zusammenhang verwendet, kann die Parametermodifikation nicht zu einer Änderung des algorithmischen Flusses des verwendeten Geolokationsverfahrens führen.

[1292] Durch die Verwendung eines Zonenkennungsansatzes für die Geolokation können Zonengrößen, die von tausenden Quadratkilometern bis zu Quadratmetern, unter anderen Größen, reichen, codiert und decodiert werden. Durch die Verwendung eines Zonenkennungsansatzes können auch Zonenkennungen für physikalische Bereiche beliebiger Größen berechnet, codiert und decodiert werden, wobei die Zonen für eine globale Verwendung geeignet sind.

[1293] Bei der Aufnahme dieser Zonen in einen Steuerkanal **21802** können die Zonenkennungen (ID) **21906** verwendet werden. Bei einigen Ausführungsformen kann ein Geolokationsverfahren zur Ressourcenentdeckung eine Nachricht **21908** ausstrahlen, welche die Zone oder die Zonen, die zur Steuerkanalnachricht **21910** gehört, und die Steuernachricht **21912** selbst aufweist. Bei einem Beispiel kann die Nachricht **21908** Nachrichten-Headerdaten aufweisen, die vor andere Daten, welche die Nachricht enthält, gehängt werden, wodurch Informationen über die Nachricht **21908** widergespiegelt werden, beispielsweise in Bezug auf die Steuerkanal-Nachrichtenidentifikation, die Anzahl der Zonen, die Lebenszeit (TTL), falls anwendbar, und die Länge des Datenfelds. Bei einem Beispiel kann die Nachricht **21908** Sicherheitsmerkmale in der Art von Autorisierungs-, Authentifizierungs- und Nachrichtenintegritätsdaten aufweisen, die an den Frame angehängt werden können. Für die Verwendung in einem Steuerkanal kann die Nachricht **21908** kurz sein und demgemäß weniger Ressourcen verwenden, was zu einer kürzeren Übertragungszeit und einem geringeren Stromverbrauch führt.

[1294] **Fig. 220** zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens **22000** zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Mitteilung des geographischen Orts unter Verwendung der Ankunftszeitdifferenz gemäß einigen Ausführungsformen. Das Verfahren **22000** aus **Fig. 220** kann durch die mit Bezug auf die **Fig. 221** und **Fig. 223** beschriebenen Systeme **22100** und **22300** implementiert werden. Das Verfahren **21400** kann unter Verwendung des mit Bezug auf **Fig. 8** beschriebenen Systems **802** ausgeführt werden. Der Prozessfluss kann in Block **22002** beginnen.

[1295] In Block **22002** kann ein Nutzinhalte mit einem Zeitstempel empfangen werden. Der empfangene Nutzinhalte kann ein einziger Nutzinhalte oder mehrere Nutzinhalte von Daten sein. In Block **22004** können die Nutzinhalte nach der Vorrichtungskennung sortiert werden. Auf diese Weise können der eine oder die mehreren Zeitstempel mehrerer Vorrichtungen verglichen werden. In Block **22006** kann festgestellt werden, ob es zwei oder mehr mit einem Zeitstempel versehene empfangene Nutzinhalte für eine Vorrichtung gibt. Falls dies nicht der Fall ist, kann der Prozessfluss zu Block **22002** zurückkehren, um auf zusätzliche mit Zeitstempeln versehene Nutzinhalte zu warten. Falls es zwei oder mehr für eine einzelne Vorrichtung empfangene mit Zeitstempeln versehene Nutzinhalte gibt, kann der Prozessfluss in Block **22008** fortgesetzt werden.

[1296] In Block **22008** kann die Zeitdifferenz zwischen den beiden Zeitstempeln für dieselbe Vorrichtung berechnet werden. In Block **22010** kann die Zeitdifferenz eine Eingabe in eine Funktion zur Ortsberechnung sein. Bei einem Beispiel kann die Funktion verwendet werden, um die geschätzten Abstände zu erhalten, und nachfolgend in Bezug auf x- und y-Koordinaten anhand der Koordinaten der Vorrichtung, die anfänglich den Nutzinhalte bereitgestellt hat, verwendet werden. Ein Beispiel einer Funktion, die in Block **22010** verwendet werden kann, kann eine nachstehend in weiteren Einzelheiten beschriebene hyperbolische Funktion sein. Die Ausgabe der Funktion können Koordinaten sein, die zur Lokalisierung einer Vorrichtung auf einer Karte im physikalischen Raum verwendet werden können.

[1297] In Block **22012** kann die Ausgabe der Funktionsberechnung aus lokalen Koordinaten in globale Koordinaten umgewandelt werden. Eine Umwandlung kann nicht erforderlich sein, falls die globalen Koordinaten für die Verwendung ausreichen. In Block **22014** kann der Knotenort mit den erzeugten Koordinaten geschätzt werden. Dieser Prozess kann so oft wie gewünscht wiederholt werden, um einen Ort einer Vorrichtung zu identifizieren, oder so oft wie die Vorrichtung durch Nachrichten in einem Netz fordert, dass ein Ort für sie identifiziert wird.

[1298] Durch die Verwendung einer hyperbolischen Berechnung beim vorstehend erörterten Prozess kann die Genauigkeit erhöht werden, wenn die Anzahl der empfangenden Gateways zunimmt. Beispielsweise kann ein standardmäßiger Gemeinsames-Netz-Geolokationsansatz mit dem gegenwärtig offenbarten zusammenarbeitenden Netz, das zusätzliche homogene oder heterogene Netzknoten aufweisen kann, verbessert werden.

[1299] **Fig. 221** ist ein schematisches Diagramm eines Netzes **22000** zur Bestimmung einer Zeitdifferenz auf der Grundlage von Ankunftszeitinformationen in einem heterogenen Netz teilweise unter Verwendung einer Zonenkennung gemäß einigen Ausführungsformen. Der geographische Ort kann allgemein ein wichtiger differenzierender Faktor für industrielle, Intelligente-Stadt-, Einzelhandels- und sicherheitsbasierte IoT sowie künftige Internetanwendungen und -dienste sein. Die Berechnung und Speicherung eines geographischen Orts kann in einem Netzserver **22102** geschehen. Der Prozess des Erhaltens von geographischen Ortsinformationen als Produkt der Kommunikationsstrecken- und Netzwegeigenschaften kann unter Verwendung hier beschriebener Techniken erweitert werden. Die Ankunftszeitdifferenz bzw. Time Difference of Arrival (TDOA) von Netzpaketen, die einen gemeinsamen zeitsynchronisierten Referenzpunkt verwenden, kann dabei helfen, einen geographischen Ort zu identifizieren. Ein Beispiel eines zeitsynchronisierten Referenzpunkts ist eine globale Positionsbestimmungsvorrichtung (GPS) oder eine andere Satellitenzeitreferenz.

[1300] Ein sendender Knoten **22104** kann Daten oder einen Nutzinhalt zu einem empfangenden Gateway **22106** senden. Auf der Grundlage der Zeitstempel, die beim Absenden an den Nutzinhalt angehängt werden, und des bei der Ankunft empfangenen Zeitstempels kann die Flugzeit (TOF) **22108** berechnet werden. Bei einem Beispiel kann ein Empfänger-Gateway **22104** einen Ortssensor in der Art von GPS **22110** aufweisen. Der GPS-Sensor ermöglicht es nicht nur, eine TOF zu berechnen, sondern kann einen Referenzabstand zur Verwendung durch den sendenden Knoten **22104** bereitstellen. Wenn die Anzahl der eine TOF und den Anfangsabstand messenden Gateways zunimmt, kann eine zusätzliche Datensammlung um die TOF und die Referenzorte vorgenommen werden, und der Ort des sendenden Knotens **22104** kann dementsprechend genauer gemacht werden. Zur Erhöhung der Anzahl der in einem Netz vorhandenen Ortmessknoten können Gateways **22106** mit zusammenarbeitenden Sensorknoten **22112**, die einen Sensor, beispielsweise einen GPS-Sensor **22110**, enthalten können, verbunden werden.

[1301] Ein zusammenarbeitender Sensorknoten **22112** kann durch Drahtlosprotokolle und/oder festverdrahtete Verfahren mit einem Gateway **22106** verbunden werden. Wenn der sendende Knoten **22104** seinen Ort unter Verwendung eines Drahtlosnetzes beurteilt, kann er einen zeitgestempelten Nutzinhalt senden, der an einem Gateway **22110** oder einem zusammenarbeitenden Sensorknoten **22112** zu empfangen ist, welcher eine zeitliche Verzögerung bzw. Time of Delay (TOD) **22114** in Bezug auf die Länge des vom Nutzinhalt zurückgelegten Wegs aufweist.

[1302] Der empfangende Gateway kann unter Verwendung eines gemeinsamen Zeitgebers zeitsynchronisiert sein. Die Verwendung eines zusammenarbeitenden Sensorknotens **22112** ist eine Erweiterung eines homogenen Netzes, wobei eine TDOA-Berechnung an einem zentralisierten Verarbeitungspunkt verwendet wird, weil diese zusätzlichen Knoten Teil eines anderen Netzes sein können und ein anderes Protokoll verwenden können als das Netz vom Server **22102** zu den Gateways **22106**. Die Sensorknoten **22112** können das interessierende Paket erfassen, beobachten und empfangen und die mit einem Zeitstempel versehenen Erfassungsinformationen einem Gateway-Knoten **22110** innerhalb der Reichweite mitteilen. Die Verwendung zusätzlicher Knoten, selbst wenn sie außerhalb des Netzes liegen, erhöht die Anzahl der Paketbeobachter. Bei einem Ortsapproximationssystem, das eine hyperbolische Projektion zur Beurteilung des Orts verwendet, kann die erhöhte Anzahl von Paketbeobachtern auch die Anzahl der hyperbolischen Loci, die für die Gateways **22100** lokal vorhanden sind, welche für den Ortsberechnungsprozess verwendet werden, erhöhen.

[1303] Bei einem Beispiel kann das in **Fig. 220** dargestellte Verfahren am in **Fig. 221** dargestellten Netzschema implementiert werden. Beispielsweise kann in Bezug auf die dargestellten Knoten die Ankunftszeit (TOA) vom sendenden Knoten **22104** zum Gateway unter Verwendung der folgenden Gleichung berechnet werden:

$$TOA_i = T_{tx} + TOF_i$$

$$0 \leq i \leq N$$

In den vorstehenden Gleichungen bezieht sich T auf die Zeit des Sendens von tx oder vom sendenden Knoten **22104**. Der Term N bezeichnet die Anzahl der vom sendenden Knoten **22104** (i) indexierten empfangenden Gateways, wobei l_i der Sichtlinien-Wegabstand zwischen dem sendenden Knoten **22104** und dem empfangenden Gateway **22106** ist. Bei einem Beispiel kann 22104 so sein, wie in **Fig. 221** dargestellt ist. Die TOA in den vorstehenden und nachstehenden Gleichungen spiegelt die Ankunftszeit wider, und TOF ist die Flugzeit, wobei T_{tx} die Sendezeit ist. Die Änderung der Ankunftszeit kann unter Verwendung der folgenden Gleichungen berechnet werden:

$$\Delta TOA_{ij} = TOA_i - TOA_j = TOF_i - TOF_j$$

$$\Delta TOA_{ij} = \frac{l_i}{v} - \frac{l_j}{v} = \frac{1}{v} (l_i - l_j)$$

In der vorstehenden Gleichung ist v die Geschwindigkeit von Licht oder Schall in Luft oder Wasser, wobei die Wahl des Mediums davon abhängen kann, wo das Senden geschieht. Bei einem Beispiel kann das Medium Luft mit klaren Sichtverbindungen zwischen Knoten sein.

[1304] Wenn Signale und Daten gesammelt werden, können die Daten unter Verwendung eines hyperbolischen Funktionsansatzes verarbeitet werden, für den die geschätzten Abstände l und nachfolgend die Abstände in Bezug auf die x und y-Koordinaten von den empfangenden Gateways verwendet werden können. Bei einem Beispiel können diese Werte folgendermaßen verwendet werden:

$$l_i - l_j = \delta_{ij} = v \Delta TOA_{ij},$$

wobei jedes δ_{ij} Positionen (x, y) entlang einer Hyperbel entspricht, wie in der nachstehenden Gleichung dargestellt ist:

$$l_i - l_j = \delta_{ij} \Rightarrow l_i^2 = l_j^2 + \delta_{ij}^2 + 2l_j \delta_{ij}$$

$$l_i^2 = (x - x_i)^2 - (y - y_i)^2$$

$$l = \sqrt{(x - x_i)^2 - (y - y_i)^2}$$

[1305] Die hyperbolische Funktion kann unter Verwendung der folgenden Gleichungen von lokalen Koordinaten (x, y) in globale Koordinaten (X, Y) übersetzt werden:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} + \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

$$a = \arctan\left(\frac{y_i - y_j}{x_i - x_j}\right)$$

[1306] Die Netzdurchlaufzeit $T_{\text{Durchlauf}}$ **22116** bezeichnet die Zeit für das Übertragen von Informationen von einem zusammenarbeitenden Sensorknoten **22112** zum empfangenden Gateway **22110**. In den nachstehenden Gleichungen ist die Zeit, zu der der zusammenarbeitende Sensorknoten das Paket erfasst, mit TOD_i für den ith zusammenarbeitenden Knoten bezeichnet. Mit diesen Informationen kann TOD_i folgendermaßen berechnet werden:

$$TOD_i = T_{tx} + TOF_i$$

$$\delta TOA_i = TOD_i - TOA_i - T_{Durchlauf}$$

[1307] Bei einem Netz mit inhomogenen Protokollen, Vorrichtungen und geographischen Orten kann es eine Variabilität in den bei diesen Berechnungen verwendeten Durchlaufzeiten geben. Diese Differenzen können dadurch kalibriert werden, dass der Server **22102** periodische zeitgestempelte Tonpakete zu den Gateways **22110** sendet, wobei die gemessene Durchlaufzeit von einer Berechnung von δTOA_i die zu subtrahieren ist. Diese Subtraktion kann dabei helfen, die erhöhte Verzögerung infolge der Zeit, die ein Paket benötigt, um sich vom zusammenarbeitenden Knoten zum empfangenden Gateway **22110** zu bewegen, herauszuheben.

[1308] **Fig. 222** ist ein schematisches Diagramm einer beispielhaften Steuerkanal-Frame-Struktur **22200**, die in einen beispielhaften Frame eines Weitbereichsnetzes mit geringer Leistungsaufnahme (LPWAN) gepackt ist, gemäß einigen Ausführungsformen. Beispielsweise kann das LPWAN-Netz ein Frame und eine Plattform eines Weitbereichsnetzes mit großer Reichweite (LoRaWAN) sein, wenngleich andere Plattformen erwogen werden und dieser Frame als ein Beispiel der Verwendung der Steuerkanal-Frame-Struktur **22200** dargestellt ist.

[1309] Die beispielhafte Steuerkanal-Frame-Struktur **22200** kann eine Nachrichten-Header **22202** mit einer Länge von X Bytes, ein Zonenkennungsfeld **22204**, das 1 bis M Bytes aufweisen kann, ein Datenfeld **22206**, das 1 bis N Bytes aufweisen kann, und ein Sicherheits-/Nachrichtenintegritätsprüfungs(MIC)-Feld **22208** mit einer Länge von Y Bytes aufweisen. Beispielsweise kann das Zonenkennungsfeld **22204** zwei bis vier Bytes aufweisen, kann das Datenfeld acht bis 256 Bytes aufweisen und kann das Integritätsprüfungsfeld zwei bis vier Bytes aufweisen. Abhängig vom Protokoll und von den Daten kann eine beliebige Anzahl von Längen verwendet werden.

[1310] Die durch diesen Frame übertragenen Steuerkanalbefehle sind Beispiele von Operationen für einen weiten Bereich einer zonenspezifischen Steuerung fern installierter Infrastruktur in Tabelle 4. Es sei bemerkt, dass der in Tabelle 4 dargestellte Befehlssatz ein Beispiel ist, weil der Befehlssatz erweiterbar ist und später modifiziert werden kann, um einer spezifischen Anwendung Rechnung zu tragen.

Tabelle 4 - Steuerkanalbefehle

Befehl	Beschreibung
0XA	Aktivieren
0XBx	Schlafen während x Stunden/Minuten, wobei x von 1 bis F reicht
0XD	Erzeugen eines neuen Sicherheitsschlüssels
0XFA	Absenden
0XCA	Rebooten
0x2	Absenden der Ortsnachricht
0x3	Senden der Beobachtungsnachricht
0x4	Wegaktualisierung (Wegpunkte, denen mobile Stellen folgen sollten)
0x5	zu vermeidende Sektoren (Wegpunkte, die mobile Knoten vermeiden sollten)

[1311] Die Befehle können durch eine LPWA-Technologie unter Verwendung von LoRaWAN durch den Befehlskanal-Frame **22200** geleitet werden. Bei einem Beispiel der Frame-Übertragung weist die LoRaWAN-Äußerer-Frame-Einkapselung einen Frame-Header (FHDR) **22210**, ein Frame-Port-Feld bzw. Frame Port Field (FPort) **22212** und ein Frame-Nutzinhaltsfeld (FRMNutzhalt) **22212** auf. Die Steuerkanalnachricht **22200** kann in das FRMNutzhalt-Feld **22214** eingekapselt werden. Dieser Verpackungsansatz bedeutet, dass die LoRaWAN-Kompatibilität nicht notwendigerweise beeinflusst werden kann, weil die Steuerkanaldaten als anwendungsspezifische Informationen behandelt werden können. Die Steuerkanaldaten können auch für Technologien einschließlich DOCSIS, WEIGHTLESS, ZWave, Ethernet, WLAN, Bluetooth, AX.25, Zigbee, IEEE **802.15.4**, IEEE **802.16** und TIA/EIA-**485**, jedoch ohne Einschränkung auf diese, in die Anwendungsnutzhalt-Frame-Felder gepackt werden.

[1312] Bei einem Beispiel kann ein Außerband-Steuerkanalsystem zur Steuerung fester, tragbarer, mobiler, luftgestützter, Raum-, Im-Körper-, unterirdischer und seegestützter/Unterwasser-Vorrichtungen verwen-

det werden. Beispiele umfassen Straßenausrüstungen, Werbeplakatwände, Fahrzeuge in der Art von Personenkraftwagen, Lastwagen, unbemannte Luftfahrzeuge, autonome Unterwasserfahrzeuge und dergleichen. Geographische Berechnungen können bei der Erzwingung von Flugverbotszonen, Wegpunktsaktualisierungen und anderen Navigationswerkzeugen besonders nützlich sein. Andere Außerband-Steuerkanalsysteme, die zur Steuerung von Vorrichtungen verwendet werden können, umfassen Untersee-Pipelineinspektionsfahrzeuge, Smart-Pig-Überwachungssysteme, die zur Ölleitungsinspektion und zur Wachsentfernung verwendet werden, Satellitensysteme in der Art von Cubesats, Umweltsteuerungs- und Sicherheitssysteme, medizinische diagnostische Sonden, Steuersysteme für industrielle Fertigungsanlagen, einschließlich Systemen, die eine Fernüberwachung und -betätigung erfordern, und dergleichen.

[1313] Fig. 223 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **22300** zur Entdeckung von Ressourcen und für die Sektoridentifikation geografischer Orte vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1314] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1315] Der Massenspeicher **808** kann einen Steuerkanal-Dateneinfüger **22402** zum Einfügen von Steuerkanaldaten in einen Weitbereichsnetz-Frame aufweisen. Der Weitbereichsnetz-Frame kann ein Anwendungsnutzzinhalts-Frame-Feld zur Einkapselung der Steuerkanaldaten vor und während der Übertragung aufweisen. Die Steuerkanaldaten können Befehle an die erste Vorrichtung, einen Nutzzinhalt von einem sendenden Knoten zu sammeln, einen Zeitstempel aus dem Nutzzinhalt zu extrahieren und Nutzzinhalte, die einen Zeitstempel und eine Zonenkennung aufweisen, zurückzugeben, aufweisen. Die Steuerkanaldaten können einen Nachrichten-Header, ein Zonenkennungsfeld, ein Steuernachrichtsdatenfeld und ein Sicherheitsfeld umfassen. Bei einem Beispiel können die Steuerkanaldaten Befehle an die erste Vorrichtung, ihre Zonenkennung und eine Anzahl von Zeitstempeln von einem sendenden Knoten zurückzugeben, umfassen. Die Steuerkanaldaten können auch Befehle an die zweite Vorrichtung, an die erste Vorrichtung ihre Zonenkennung und einen Zeitstempel von einem sendenden Knoten zurückzugeben, umfassen, wobei die zweite Vorrichtung gemäß einigen Ausführungsformen keinen Kommunikationsweg zu einer Servervorrichtung hat, außer indem durch die erste Vorrichtung geleitet wird.

[1316] Der Massenspeicher **808** kann einen Frame-Sender **22404** zum Senden des Weitbereichsnetz-Frames zu einer ersten Vorrichtung mit einem ersten Übertragungsprotokoll, beispielsweise unter Verwendung des Mesh-Sendeempfängers **810**, und zu einer zweiten Vorrichtung mit einem zweiten Protokoll, beispielsweise unter Verwendung des Uplink-Sendeempfängers **814**, aufweisen. Die Steuerkanaldaten können Zugriff sowohl auf die erste Vorrichtung und das erste Vorrichtungsprotokoll als auch auf die zweite Vorrichtung und das zweite Vorrichtungsprotokoll erhalten. Der Frame-Sender **22404** kann den Weitbereichsnetz-Frame durch Befehle an die erste Vorrichtung, als ein Zwischensender zu wirken und zur zweiten Vorrichtung zurückzusenden, zur zweiten Vorrichtung senden.

[1317] Der Massenspeicher **808** kann einen Datenmodifizierer **22406** zum Modifizieren von Daten eines sendenden Knotens ansprechend darauf, dass die erste Vorrichtung eine erkannte Übertragung von einem sendenden Knoten zurückgibt, aufweisen. Die Modifikation der Daten des sendenden Knotens kann auf dem Extrahieren einer Zonenkennung und eines Zeitstempels, wodurch eine erkannte Übertragung von einem sendenden Knoten zurückgegeben wird, beruhen.

[1318] Bei einem Beispiel kann die Vorrichtung auch einen Berechner zum Berechnen des Orts des sendenden Knotens unter Verwendung einer hyperbolischen Funktion und einer Anzahl von Zeitstempeln von Paketen, die von der ersten Vorrichtung und der zweiten Vorrichtung empfangen werden, sowie der Zonenkennung der ersten Vorrichtung und der zweiten Vorrichtung aufweisen. Die Vorrichtung kann einen Berechner zum Berechnen des Orts der sendenden Vorrichtung auf der Grundlage der Sendedurchlaufzeit zwischen einer das Signal erfassenden Vorrichtung und einer das Signal empfangenden Vorrichtung aufweisen.

[1319] Fig. 224 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **22400**, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß eini-

gen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1320] Das nichtflüchtige maschinenlesbare Medium **22400** kann einen Code **22402** aufweisen, um den Prozessor **902** anzuweisen, Steuerkanaldaten in einen Weitbereichsnetz-Frame einzufügen. Der Weitbereichsnetz-Frame kann ein Anwendungsnutzzinhalts-Frame-Feld zur Einkapselung der Steuerkanaldaten vor und während der Übertragung aufweisen. Die Steuerkanaldaten können Befehle an die erste Vorrichtung, einen Nutzzinhalt von einem sendenden Knoten zu sammeln, einen Zeitstempel aus dem Nutzzinhalt zu extrahieren und Nutzzinhalte, die einen Zeitstempel und eine Zonenkennung aufweisen, zurückzugeben, aufweisen. Die Steuerkanaldaten können einen Nachrichten-Header, ein Zonenkennungsfeld, ein Steuernachrichtsdatenfeld und ein Sicherheitsfeld umfassen. Bei einem Beispiel können die Steuerkanaldaten Befehle an die erste Vorrichtung, ihre Zonenkennung und eine Anzahl von Zeitstempeln von einem sendenden Knoten zurückzugeben, umfassen. Die Steuerkanaldaten können auch Befehle an die zweite Vorrichtung, an die erste Vorrichtung ihre Zonenkennung und einen Zeitstempel von einem sendenden Knoten zurückzugeben, umfassen, wobei die zweite Vorrichtung keinen Kommunikationsweg zu einer Servervorrichtung hat, außer indem durch die erste Vorrichtung geleitet wird.

[1321] Das nichtflüchtige maschinenlesbare Medium **22400** kann einen Code **22404** aufweisen, um den Prozessor **902** anzuweisen, den Weitbereichsnetz-Frame mit einem ersten Übertragungsprotokoll zu einer ersten Vorrichtung und mit einem zweiten Protokoll zu einer zweiten Vorrichtung zu senden, wobei die Steuerkanaldaten dazu dienen, Zugang sowohl zur ersten Vorrichtung und zum ersten Vorrichtungsprotokoll als auch zur zweiten Vorrichtung und zum zweiten Vorrichtungsprotokoll zu erhalten. Der Frame-Sender kann den Weitbereichsnetz-Frame durch Befehle an die erste Vorrichtung, als ein Zwischensender zu wirken und zur zweiten Vorrichtung zurückzusenden, zur zweiten Vorrichtung senden.

[1322] Das nichtflüchtige maschinenlesbare Medium **22400** kann einen Code **22404** aufweisen, um den Prozessor **902** anzuweisen, Daten eines sendenden Knotens ansprechend darauf zu modifizieren, dass die erste Vorrichtung eine erkannte Übertragung von einem sendenden Knoten zurückgibt. Die Modifikation der Daten des sendenden Knotens kann auf dem Extrahieren einer Zonenkennung und eines Zeitstempels, wodurch eine erkannte Übertragung von einem sendenden Knoten zurückgegeben wird, beruhen.

[1323] Bei einem Beispiel kann das maschinenlesbare Medium auch einen Berechner zum Berechnen des Orts des sendenden Knotens unter Verwendung einer hyperbolischen Funktion und einer Anzahl von Zeitstempeln von Paketen, die von der ersten Vorrichtung und der zweiten Vorrichtung empfangen werden, sowie der Zonenkennung der ersten Vorrichtung und der zweiten Vorrichtung aufweisen. Das maschinenlesbare Medium kann einen Berechner zum Berechnen des Orts der sendenden Vorrichtung auf der Grundlage der Sendedurchlaufzeit zwischen einer das Signal erfassenden Vorrichtung und einer das Signal empfangenden Vorrichtung aufweisen.

[1324] Die hier beschriebenen IoT-Vorrichtungen und Netze können eine große Menge an Daten erzeugen, die gemeint werden können, um Trends zu identifizieren, Probleme zu lösen und dergleichen. Die Datenanalytiken können jedoch von Praktikern und Forschern unterschiedlich interpretiert werden. Auf der Grundlage der breiten Vielfalt von datenanalytischen Techniken kann es schwierig sein, die technischen Herausforderungen der Datenanalytik in Zusammenhang mit IoT zu verallgemeinern. Eine Herausforderung für die IoT-Technologie besteht darin, ein Framework bereitzustellen, das es IoT-Praktikern und -Forschern ermöglicht, Analytiken über ihre IoT-Plattform leichter zu installieren.

[1325] Hier wird ein Architektur-Framework für die Analytik beschrieben, das Einflüsse der visuellen Analytik, der Entwurfswissenschaft und des Wissens-Engineerings übernimmt. Das in **Fig. 226** dargestellte Modell weist Aufgaben auf, die bei der Ausführung einer Analytik allgemein, einschließlich Edge-basierter Analytik, beteiligt sind. Durch Anbieten einer einheitlichen Sicht der Datenanalytik wirkt dieses Architektur-Framework als Referenz für Praktiker und Forscher der Datenanalytik, wodurch eine bessere Zusammenarbeit zwischen diesen Agenten ermöglicht wird. Das Architektur-Framework kann dabei helfen, technische Herausforderungen hervorzuheben, welche die Datenanalytik beeinflussen. Das Architektur-Framework kann auch verwendet werden, um die Beschreibung von Erfindungen in Bezug auf Forschungsrichtungen zu ermöglichen.

[1326] Die Datenanalytik kann grob in online-basierte, wie Edge-basierte Analytik und Offline-Analytik unterteilt werden. Die unterschiedlichen Typen der Analytik können verwendet werden, um zwischen der Erzeugung von Wissen aus repräsentativen Daten, während ein Offline-Zustand gegeben ist, und der Installation dieses Wissens in einem Online-System in der Art einer IoT-Plattform zu unterscheiden. Der Offline-Analytikabschnitt

entspricht breit der Edge-basierten Analytik. Ferner kann bei einem Beispiel die hierarchische Struktur der Offline-Analytik einer Ableitungshierarchie entsprechen, wobei mehr Wissen abgeleitet wird oder ein größerer Zusammenhang gelernt wird, wenn sich die Daten durch die verschiedenen Analytik-Aufgaben bewegen. Die hierarchische Struktur der Offline-Analytik kann in einer Entscheidung oder einer Aktion kulminieren, welche für die Steuerung eines Betätigungselements verwendet werden kann. Dieses hierarchische Analytikmodell bietet ein abgestuftes Framework, in dem komplexe Analytikaufgaben in Klassen von Teilaufgaben in der Art von Merkmalsextraktions-, Klassifizierungs- und Entscheidungsalgorithmen zerlegt werden können. Zusätzlich kann, wenn Teilaufgaben verteilbar sind, die Klassifizierung durch das Offline-Analytik-Framework als allgemeine Anleitung für die Platzierung von Arbeitslasten über eine IoT-Vorrichtung wirken. Bei einem Beispiel können Hardwareplattformen in Plattformgruppen ähnlicher Hardware-Ressourcen, wie Sensorknoten und Gateways, klassifiziert werden. In Fällen, in denen Plattformen in Gruppen ähnlicher Hardware-Ressourcen klassifiziert werden können, kann eine Klasse von Analytikaufgaben in einer bestimmten Klasse von Hardwareplattformen installiert werden, indem Arbeitslasteigenschaften mit Analytikanforderungen unterliegenden Plattformeneigenschaften abgestimmt werden.

[1327] Fig. 225 ist ein schematisches Diagramm eines Konzeptmodells **22500** der Datenanalytik gemäß einigen Ausführungsformen. Bei einem Beispiel weist das Konzeptmodell **22500** ein Daten-Mining **22502** und eine Datenfusion **22504** auf. Das Konzeptmodell **22500** kann eine Datenstruktur in einem Speicher eines oder mehrerer Netzsysteme repräsentieren. Beispielsweise kann das Daten-Mining **22502** auf einem Daten-Aggregator in einem IoT-Netz oder einer Fog-Vorrichtung stattfinden, während die Datenfusion **22504** an einem Datensammelsystem ausgeführt werden kann.

[1328] Das Daten-Mining **22502** kann als die Erzeugung von Wissen in einer Stapel- oder Offline-Art angesehen werden. Die Datenfusion **22504** kann als das Anwenden des Wissens in einer Online-Art betrachtet werden. Das Konzeptmodell **22500** beschreibt das Daten-Mining **22502** durch Kombinieren des Sichtanalytikparadigmas in einem kombinierten Modell **22506**, wobei die Interaktion des Datenanalytikers mit den Daten durch Datenvisualisierungstechniken **22508** als zentrale Aktivität eingesetzt wird. Das Entwurfswissenschafts-Forschungsparadigma beim Daten-Mining **22502** modelliert die Wissenserzeugung als einen iterativen Prozess von Modellbewertungszyklen durch einen Entwurf und bildet ein Analytikmodell **22510**, das existierendes Wissen aus einer Wissensdatenbank **22512** und Geschäftsanforderungen aus einer Anwendungs-Domain **22514** verwendet. Die aus der Umgebung der Datenfusion **22504** ankommenden Daten können für das Daten-Mining **22502** durch ein Datenlagerhaus **22516** empfangen werden. Die Daten können dann in das kombinierte Modell **22506** für die Verwendung in den Zyklen zwischen dem Entwurfs- und Bildungsanalytikmodell **22510** und dem Visualisierungs- und Bewertungsmodell **22508** gereinigt, transformiert und geladen **22518** werden.

[1329] Das Datenfusionsmodell **22504** kann Daten vom Konzeptmodell **22500** empfangen und sie auf die Arbeitslast- und Ressourcenbehandlung **22520** anwenden. Vom Arbeitslast- und Ressourcenbehandlungsmodell können Daten zu einem Signalfusionsmodell **22522** und einem Signalverfeinerungsmodell **22524** gesendet werden.

[1330] Innerhalb des Signalverfeinerungsmodells **22524** kann das Signal mit einer Signalverarbeitung **22526** verarbeitet werden, indem die empfangenen Daten genommen werden und diese Daten mit Daten, die mit einem Sensorbetätigungselement **22528** erfasst wurden, verglichen und dadurch ausgetauscht werden. Wenn gleich nur ein Sensorbetätigungselement **22528** und ein Signalprozessor **22526** dargestellt sind, können mehrere Paarungen existieren. Ferner können jegliche von einem Signalbetätigungselement erzeugte Daten für das Daten-Mining **22502** zum Datenlagerhaus **22516** zurückgegeben werden. Daten vom Arbeitslast- und Ressourcenmanager **22520** können auch zu einem Merkmalsextraktions- oder Objektverfeinerungsmodell **22530** gesendet werden, das die Daten sowie zusätzliche Daten von der Signalverfeinerung **22524** empfangen kann. Daten vom Arbeitslast- und Ressourcenmanager **22520** können auch zu einem Erkennungs-, Klassifizier- oder Vorhersageereignis- oder Szenariomodell **22532** gesendet werden, das die Daten sowie zusätzliche Daten vom Merkmalsextraktions- oder Objektverfeinerungsmodell **22530** empfangen kann. Daten vom Arbeitslast- und Ressourcenmanager **22520** können zum Handeln-nach-Entscheidung-Modell **22534** gesendet werden, das die Daten sowie zusätzliche Daten vom Erkennungs-, Klassifizier- oder Vorhersageereignis- oder Szenariomodell **22532** empfangen kann. Auf der Grundlage der Entscheidung kann eine Aktion vom Handeln-nach-Entscheidung-Modell **22534** zum Signalverfeinerungsmodell **22526** gesendet werden. Auf der Grundlage der Entscheidung kann eine Aktion vom Handeln-nach-Entscheidung-Modell **22534** zum Visualisierungs- und Handlungsmodul **22536** gesendet werden.

[1331] Fig. 226 zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens **22600** zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung zur Bereitstellung einer Datenanalytik von IoT-Systemen gemäß eini-

gen Ausführungsformen. Das Verfahren **22600** aus **Fig. 226** kann durch die mit Bezug auf **Fig. 227** beschriebene IoT-Vorrichtung **22700** implementiert werden. Das Verfahren **22600** kann unter Verwendung des mit Bezug auf **Fig. 8** beschriebenen Systems **802** ausgeführt werden. Der Prozessfluss kann in Block **22602** beginnen. Der Beginn dieses Prozesses kann durch eine Anforderung von einer Client-Vorrichtung, einer Anwendung oder einer dritten Partei, welche die Datenanalytik eines Zielsystems bewertet, ausgelöst werden.

[1332] In Block **22604** können repräsentative Daten von einer Installationsumgebung in der Art verteilter Sensoren in einer Fabriküberwachungsumgebung erhalten werden. Diese Daten können durch physikalisches Verbinden mit einer Vorrichtung und Ziehen von Daten über UART oder andere Schnittstellen in einem Kommagetrennter-Wert- bzw. Comma-Separated-Value(CSV)-Dateiformat erhalten werden. Es können auch andere Formate verwendet werden, wie unter anderem Extensible Markup Language (XML) oder proprietäre Formate in der Art von Microsoft Excel.

[1333] In Block **22606** können Daten in den verschiedenen CSV-Dateien in abfragbare Daten in der Art von Structured Query Language (SQL) oder anderer Datenstrukturen eingefügt werden. Die Daten können gemäß einem Standardzeitreihenschema eingegeben werden, wobei ein Sensor seine eigene Spalte in der Tabelle haben kann und jede Zeile einen eindeutigen Zeitstempel haben kann.

[1334] In Block **22608** kann die Datenbank durchsucht werden, um schlechte Daten in der Art von Daten mit duplizierten Zeitstempeln oder fehlende Daten zu identifizieren. Dies kann von einem Benutzer ausgeführt werden oder unter Verwendung automatisierter Werkzeuge ausgeführt werden, welche duplizierte Werte und andere Fehler oder beide suchen. Diese duplizierten Daten können entfernt werden. In Block **22610** kann eine Analytikumgebung gewählt werden, in der die Datenanalytik ausgeführt werden kann. Dies könnte beispielsweise ein R-basiertes Datenwerkzeug, ein Python-basiertes Datenwerkzeug und andere Big-Data-Werkzeuge sein. In Block **22612** können Daten in eine Umgebung für die Analytik geladen werden. In Block **22614** können Daten innerhalb der Umgebung für die Analytik visualisiert werden, um die Gesamtstruktur der Daten zu verstehen, so dass ein Modell hypothesisiert werden kann. Die Struktur weist die zeitlichen Beziehungen zwischen den Proben und die Beziehung zwischen Proben von verschiedenen Sensoren auf. Verschiedene mathematische Werkzeuge in der Art einer Kreuzkorrelation/Autokorrelation können verwendet werden, um die Beziehungen zwischen den Daten zu verstehen.

[1335] In Block **22616** kann ein Modell vorgeschlagen werden, das die Daten erfasst oder im Fall eines Klassifizierungsproblems die nützlichen Daten aus den Daten extrahiert. In Block **22618** kann der Erfolg des vorgeschlagenen Modells bewertet werden, um Anwendungsziele zu erreichen. Falls das Modell beispielsweise ein Klassifizierer ist, kann die Funktionsweise des Modells in Bezug auf die Anzahl der falsch Positiven, der wahren Positiven und der falsch Negativen bewertet werden.

[1336] Falls in Block **22620** die Funktionsweise nicht mit Anwendungszielen übereinstimmt, kann der Prozessfluss zu Block **22614** für die Neuvisualisierung und -untersuchung von Daten zurückkehren. Falls in Block **22620** jedoch festgestellt wird, dass das Maß des Modellerfolgs oberhalb einer benutzerspezifisierten Schwelle liegt, welche durch die Anwendungsziele festgelegt werden kann, kann der Prozessfluss in Block **22622** fortgesetzt werden.

[1337] In Block **22622** kann das Modell analysiert werden, um eine Arbeitslast zu berechnen. Die Berechnung der Arbeitslast kann das Profilieren des RechenSpeichers und des Netzes und der Energieanforderungen der Arbeitslasten umfassen. In Block **22624** können die Mittel für die Zerlegung der Arbeitslast identifiziert werden. Bei einem Beispiel können die Mittel für die Zerlegung der Arbeitslast durch Parallelität oder modulare seriell abhängige Aufgaben funktionieren. Ein anderes Beispiel der Zerlegung einer Arbeitslast könnte sich auf eine Klassifikationsaufgabe beziehen, bei der die Vorverarbeitungs-, die Merkmalsextraktions- und die Klassifizierungsaufgabe getrennt werden können. Eine Ausführungsarbeitslastcharakterisierung der Teilarbeitslasten könnte zumindest teilweise unter Verwendung der vorstehend angegebenen Metriken ausgeführt werden.

[1338] In Block **22626** können die verfügbaren Vorrichtungen im verteilten Netz spezifiziert werden und können die verfügbaren Vorrichtungen Eigenschaften erhalten. Bei einem Beispiel können bei der Spezifikation und beim Erhalten von Eigenschaften eine Reihe von Benchmarks auf jede Vorrichtung angewendet werden und die Speicher-, Rechen-, Netz- und Energiekapazität gemessen werden. Eine Arbeitslast kann Teilarbeitslasten an Vorrichtungen, die eine enge Übereinstimmung mit den Vorrichtungen aufweisen, umfassen. Das Übereinstimmungsniveau kann durch Sortieren der Arbeitslasten in Bezug auf benötigte Ressourcen und Sortieren der Vorrichtungen in Bezug auf verfügbare Ressourcen und anschließendes Paaren der Arbeitslasten und Vorrichtungen in Abhängigkeit von ihrem Rang auf dieser Liste erhalten werden.

[1339] In Block **22628** können die Arbeitslasten unter Verwendung eines geeigneten Protokolls in der Art eines Sichere-Kopie-Protokolls (SCP) den Vorrichtungen auferlegt werden. In Block **22630** kann die Funktionsweise der Systeme in Bezug auf die Ressourcenverwendung und Anwendungsziele kontinuierlich überwacht werden. In Block **22632** kann der Prozess enden oder neu starten.

[1340] **Fig. 227** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **22700** zum Bereitstellen einer Datenanalytik von IoT-Systemen vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1341] Wie vorstehend auch mit Bezug auf **Fig. 8** gezeigt wurde, kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation der hier beschriebenen Gruppenerzeugungsfunktionen aufweisen. Wenn gleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen Datenlader **22702** zum Laden von Daten in eine ausgewählte interaktive Analytikumgebung aufweisen. Bei einem Beispiel können die Daten auf der Grundlage eines repräsentativen Datensatzes aus einer Umgebung geladen werden, wobei die interaktive Analytikumgebung wenigstens eine von einer Daten-Mining-Umgebung und einer Datenfusionsumgebung ist. Die Daten können vor dem Laden in eine ausgewählte interaktive Analytikumgebung gereinigt und organisiert werden. Die interaktive Analytik kann eine zeitliche Beziehung zwischen einer ersten Datenprobe und einer zweiten Datenprobe angeben. Ferner kann die interaktive Analytik eine Beziehung zwischen einer ersten Probe und einem ersten Sensor und der zweiten Datenprobe und einem zweiten Sensor angeben. Die Arbeitslast kann ein Profil der Rechenanforderungen, Speicheranforderungen und Netzanforderungen der Arbeitslast einschließen. Bei einem Beispiel kann die Arbeitslast eine Anzahl von Teilarbeitslasten an der Vorrichtung einschließen, wobei die Arbeitslast eine durch Sortieren der Arbeitslasten in Bezug auf eine benötigte Ressource und eine verfügbare Ressource erhaltene Übereinstimmung mit den Vorrichtungen und eine Paarung einer Arbeitslast mit einer Vorrichtung auf der Grundlage der benötigten Ressource und der verfügbaren Ressource einschließt.

[1342] Der Massenspeicher **808** kann einen Übereinstimmungsfeststeller **22704** zur Feststellung, ob ein vorgeschlagenes Modell mit Daten übereinstimmt, aufweisen, wobei das vorgeschlagene Modell eine Arbeitslast zerlegbarer Abschnitte zur Verarbeitung auf einer Anzahl von Vorrichtungen aufweist. Die Arbeitslast wird durch eine parallele und/oder eine modulare seriell abhängige Aufgabe zerlegt. Die Arbeitslast kann auch für eine Klassifizierungsaufgabe zerlegt werden, wobei die Vorverarbeitungs-, die Merkmalsextraktions- und die Klassifizierungsaufgabe getrennt werden können. Bei einem Beispiel kann die IoT-Vorrichtung einen Prozessor zum Zerlegen der Arbeitslast vor der Zuordnung der Arbeitslast zur Vorrichtung für die Ausführung durch die Vorrichtung aufweisen. Bei einem Beispiel kann der Prozessor auch die Funktionsweise einer Plattform in Bezug auf die nach der Zuordnung verwendeten Ressourcen und die Ausführung der Arbeitslast auf der Vorrichtung berechnen.

[1343] Bei einem Beispiel kann der Massenspeicher **808** eine Arbeitslast-Zuordnungseinrichtung **22706** zum Zuordnen der Arbeitslast zu einer Vorrichtung für die Ausführung durch die Vorrichtung aufweisen. Bei einem Beispiel nimmt die Daten-Mining-Umgebung Daten von einer Anwendungsdomain und von einer gespeicherten Wissensdatenbank entsprechend dem repräsentativen Datensatz aus einer Umgebung auf.

[1344] **Fig. 228** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **22800**, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1345] Das nichtflüchtige maschinenlesbare Medium **22400** kann einen Code **22802** aufweisen, um den Prozessor **902** anzuweisen, Daten in eine ausgewählte interaktive Analytikumgebung zu laden. Bei einem Beispiel können die Daten auf der Grundlage eines repräsentativen Datensatzes aus einer Umgebung geladen werden, wobei die interaktive Analytikumgebung wenigstens eine von einer Daten-Mining-Umgebung und einer Datenfusionsumgebung ist. Die Daten können vor dem Laden in eine ausgewählte interaktive Analytikumgebung gereinigt und organisiert werden. Die interaktive Analytik gibt eine zeitliche Beziehung zwischen einer ersten Datenprobe und einer zweiten Datenprobe an, und die interaktive Analytik gibt ferner eine Beziehung zwischen einer ersten Probe und einem ersten Sensor und der zweiten Datenprobe und einem zweiten Sensor an. Die Arbeitslast kann ein Profil der Rechenanforderungen, Speicheranforderungen und Netzanforderungen der Arbeitslast einschließen. Bei einem Beispiel kann die Arbeitslast eine Anzahl von Teilarbeitslasten an der Vorrichtung einschließen, wobei die Arbeitslast eine durch Sortieren der Arbeitslasten in Bezug auf eine benö-

tigte Ressource und eine verfügbare Ressource erhaltene Übereinstimmung mit den Vorrichtungen und eine Paarung einer Arbeitslast mit einer Vorrichtung auf der Grundlage der benötigten Ressource und der verfügbaren Ressource einschließt.

[1346] Das nichtflüchtige maschinenlesbare Medium **22800** kann einen Code **22804** aufweisen, um den Prozessor **902** anzuweisen, festzustellen, ob ein vorgeschlagenes Modell mit Daten übereinstimmt, wobei das vorgeschlagene Modell eine Arbeitslast zerlegbarer Abschnitte zur Verarbeitung auf einer Anzahl von Vorrichtungen umfasst. Die Arbeitslast kann durch eine parallele und/oder eine modulare seriell abhängige Aufgabe zerlegt werden. Die Arbeitslast kann auch für eine Klassifizierungsaufgabe zerlegt werden, wobei die Vorverarbeitungs-, die Merkmalsextraktions- und die Klassifizierungsaufgabe getrennt werden können. Bei einem Beispiel kann die IoT-Vorrichtung einen Prozessor zum Zerlegen der Arbeitslast vor der Zuordnung der Arbeitslast zur Vorrichtung für die Ausführung durch die Vorrichtung aufweisen. Bei einem Beispiel kann der Prozessor auch die Funktionsweise einer Plattform in Bezug auf die nach der Zuordnung verwendeten Ressourcen und die Ausführung der Arbeitslast auf der Vorrichtung berechnen.

[1347] Das nichtflüchtige maschinenlesbare Medium **22800** kann einen Code **22706** aufweisen, um den Prozessor **902** anzuweisen, die Arbeitslast einer Vorrichtung zur Ausführung durch die Vorrichtung zuzuordnen. Bei einem Beispiel nimmt die Daten-Mining-Umgebung Daten von einer Anwendungsdomain und von einer gespeicherten Wissensdatenbank entsprechend dem repräsentativen Datensatz aus einer Umgebung auf.

[1348] Zusätzlich zur Verwendung einer aktiven Datensammlung und Modellierung innerhalb eines IoT-Netzes können IoT-Vorrichtungen passive Erzeuger von Daten sein, die von anderen Systemen, Vorrichtungen oder Benutzern fern verarbeitet und verbraucht werden. Bei einigen Frameworks fließen Daten durch ein Netz, um sie entweder in der Fog oder in der Cloud fern zu speichern und zu verarbeiten. Auf der Grundlage der Anwendung können die verarbeiteten Informationen einer IoT-Vorrichtung oder -Vorrichtungen in einer geographischen Nähe zu den IoT-Knoten, welche die Daten erzeugt haben, zugeführt werden.

[1349] Bei der vorliegenden Offenbarung kann ein Im-Netz-Verarbeitungsparadigma ein IoT-Netz ausnutzen, um als ein integriertes Rechen- und Kommunikationssystem zu wirken. Ein Verfahren ermöglicht es, dass ein IoT-Netz als Parallelprozessor wirkt, indem Daten zusammenarbeitend verarbeitet werden, wenn sie das Netz durchlaufen. Durch parallele Datenverarbeitung kann die Abhängigkeit von der Weiterverarbeitung an der Ziel-IoT-Vorrichtung verringert oder beseitigt werden.

[1350] Durch Bereitstellen einer nähebasierten Parallelverarbeitung in einem IoT-Netz wird es ermöglicht, dass in einem Netz erzeugte Daten lokal an diesem Netz verbleiben. Eine nähebasierte Parallelverarbeitung kann auch Prozesse verringern oder beseitigen, die Daten zu externen Systemen und Netzen weiterleiten, wodurch inhärente mögliche Sicherheits- und Datenschutzlücken bei der externen Datenoffenlegung verringert werden. Die nähebasierte Parallelverarbeitung kann die in einem IoT-System vorhandene Latenz verringern und die Lokalität der erzeugten Informationen bewahren. Die Verringerung der Latenz für die Berechnungen und die Bewahrung der Lokalität können automatische und halbautomatische Steueranwendungen unterstützen, bei denen sich die Verbraucher der verarbeiteten Informationen wahrscheinlich in der Nähe der messenden Vorrichtungen befinden.

[1351] Bei einem Beispiel einer Parallelverarbeitung in einem IoT-Netz können künstliche neuronale Netze bzw. Artificial Neural Networks (ANN) als durch ein IoT-Netz implementierter generischer Parallelprozessor verwendet werden. ANN können jede messbare Funktion approximieren. Bei einem Beispiel kann eine Berechnung, die durch ein neuronales Feed-Forward-Netz ausgeführt wird, in verschiedene Aufgaben, die gleichzeitig auszuführen sind, unterteilt werden. Die Verarbeitung kann eine verteilte Verarbeitung, eine bewahrte Lokalität, eine verringerte Latenz und ähnliche Eigenschaften zur Verarbeitung der Daten ausnutzen, während die Ressourcenverwendung im Netz optimiert wird. Die verschiedenen Rechenaufgaben können unter Berücksichtigung der an jedem Knoten verfügbaren Ressourcen, der Verbindbarkeit des IoT-Netzes, von Erzeugern und Verbrauchern von Informationen im Netz auf mehrere IoT-Knoten verteilt werden.

[1352] Bei einem Beispiel können Rechenaufgaben zur Installation über eine Fog-Ressource, die eine Rechenpraxis anwendet, Rechenaufgaben in Bestandteile zu zerlegen, die für die Installation auf eine Anzahl von Plattformen innerhalb eines Netzes geeignet sind, zerlegt werden. Bei einem Beispiel kann der Installations- und Berechnungsansatz auf einer Installation über eine Fog beruhen, was zur Verwendung eines anderen Kommunikationsnetzes als Werkzeug für den Austausch von Daten zwischen Rechenplattformen führt. Bei diesem Beispiel werden die Berechnung und Kommunikationen als getrennte Prozesse angesehen. Dementsprechend erfolgt die Installation der verteilten Berechnung bei diesem beispielhaften System ohne Berück-

sichtigung der Netztopologie oder des Betriebs zur Unterstützung der Berechnung. Die gegenwärtig offenbarten Techniken berücksichtigen gemeinsam die Kommunikation und Berechnung gemäß einigen Ausführungsformen.

[1353] Fig. 229 zeigt ein Prozessflussdiagramm eines beispielhaften Verfahrens 22900 zur Verwendung durch eine Internet-der-Dinge(IoT)-Vorrichtung bei einer verteilten Zuordnung eines neuronalen Netzes und der Ressourcenverwaltung gemäß einigen Ausführungsformen. Das Verfahren 22900 aus Fig. 229 kann durch die mit Bezug auf Fig. 231 beschriebene IoT-Vorrichtung 23100 implementiert werden. Das Verfahren 22900 kann unter Verwendung des mit Bezug auf Fig. 8 beschriebenen Systems 802 ausgeführt werden. Der Prozessfluss kann in Block 22002 beginnen.

[1354] In Block 22002 kann eine IoT-Vorrichtung eine Netztopologiekarte und Auflistung durch Identifizieren verbundener Knoten und physikalischer Netzeigenschaften erhalten. Die physikalischen Netzeigenschaften können den genauen Ort oder die relative Position zueinander einschließen. Physikalische Netzeigenschaften können auch den Abstand zwischen Knoten, die Clusterbildung, Verteilungsinformationen, empfangene Signalstärken und Signal-Rausch-Verhältnisse einschließen. Durch Erhalten einer Netztopologie durch die IoT-Vorrichtung kann zusätzlich eine Abstraktion der IoT-Netztopologie für die Weiterverwendung durch das Zuordnungssystem des neuronalen Netzes bereitgestellt werden. Hierbei können die Nähe der Vorrichtungen zueinander und die aktuellen Energieniveaus der Vorrichtungen festgestellt werden. Als Teil der Abstraktion können Signalmessungen von IoT-Vorrichtungen abgerufen werden. Ein Beispiel von Signalmessungen kann einen Indikator der empfangenen Signalstärke bzw. Received Signal Strength Indicator (RSSI) und die Ausstrahlungsleistung einschließen. Sobald eine Topologie erhalten wurde, kann die IoT-Vorrichtung einen erwarteten Wegverlust und Interferenzen im Netz zwischen Vorrichtungen modellieren. Die Ergebnisse der Abstraktion können in einer IoT-Datenbank 22904 gespeichert werden.

[1355] In Block 22906 können beim Verfahren die IoT-Knotenressourcen abstrahiert werden. Diese Ressourcen können Energieniveaus, den verfügbaren Speicherplatz, aktuelle Verarbeitungslasten, Netzfähigkeiten, die verfügbare Betriebszeit und Knotenzuverlässigkeitsinformationen einschließen. Bei einem Beispiel können Netzfähigkeiten Schnittstellentypen, Latenz und Datenkapazität einschließen. Bei einem Beispiel können die IoT-Ressourcen, die abstrahiert werden, über eine Anwendungsprogrammierschnittstellen(API)-Wrapper-Funktion oder Representational-State-Aufrufe offengelegt werden. Die Abstraktion der IoT-Knotenressourcen kann eine Softwareabstraktion des restlichen flüchtigen Speichers, der Restenergie und des restlichen permanenten Speichers einschließen. Bei einer anderen Teilaufgabe kann die API-Wrapper-Funktion für die Systemressourceninformationen eingesetzt werden. Sobald abstrahiert wurde, kann die IoT-Vorrichtung Ressourceninformationen abrufen, auf die sie zugreifen kann.

[1356] In Block 22908 kann eine Topologie des neuronalen Netzes durch Ausführen von Teilaufgaben bestimmt werden, wie das Bestimmen des Orts von Eingangsknoten mit Quelldaten für die Verarbeitung, verborgener Knoten für die Zwischenverarbeitung des neuronalen Netzes und Ausgangsknoten unter Verwendung von Sink-Daten. Wie mit Bezug auf andere Schritte bei diesem Verfahren erörtert, können Daten entweder lokal oder fern in der Datenbank 22904 oder einem entsprechenden Speichermedium gespeichert werden. Die Abstrahierung der Topologie des neuronalen Netzes kann beispielsweise unter anderem eine Identifikation eines Orts unter Ausnutzung des euklidischen Abstands unter Verwendung einer Signaltiangulation oder einer direkten Positionsmitteilung durch das globale Positionsbestimmungssystem (GPS) einschließen.

[1357] In Block 22910 führt das Verfahren eine Zuordnungsoptimierung aus. Bei der Zuordnungsoptimierung kann eine multivariate Zielfunktion mit dem Ziel einer optimalen Zuweisung von Knotenaufgaben auf der Grundlage aktueller und historischer Netz- und Knoteneigenschaften ausgewählt und verfeinert werden. Die Zielfunktion kann beispielsweise die Kosten, die Zuverlässigkeit, die Verarbeitungsgeschwindigkeit und die Ergebniserzeugungszeit oder die Verbreitung über einen geographischen Bereich begünstigen. Eine andere Teilaufgabe von Block 22910 kann die Formulierung eines ganzzahlig-linearen Programms, eine Zielfunktionsauswahl, das Verfeinern von Randbedingungen und eine Modellentwicklung einschließen.

[1358] In Block 22912 kann das Verfahren die Topologie des neuronalen Netzes dem Netz überlagern. Hierbei können in der Optimierungsstufe erhaltene Rollen und Aufgaben physikalischen Knoten im Netz zugeordnet werden. Aufgaben für einen Knoten werden erzeugt, präpariert und auf die physikalischen Knoten oder Vorrichtungen übertragen. Eine Teilaufgabe von Block 22912 kann eine erfolgreiche Installation der Rollen und Aufgaben aus der Optimierungsstufe bestätigen. Nach der erfolgreichen Übertragung der Aufgaben und Rollen kann das System als Vorbereitung für nachfolgende Arbeitslastzuweisungsanforderungen eine aktualisierte Netz- und Knotenzuordnungsübung einleiten. Der Prozessfluss kann dann enden und bei Bedarf auch wieder

starten, um das IoT in eine Reihe von Aufgaben zu abstrahieren, die entlang Knoten zur Kommunikation und Verarbeitung verteilbar sind, wie mit dem vorstehenden Sprachgebrauch erörtert wurde.

[1359] **Fig. 230** ist ein schematisches Diagramm für eine verteilte Zuordnung eines neuronalen Netzes **23000** zur Ressourcenverwaltung gemäß einigen Ausführungsformen. Eine Legende von Symbolen ist in Block **23002** bereitgestellt, wodurch die Eingangsknoten, Ausgangsknoten und verborgenen Knoten in den beispielhaften dargestellten IoT-Netztopologien identifiziert sind.

[1360] Bei der Eingangs-IoT-Netztopologie **23004** sind drei Eingangsschichtknoten, vier Verborgene-Schicht-Knoten und zwei Ausgangsschichtknoten im Gitter der IoT-Verbindungen gestreut dargestellt. Bei einem Beispiel repräsentiert jeder Punkt der IoT-Topologie einen Knoten. Knoten können wie vorstehend und in dieser gesamten Anmeldung beschrieben sein und eine IoT-Vorrichtung, einen Server oder ein anderes verbindbares Werkzeug für die Kommunikation und Verarbeitung repräsentieren. Bei einem Beispiel kann die Eingangs-IoT-Netztopologie eine Visualisierung von Knotenverbindungen vor der Optimierung der Abbildung eines neuronalen Netzes in ein physikalisches IoT-Netz repräsentieren. Jeder der dargestellten Knoten oder jede der dargestellten Vorrichtungen kann als ein oder mehrere Neuronen wirken, und Verbindungen sind durch Drahtlosstrecken verwirklicht.

[1361] Ein Zuordnungs-Framework **23006** kann einen Versuch einer Zuordnung zwischen Knoten zum Minimieren der Sendeleistung und der Übertragungszeit für die Übertragung von Informationen von den Eingangsknoten zu den Ausgangsknoten repräsentieren. Die Zuordnung kann die an jeder Vorrichtung verfügbaren Ressourcen, die Fähigkeiten jeder Vorrichtung und die Verbindbarkeit des IoT-Netzes berücksichtigen. Die im Gitter der visualisierten IoT-Knoten-Netze dargestellten Knotenverbindungen können jeweils ein Gewicht und die Übertragungszeit von Informationen über einen bestimmten Knoten repräsentieren.

[1362] Das Zuordnungs-Framework **23006** kann eine Zuordnung ergeben, die Datenwege für eine Ausgangs-IoT-Netztopologie **23008** zeigt. Die Ausgangs-IoT-Netztopologie kann eine Identifikation physikalischer Knoten, denen Neuronen zuzuordnen sind, aufweisen. Die Zuordnung kann durch Formulieren eines Optimierungsmodells erreicht werden, das alle Eingaben in Zusammenhang mit dem zugrunde liegenden IoT-Netz und dem überlagerten neuronalen Netz verwendet. Eingaben in das Optimierungsmodell können die IoT-Topologie und die an jedem Knoten verfügbaren Ressourcen, die Topologie des neuronalen Netzes, die auf die IoT-Topologie abzubilden ist, den Satz von Quellknoten und den Satz von Ausgangsknoten einschließen. Für die Zwecke der IoT-Netztopologie können sich die Ressourcen an jedem Knoten beispielsweise unter anderem auf Ressourcen flüchtigen Speichers, Energieressourcen, Sensorressourcen oder Ressourcen permanenten Speichers beziehen. Ähnlich kann die Topologie des neuronalen Netzes auf eine IoT-Topologie abgebildet werden, einschließlich der Anzahl von Schichten und verborgenen Neuronen, wie zumindest in **Fig. 230** dargestellt ist.

[1363] **Fig. 231** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **23100** zur verteilten Zuordnung eines neuronalen Netzes und zur Ressourcenverwaltung vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die in **Fig. 8** beschrieben sind.

[1364] Bei einem Beispiel kann der Massenspeicher **808** eine Anzahl von Modulen zur Implementation des hier beschriebenen Zuordnungs-Frameworks aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind. Der Massenspeicher **808** kann einen IoT-Netztopologieidentifizierer **23102** zum Identifizieren einer IoT-Netztopologie, wodurch die Verbindungen zwischen einer Anzahl von IoT-Knoten in einem IoT-Netz dargestellt werden, aufweisen. Die IoT-Netztopologie zeigt Knoteneigenschaften, einschließlich beispielsweise Abständen zwischen Knoten, einer Clusterbildung, Dispersionsinformationen, empfangener Signalstärken und/oder Signal-Rausch-Verhältnisse. Bei der Identifikation der IoT-Netztopologie können die Nähe der Anzahl der Knoten zueinander, die aktuellen Energieniveaus der Anzahl von Knoten und eine Signalmessung für die Anzahl von Knoten bestimmt werden. Bei einem Beispiel kann die Signalmessung der Anzahl von Knoten durch Abrufen des Indikators der empfangenen Signalstärke und/oder der Ausstrahlungsleistung geschehen.

[1365] Der Massenspeicher **808** kann einen IoT-Knotenressourcenidentifizierer **23104** zum Identifizieren von IoT-Knotenressourcen für jeden in der IoT-Netztopologie identifizierten IoT-Knoten aufweisen. Die IoT-Knotenressourcen können das Energieniveau, den verfügbaren Speicherplatz, die aktuelle Verarbeitungslast, die Netzkapazitäten, die verfügbare Betriebszeit und/oder Knotenzuverlässigkeitsinformationen umfassen.

[1366] Der Massenspeicher **808** kann einen Identifizierer **23106** der Topologie des neuronalen Netzes zur Identifikation der Topologie des neuronalen Netzes von Knotenabständen und Knotenorten aufweisen. Das neuronale Netz kann ein künstliches neuronales Netz sein.

[1367] Der Massenspeicher **808** kann einen Zuordnungsoptimierer **23108** zur Optimierung einer Zuordnung unter anderem auf der Grundlage der IoT-Knotenressourcen, der Knotenabstände und der Knotenorte aufweisen. Die optimierte Zuordnung bewahrt den Ort der Verarbeitung der zerlegbaren Aufgabe über das IoT-Netz durch die Verwendung der Knotenorte der Anzahl von Knoten zur Identifikation eines Knotens oder der Anzahl der Knoten, die sich am selben physikalischen Ort befinden. Bei der Optimierung der Zuordnung wird die Übertragungszeit zur Übertragung der Informationen von den Eingangsknoten zu den Ausgangsknoten bestimmt.

[1368] Der Massenspeicher **808** kann einen Prozessor **23110** für zerlegbare Aufgaben zur Verarbeitung einer zerlegbaren Aufgabe in eine Anzahl von IoT-Knoten auf der Grundlage einer Überlagerung der Topologie des neuronalen Netzes auf dem IoT-Netz aufweisen. Die Verarbeitung einer zerlegbaren Aufgabe in der Anzahl von IoT-Knoten umfasst das Senden von Teilen der zerlegbaren Aufgabe zu der Anzahl von IoT-Knoten auf der Grundlage davon, ob festgestellt wurde, dass sich die IoT-Knoten an einem physikalischen Ort auf demselben Netz befinden. Die Überlagerung der Topologie des neuronalen Netzes kann drei Schichten aufweisen, beispielsweise einschließlich einer Eingangsschicht, einer verborgenen Schicht und einer Ausgangsschicht.

[1369] **Fig. 232** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **23200**, das Code aufweist, um über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen zu berichten, gemäß einigten Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 9** beschrieben wurden.

[1370] Das nichtflüchtige maschinenlesbare Medium **23200** kann einen Code **23202** aufweisen, um den Prozessor **902** anzuweisen, eine IoT-Netztopologie zu identifizieren, welche die Verbindungen zwischen einer Anzahl von IoT-Knoten in einem IoT-Netz zeigt. Die IoT-Netztopologie kann beispielsweise Knoteneigenschaften zeigen, einschließlich beispielsweise Abständen zwischen Knoten, einer Clusterbildung, Dispersionsinformationen, empfangener Signalstärken und/oder Signal-Rausch-Verhältnisse. Bei der Identifikation der IoT-Netztopologie können die Nähe der Anzahl der Knoten zueinander, die aktuellen Energieniveaus der Anzahl von Knoten und eine Signalmessung für die Anzahl von Knoten bestimmt werden. Bei einem Beispiel kann die Signalmessung der Anzahl von Knoten durch Abrufen des Indikators der empfangenen Signalstärke und/oder der Ausstrahlungsleistung geschehen.

[1371] Das nichtflüchtige maschinenlesbare Medium **23200** kann einen Code **23204** aufweisen, um den Prozessor **902** anzuweisen, IoT-Knotenressourcen jedes in der IoT-Netztopologie identifizierten IoT-Knotens zu identifizieren. Die IoT-Knotenressourcen umfassen das Energieniveau, den verfügbaren Speicherplatz, die aktuelle Verarbeitungslast, die Netzkapazitäten, die verfügbare Betriebszeit und/oder Knotenzuverlässigkeitsinformationen.

[1372] Das nichtflüchtige maschinenlesbare Medium **23200** kann einen Code **23206** aufweisen, um den Prozessor **902** anzuweisen, eine Topologie eines neuronalen Netzes von Knotenabständen und Knotenorten zu identifizieren. Das neuronale Netz kann ein künstliches neuronales Netz sein.

[1373] Das nichtflüchtige maschinenlesbare Medium **23200** kann einen Code **23208** aufweisen, um den Prozessor **902** anzuweisen, eine Zuordnung auf der Grundlage der IoT-Knotenressourcen, der Knotenabstände und der Knotenorte zu optimieren. Die optimierte Zuordnung bewahrt den Ort der Verarbeitung der zerlegbaren Aufgabe über das IoT-Netz durch die Verwendung der Knotenorte der Anzahl von Knoten zur Identifikation eines Knotens oder der Anzahl der Knoten, die sich am selben physikalischen Ort befinden, beispielsweise in einem Gebiet einer Stadt in der Art einer Kreuzung, eines Gebäudes, eines Raums in einem Gebäude und dergleichen. Die Optimierung der Zuordnung umfasst die Übertragungszeit für die Übertragung der Informationen von den Eingangsknoten zu den Ausgangsknoten.

[1374] Das nichtflüchtige maschinenlesbare Medium **23200** kann einen Code **23210** aufweisen, um den Prozessor **902** anzuweisen, eine zerlegbare Aufgabe in der Anzahl von IoT-Knoten auf der Grundlage einer Überlagerung der Topologie des neuronalen Netzes auf dem IoT-Netz zu verarbeiten. Bei der Verarbeitung einer zerlegbaren Aufgabe in der Anzahl von IoT-Knoten können Teile der zerlegbaren Aufgabe auf der Grundlage davon, ob die IoT-Knoten als sich an einem physikalischen Ort oder am selben lokalen Netz befindend identifiziert wurden, beispielsweise durch einen Router oder eine Peer-to-Peer-Verbindung gekoppelt, zu der Anzahl von IoT-Knoten gesendet werden. Die Überlagerung der Topologie des neuronalen Netzes kann bei-

spielsweise drei Schichten, einschließlich einer Eingangsschicht, einer verborgenen Schicht und/oder einer Ausgangsschicht, aufweisen.

[1375] Gemäß einigen Ausführungsformen können IoT-Netze Blockchains für mehrere Funktionen verwenden. Diese können beispielsweise unter anderem das Erzeugen von Gruppenidentitäten, das Erzeugen von Typidentitäten, das Archivieren von Vertrauensmessungen, das Registrieren von Objektidentifizierern, eine sichere Vorrichtungseinbringung, eine Ereignisverfolgung und eine Datenprotokollierung einschließen. Die Blockchain-Synchronisation führt jedoch zusätzlichen Overhead herbei, was für beschränkte Vorrichtungen schwierig sein kann. Die Verwendung nicht lokalisierter Blockchains in der Art solcher, die Transaktionen von irgendwo akzeptieren, kann zur Sättigung von IoT-Subnetzen mit beschränkter Bandbreite führen, was unter anderen Problemen zu funktionellen Verzögerungen oder zum Verlust von Daten führen kann. Folglich kann eine Strategie zur lokalisierenden Blockchain-Verarbeitung erforderlich sein, um den Bedarf zu verringern. Ferner können kleinere Blockchains infolge einer geringeren Anzahl von Knoten weniger vertrauenswürdig sein.

[1376] **Fig. 233** ist ein schematisches Diagramm einer Hierarchie von Blockchains **23302** in Zusammenhang mit Ebenen in einer Netzhierarchie **23304** gemäß einigen Ausführungsformen. Die Hierarchie von Blockchains **23302** kann die Effizienz des lokalen Subnetzverkehrs für die Aufrechterhaltung und Verwendung der Blockchains **23302** erhöhen. Um die Effizienz weiter zu verbessern, können die Blockchains **23304** durch eine zugeordnete Hierarchie von Merkle-Bäumen **23306** indexiert werden, wie weiter mit Bezug auf **Fig. 235** beschrieben. Wie hier verwendet, ist ein Merkle-Baum allgemein eine Form eines Hash-Baums, in dem jeder Nicht-Blattknoten mit einem Hash der Bezeichnungen oder Werte von zwei nachgeordneten Knoten bezeichnet ist.

[1377] IoT-Subnetze können jeweils eine Blockchain aufweisen, die für das Subnetz lokal ist, so dass Blockchain-Operationen in das Subnetz eingeschlossen sind. Demgemäß kann eine häufige Verwendung der lokalen Blockchain Subnetze, die sich mit dem lokalen Subnetz verbinden, nicht sättigen.

[1378] Wie in **Fig. 233** dargestellt ist, kann ein lokales IoT-Netz (R1) **23308**, das sich beispielsweise in einem Raum oder einer lokalen Umgebung befindet, eine zugeordnete Blockchain **23310** aufweisen. Aktionen, die in R1 **23308** ausgeführt werden, werden in Transaktionen **23312** aufgenommen, die an die Blockchain **23310** übergeben werden können, um Aktivitäten in der Art von Identitäten, Gruppenzusammensetzung, Sicherheit, Operationsverfolgung und dergleichen aufzuzeichnen. Eine Transaktion **23312** kann in einem Block in der Blockchain **23310** gespeichert werden. Ein zugeordneter Hash-Code kann für den Block berechnet und in einem Merkle-Baum **23306** gespeichert werden. Im Merkle-Baum **23306** repräsentiert ein Dreieck einen übergeordneten Knoten oben und zwei nachgeordnete Knoten darunter. Bei diesem Beispiel können ein R1-Merkle-Baum **23314** und ein R2-Merkle-Baum **23316** mit verschiedenen Blöcken in der Blockchain **23310** assoziiert werden.

[1379] Das lokale IoT-Netz, R1 **23308**, kann mit einem IoT-Netz höherer Ebene in der Art eines Heimnetzes (H1) **23318** durch eine Brücke oder einen Router **23320** gekoppelt werden. H1 **23318** kann eine Blockchain **23322** zur Aufzeichnung von Transaktionen **23324** von H1 **23318** aufweisen. Eine Kontrollpunkttransaktion **23326** kann periodisch, beispielsweise jede Sekunde, Minute oder bei anderen Wiederholungszeiträumen, in der zum übergeordneten Netz H1 **23318** gehörenden Blockchain **23322** erzeugt werden. Die Kontrollpunkttransaktion **23326** kann den Hash-Wert für die R1-Merkle-Bäume **23314** oder **23316** unter anderen Merkle-Bäumen oder eine Probe an die Blockchain **23310** unterer Ebene übermittelter Blöcke aufweisen.

[1380] Die höchsten Vertices **23328** für die Merkle-Bäume R1 **23314** und R2 **23316** sind durch Netzreferenzen **23330** mit der untersten Ebene **23332** der H1-Merkle-Bäume **23334** verknüpft. Ähnlich kann H1 **23318** mit einem nächsthöheren Netz in der Art einer IoT-Netz-Cloud (C1) **23336** durch eine andere Brücke oder einen anderen Router **23338** gekoppelt werden. Konsolidierte Kontrollpunkttransaktionen **23340** können in der mit C1 **23336** assoziierten öffentlichen oder privaten Blockchain **23342** erzeugt werden. Ferner kann C1 **23336** Transaktionen **23344** in der Blockchain **23342** speichern. Die unterste Ebene **23346** des C1-Merkle-Baums **23348** kann Netzreferenzen **23350** aufweisen, die anhand des Hash-Codes der Scheitelpunkte **23352** höchster Ebene der nächstniedrigeren Ebene von Merkle-Bäumen in der Art der H1-Merkle-Bäume **23334** erzeugt werden.

[1381] Wenngleich er als ein einfacher Satz kaskadierender Blockchains und zugeordneter Merkle-Bäume über drei Ebenen dargestellt ist, kann der Prozess eine Kaskade bis zu einer Root-Blockchain für eine große Anzahl von Teilnehmern und Ebenen aufweisen. Die periodischen Kontrollpunkte ermöglichen es, einen großen Teil des lokalen Verkehrs von den übergeordneten Blockchains zu isolieren, wodurch eine Skalierbarkeit von IoT-Netzen ermöglicht wird, während weiter die Integrität durch die Verwendung von Blockchains ge-

schützt wird. Mit der umfangreichen Verwendung von Blockchains kann es nützlich sein, wenn ein definiertes Verfahren zur Instanziierung und Permissionierung neuer Blockchains zur Verfügung steht.

[1382] Fig. 234 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **23400** zur Bildung einer Blockchain-Hierarchie gemäß einigen Ausführungsformen. Das Verfahren **23400** aus Fig. 234 kann durch die mit Bezug auf Fig. 242 beschriebene IoT-Vorrichtung **24200** implementiert werden. Das Verfahren **23400** kann in Block **23402** beginnen, wenn beispielsweise eine IoT-Vorrichtung hochgefahren wird oder sich einem lokalen Netz anschließt.

[1383] In Block **23404** schreibt eine Vorrichtung im aktuellen oder lokalen IoT-Subnetz Transaktionsdaten in die aktuelle Blockchain. Wie hier beschrieben, können die Transaktionsdaten IoT-Betriebsereignisse, vertrauenswürdige Berechnungsmessungen, Vorrichtungs- oder Gruppenidentitätsinformationen und dergleichen sein.

[1384] In Block **23406** kann festgestellt werden, ob der Blockchain-Block ein „Sync“-Block ist. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **23404** zurück. Falls der Block, wie in Block **23406** festgestellt, ein Sync-Block ist, bildet in Block **23408** ein Gateway-Blockchain-Knoten eine Nachricht, die den Hash-Code des Sync-Blocks enthält. Die Nachricht wird zu einer Blockchain auf der nächsten Ebene übertragen.

[1385] In Block **23410** übermitteln Miner in der Blockchain der nächsten Ebene die Nachricht zusammen mit einer Netzreferenz, die auf die untere Blockchain zeigt, zu einem aktuellen Block. In Block **23412** wird festgestellt, ob es eine Blockchain der nächsten Ebene gibt. Falls dies der Fall ist, kehrt der Prozessfluss zu Block **23406** zurück, um festzustellen, ob der Block ein Sync-Block ist. Falls dies nicht der Fall ist, endet das Verfahren **23400** in Block **23414**, wenn die IoT-Vorrichtungen zu normalen Operationen zurückkehren, um auf ein anderes periodisches Blockchain-Schreiben zu warten.

[1386] Fig. 235 ist eine erweiterte Ansicht der mit Bezug auf Fig. 233 beschriebenen Merkle-Bäume gemäß einigen Ausführungsformen. Wie beschrieben, erfordern viele Blockchains verwendende IoT-Verwendungsfälle das Abrufen von Informationen, die unter Verwendung der Blockchain-Blöcke integritätsgeprüft werden. Infolge der Natur von Blockchain-Transaktionen brauchen wichtige Messungen in der Chain jedoch nicht nahe beieinander liegen. Demgemäß ist eine wirksame Indexierung für die Blockchain erforderlich. Dies kann durch die Verwendung von Merkle-Bäumen zur Indexierung der Chain geschehen. Wie mit Bezug auf Fig. 233 beschrieben, können Blockchains, die Netzhierarchien kreuzen, Merkle-Bäume zusammen mit Netzreferenzen zur Indexierung der Transaktionen verwenden. Eine Blockchain kann ihren eigenen Merkle-Baum oder Index aufweisen. Die Kontrollpunkttransaktionen erfordern einen Kontextschalter zur nachgeordneten Blockchain, die den Kontrollpunkt ursprünglich erzeugt hatte. Eine Suchmaschine, die versucht, Einblicke hinsichtlich der mit Kontrollpunkten versehenen Blöcke zu erhalten, kann das Netz durchlaufen müssen, beispielsweise indem es unter anderem den Netzreferenzen **23330** oder **23350** folgt, um die Merkle-Baumindizes für niedrigere Ebenen in der Blockchain-Hierarchie zu durchsuchen.

[1387] Fig. 236 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **23600** zum Durchsuchen einer Blockchain-Hierarchie unter Verwendung von Merkle-Baumindizes gemäß einigen Ausführungsformen. Das Verfahren **23600** aus Fig. 236 kann durch die mit Bezug auf Fig. 242 beschriebene IoT-Vorrichtung **24200** implementiert werden. Das Verfahren kann in Block **23602** beginnen, beispielsweise wenn eine Anfrage zur Lokalisierung von Daten empfangen wird. In Block **23604** können die Anfragedaten in einer hierarchischen Blockchain lokalisiert werden. In Block **23606** kann ein Datenwert verwendet werden, um einen Index oder eine Nachschlagetabelle zu konsultieren, worin Datenwerte mit Block-Hash-Werten assoziiert sind. In Block **23608** wird veranlasst, dass die aktuelle Blockchain auf die Root einer hierarchischen Blockchain zeigt. In Block **23610** wird der Block-Hash-Wert zur Konsultierung eines Merkle-Baums für eine aktuelle Blockchain verwendet, um den Ort eines Zielblocks in einer Kette von Blöcken in der Blockchain zu bestimmen.

[1388] In Block **23612** wird festgestellt, ob der Zielblock einen Sync-Block-Hash von einer nachgeordneten Blockchain enthält. Falls dies der Fall ist, wird die aktuelle Blockchain in Block **23614** so festgelegt, dass sie auf die nachgeordnete Blockchain zeigt, um die nachgeordnete Blockchain zu durchsuchen. Der Prozessfluss kehrt dann zur Block **23610** zurück, um die Suche in der nachgeordneten Blockchain wiederaufzunehmen.

[1389] Falls der Zielblock keinen Sync-Block-Hash enthält, wird der Zielblock in Block **23616** abgerufen und der suchenden Entität bereitgestellt. Das Verfahren endet dann in Block **23618**, beispielsweise wenn normale Operationen wiederaufgenommen werden.

[1390] Das Durchsuchen der Merkle-Baumindizes in niedrigere Ebenen der Blockchain-Hierarchie hinein unter Verwendung der Netzreferenzen kann zu einer erhöhten Netzlatenz führen. Das Caching von Merkle-Baumindizes für die nachgeordneten-Knoten-Blockchains kann ein Weg zum Verringern des Overheads der Indexsuchen sein, indem die Suchen auf die Root-Blockchain beschränkt werden. Ferner können Cloud-Server genügend Verarbeitungsressourcen haben, um die Merkle-Bäume für alle nachgeordneten Blockchains in einem IoT-Netz zu unterhalten.

[1391] **Fig. 237** ist ein schematisches Diagramm eines in einem Cloud-Server gespeicherten gecachten Merkle-Baums gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf **Fig. 233** beschrieben wurden. Bei diesem Beispiel gleicht der C1-Merkle-Baum **23348** den hierarchischen Merkle-Bäumen aus **Fig. 233**. Die unterste Ebene **23702** im C1-Merkle-Baum **23348** weist jedoch keine Netzreferenzen auf, sondern weist vielmehr Cache-Referenzen **23704** zu gecachten Merkle-Bäumen **23706** für IoT-Netze niedrigerer Ebene auf.

[1392] Beispielsweise können die gecachten Merkle-Bäume **23706** eine H1-Merkle-Baumkopie **23708** des mit Bezug auf **Fig. 233** beschriebenen H1-Merkle-Baums **23334** aufweisen. Bei der H1-Merkle-Baumkopie **23708** kann die unterste Ebene **23710** Referenzen **23712** zu Kopien **23714** von Merkle-Bäumen noch niedrigerer Ebenen aufweisen.

[1393] Ähnlich können Zwischen-Blockchains einen Teilbaum-Cache unterhalten, um zu ermöglichen, wirksamere regionale Suchen auszuführen. Beispielsweise zeigt **Fig. 238** ein schematisches Diagramm eines verteilten Merkle-Baum-Caches **23800** auf der IoT-Netzebene H1 **23318**, wie mit Bezug auf **Fig. 233** beschrieben. Der H1-Merkle-Baum **23334** kann der gleiche sein wie mit Bezug auf **Fig. 233** beschrieben. Die unterste Ebene **23802** kann jedoch Cache-Referenzen **23804** zu Kopien der Rn-Merkle-Bäume **23806** an Stelle von Netzreferenzen aufweisen.

[1394] **Fig. 239** ist ein schematisches Diagramm einer Technik zum Unterhalten eines verteilten Caches **23900** mit Kohärenz gemäß einigen Ausführungsformen. Weil die Caches nicht direkt mit den Blockchains gespeichert werden, auf die sie sich beziehen, kann es nützlich sein, ein Verfahren zum Aufrechterhalten der Cache-Kohärenz für nachgeordnete Blockchain-Merkle-Bäume zu implementieren. IoT-Frameworks können zur wirksamen Implementierung einer Veröffentlichen-Subskribieren-Signalisierung verwendet werden. Nachgeordnete Blockchains können Merkle-Bäume **23904** niedrigerer Ebene für eine übergeordnete Blockchain veröffentlichen, die einen Merkle-Baum **23906** höherer Ebene hält. Ähnlich kann die übergeordnete Blockchain den Merkle-Baum **23906** höherer Ebene für eine Root-Blockchain veröffentlichen **23908**, wie denn mit Bezug auf **Fig. 233** erörterten C1-Merkle-Baum **23348**.

[1395] **Fig. 240** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **24000** zur Bildung eines kohärenten Caches für eine Hierarchie von Blockchains gemäß einigen Ausführungsformen. Das Verfahren **24000** kann beispielsweise durch IoT- oder Cloud-Vorrichtungen auf jeder Ebene der Hierarchie implementiert werden. Beispielsweise kann das Verfahren **24000** aus **Fig. 240** durch die mit Bezug auf **Fig. 242** beschriebene IoT-Vorrichtung **24200** implementiert werden. Das Verfahren kann in Block **24002** beginnen, wenn ein IoT-Netz beispielsweise zum ersten Mal gestartet wird oder wenn sich eine IoT-Vorrichtung dem IoT-Netz anschließt.

[1396] In Block **24004** subskribiert eine aktuelle Blockchain einen nachgeordnete-Blockchain-Veröffentlicher-agenten. In Block **24006** akzeptiert die nachgeordnete Blockchain eine Registrierung des Übergeordnete-Blockchain-Subskriptionsagenten. Die Veröffentlichung und die Subskription (Pub-Sub) können nur die Indizes oder Merkle-Bäume zur Unterhaltung des kohärenten Caches aufweisen. Bei einigen Beispielen kann das Pub-Sub die vollständige Blockchain von der nachgeordneten Blockchain aufweisen.

[1397] In Block **24008** setzt die aktuelle Blockchain ihren aktuellen Zeiger auf ihre übergeordnete Blockchain. In Block **24010** wird festgestellt, ob die aktuelle Blockchain die Root-Blockchain ist. Falls dies der Fall ist, werden in Block **24012** die kohärenten Cache-Verbindungen eingerichtet und wartet das System darauf, dass Veröffentlichungsereignisse stattfinden, wie beispielsweise mit Bezug auf **Fig. 241** beschrieben. Falls die aktuelle Blockchain nicht die Root-Blockchain ist, kehrt der Prozessfluss zu Block **24004** zurück, um die Pub-Sub-Verbindungen für die nächste Ebene in der Hierarchie zu bilden.

[1398] **Fig. 241** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **24100** zur Aufrechterhaltung eines kohärenten Caches für eine Hierarchie von Blockchains gemäß einigen Ausführungsformen. Das Verfahren **24100** kann beispielsweise durch IoT- oder Cloud-Vorrichtungen auf einer beliebigen Hierarchieebene implementiert werden. Beispielsweise kann das Verfahren **24100** aus **Fig. 241** durch die mit Bezug auf **Fig. 242**

beschriebene IoT-Vorrichtung **24200** implementiert werden. Das Verfahren kann in Block **24102** beginnen, nachdem der kohärente Cache beispielsweise nach den Techniken aus **Fig. 240** gebildet wurde.

[1399] In Block **24104** empfängt der Blockchain-Cache-Agent ein Cache-Kohärenzereignis. Das Cache-Kohärenzereignis kann beispielsweise eine Veröffentlichung einer Änderung sein, die im Merkle-Baum für eine Blockchain niedrigerer Ebene stattgefunden hat. Bei einigen Beispielen kann ein periodisches Wiederauffrischen verwendet werden, um zu bestätigen, dass die Informationen im Merkle-Baum höherer Ebene korrekt sind.

[1400] In Block **24106** wird der Merkle-Baum-Weg von der Quell-Blockchain kopiert und für den Subskribenten-Cache-Agenten veröffentlicht. In Block **24108** ersetzt der Cache-Agent in der Subskribenten-Blockchain den aktuellen gecachten Merkle-Baum-Weg im dem nachgeordneten Baum und Block entsprechenden Teilbaum. In Block **24110** wird festgestellt, ob der Weg einen neuen Zweig des Merkle-Baums bildet. Falls dies nicht der Fall ist, kehrt der Prozessfluss zu Block **24104** zurück, um mit normalen Aktualisierungen zur Aufrechterhaltung der Cache-Kohärenz fortzufahren.

[1401] Falls in Block **24110** der Weg einen neuen Zweig des Merkle-Baums bildet, wird in Block **24112** eine neue lokale Root im Teilbaum gebildet. In Block **24114** wird die aktuelle Referenz gleich der lokalen Root gemacht. In Block **24116** wird festgestellt, ob die lokale Root die globale Root ist. Falls dies nicht der Fall ist, geht der Prozessfluss zu Block **24112** zurück, um eine neue lokale Root im nächsten Teilbaum zu bilden.

[1402] Falls in Block **24116** die lokale Root gleich der globalen Root ist, endet das Verfahren **24100** in Block **24118**. An diesem Punkt kann der Prozess in Block **24102** neu starten.

[1403] **Fig. 242** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **24200** zur Implementation hierarchischer Blockchains mit zugeordneten Indizes vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **24200** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1404] Die IoT-Vorrichtung **24200** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Eine Messung kann sich auf einen Hash-Code beziehen, der für einen Code oder ein Datensegment berechnet wird, der oder das im permanenten Speicher **808** oder im flüchtigen Speicher **804** gespeichert ist. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1405] Das TPM kann verwendet werden, um eine TEE oder eine sichere Enklave für das Ausführen von Programmen festzulegen. Das TPM kann auch für eine Anzahl anderer Funktionen, einschließlich der Bereitstellung einer kryptographischen Unterstützung für sichere Kommunikationen und Schlüssel für die Identifikation verwendet werden. Das TPM kann in stärker beschränkten Vorrichtungen in der Art von Sensoren ganz am Rand der IoT-Netze nicht vorhanden sein. Bei diesen Vorrichtungen kann Sicherheit durch eine Blockchain selbst, durch vorgeschaltete Vorrichtungen, durch ein virtuelles TPM und dergleichen bereitgestellt sein.

[1406] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Schlüsselbehandlungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1407] Der Massenspeicher **808** kann eine Blockchain-Logik **24202** aufweisen, die aufgenommen sein kann, um eine Blockchain **24204** zu unterhalten, welche Dienste, Attribute, Identitäten von Vorrichtungen, Contracts,

Münzguthabenbeträge und dergleichen hält. Die Blockchain-Logik **24202** kann verwendet werden, um die Blockchain-Transaktionen zu anderen IoT-Vorrichtungen zu übertragen. Ferner kann die Blockchain-Logik **24202** verwendet werden, um Netzreferenzen zu Blockchains auf niedrigeren oder höheren Ebenen einer Netzhierarchie einzurichten. Beispielsweise können die Netzreferenzen eine Verbindung durch einen Gateway oder Router zu einem IoT-Netz niedrigerer Ebene einschließen.

[1408] Ein Indexierer **24206** kann zur Erzeugung eines Merkle-Baums **24208** verwendet werden, der Hash-Codes von Blöcken in der Blockchain **2422** umfasst. Die untersten Ebenen des Merkle-Baums **24208** können Netzreferenzen zu Merkle-Bäumen in IoT-Vorrichtungen in einem IoT-Netz niedrigerer Ebene, die durch die Blockchain-Logik **24202** erzeugt werden, aufweisen.

[1409] Ein Lokalisierer **24210** kann aufgenommen sein, um eine Blockchain-Hierarchie zu durchsuchen. Der Lokalisierer **24210** kann diese Funktion wie mit Bezug auf Fig. 236 beschrieben ausführen, wobei der Lokalisierer **24210** relevante Merkle-Bäume nach Hash-Code der Zieldaten durchsucht, der verwendet werden kann, um den Block in den Blockchains zu finden.

[1410] Ein Cache-Erzeuger **24212** kann verwendet werden, um einen Cache **24214** von Merkle-Bäumen in IoT-Netzen, die auf einer niedrigeren Ebene in der Hierarchie sind, zu bilden. Beispielsweise kann der Cache-Erzeuger **24212** das mit Bezug auf Fig. 240 beschriebene Verfahren **24000** ausführen. Der Lokalisierer **24210** kann dann die Durchsuchung der Blockchain-Hierarchie im Cache **24214** ausführen, wodurch die den IoT-Netzen auferlegte Last verringert wird.

[1411] Die Kohärenz des Caches **24214** kann durch einen Cache-Agenten **24216** aufrechterhalten werden. Der Cache-Agent **24216** kann das mit Bezug auf Fig. 241 beschriebene Verfahren **24100** ausführen. Ferner kann der Cache-Agent **24216** Cache-Agenten in IoT-Vorrichtungen niedrigerer Ebene subskribieren, um eine Benachrichtigung über Cache-Kohärenzereignisse von diesen Cache-Agenten zu empfangen. Der Cache-Agent **24216** kann auch Cache-Kohärenzereignisse für die aktuelle IoT-Vorrichtung **24200** für IoT-Vorrichtungen höherer Ebene veröffentlichen, die den Cache-Agenten **24216** subskribiert haben, um Änderungsbenachrichtigungen zu empfangen.

[1412] Fig. 243 ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **24300**, das Code aufweist, um einen Prozessor **902** anzuweisen, Schlüssel für sichere Kommunikationen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **24300** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf Fig. 9 beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **24300** kann für den Massenspeicher **808** aus Fig. 8 beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1413] Das nichtflüchtige maschinenlesbare Medium **24300** kann einen Code **24302** aufweisen, um den Prozessor **902** anzuweisen, eine Blockchain-Hierarchie über ein hierarchisches IoT-Netz zu bilden, das sich beispielsweise von Vorrichtungen der untersten Ebene in der Art von Sensoren in einem Raum zu einem breiteren IoT-Netz in der Art eines Hauses oder eines Fabriknetzes und zu noch breiteren IoT-Netzen, die sich beispielsweise in der Cloud befinden, erstreckt. Der Code **24302** kann diese Funktion gemäß dem mit Bezug auf Fig. 243 beschriebenen Verfahren ausführen.

[1414] Ein Code **24304** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen hierarchischen Index der Blockchains zu bilden. Der hierarchische Index kann ein Merkle-Baum auf der Grundlage von Hash-Codewerten der Inhalte der Blöcke in den Blockchains sein.

[1415] Ein Code **24306** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen kohärenten Cache von Merkle-Bäumen auf der gegenwärtigen IoT-Netzebene zu bilden, wobei der kohärente Cache die Merkle-Bäume niedrigerer Ebenen im IoT-Netz aufweist. Der Code **24306** kann die Bildung des kohärenten Caches unter Verwendung des mit Bezug auf Fig. 240 beschriebenen Verfahrens ausführen.

[1416] Ein Code **24308** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Kohärenz des Caches aufrechtzuerhalten. Der Code **24308** kann diese Funktion unter Verwendung des mit Bezug auf Fig. 241 beschriebenen Verfahrens ausführen.

[1417] Eine Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **24300** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur

Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Boot-/Messeinrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1418] Veröffentlichen-Subskribieren (Pub-Sub) ist eine Teilmenge des inhaltszentrischen Netzwerks bzw. Content-Centric Networking (CCN), wobei eine Netzweiterleitungsfunktion auf die wirksame Weiterleitung von Inhalt verglichen mit dem Weiterleiten durch Senden von Paketen oder Frames zu einer spezifischen Netzadresse angewendet wird. Der Brennpunkt von Pub-Sub liegt auf der wirksamen Weiterleitung eines einzigen veröffentlichten Inhalts zu mehreren Subskribenten, mehreren Veröffentlichungen zu einem einzigen Subskribenten oder von beiden. Zusätzlich dazu, dass es für ein spezifisches Inhaltselement verwendet wird, kann sich Pub-Sub auf ein Thema beziehen, das ein logischer Titel oder eine Gegenstandszeile ist, worunter mehrere Inhaltselemente ausgetauscht werden können.

[1419] Das Pub-Sub-Modell ermöglicht die Definition eines Themas, das Netzvorrichtungen subskribieren oder veröffentlichen können oder beides davon ausführen können. Veröffentlichender können mehrere Themen veröffentlichen, und Subskribenten können mehrere Themen subskribieren. Demgemäß können Skalierbarkeitsbedenken auftreten, wenn der Themenverkehr weitergeleitet wird. Themen können in einer Vielzahl von Datenformaten, einschließlich Zeichenketten, Zahlen, Netz(Multicast)-Adressen, UUID und Objektkennungshierarchien, ausgedrückt werden. Die Wirksamkeit der Weiterleitung kann jedoch dadurch beeinflusst werden, wie Themen ausgedrückt und formatiert werden.

[1420] Wie hier beschrieben, können Bloomfilter ein wirksames Verfahren zur Repräsentation von Pub-Sub-Themen für die Weiterleitung bereitstellen. Das Bloomfilter gibt eine Übereinstimmung an, falls alle gesetzten Bits des Zielwerts, beispielsweise Bits mit einem Wert von eins, mit gesetzten Bits im Bloomfilter übereinstimmen. Bits, die nicht gesetzt wurden, beispielsweise einen Wert von Null haben, werden ignoriert. Falls demgemäß ein Bit im Bloomfilter gesetzt wird, jedoch einen Wert von Null im Zielwert hat, kann dennoch eine Übereinstimmung auftreten, solange alle gesetzten Bits im Zielwert im Bloomfilter gesetzt sind. Andere Techniken können für die Verfolgung der Themen für die Pub-Sub-Verteilung verwendet werden, wie unter anderem das Speichern von Bitmustern für die Themen in verteilten Hash-Tags bzw. Distributed Hash Tags (DHT) oder das Speichern eines Themas und des zugeordneten Status in einer Blockchain.

[1421] **Fig. 244** ist ein schematisches Diagramm der Verwendung der Pub-Sub-Weiterleitung auf der Grundlage von Bloomfiltern gemäß einigen Ausführungsformen. Die Vorrichtung bildet ein Bloomfilter, wobei ein Thema gehasht wird und dann dem Bitraum für das Bloomfilter überschrieben wird, beispielsweise durch eine Exklusiv-ODER-Funktion. Mehrere Themenformate können dasselbe Bloomfilter verwenden, weil alle unterschiedlich hashen würden. Die Länge des für das Bloomfilter verwendeten Bitraums kann auf der Grundlage der Informationstheorie bestimmt werden. Beispielsweise kann ein längeres Bloomfilter, das eine höhere Anzahl von Bitwerten aufweist, dafür sorgen, dass mehr Themen aufgenommen werden, während die Möglichkeiten verringert werden, dass Bits überlappen können, woraus sich ein inkorrektes Abrufen von Themen ergeben könnte.

[1422] Wie in **Fig. 244** dargestellt ist, kann ein Inhaltsveröffentlicher **24402** einen Hash-Code für ein Thema erzeugen. Der Veröffentlicher kann den Inhalt dann durch einen Router **24404** (U2) senden, der alle gesetzten Bits des Hash-Codes für den Inhalt aufweist. Der Router **24404** kann dann das Bloomfilter einem lokalen Veröffentlicher **24406** (P4) veröffentlichen. Andere Router **24404** in der Art von U2 und U3 können den ersten Router U2 subskribieren, beispielsweise eine Bitmap präsentieren, die Hash-Codes für Zielthemen aufweist.

[1423] Falls ein Router **24404** in der Art von U3 nicht alle im Bloomfilter gesetzten Bits aufweist, wird der Hash-Code für das Thema nicht durch diesen Baum gesendet. Dies kann angeben, dass kein Subskribent **24408** im vom Router für **24404** U3 unterhaltenen Baum dieses Thema subskribiert hat. Vom Router **24404** U2 kann der Hash-Code den anderen Veröffentlicher **24406** in der Art von P2 bereitgestellt werden. Ferner kann sich der Hash-Code für das Thema durch andere Router **24404** in der Art von U1 bewegen, solange alle gesetzten Bits im Bloomfilter in U1 übereinstimmen. Subskribenten **24408** in der Art von S1, S2 und S5 können Inhalt von einem Veröffentlicher **24406** in der Art von P1 oder von einem Router **24404** in der Art von U1 empfangen. Bei diesem Beispiel hat ein Subskribent **24408**, S2, einen Hash-Code für ein Zielthema, das mit dem Hash-Code vom Inhaltsveröffentlicher **24402** übereinstimmt.

[1424] Bei diesem Ansatz kann ein Subskribent **24408** ein Bloomfilter bilden, das überschriebene Hash-Codes für alle Themen, die er subskribieren möchte, enthält. Der Subskribent **24408** kann dann das Bloomfilter beim Router-Gewebe registrieren. Ein Veröffentlicher **24406** kann auch ein Bloomfilter, das überlappende Hash-

Codes für alle Themen, für die er Inhalt bereitstellen möchte, enthält, anliefern. Falls es beispielsweise mehrere demselben semantischen Thema entsprechende Formatierungsverfahren gibt, kann das Bloomfilter eines oder mehrere vergleichen, welche die Weiterleitungsanforderungen erfüllen.

[1425] Bei einem gegebenen Weiterleitungsschema, welches das Veröffentlichen-Subskribieren-Modell verwendet, ist es möglich, dass eine Sicherheitsrichtlinie der Themenmenge, die für ein Subnetz, einer Vorrichtung oder einem Gateway zu einem fremden Netz offengelegt werden kann, Restriktionen auferlegen möchte. Eine Bloomfilter-Maske kann zu einem Weiterleitungsknoten hinzugefügt werden, wobei die Maske einen Whitelist-Ausdruck von Themen, die weitergeleitet werden können, repräsentiert. Eine Maske kann auch verwendet werden, um eine Blacklist von Themen, die gefiltert werden, zu repräsentieren.

[1426] **Fig. 245** ist ein schematisches Diagramm der Verwendung eines Whitelist-Bloomfilters zum Ermöglichen der Verteilung von Inhalt gemäß einigen Ausführungsformen. In **Fig. 245** wird ein Schnitt für ein Themen-Bloomfilter **24502** und ein Whitelist-Bloomfilter **24504**, beispielsweise durch Ausführen einer UND-Funktion, berechnet. Falls das End-Bloomfilter **24506** null ist, ist das Themen-Bloomfilter **24502** im Whitelist-Bloomfilter **24504**, und der Inhalt kann zum Verbraucher weitergeleitet werden.

[1427] **Fig. 246** ist ein schematisches Diagramm der Verwendung eines Blacklist-Bloomfilters zum Verhindern der Verteilung von Inhalt gemäß einigen Ausführungsformen. In **Fig. 246** wird ein Schnitt für ein Themen-Bloomfilter **24602** und ein Blacklist-Bloomfilter **24604** berechnet, beispielsweise durch Ausführen einer NAND-Funktion zur Erzeugung eines Zwischen-Bloomfilters **24608**. Das Zwischen-Bloomfilter **24608** kann dann mit dem Themen-Bloomfilter **24602** zum Schnitt gebracht werden, beispielsweise unter Verwendung einer UND-Funktion **24610**. Falls das End-Bloomfilter **24612** nicht null ist, ist das Themen-Bloomfilter **24602** im Blacklist-Bloomfilter **24604**, und es wird verhindert, dass der Inhalt zum Verbraucher weitergeleitet wird.

[1428] **Fig. 247** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **24700** zur Implementation von Pub-Sub mit Blacklist- oder Whitelist-Bloomfiltern zur Inhaltssteuerung gemäß einigen Ausführungsformen. Das Verfahren **24700** aus **Fig. 247** kann durch die mit Bezug auf **Fig. 248** beschriebene IoT-Vorrichtung **24800** implementiert werden. Das Verfahren **24700** kann in Block **24702** beginnen, beispielsweise wenn ein Subskribent ein Bloomfilter berechnet, das Hash-Codes für eine Anzahl von Themen aufweist.

[1429] In Block **24706** registriert ein Administrator das Blacklist-Bloomfilter, das Whitelist-Bloomfilter oder beide bei Routern im System. In Block **24708** veröffentlicht ein Veröffentlichender Inhalt unter Verwendung eines Veröffentlichungs-Bloomfilters. Das Veröffentlichungs-Bloomfilter kann ein direkter Hash-Code des Themas oder der Veröffentlichung sein, wobei seine Bitlänge mit der Länge der für die Verteilung zu verwendenden Bloomfilter übereinstimmt.

[1430] In Block **24710** wird der Inhalt an einen Router übergeben. Der Router berechnet dann einen Pub-Sub-Schnitt für das Veröffentlichender- und das Subskribenten-Bloomfilter. In Block **24712** wird festgestellt, ob der Pub-Sub-Schnitt gleich null ist. Falls der Pub-Sub-Schnitt gleich null ist, wodurch angegeben wird, dass keine Überlappung zwischen dem Veröffentlichender- und dem Subskribenten-Bloomfilter vorhanden ist, endet das Verfahren **24700** in Block **24714**.

[1431] Falls in Block **24712** festgestellt wird, dass der Pub-Sub-Schnitt nicht gleich null ist, wird in Block **24716** ein Whitelist-Schnitt des Pub-Sub-Schnitts mit dem Bloomfilter für die Whitelist-Themen berechnet. In Block **24718** wird ein Blacklist-Schnitt des Pub-Sub-Schnitts mit dem Bloomfilter für die Blacklist-Themen berechnet.

[1432] In Block **24720** wird festgestellt, ob sowohl der Whitelist-Schnitt gleich null ist als auch der Blacklist-Schnitt nicht gleich null ist. Falls beide Bedingungen wahr sind, wird in Block **24722** der Inhalt nicht zum Subskribenten weitergeleitet. Falls eine der Bedingungen nicht wahr ist, wird in Block **24724** der Inhalt zum Subskribenten weitergeleitet. Das Verfahren **24700** endet dann in Block **24714**.

[1433] **Fig. 248** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **24800** zur Implementation eines Pub-Sub-Inhaltsverteilungssystems unter Verwendung von Bloomfiltern vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **24800** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1434] Die IoT-Vorrichtung **24800** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datenssegment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1435] Das TPM kann verwendet werden, um eine TEE oder eine sichere Enklave für das Ausführen von Programmen festzulegen. Das TPM kann auch für eine Anzahl anderer Funktionen, einschließlich der Bereitstellung einer kryptographischen Unterstützung für sichere Kommunikationen und Schlüssel für die Identifikation verwendet werden. Das TPM kann in stärker beschränkten Vorrichtungen in der Art von Sensoren ganz am Rand der IoT-Netze nicht vorhanden sein. Bei diesen Vorrichtungen kann Sicherheit durch eine Blockchain selbst, durch vorgeschaltete Vorrichtungen, durch ein virtuelles TPM und dergleichen bereitgestellt sein.

[1436] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen Pub-Sub-Funktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1437] Der Massenspeicher **808** kann einen Hash-Code-Berechner **24802** aufweisen, der Hash-Codes für Themen erzeugen kann. Der Hash-Code-Berechner **24802** kann die Hash-Codes in ein Bloomfilter schreiben, beispielsweise unter Verwendung einer Exklusiv-ODER-Funktion. Hierdurch kann eine Bloomfilter-Themenliste **24804** erzeugt werden. Bei einigen Beispielen kann die IoT-Vorrichtung **24800** als ein Veröffentlichender oder Router wirken. Bei diesen Beispielen kann die Bloomfilter-Themenliste **24804** von anderen Veröffentlichern oder Routern erhalten werden, beispielsweise in der Fog **812** oder in der Cloud **302**.

[1438] Eine Whitelist-Maske **24806** kann aufgenommen werden, um von einem Administrator als für die Weiterverteilung akzeptabel identifizierte Themen zu speichern. Eine Blacklist-Maske **24808** kann aufgenommen werden, um Themen zu speichern, die als für die Weiterverteilung nicht akzeptabel identifiziert wurden.

[1439] Ein Subskriptions-Manager **24810** kann aufgenommen sein, um die Bloomfilter-Themenliste **24804** bei Routern und anderen Vorrichtungen in der Fog **812** oder der Cloud **302** zu registrieren. Falls die IoT-Vorrichtung **24800** als Router oder Veröffentlichender wirkt, kann der Subskriptions-Manager **24810** feststellen, ob Themen in der Bloomfilter-Themenliste **24804** in der Whitelist-Maske **24806** oder in der Blacklist-Maske **24808** sind, um festzustellen, ob der Inhalt weitergeleitet werden sollte, wie mit Bezug auf **Fig. 247** beschrieben.

[1440] Ein Inhaltslokalisierer **24812** kann aufgenommen sein, um Inhalt in Zusammenhang mit dem Thema zu lokalisieren und bereitzustellen. Beispielsweise kann Inhalt von anderen Veröffentlichern oder Routern bereitgestellt werden und durch den Inhaltslokalisierer **24812** gespeichert werden, um eine Feststellung zu ermöglichen, ob der Inhalt in der Whitelist **24806** oder der Blacklist **24808** ist, bevor der Inhalt anderen Vorrichtungen in der Fog **812** oder der Cloud **302** bereitgestellt wird.

[1441] **Fig. 249** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **24900**, das Code aufweist, um einen Prozessor **902** anzuweisen, ein Pub-Sub-System unter Verwendung von Bloomfiltern zur Inhaltsverteilung zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **24900** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **24900** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1442] Das nichtflüchtige maschinenlesbare Medium **24900** kann einen Code **24902** aufweisen, um den Prozessor **902** anzuweisen, eine Bloomfilter-Themenliste zu erzeugen, beispielsweise durch Berechnen eines

Hash-Codes von jedem von der Anzahl von Themen und anschließendes Überschreiben eines Bloomfilters mit den Hash-Codes.

[1443] Ein Code **24904** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Whitelist-Maske, eine Blacklist-Maske oder beide bei Routern in einem IoT-Netz zu registrieren. Bei einigen Beispielen kann der Code **24904** die Whitelist-Maske, die Blacklist-Maske oder beide von einer anderen Vorrichtung für die Verwendung durch den Prozessor bei der Feststellung, ob Inhalt weiterzuleiten ist, akzeptieren.

[1444] Ein Code **24906** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Subskriptions-Bloomfilter beim Router zu registrieren. Der Code **24906** kann den Prozessor **902** anweisen, ein Subskriptions-Bloomfilter von einer anderen Vorrichtung zu akzeptieren, beispielsweise falls das maschinenlesbare Medium **24900** Teil eines Routers ist.

[1445] Ein Code **24908** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Inhaltsschnitt des Inhaltsfilters mit dem Subskriptions-Bloomfilter zu berechnen, um festzustellen, ob der Inhalt am Netz zugänglich ist. Ein Code **24910** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Schnitt des Inhaltsschnitts mit einer Whitelist-Maske zu berechnen, um festzustellen, ob der Inhalt zulässig ist. Ein Code **24912** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Schnitt des Inhaltsschnitts mit einer Blacklist-Maske zu berechnen, um festzustellen, ob der Inhalt verboten ist.

[1446] Ein Code **24914** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Inhalt zu einem Subskribenten weiterzuleiten, falls der Inhalt beispielsweise von der Whitelist-Maske autorisiert wurde und von der Blacklist-Maske nicht verboten wurde. Falls eine dieser Bedingungen wahr ist, kann der Code **24914** den Inhalt löschen.

[1447] Eine Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **24900** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Boot-/Messeinrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1448] Bei einem gegebenen Pub-Sub-Netz, das aus einem Satz von Veröffentlichern, Subskribenten und Weiterleitungsknoten besteht, können es Veröffentlichern wünschen, vertraulichen Inhalt in die Themenbenachrichtigungsnachricht aufzunehmen. Der Inhalt kann mit einem Themenverschlüsselungsschlüssel, der ein Gruppen- oder geteilter Schlüssel sein kann, verschlüsselt sein. Eine Herausforderung für diesen Verwendungsfall besteht darin, dass Subskribenten nach dem Empfang der Themenbenachrichtigung den Inhaltsverschlüsselungsschlüssel erhalten müssen, bevor sie den Inhalt verbrauchen können.

[1449] Das Pub-Sub-Netz kann zur Auslieferung von Schlüsselbehandlungs-Benachrichtigungsnachrichten verwendet werden, die mit der Netzdynamik skalieren, weil die Weiterleitungsknoten als Schlüsselbehandlungsknoten wirken können. Schlüsselbehandlungsthemen können automatisch erzeugt und ausgeliefert werden, beispielsweise wenn ein ursprüngliches Thema verschlüsselten Inhalt enthält, was die Notwendigkeit eines Schlüssels signalisiert. Nach dem Empfang eines verschlüsselten Inhalts geben Subskribenten eine Schlüsselbehandlungs-GET-Anforderung aus, um den Verschlüsselungsschlüssel zu erhalten. Der Weiterleitungsknoten sieht dies voraus und subskribiert das Schlüsselbehandlungsthema, das den Verschlüsselungsschlüssel vorab abrufen.

[1450] Fig. 250 ist ein schematisches Diagramm einer Themenbenachrichtigung mit verschlüsseltem Inhalt gemäß einigen Ausführungsformen. In Fig. 250 können Weiterleitungsknoten **25002** Themenschlüssel in der Art von K_{T_1} cachen, so dass sie einer Gemeinschaft von Subskribenten **25004**, die vom Weiterleitungsknoten **25002** bedient wird, lokal verfügbar sind. Das Schlüsselbehandlungsthema benachrichtigt Weiterleitungsknoten **25002**, die als Subskribenten wirken, über die Schlüsselbehandlungsthemen.

[1451] Fig. 251 (A) ist ein schematisches Diagramm einer Gruppe von Routern, die Benachrichtigungen über ein Thema empfangen, das verschlüsselten Inhalt aufweist, gemäß einigen Ausführungsformen. Ein Weiterleitungsknoten **25102** mit dem in seinem Cache enthaltenen Schlüssel K_{T_1} kann durch Veröffentlichen des Schlüsselbehandlungsthemas $T[K_{T_1}]$ **25106** für das Thema **T1 25108** auf eine Subskription **25104** antworten. Eine Themenbenachrichtigung **T1 25108** kann den verschlüsselten Inhalt C aufweisen, wobei der Inhaltsverschlüsselungsschlüssel K_{T_1} ist. Der Empfang der Themenbenachrichtigung **T1 25108** kann einen Router **25102**, **25110**, **25112** oder **25114** veranlassen, das Thema $T[K_{T_1}]$ **25106** zu definieren.

[1452] Der Router **25110**, der das Thema $T[T_1]$ subskribiert **25104**, kann auf den Empfang eines Schlüsselbehandlungsereignisses für dieses Thema warten. Der Veröffentlicher P **25116** des Themas T_1 **25108** kann den Schlüssel K_{T_1} dem Router **25102** zur Verfügung stellen, welcher als Schlüssel-Cache-Manager für andere Router im System wirken kann. Nach dem Empfang der Schlüssels benachrichtigt der Weiterleitungsknoten **25102** seine Subskribenten über die Verfügbarkeit des Schlüssels K_{T_1} .

[1453] Fig. 251 (B) ist ein schematisches Diagramm einer Gruppe von Routern, welche ihre Caches in Erwartung von Subskribenten, die ein verschlüsseltes Thema anfordern, vorwärmen, gemäß einigen Ausführungsformen. Die Subskribenten des Weiterleitungsknotens **25102** in der Art des Weiterleitungsknotens **25110** können ihren Cache **25120** vorwärmen, beispielsweise indem sie präemptiv den Schlüssel K_{T_1} unter Verwendung einer Schlüssel-GET-Anforderung **25122** erhalten. Dies kann in Erwartung darauf geschehen, dass nachgeschaltete Router in der Art des Weiterleitungsknotens **25112** und Subskribentenknoten in der Art eines Subskribentenknotens **25118** durch Ausgabe einer Schlüssel-GET-Anforderung **25124** auf die T_1 -Benachrichtigung **25108** reagieren. Es sei bemerkt, dass Inhaltsverschlüsselungsschlüssel ferner durch einen seitenspezifischen Schlüssel oder durch eine VPN-Sitzung, die den Themenschlüssel über Hops zwischen Weiterleitungsknoten und Subskribentenknoten schützt, verschlüsselt werden können.

[1454] Fig. 252 ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **25200** zur Verwendung einer Schlüsselbehandlungsbenachrichtigung und zum Vorwärmen des Schlüssel-Cachings gemäß einigen Ausführungsformen. Das Verfahren **25200** aus Fig. 252 kann durch die mit Bezug auf Fig. 253 beschriebene IoT-Vorrichtung **25300** implementiert werden. Das Verfahren kann in Block **25202** beginnen, beispielsweise wenn sich eine IoT-Vorrichtung einem Weiterleitungsnetz zur Inhaltsverteilung anschließt, welches Bloomfilter zur Inhaltsverteilung verwenden kann. In Block **25204** erzeugt ein Veröffentlicher Inhalt (C) und einen Inhaltsverschlüsselungsschlüssel (K_{T_1}). Der verschlüsselte Inhalt $E = \{C\}K_{T_1}$ kann dann zum Herunterladen von einem öffentlichen Ablagespeicher verfügbar sein.

[1455] In Block **25206** kann der Veröffentlicher den Inhalt unter einem Thema T_1 , das Pub-Sub-Subskribenten hat, verfügbar machen. In Block **25208** kann der Veröffentlicher einen ersten Weiterleitungsknoten (R1) über T_1 benachrichtigen. Der Weiterleitungsknoten (R1) kann ein Bloomfilter bilden, das verfügbare veröffentlichte Themen aufweist. Der Weiterleitungsknoten (R1) weist ein Tag auf, das angibt, dass der verschlüsselte Inhalt (E) verfügbar ist.

[1456] In Block **25210** empfängt ein zweiter Weiterleitungsknoten (R2) mit Subskriptionen für T_1 die Themenbenachrichtigung vom ersten Weiterleitungsknoten (R1), einschließlich des E-Tags. In Block **25212** bildet der erste Weiterleitungsknoten (R1) ein Schlüsselbehandlungsthema $T[K_{T_1}]$ zur Benachrichtigung des zweiten Weiterleitungsknotens (R2) über die Verfügbarkeit des Schlüssels K_{T_1} .

[1457] In Block **25214** subskribiert der zweite Weiterleitungsknoten (R2) nach dem Empfang der T_1 -Benachrichtigung mit dem E-Tag das Schlüsselbehandlungsthema $T[K_{T_1}]$. Ferner können die T_1 -Benachrichtigung und das Schlüsselbehandlungsthema durch aufeinander folgende Router in der Kette verbreitet werden.

[1458] In Block **25216** führt der Veröffentlicher den Themenverschlüsselungsschlüssel K_{T_1} dem ersten Weiterleitungsknoten zu. Nach dem Empfang des Themenverschlüsselungsschlüssels werden alle Subskribenten von T_1 benachrichtigt.

[1459] In Block **25218** fordert, wenn ein Subskribent (S) des Themas T_1 E unter Verwendung des Themenverschlüsselungsschlüssels K_{T_1} entschlüsseln möchte, der Subskribent K_{T_1} von einem als Schlüssel-Cache-Manager wirkenden Router an. Der Schlüssel-Cache-Manager für den Subskribenten kann der nächstgelegene Router in der Kommunikation mit dem Subskribenten sein oder ein anfänglicher Router sein, der die Schlüssel-Cache-Behandlungsdienste für die gesamte Gruppe von Routern bereitstellt.

[1460] In Block **25220** wird festgestellt, ob der Themenverschlüsselungsschlüssel im Cache ist. Falls dies nicht der Fall ist, fordert in Block **25222** der als Schlüssel-Cache-Manager wirkende Router den Themenverschlüsselungsschlüssel von einem Peer-Knoten an. Der Prozessfluss kehrt dann zu Block **25220** zurück, um festzustellen, ob der Themenverschlüsselungsschlüssel nun im Cache ist. Falls in Block **25220** festgestellt wird, dass der Themenverschlüsselungsschlüssel im Cache ist, wird der Prozessfluss mit Block **25224** fortgesetzt, wo der Themenverschlüsselungsschlüssel zum Antragsteller gesendet wird, wie dem Subskribenten bei diesem Beispiel.

[1461] In Block **25226** wird der verschlüsselte Inhalt E unter Verwendung des Themenverschlüsselungsschlüssels K_{T1} entschlüsselt. Das Verfahren **25200** endet dann in Block **25228**.

[1462] **Fig. 253** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **25300** zum Verwalten der Themenbenachrichtigung mit verschlüsseltem Inhalt vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3** und **Fig. 8** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **25300** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1463] Die IoT-Vorrichtung **25300** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group 2009 als ISO/IEC 11889 herausgegebenen Spezifikation kompatibel ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datenssegment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1464] Das TPM kann verwendet werden, um eine TEE oder eine sichere Enklave für das Ausführen von Programmen festzulegen. Das TPM kann auch für eine Anzahl anderer Funktionen, einschließlich der Bereitstellung einer kryptographischen Unterstützung für sichere Kommunikationen und Schlüssel für die Identifikation verwendet werden. Das TPM kann in stärker beschränkten Vorrichtungen in der Art von Sensoren ganz am Rand der IoT-Netze nicht vorhanden sein. Bei diesen Vorrichtungen kann Sicherheit durch eine Blockchain selbst, durch vorgeschaltete Vorrichtungen, durch ein virtuelles TPM und dergleichen bereitgestellt sein.

[1465] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen verschlüsselten Inhaltsverteilungsfunktionen aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1466] Der Massenspeicher **808** kann einen Themenklassifizierer **25302** aufweisen, der Themen identifizieren kann, die verschlüsselten Inhalt aufweisen. Der Themenklassifizierer **25302** kann eine Bloomfilter-Themenliste für verfügbare Themen, einschließlich Themen, die verschlüsselten Inhalt und nicht verschlüsselten Inhalt enthalten, schreibezeugen.

[1467] Ein Benachrichtiger **25304** kann andere Vorrichtungen in der Fog **812** über das Thema benachrichtigen, das verschlüsselten Inhalt aufweist. Ein Schlüssel-Subskribent **25306** kann aufgenommen sein, um ein Thema, das einen Themenschlüssel **25308** für verschlüsselten Inhalt **25310** aufweist, zu subskribieren. Der Schlüssel-Subskribent **25306** kann den verschlüsselten Inhalt **25310** von Vorrichtungen in der Fog **812** in der Art von Veröffentlichen oder Routern ziehen oder empfangen und den verschlüsselten Inhalt **25310** anderen Vorrichtungen in der Fog **812** in der Art von Routern oder Subskribenten bereitstellen. Ein Entschlüsseler **25312** kann aufgenommen sein, um verschlüsselten Inhalt zu entschlüsseln, beispielsweise unter Verwendung des Themenschlüssels **25308**.

[1468] **Fig. 254** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **25400**, das Code aufweist, um einen Prozessor **902** anzuweisen, eine Themenbenachrichtigung mit verschlüsseltem Inhalt zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **25400** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **25400** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1469] Das nichtflüchtige maschinenlesbare Medium **25400** kann einen Code **25402** aufweisen, um den Prozessor **902** anzuweisen, eine Benachrichtigung über verschlüsselten Inhalt zu empfangen. Die Benachrichtigung kann vom Veröffentlicher, einem primären Router in Kontakt mit dem Veröffentlicher oder anderen Routern in Kontakt mit dem primären Router gesendet werden.

[1470] Ein Code **25404** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Bloomfilter von verfügbarem Inhalt zu bilden. Das Bloomfilter kann Hash-Codes für Themen, die verschlüsselten Inhalt aufweisen, sowie Hash-Codes für Themen, die keinen verschlüsselten Inhalt aufweisen, aufweisen.

[1471] Ein Code **25406** kann aufgenommen sein, um den Prozessor **902** anzuweisen, die Themenbenachrichtigung zu Routern zu senden. Die Themenbenachrichtigung kann die Information aufweisen, dass das Thema verschlüsselten Inhalt hat.

[1472] Ein Code **25408** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein Thema nach dem Empfang der Themenbenachrichtigung, einschließlich beispielsweise eines Schlüsselbehandlungsthemas, zu abonnieren. Ein Code **25410** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Teilnehmer über die Verfügbarkeit eines Schlüssels für den verschlüsselten Inhalt zu benachrichtigen.

[1473] Ein Code **25412** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Schlüssel von einem Peer-Knoten zu erhalten, beispielsweise falls der Abonnent ein Router ist. Der Code **25412** kann den Prozessor **902** anweisen, den Schlüssel von einem Router, der in Kommunikation mit einem Abonnenten steht, zu erhalten.

[1474] Ein Code **25414** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Schlüssel zu einem Abonnenten zu senden, wobei der Abonnent einen Router oder einen Endverbraucher des Inhalts einschließen kann. Der Code **25414** kann den Prozessor anweisen, den Inhalt zu entschlüsseln, falls der Abonnent beispielsweise ein Verbraucher des Inhalts ist.

[1475] Eine Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **25400** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Boot-/Messeinrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1476] Zusätzlich zur Themenniveauverschlüsselung zum Schützen eines Kontakts können Themen innerhalb einer Gruppe von Veröffentlicern und Abonnenten datenschutzsensitiv sein. Um das erhöhte Schutzniveau zu erreichen, kann ein Gruppenschlüssel zum Verschlüsseln von Themen verwendet werden. Der Gruppenschlüssel kann eine doppelte Funktion als Inhaltsverschlüsselungsschlüssel erfüllen oder ein zweiter Gruppenschlüssel sein. Die Verteilung des Gruppenschlüssels kann nach dem mit Bezug auf **Fig. 252** beschriebenen Verfahren **25200** sowie durch andere Techniken ausgeführt werden.

[1477] **Fig. 255** ist ein schematisches Diagramm eines Abonnenten **25502**, der einen Themengruppenschlüssel **25504** erhält, gemäß einigen Ausführungsformen. Der Themengruppenschlüssel **25504** kann erhalten werden, indem der Abonnent **25502** zuerst in eine von einem Themenbenennungsserver **25506** verwaltete Themengruppe eingeschrieben wird, beispielsweise indem der Abonnent **25502** veranlasst wird, eine Beitrittsanforderung **25508** zum Themenbenennungsserver **25506** zu senden, um sich einem Thema anzuschließen oder dieses zu erzeugen. Der Themenbenennungsserver **25506** kann dann eine Nachricht **25510** ausgeben, die einen Gruppenmitgliedschafts-Berechtigungsnachweis aufweist. Ähnlich können Veröffentlicher **25512** der Gruppe beitreten, indem sie eine Beitrittsanforderung **25514** senden und eine Nachricht **25516** vom Themenbenennungsserver **25516** empfangen, welche die Gruppenmitgliedschafts-Berechtigungsnachweise in der Art des Themengruppenschlüssels **25504** aufweist.

[1478] Der Veröffentlicher **25512** und der Abonnent **25502** können sich dann als Gruppenmitglieder authentifizieren. Sobald die Authentifizierung erfolgreich war, kann der Themenbenennungsserver **25506** eine sichere Sitzung einleiten, um den Themengruppenschlüssel **25504** an die Mitglieder zu verteilen.

[1479] **Fig. 256** ist ein schematisches Diagramm eines Veröffentlichers, der ein Subskriptions-Bloomfilter **25602** zur Benachrichtigung von Abonnenten über verfügbare Themen erzeugt, gemäß einigen Ausführungsformen. Das Bloomfilter-Benachrichtigungssystem kann wie mit Bezug auf **Fig. 244** beschrieben funktionieren. Sobald die Veröffentlicher und Abonnenten Themengruppenschlüssel **25604** zum Verschlüsseln von

Themen **25606** besitzen, können die Themen **25606** von den Veröfentlichern verschlüsselt werden, um verschlüsselte Themen **25608** zu bilden. Nicht private Themen **25610** können in einer Hash-Funktion **25612** mit den verschlüsselten Themen **25608** kombiniert werden, um das Benachrichtigungs-Bloomfilter **25602** zu bilden. Ähnlich können Subskribenten ein Subskriptions-Bloomfilter durch Verschlüsseln eines interessierenden Themas und anschließendes Verwenden des verschlüsselten Werts als Eingabe in das Bloomfilter berechnen.

[1480] Fig. 257 ist ein Leiterdiagramm eines beispielhaften Verfahrens **25700** zur Themenverschlüsselung gemäß einigen Ausführungsformen. Das Verfahren **25700** aus Fig. 257 kann durch die mit Bezug auf Fig. 260 beschriebene IoT-Vorrichtung 26000 implementiert werden. Die Themenverschlüsselungsschlüssel können von einem Schlüsselverteilungszentrum (KDC) **25702** behandelt werden, das eine Liste bekannter Themen von einem Themennamensgebungsdienst (TNS) empfängt, wie beispielsweise mit Bezug auf Fig. 255 beschrieben wurde. Das **25702** kann ein vom TNS erteiltes Zertifikat oder einen davon erteilten Attestierungsschlüssel **25704** verwenden, um zu verifizieren, dass Subskribenten in der Art eines Subskribenten **25706** und Veröfentlichlicher in der Art eines Veröfentlichers **25708** Mitglieder der Themengruppe sind, bevor der Themengruppenschlüssel **25710** bereitgestellt wird.

[1481] Das TNS kann einen Erhöhter-Datenschutz-Kennung bzw. Enhanced-Privacy-ID (EPID)-Schlüssel als Themengruppenschlüssel **25710** und ein EPID-Beitrittsprotokoll zum Aufnehmen von Mitgliedern verwenden. Das KDC **25702** kann ein signiertes Diffie-Hellman-Protokoll zum Einrichten eines sicheren Kanals zum Subskribenten **25704** oder zum Veröfentlichlicher **25706** verwenden, wenn der Themengruppenschlüssel **25704** verteilt wird. Der Themengruppenschlüssel **25704** kann ein symmetrischer Schlüssel sein.

[1482] Das Verfahren **25700** kann beginnen, wenn der Veröfentlichlicher **25708** den Themenverschlüsselungsschlüssel **25710** erzeugt **25712**. Der Veröfentlichlicher **25708** verschlüsselt dann den Themenverschlüsselungsschlüssel **25710** mit einem vom KDC **25702** bereitgestellten Schlüssel und übergibt den Themenverschlüsselungsschlüssel **25710** in einer Attestierungsnachricht **25714** an das KDC **25702**.

[1483] Der Veröfentlichlicher **25708** kann das Thema dann zusammen mit dem Inhalt, der unter Verwendung des Themenverschlüsselungsschlüssels **25710** verschlüsselt ist, einem Router **25718** bekannt machen **25716**. Der Subskribent **25706** kann eine Subskriptionsnachricht **25720** zum Router **25718** senden, worin sich beispielsweise ein Bloomfilter befindet, das Hash-Codes von Themen aufweist, welche der Subskribent **25706** zu empfangen wünscht.

[1484] Nach dem Empfang der veröffentlichten Themennachricht **25716** kann der Router **25718** feststellen, dass der Inhalt vom Subskribenten **25706** angefordert wurde. Der Router **25718** kann dann eine Benachrichtigungsnachricht **25722**, die den verschlüsselten Inhalt aufweist, zum Subskribenten **25706** senden. Der Subskribent **25706** kann dann eine Attestierungsnachricht **25724** mit einer Anforderung, den Themenverschlüsselungsschlüssel **25710** zu erhalten, zum KDC **25702** senden.

[1485] Das KDC **25702** kann dann eine Nachricht **25726**, welche den Themenverschlüsselungsschlüssel **25710** aufweist, welche beispielsweise mit einem Kommunikationsschlüssel verschlüsselt ist, senden. Der Subskribent **25706** kann den Inhalt dann zur Verwendung entschlüsseln **25728**.

[1486] Zusätzlicher Inhalt kann für das Thema unter Verwendung desselben Themenverschlüsselungsschlüssels **25710** bereitgestellt werden. Beispielsweise kann der Veröfentlichlicher **25708** eine Nachricht **25730** zum Router **25718** senden, welche den zusätzlichen verschlüsselten Inhalt aufweist. Der Router **25718** kann dann eine Benachrichtigungsnachricht **25732**, welche den zusätzlichen verschlüsselten Inhalt aufweist, zum Subskribenten **25706** senden. Der Subskribent **25706** kann den zusätzlichen Inhalt dann zur Verwendung entschlüsseln **25734**.

[1487] IoT-Netze werden häufig entsprechend einer Klassifikation von Sicherheit, Privatheit, Integrität oder Sicherung partitioniert. Ein Mehrniveau-Sicherheitsetikett kann verwendet werden, um Mehrdeutigkeiten in der Klassifikation zu beseitigen. Es kann einen Satz von Themen oder Kategorien geben, die zu jedem Niveau gehören. Ein Bloomfiltermechanismus, wie beispielsweise mit Bezug auf Fig. 244 beschrieben, kann verwendet werden, um Benachrichtigungen, die das Sicherheitsetikett aufweisen, auszuliefern. Das Sicherheitsetikett kann dann von Weiterleitungsknoten und anderen Subskribenten und Veröfentlichern verfolgt werden. Die Mehrniveau-Sicherheitsetikette können auf einer Anzahl verschiedener Modelle beruhen, einschließlich beispielsweise unter anderem auf dem Biba-Integritätsmodell und dem Bell-LaPadula-Sicherheitsmodus.

[1488] **Fig. 258** ist ein schematisches Diagramm der Verwendung von Mehrniveau-Sicherheitsetiketten in einer Veröffentlichen-Subskribieren-Umgebung gemäß einigen Ausführungsformen. Die Figur zeigt die Verwendung eines Biba-Integritätsmodells und eines Bell-LaPadula-Vertraulichkeitsmodells. Bei einem Biba-Integritätsmodell gibt es im Allgemeinen keine erlaubten Hochschreibungen (Randbedingung I) auf ein höheres Niveau, beispielsweise von L1 zu L2, und kein erlaubtes Herunterlesen (Randbedingung II) von einem niedrigeren Niveau, beispielsweise von L0 zu L1. Bei einem Bell-LaPadula-Vertraulichkeitsmodell gibt es im Allgemeinen kein erlaubtes Herunterschreiben (Randbedingung III) zu einem niedrigeren Niveau, beispielsweise von L1 zu L0, und kein erlaubtes Hochlesen (Randbedingung IV) zu einem höheren Niveau, beispielsweise von L0 zu L1.

[1489] Ein Veröffentlicher **25802** kann das Sicherheitsetikett als Bloomfilter durch Zuordnen von Etikettkategorien zu Bloom-Themen C_x codieren. Das Etikettniveau kann selbst ein Thema sein, das vorhanden ist, wenn jegliche der Etikettkategorien zugeführt werden. Es kann geeignet sein, um das Niveauthema mit jedem der Kategoriethemen zu codieren, um zu gewährleisten, dass eine Kategorie eines anderen Niveaus nicht mit der Kategorie eines ersten Niveaus verwechselt wird. Die Codierung kann beispielsweise durch einen kryptographischen Hash der beiden Werte oder durch Anwenden irgendeiner Funktion $f()$, so dass der Ausgangswert nicht mit einem der Eingangswerte kollidiert, erreicht werden. Beispielsweise kann ein Bitmuster, das ein Niveau repräsentiert, mit dem Bitmuster für ein Thema durch Ausführen einer Exklusiv-ODER-Operation der Bitmuster verarbeitet werden. Die restlichen Bitmuster können dann im Bloomfilter verwendet werden.

[1490] Weiterleitungsknoten **25804** wenden die Sicherheitsrichtliniensemantik durch Erkennen des Sicherheitsniveauthemas und anschließendes Anwenden der geeigneten Sicherheitsmodell-Verhaltensrandbedingung an. Falls beispielsweise einer Randbedingung I gefolgt wird, kann ein Router erlauben, dass ein Subskribent S0 autorisiert wird, auf einem Niveau L0 zu arbeiten, um eine Benachrichtigung von einem Veröffentlicher zu empfangen, der autorisiert ist, auf einem Niveau L1 zu arbeiten. Ähnlich würde, falls ein Subskribent S2 autorisiert ist, auf dem Niveau L2 zu arbeiten, die Benachrichtigung vom L1-Veröffentlicher blockiert werden.

[1491] Die in **Fig. 258** dargestellten Mehrniveau-Sicherheitsrichtlinien sind nicht ausschließlich. Andere Mehrniveau-Sicherheitsrichtlinien können verwendet werden. Ferner sei bemerkt, dass, wenngleich die hier verwendeten Beispiele die Anwendung der Mehrniveau-Sicherheitsrichtlinien auf mehrere Ebenen in einem Netz beschreiben, die Sicherheitsniveaus als verschiedene Sicherheitsniveaus in einer einzigen Netzebene sowie durch andere Definitionen definiert werden können.

[1492] **Fig. 259** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **25900** zur Implementation von Bloomfiltern für die Anwendung von Mehrniveau-Sicherheitsrichtlinien auf Benachrichtigungsnachrichten gemäß einigen Ausführungsformen. Das Verfahren **25900** aus **Fig. 259** kann durch die mit Bezug auf **Fig. 260** beschriebene IoT-Vorrichtung 26000 implementiert werden. Das Verfahren **25900** kann in Block **25902** beginnen, beispielsweise wenn ein Subskribent Inhalt fordert, wobei ein Veröffentlicher Inhalt zum Teilen hat. In Block **25904** wird festgestellt, ob dies eine Veröffentlichung ist. Falls dies nicht der Fall ist, kann die Aktivität die Registrierung einer Subskription sein, und der Prozessfluss wird in Block **25906** fortgesetzt.

[1493] In Block **25906** attestiert ein Subskribent seine Identität Router-Knoten und offenbart das Sicherheitsniveau, bei dem die Subskription registriert ist. In Block **25908** führt der Subskribent ein Bloomfilter einschließlich des interessierenden Inhalts zu. Wie hier erörtert, kann das Bloomfilter überschriebene Bit-Hash-Codes unter anderem für interessierende Themen, Kategorien und Sicherheitsniveaus aufweisen.

[1494] In Block **25910** wird festgestellt, ob der Router eine Integritätsrichtlinie erzwingt. Falls dies der Fall ist, kann der Router in Block **25912** Filterwerte, die ein Herunterlesen zu niedrigeren Netzebenen ermöglichen, fort maskieren. In Block **25914** wird festgestellt, ob der Router eine Vertraulichkeitsrichtlinie erzwingt. Falls dies der Fall ist, kann der Router in Block **25916** Filterwerte, die ein Herauflesen zu höheren Netzebenen ermöglichen, fort maskieren. In Block **25918** wird die Subskription beim Router registriert. Das Verfahren **25900** endet dann in Block **25920**.

[1495] Falls in Block **25904** festgestellt wird, dass die Aktivität eine Veröffentlichung ist, wird der Prozessfluss in Block **25922** fortgesetzt. In Block **25922** attestiert der Veröffentlicher dem Router-Knoten das Sicherheitsniveau, bei dem die Veröffentlichung vorgenommen wird. In Block **25924** werden die Sicherheitsniveaus in Kategorien in ein dem Veröffentlichungsinhalt entsprechendes Bloomfilter codiert. Wie hier beschrieben, kann das Bloomfilter beispielsweise öffentliche Themen, private Themen, Schlüsselbehandlungsthemen, Sicherheitsniveauthemen und andere Themen zusätzlich zur aktuellen Veröffentlichung aufweisen.

[1496] In Block **25926** wird festgestellt, ob der Router eine Integritätsrichtlinie erzwingt. Falls dies der Fall ist, kann der Router in Block **25928** Filterwerte, die ein Heraufschreiben zu höheren Netzebenen ermöglichen, fort maskieren. In Block **25930** wird festgestellt, ob der Router eine Vertraulichkeitsrichtlinie erzwingt. Falls dies der Fall ist, kann der Router in Block **25932** Filterwerte, die ein Herunterschreiben zu niedrigeren Netzebenen ermöglichen, fort maskieren. In Block **25934** wird die veröffentlichte Benachrichtigung zum Router, zum Subskribenten oder zu beiden gesendet. Das Verfahren **25900** endet dann in Block **25920**.

[1497] **Fig. 260** ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer IoT-Vorrichtung **26000** zum Verwalten der Themenbenachrichtigung mit verschlüsseltem Inhalt vorhanden sein können, gemäß einigen Ausführungsformen. Die IoT-Vorrichtung **26000** kann als ein Veröffentlichender, ein Router oder ein Subskribent in einem IoT-Netz in der Art der Fog **812** wirken. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die **Fig. 3**, **Fig. 8**, **Fig. 255** und **Fig. 257** beschrieben wurden. Es sei bemerkt, dass für die IoT-Vorrichtung **26000** andere Komponenten als für die mit Bezug auf **Fig. 8** erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt und verwendet werden können.

[1498] Die IoT-Vorrichtung **26000** kann ein vertrauenswürdiges Plattformmodul bzw. Trusted Platform Module (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datenssegment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1499] Das TPM kann verwendet werden, um eine TEE oder eine sichere Enklave für das Ausführen von Programmen festzulegen. Das TPM kann auch für eine Anzahl anderer Funktionen, einschließlich der Bereitstellung einer kryptographischen Unterstützung für sichere Kommunikationen und Schlüssel für die Identifikation verwendet werden.

[1500] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der hier beschriebenen verschlüsselten Inhaltsverteilungsfunktion aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1501] Der Massenspeicher **808** kann einen Themenbenennungsserver **25506**, wie mit Bezug auf **Fig. 255** beschrieben, aufweisen. Wie beschrieben, kann der Themenbenennungsserver **25506** Themen in der Art des Erzeugens von Themengruppen und des Ausgebens von Schlüsseln für Themengruppen, Sicherheitsniveaus und dergleichen behandeln. Der Themenbenennungsserver **25506** kann eine beliebige Anzahl von Techniken zur Erzeugung des Schlüssels, einschließlich hier beschriebener Techniken, verwenden, wie das Zusammenstellen zirkularer fraktioneller Schlüssel, eine EPID-Schlüsselerzeugung, eine Blockchain-Schlüsselerzeugung und dergleichen.

[1502] Ein Subskribent **26002** kann ein interessierende Kategorien, Themen und Sicherheitsniveaus sowie andere Elemente enthaltendes Bloomfilter Routern und anderen Vorrichtungen in der Art von Veröffentlichern zuführen. Ein Attestierer **26004** kann die Identifikation des Veröffentlichers oder Subskribenten, beispielsweise für ein Schlüsselverteilungszentrum **25702**, wie mit Bezug auf **Fig. 257** beschrieben, attestieren. Das Schlüsselverteilungszentrum **25702** kann sich in einer anderen Vorrichtung, beispielsweise in der Fog **812** oder der Cloud **302** befinden. Das Schlüsselverteilungszentrum **25702** kann die Identitäten von Vorrichtungen in der Fog **812** oder der Cloud **302** bestätigen und anderen Vorrichtungen Themenschlüssel, Niveauschlüssel oder beides bereitstellen. Der Subskribent **26002** kann eine Benachrichtigung über interessierenden Inhalt empfangen und die vom Schlüsselverteilungszentrum **25702** empfangenen Schlüssel verwenden, um den interessierenden Inhalt zu entschlüsseln.

[1503] Ein Integritätserzwinger **26006** kann Filterwerte, die Herunterleseoperationen zu niedrigeren Sicherheitsniveaus oder Netzebenen, Heraufschreiboperationen zu höheren Sicherheitsniveaus oder Netzebenen oder beides ermöglichen, fort maskieren. Ein Vertraulichkeitserzwinger **26008** kann Filterwerte, die Heraufleseoperationen zu höheren Sicherheitsniveaus oder Netzebenen, Herunterschreiboperationen zu niedrigeren Sicherheitsniveaus oder Netzebenen oder beides ermöglichen, fort maskieren.

[1504] **Fig. 261** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **26100**, das Code aufweist, um einen Prozessor **902** anzuweisen, eine Themenbenachrichtigung mit verschlüsseltem Inhalt zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **26100** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **26100** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1505] Das nichtflüchtige maschinenlesbare Medium **26100** kann einen Code **26102** aufweisen, um den Prozessor **902** anzuweisen, einen Verschlüsselungsschlüssel für ein Thema, ein Sicherheitsniveau oder beides zu erzeugen. Ein Code **26104** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Verschlüsselungsschlüssel zu einem Schlüsselverteilungszentrum zu übertragen. Das Schlüsselverteilungszentrum kann den Schlüssel dann Vorrichtungen bereitstellen, die dem Schlüsselverteilungszentrum eine Attestierung zur Bestätigung der Identifikation bereitstellen. Die Vorrichtungen können unter anderem Veröffentlicher, Router und Subskribenten einschließen.

[1506] Ein Code **26106** kann aufgenommen sein, um den Prozessor **902** anzuweisen, ein verschlüsseltes Thema beispielsweise einem Router bekanntzumachen. Ein Code **26108** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Benachrichtigung über das Thema zu anderen Vorrichtungen in der Art von Routern, Subskribenten oder beiden zu senden. Die Themenbenachrichtigung kann die Information aufweisen, dass das Thema verschlüsselten Inhalt hat.

[1507] Ein Code **26110** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Verschlüsselungsschlüssel vom Schlüsselverteilungszentrum zu erhalten, beispielsweise durch Senden einer Attestierungsnachricht, die den Verschlüsselungsschlüssel anfordert. Ein Code **26112** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Inhalt unter Verwendung des Verschlüsselungsschlüssels zu entschlüsseln.

[1508] Ein Code **26114** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Integritätsrichtlinie zu erzwingen, beispielsweise Filterwerte fort zu maskieren, die ein Herunterlesen zu einem niedrigeren Sicherheitsniveau oder einer niedrigeren Netzebene oder ein Heraufschreiben zu einem höheren Sicherheitsniveau oder einer höheren Netzebene ermöglichen. Ein Code **26116** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Vertraulichkeitsrichtlinie zu erzwingen, beispielsweise Filterwerte fort zu maskieren, die ein Herunterlesen zu einem niedrigeren Sicherheitsniveau oder einer niedrigeren Netzebene oder ein Heraufschreiben zu einem höheren Sicherheitsniveau oder einer höheren Netzebene ermöglichen.

[1509] Eine beliebige Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **26100** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Boot-/Mess-einrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1510] Die hier beschriebenen Techniken können zur Implementation einer Anzahl von IoT-Netzen für verschiedene Zwecke verwendet werden. Die folgenden Abschnitte beschreiben zusätzliche Anwendungen, die implementiert werden können.

[1511] Bei einem ersten Beispiel kann ein IoT-Netz verwendet werden, um einen Schwarm von Robotern für unauffällige Arbeitsvorgänge in der Art einer Reinigung, Wartung, Müllbeseitigung oder dergleichen zu implementieren. Die Roboter können als scheue Roboter beschrieben werden, weil sie dafür ausgelegt werden können, die Bewegung zu minimieren, wenn sich Personen dem Einsatzbereich nähern oder in die Nähe der Roboter gelangen. Der scheue Roboter kann auf einem autonomen gleichzeitigen Lokalisierungs- und Zuordnungs- bzw. Autonomous-Simultaneous-Localization-and-Mapping(SLAM)-System zur Ausführung von Funktionen unter einer Vielzahl von Modalitäten, einschließlich Vorhandenseins-, Bewegungserkennung und

Interaktion mit anderen scheuen Robotern, Personen und anderen cybernetischen Systemen, um spezifische Ziele zu erreichen, beruhen.

[1512] Fig. 262 ist eine Zeichnung eines Beispiels eines scheuen Roboters **26200** gemäß einigen Ausführungsformen. Bei diesem Beispiel ist der scheue Roboter eine Recyclingtonne oder Mülltonne **26202**, die auch von vorbeigehenden Personen fallen gelassenen Abfall oder Müll aufnehmen kann. Wenngleich ein autonomes Sammeln des Mülls eine Primärfunktion sein kann, kann der scheue Roboter eine Anzahl erweiterter Funktionen und Sekundärfunktionen aufweisen. Beispielsweise kann sich die Mülltonne in einem Bereich, beispielsweise einem Park oder Stadtblock, herum bewegen und einen normalerweise verborgenen Arm zum Aufsammeln von Müll verwenden.

[1513] Ferner können die scheuen Roboter Sekundärfunktionen aufweisen, die durch die enthaltenen Rechen-, Speicher- und Vernetzungsfähigkeiten bereitgestellt werden. Die scheuen Roboter können als Edge- oder Fog-Knoten wirken, um beispielsweise Parkzahlungs-Zugangspunkte, Internetzugangspunkte und dergleichen bereitzustellen. Als Internetzugangspunkte können sie dafür ausgelegt sein, als Proxy-Server zu wirken, die Webseiten, auf die üblicherweise zugegriffen wird, lokal cachen, um die Interneterfahrung von Benutzern zu verbessern. Sie können jegliche Daten, die lokal gesammelt werden, speichern und verarbeiten und diese Daten zur Cloud senden, beispielsweise um als verbesserte Sicherheitsmechanismen für Stadt- oder öffentliche Stellen zu wirken. Dies kann eine Überwachung, Warnung und Abschreckung illegaler oder antisozialer Aktivitäten einschließen. Sie können auch für Spracherkennungskommunikationen ausgelegt werden, beispielsweise Anleitungen bieten, falls jemand die Orientierung verloren hat, Behörden kontaktieren und dergleichen.

[1514] Zur Implementation dieser Funktionen kann der scheue Roboter **26200** mit einer Anzahl verschiedener Systeme ausgerüstet sein. Die Systeme können als ein IoT-Netz in jedem scheuen Roboter **26200** bildende einzelne IoT-Vorrichtungen implementiert sein.

[1515] Der scheue Roboter **26200** kann eine Anzahl von Sensoren **26204** aufweisen. Die Sensoren **26204** können Daten über die Umgebung sammeln, um zu ermöglichen, dass der scheue Roboter Kontextinformationen zusammenstellt und lokale Entscheidungen für Bewegungs- und andere Funktionen trifft.

[1516] Eine Stromquelle **26206** kann verwendet werden, um dem scheuen Roboter **26200** Strom bereitzustellen. Die Stromquelle kann eine Anzahl von Konfigurationen aufweisen, beispielsweise als Batterie, Ladegerät, Solarkollektoren und dergleichen.

[1517] Der scheue Roboter **26200** kann Verarbeitungs- und Analytiksysteme **26208** aufweisen, um zu ermöglichen, dass die Roboter als Fog-Knoten wirken, beispielsweise mit anderen scheuen Robotern und anderen lokalen Vorrichtungen. Dies kann es ihnen ermöglichen, unter anderem Daten und Verarbeitungsdienste anzubieten oder die Ergebnisse einer Analyse für weitere Aktionen zur Cloud oder zu Personen zu senden. Aktionen, die ausgeführt werden können, werden mit Bezug auf Fig. 264 detaillierter erörtert.

[1518] Eine Anzahl einziehbarer Geräte **26210** kann aufgenommen sein, um die Ausführung von Funktionen zu unterstützen. Beispielsweise kann die Mülltonne **26202** einen verborgenen einziehbaren Arm aufweisen, der dafür ausgelegt sein kann, entlang seinem Weg Abfall aufzusammeln. Andere einziehbare Geräte **26210** können verborgene Ladestecker, die sich autonom mit Ladestationen verbinden, Ladestationen für Personenvorrichtungen und dergleichen einschließen.

[1519] Antriebssysteme **26212** können aufgenommen sein, um den scheuen Roboter **26200**, abhängig von Bewegungsregeln, die feststellen, ob sich Personen, Tiere oder andere Hindernisse in der Nähe befinden, zu verschiedenen Orten zu bewegen. Die Antriebssysteme **26212** können abhängig von der Verwendung motorbetriebene Raupen, Räder oder andere Systeme, aufweisen.

[1520] Fig. 263 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einem scheuen Roboter **26200** vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 8 und Fig. 262 beschrieben wurden. Es sei bemerkt, dass für den scheuen Roboter **26200** andere Komponenten ausgewählt und verwendet werden können als jene, die für die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung 800 und andere hier erörterte IoT-Vorrichtungen ausgewählt wurden.

[1521] Der scheue Roboter **26200** kann ein vertrauenswürdiges Plattformmodul (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group 2009 als ISO/IEC 11889 herausgegebenen Spezifika-

tion kompatibel ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Daten-segment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) **26302** durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1522] Die TEE **26302** stellt einen Bereich bereit, worin Prozesse ausgeführt werden können, die hardware-gestützte Enklaven für den Schutz benötigen. Durch die Verwendung des TPM und anderer hier beschriebener Verfahren kann eine Attestierung der Funktion und Identität des scheuen Roboters anderen Systemen einschließlich anderer scheuer Roboter, cloudbasierter Systeme und tragbarer Benutzervorrichtungen in der Art von Smartphones u. a. bereitgestellt werden.

[1523] Der scheue Roboter **26200** kann einen Zeitgeber **26304** zum Verfolgen der Tageszeit aufweisen. Hierbei kann es sich um einen Echtzeit-Zeitgeber (RTC), eine Atomuhr oder einen Typ eines Zeitgebers, der in der Lage ist, dem Roboter die Tageszeit mitzuteilen, handeln. Aktivitäten in Städten, auf Bauernhöfen oder in anderen Umgebungen können direkt mit der Tageszeit korreliert sein.

[1524] Wie beschrieben, kann der scheue Roboter **26200** eine beliebige Anzahl von Sensoren **26306** aufweisen. Die Sensoren **26306** können Infrarot(IR)- und passive Infrarot(PIR)-Sensoren einschließen, die verwendet werden können, um das Vorhandensein von Personen oder Wildtieren durch die vom Körper abgegebene Infrarotenergie zu erkennen. Ultraschalldetektoren können verwendet werden, um nahe Objekte und den Abstand von diesen Objekten zu erkennen. Die vom Ultraschall erhaltenen Daten können dem Roboter helfen, zu navigieren und eine Bewegung von Objekten in seiner Reichweite zu erkennen.

[1525] Die Sensoren **26306** können audiovisuelle (AV) Sensoren einschließen. Die Audiodaten können verwendet werden, um das Vorhandensein von Personen zu erkennen, wie beispielsweise mit Bezug auf **Fig. 264** beschrieben. Ein Mikrofonfeld kann auch den Ankunftsinkel des Schalls als Schätzung für den Abstand der Schallquelle verwenden. Eine Kamera kann verwendet werden, um Bilddaten zu erhalten. Die Bilddaten können durch eine Anzahl von Techniken, einschließlich beispielsweise Kantenerkennung, dreidimensionale Bilderkennung und dergleichen, analysiert werden. Beispielsweise kann die Kantenerkennung verwendet werden, um Objekte im Bild zu identifizieren und um die Größe der Objekte zu bestimmen, was eine Bestimmung der Entfernung zu den Objekten ermöglichen kann. Eine visuelle Messung kann auch die Erfassung von Lichtpegeln einschließen. Diese können einer Kontextmaschine zugeführt werden, um sie mit der Tageszeit oder der Annäherung von Lichtern von einem Fahrzeug in der Art eines Blitzlichts oder einer Straßenbeleuchtung zu korrelieren, wie mit Bezug auf **Fig. 264** beschrieben.

[1526] Die Sensoren **26306** können D6T- oder thermische Sensoren einschließen. Der D6T-Sensor ist ein Sensor, der dafür ausgelegt werden kann, die Wärmesignatur einer Person zu erkennen. Er kann verwendet werden, um das Vorhandensein oder eine Bewegung von Personen zu erkennen, wobei verschiedene Erfassungswinkel verfügbar sind. Eine Kombination dieser Erfassungswinkel kann dann einen breiten Abdeckungsbereich bereitstellen. Die D6T-Sensoren können jedoch auch für andere interessierende Wärmesignaturen abgestimmt werden.

[1527] Die Sensoren **26306** können ein Magnetometer (mag) zur Erfassung beispielsweise von Telefonen emittierter elektronischer Felder einschließen. Ein Mikrowellen-Radarsensor kann aufgenommen sein, um die Entfernung zu Objekten zu bestimmen. Die Mikrowellen-Radarsensoren können kostengünstige Erweiterungen des scheuen Roboters **26200** sein, wodurch sowohl die Funktionalität als auch wirtschaftliche Gesichtspunkte verbessert werden.

[1528] Ein Antriebssystem **26308** kann in den Roboter aufgenommen sein, um seine Bewegung zu steuern. Das Antriebssystem **26308** kann eine Anzahl bekannter Roboterantriebssysteme in der Art von Doppelraupenantrieben einschließen, die verwendet werden können, um den Roboter zu bewegen oder ihn in verschiedene Richtungen zu drehen, beispielsweise wenn eine Raupe in eine andere Richtung als die andere Raupe bewegt wird. Andere Antriebssysteme können ein Dreiradsystem einschließen, wobei zwei Räder angetrieben

werden und ein drittes Rad ein Läufer ist, der verwendet wird, um ein Dreibein zu vervollständigen. Bei diesem System kann die Richtung, in welche die beiden Räder angetrieben werden, verwendet werden, um den Roboter zu lenken. Bei anderen Beispielen können abhängig von der Anwendung andere Antriebssysteme gewählt werden. Beispielsweise kann der Roboter ein autonomes Seefahrzeug sein, das beispielsweise durch einen Strahlantrieb angetrieben wird und ein Ruder zum Lenken verwendet. Bei anderen Beispielen kann der Roboter ein unbemanntes Luftfahrzeug unter Verwendung von Antriebssystemen in der Art eines Hexakopters oder eines Quadrokopters oder ein aerodynamisches System unter Verwendung eines Antriebssystems zur Entwicklung von Auftrieb über Flügeln sein.

[1529] Ein Geolokationssystem **26310** kann Satellitenempfänger des globalen Positionsbestimmungssystems (GPS) zur Bestimmung der Position, der Orientierung oder von beiden verwenden, um das Navigieren des scheuen Roboters **26200** zu unterstützen. Die Ortsdaten können von einem Schwarm scheuer Roboter verwendet werden, um ihre Bewegungen zu koordinieren und um das Vorhandensein von Personen/Wildtieren, die in Zonen innerhalb des Einsatzgebiets der Roboter eindringen und diese verlassen, mitzuteilen. Dies wird weiter mit Bezug auf **Fig. 265** erörtert.

[1530] Ein Robotiksystem **26312** kann aufgenommen sein, um jegliche Robotik, die in der Vorrichtung vorhanden ist, zu steuern. Beispielsweise können Robotiksysteme Vorrichtungen zur Ausführung von Funktionen in der Art des Einziehens von Armen zur Müllaufnahme oder der automatischen Verbindung mit der Ladestation und dergleichen aufgenommen sein. Die Vorrichtungen können wetterfeste Türen, die beispielsweise externe Ladeanschlüsse für Benutzervorrichtungen abdecken, aufweisen. Das Robotiksystem **26312** kann Treiber aufweisen, die es dem Prozessor ermöglichen, Vorrichtungen in der Art von Schrittmotoren, Solenoiden und dergleichen zu steuern.

[1531] Der scheue Roboter **26200** kann ein Ladesystem **26314** aufweisen. Das Ladesystem **26314** kann eine Anzahl von Batterieüberwachungs- und Ladevorrichtungen, wie hier beschrieben, aufweisen. Ferner kann das Ladesystem **26314** automatische Kopplungssysteme zum Anschluss des Ladesystems **26314** an eine externe Stromversorgung zum Laden der Batterie **824** aufweisen. Wie hier beschrieben, kann eine beliebige Anzahl anderer Vorrichtungen aufgenommen sein, um dies zu unterstützen, wie Solarkollektoren und dergleichen. Abhängig von der Stromentnahme und der Sparsamkeit kann das Ladesystem **26314** fortgelassen werden und kann die Vorrichtung die Batterie überwachen und Betreiber informieren, wenn die Batteriereserve ausgewählte Untergrenzen, wie 10 % verbleibende Energie, 5 % verbleibende Energie oder weniger, erreicht. Bei diesem Beispiel kann die Batterie leicht ersetzbar sein, beispielsweise indem ein Betreiber die entleerte Batterie entfernt und eine geladene Batterie einschiebt. Andere Vorrichtungen in der Art von Superkondensatoren können aufgenommen sein, um Zeitgebereinstellungen, Speichereinstellungen und dergleichen aufrechtzuerhalten, während die Batterie ersetzt wird.

[1532] Ein Stromausgabesystem **26316** kann aufgenommen sein, um externen Vorrichtungen Strom bereitzustellen. Beispielsweise kann der scheue Roboter **26200** auch als Ladestation für Benutzervorrichtungen wirken. Die Ladestation kann USB-Anschlüsse, drahtlose Stromversorgungsstellen und dergleichen aufweisen.

[1533] Ein Netzsystem **26318** kann aufgenommen sein, um Kommunikationen für den scheuen Roboter **26200** bereitzustellen. Das Netzsystem **26318** kann drahtlose und/oder festverdrahtete Technologien aufweisen. Der scheue Roboter **26200** kann mit der Cloud verbunden sein und/oder als Mitglied eines Peer-to-Peer-Netzes und/oder Schwarms wirken. Während sie als Schwarm wirken, können scheue Roboter, die sich in einem ähnlichen geographischen Gebiet befinden, interessierende Informationen austauschen. Bei diesem Szenario können die Roboter Informationen über menschliche Aktivität zu anderen Robotern übertragen.

[1534] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der Funktionen für den scheuen Roboter **26200**, wie hier beschrieben, aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1535] Der Massenspeicher **808** kann ein Betriebssystem und ein einfaches Ein-Ausgabe-System (OS/BIOS) **26318** aufweisen. Das OS/BIOS **26318** ist das Betriebssystem für den scheuen Roboter **26200**. Das OS/BIOS **26318** kann ein vollständiges OS sein, beispielsweise von Typen, die in höher entwickelten Systemen verwendet werden, oder es kann ein erweitertes BIOS sein.

[1536] Der Massenspeicher **808** kann Firmware und Treiber (FW/Treiber) **26320** für den scheuen Roboter **26200** aufweisen. Die FW/Treiber **26320** können die Treiber einschließen, die für den Betrieb der Sensoren **26306**, des Antriebs **26308**, der Geolokation **26310**, der Robotik **26312** und von Netzsystemen **26318** erforderlich sind. Ferner können die FW/Treiber **26320** Treiber für die Entdeckung anderer Vorrichtungen in der Art jener, die von der Open Connectivity Foundation (OCF) ausgegeben wurden, oder jener gemäß den AllJoyn-Spezifikationen, die von der AllSeen Alliance ausgegeben wurden, einschließen. Die FW/Treiber **26320** können Code zur Bildung von Fog-Vorrichtungen in der Art jener für das Open Fog Consortium (OFC), wie hier beschrieben, aufweisen.

[1537] Der Massenspeicher **808** kann die Kontextmaschine **26322** aufweisen. Die Kontextmaschine **26322** kann dem System Funktionalität bereitstellen, in der Art des Steuerns von Bewegungen, des Bereitstellens von Kommunikationen zwischen Robotern, des Sendens von Batteriewarnungen und dergleichen. Wie hier beschrieben, kann die Kontextmaschine **26322** aus dem Systemspeicher ausgeführt werden oder sich in einer anwendungsspezifischen integrierten Schaltung (ASIC) oder einem Gleitkomma-Gate-Array (FPGA) befinden.

[1538] Fig. 264 ist ein schematisches Diagramm des Betriebs **26400** der Kontextmaschine **26322** gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf Fig. 263 beschrieben wurden. Der Betrieb **26400** kann in Block **26402** beginnen, beispielsweise wenn der Roboter online gebracht wird.

[1539] Messungen **26404** können ausgeführt werden, um Informationen zur Bestimmung des Vorhandenseins von Personen, von Wildtieren oder von anderen Objekten in der Art von Müll oder anderen Zielgegenständen in der Umgebung zu sammeln. Die mit Bezug auf Fig. 263 beschriebene Messeinheit **26306** stellt die für das Feststellen des Vorhandenseins von Personen oder Wildtieren in der Umgebung benötigten Informationen bereit.

[1540] Beispielsweise können die Bewegungs- oder Nähedetektoren feststellen, ob es in der Nähe eine Bewegung gibt. Die Bewegung kann eine Bewegung von Personen, von Wildtieren oder anderen Objekten in der Art einer im Wind flatternden Plastiktüte einschließen. Die Bewegungsbestimmung kann den Bewegungstyp klassifizieren, um etwas, das lebendig ist, von etwas, das nicht lebendig ist, wie die Plastiktüte, zu unterscheiden.

[1541] Bei der Messung **26404** kann die Tageszeit bestimmt werden. Verschiedene Bereiche der unmittelbaren Umgebung in der Art eines Parks oder einer Straße in einer Stadt können zu verschiedenen Tageszeiten belebt sein. Ein Lernalgorithmus kann aufgenommen sein, um den Roboter zu Wochenzeiten zu trainieren, die belebt sind oder dies nicht sind. Falls bestimmte Zeiten immer belebt sind, kann der Roboter dafür ausgelegt werden, die Bewegung während dieser Zeiträume zu begrenzen.

[1542] Bei der Messung **26404** können Audiodetektoren verwendet werden, um die Pegel und Richtungen von Tönen zu erfassen und zu klassifizieren, festzustellen, ob die Töne von menschlichen oder tierischen Quellen hervorgerufen werden, und die Entfernung zu den Tönen festzustellen, um festzustellen, ob sie von menschlichen oder tierischen Quellen hervorgerufen werden, und den Abstand festzustellen.

[1543] Sobald die Messung **26404** abgeschlossen wurde, kann die Kontextmaschine **26322** die Sensormesswerte zur Weiterverarbeitung auf Werte zwischen 0 und 1 normieren. Sie kann diese Funktion ausführen, indem sie den beobachteten Wert in den Bereich möglicher vom Sensor erfassbarer Werte legt, beispielsweise unter Verwendung eines Gewichtskoeffizienten, der Nachdruck auf Werte legt, die sich innerhalb interessierender Bereiche befinden. Unter Verwendung eines Audiosensors als Beispiel sei bemerkt, dass, falls der Audiosensor Töne im Bereich von 0 Hz bis 40 kHz erfassen kann und der interessierende Bereich jener des menschlichen Gehörs ist, beispielsweise von 20 Hz bis 20 kHz, ein beobachteter Wert von 200 Hz als $(200/40 \text{ kHz}) \cdot 1$ berechnet werden kann. Falls die Beobachtung außerhalb des interessierenden Bereichs liegt, kann ein Koeffizient kleiner als 1 verwendet werden.

[1544] Für jede in die Kontextmaschine einspeisende Sensorquelle S_n kann der Algorithmus auf der folgenden Gleichung beruhen:

$$S_n = \mu \left(\frac{O_{sn}}{\text{Max}_{sn}} \right).$$

In dieser Gleichung sind S_n die vom Sensor n kommenden Werte. Der Term μ ist der Gewichtungskoeffizient, der einen Wert zwischen 0 und 1 hat. Der Gewichtungskoeffizient kann durch eine Transformationsfunktion über alle möglichen Werte angewendet werden, so dass interessierende Bereiche Werte von 1 oder nahe bei 1 aufweisen und Bereiche, die weniger interessieren, Werte von 0 oder nahe bei 0 aufweisen. Der Term 0_{sn} ist der spezifische Beobachtungswert für den aktuellen Messwert vom Sensor n . Der Term Max_{sn} ist der maximal mögliche Wert vom Sensor.

[1545] Eine Entscheidung zum Bewegen **26406** kann auf der Grundlage der bestimmten Messwerte von den Sensoren gefällt werden. Der Prozess endet bei 26408. Gewichte können auch auf einen Sensormesswert angewendet werden, um die Bedeutung des Sensormesswerts gegenüber anderen Sensormesswerten zu gewichten. Beispielsweise kann ein Systementwickler festgestellt haben, dass Audioinformationen ein besserer Indikator für das Vorhandensein von Personen oder Tieren ist als Bewegungsdetektoren. Die Entscheidung zur Bewegung von M kann einen Wert zwischen 0 und 1 aufweisen. Die Entscheidungsfällung kann auf mehrere Arten ausgeführt werden, beispielsweise kann ein gewichteter Mittelwert von M verwendet werden. Bei dieser Technik kann einem Eingangswert ein Gewicht w zugewiesen werden und ist der endgültige Wert, der verwendet wird, um die Entscheidung zur Bewegung oder Nichtbewegung zu fällen, ein gewichteter Mittelwert dieser Werte:

$$M = \frac{\sum_{t=1}^n w_t S_{n_t}}{\sum_{t=1}^n w_t}.$$

Eine Größter-gewichteter-Wert-Technik kann verwendet werden, wobei der größte gewichtete Wert als Wert für M genommen wird:

$$M = \max(w_1 S_{n_1}, \dots, w_i S_{n_i}).$$

[1546] Zusätzlich kann die Kontextmaschine **26322** eine Warnungsüberwachungseinrichtung zum Entgegennehmen von Warnungen von benachbarten Robotern in einem Schwarm implementieren. Was andere Sensormesswerte angeht, kann die Ausgabe der Warnungsüberwachungseinrichtung ein Wert zwischen 0 und 1 sein. Der Wert kann vom Abstand vom die Warnung sendenden Peer-Roboter abhängen. Beispielsweise kann ein weiter entfernter Peer-Roboter zu einem geringeren Messwert führen, während ein dichter benachbarter Peer-Roboter zu einem höheren Messwert führen kann. Demgemäß kann ein höherer Wert einer höheren Wahrscheinlichkeit entsprechen, dass eine Person oder ein Tier in das Gesichtsfeld des scheuen Roboters eintreten wird, und er kann einen höheren Einfluss auf die Entscheidungsfällung der Kontextmaschine haben.

[1547] Eine Anzahl anderer Techniken kann zusätzlich zu diesen Techniken oder an Stelle von diesen verwendet werden. Diese können beispielsweise Maschinenlernen, neuronale Netze, hardwareunterstütztes Lernen und dergleichen umfassen. Beispielsweise können die Sensormesswerte als Eingaben für ein vortrainiertes neuronales Netz oder eine hardwareunterstützte Lernplattform in der Art eines ARC-Prozessors in den Intel-Atlas-Peak-Prozessoren verwendet werden.

[1548] Sobald die Entscheidung zur Bewegung getroffen wurde, navigiert sich der Roboter selbst zum neuen Ort. Eine Anzahl von Techniken kann verwendet werden, um die Bewegung zu steuern. Beispielsweise können sich die Roboter in einem festen Bereich um bekannte, feste Hindernisse in der Art von Parkbänken, Lampenpfählen, bestimmten Bürgersteiggebieten und dergleichen bewegen. Die Roboter können gemalten Linien oder eingelassenen Linien auf Gehsteigen oder Parkwegen folgen. Ferner können die Roboter auf dem ADAS-Gebiet entwickelte Technologien für die Navigation aufweisen. Die Roboter können durch GPS entlang einem vorab festgelegten Weg mit festgelegten „Haltepunkten“ oder Basen geleitet werden, wo sich der Roboter nicht zur nächsten Basis bewegen kann, es sei denn, dass eine hohe Wahrscheinlichkeit besteht, dass er dies kann, ohne Störungen hervorzurufen. Alternativ kann dem Roboter erlaubt werden, an einer beliebigen Stelle entlang einem Weg zu halten.

[1549] Roboter, die auf diese Weise arbeiten, können größere Vorteile aus der Kombination ihrer jeweiligen Beobachtungen und dem Warnen benachbarter Roboter erhalten, wenn in ihrem Gebiet eine Aktivität bemerkt wird. Dies kann als Schwarmbetrieb bezeichnet werden. Dies kann es Robotern ermöglichen, Informationen über die direkte Reichweite ihrer Sensoren hinaus zu erhalten und diese anderen Robotern zuzuführen, wie mit Bezug auf **Fig. 265** beschrieben.

[1550] **Fig. 265** ist ein schematisches Diagramm des Betriebs **26500** eines Schwarms scheuer Roboter gemäß einigen Ausführungsformen. Die Zusammenarbeit eines Roboters mit anderen autonomen SLAM-Robotern kann die Ausführung von Operationen ermöglichen, die von einer einzigen Einheit nicht erreicht werden können. Bei diesem Prozess können eine gemeinsame Kommunikation, ein gemeinsames Verhalten und eine gemeinsame Sicherheitsprotokollerzeugung in einer Peer-to-Peer- und Rundsendeweise ausgeführt werden. Jeder Roboter **R** kann zwei allgemeine Erfassungszonen aufweisen, nämlich eine innere Zone **26502**, in der die Messung von Aktivitäten mit hoher Erfassungsgewissheit ausgeführt werden kann, und eine äußere Zone **26504**, in der die Messung von Aktivitäten mit geringerer Gewissheit ausgeführt werden kann. Falls ein Roboter **26506** ein sich durch eine der Zonen **26502** oder **26504** bewegendes Objekt **26508** erfasst, kann er eine Warnnachricht **26510** zu den nächstgelegenen Nachbarrobotern **26512** senden, um sie über die Erfassung zu informieren. Die Warnnachricht **26510** kann Informationen in der Art der Bewegung und des Orts des Objekts **26508** sowie der Gewissheit der Erfassung aufweisen.

[1551] Als Schwarm können die Roboter eine Anzahl von Gruppenaufgaben zusätzlich zu individuellen Aufgaben ausführen. Diese können solche Operationen wie eine Hypotheseerzeugung und -testung, ein Delegieren von Rollen und Teilaufgaben an die Gemeinschaft, eine Ausführung von Aufgaben, eine Validierung vollständiger Teilaufgaben, eine Datenzusammenstellung und -analyse, eine rufbasierte Rückmeldung und Belohnungsmechanismen einschließen.

[1552] **Fig. 266** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **26600** für den Betrieb eines scheuen Roboters in einem Schwarm gemäß einigen Ausführungsformen. Das Verfahren **26600** aus **Fig. 266** kann durch den mit Bezug auf die **Fig. 262** und **Fig. 263** beschriebenen scheuen Roboter **26200** implementiert werden. Das Verfahren kann in Block **26602** beginnen, beispielsweise wenn der scheue Roboter hochgefahren wird.

[1553] In Block **26604** bootet der scheue Roboter das Steuersystem. Dies kann das Laden des Betriebssystems oder des erweiterten BIOS und anschließend das Laden von Code oder das Initialisieren von Firmware einschließen. Der scheue Roboter kann dann Treiber oder andere Software laden, die für den Betrieb erforderlich sein können. Dies kann als Teil eines sicheren Boot-Prozesses, beispielsweise unter Verwendung eines TPM zum Erhalten von Messungen für die Erzeugung einer TEE im scheuen Roboter, geschehen.

[1554] In Block **26606** kann der Roboter alle ihm verfügbaren Sensoren spezifizieren. Diese können einen beliebigen Satz der mit Bezug auf **Fig. 263** beschriebenen Sensoren einschließen, einschließlich Sensoren, die sich nicht am Roboter selbst befinden können. Hierbei kann es sich um ferne Sensoren in der Art von an Lichtmasten, Gebäuden und anderen Objekten angeordneten Bewegungssensoren handeln. Ferner kann der Roboter Sensoren in anderen Robotern in einem Schwarm ausnutzen. Demgemäß können einige der Roboter im Schwarm höher entwickelte Sensorsysteme als andere Roboter im Schwarm aufweisen. Wenngleich die Sensoren in den Roboter eingebaut sein können, können Sensoren während des Betriebs dynamisch hinzugefügt werden, beispielsweise durch Einfügen in USB-Ports oder ähnliche Erweiterungsports oder -schlitze und/oder durch drahtloses Anschließen von ihnen.

[1555] In Block **26608** initialisiert der Roboter seinen Betrieb. Als Teil der Initialisierung werden Richtlinien geladen. Die Richtlinien können Regeln für die Warnung und andere fixe oder dynamische Schwellen oder Koeffizienten, die von der Kontextmaschine verwendet werden, aufweisen. Richtlinien können auch implementiert werden, um zu gewährleisten, dass die scheuen Roboter Vorschriften in der Art der FCC-Vorschriften über unbemannte Luftfahrtsysteme bzw. Unmanned Aircraft Systems (UAS) entsprechen. Die Kontextmaschine kann auch an diesem Punkt initialisiert werden.

[1556] In Block **26610** kann der Roboter mit normalen Betriebsvorgängen beginnen, beispielsweise durch Erkennen eines Vorhandenseins oder einer Bewegung, durch Implementieren eines zielorientierten Verhaltens und dergleichen. Die angeschlossenen Sensoren führen der Kontextmaschine Daten zu, die analysiert werden, wie mit Bezug auf **Fig. 264** beschrieben.

[1557] In Block **26612** wird festgestellt, ob ein Vorhandensein erkannt wird. Falls dies der Fall ist, wird in Block **26614** eine Warnrichtlinie geprüft, um die Aktion zu bestimmen, die der scheue Roboter beim Erkennen des Vorhandenseins, beispielsweise in Bezug auf das Warnen von Peers und das Unterbrechen der Bewegung, ausführen sollte.

[1558] In Block **26616** kann der Roboter Befehle an sein Antriebssystem ausgeben, um es zu einem Halt zu bringen. Die Roboter können sofort anhalten, oder sie können sich zu einem nächstgelegenen vordefinierten

Halteort bewegen, wobei dies von ihrer Funktion abhängt. Falls sich ein Roboter an einem problematischen Ort befindet, beispielsweise mitten auf einem Fahrradweg, kann ihm vorgeschrieben werden, sich zu einem Punkt außerhalb des problematischen Orts zu bewegen.

[1559] In Block **26618** kann der scheue Roboter andere Roboter im Schwarm warnen. Die Warnrichtlinie kann Regeln und Schwellen in Bezug darauf, welche Peers zu warnen sind, beispielsweise die nächstgelegenen Nachbarn zuerst, oder direkte Rundsendewarnungen an alle Roboter aufweisen. Dadurch kann die Warnrichtlinie die Wirksamkeit des Netzverkehrs zwischen den Schwarmknoten verbessern, indem bevorzugt Warnungen zu Peer-Robotern gesendet werden, bei denen die höchste Wahrscheinlichkeit besteht, dass sie die Warnungen verwenden.

[1560] Der Prozessfluss kehrt dann zu Block **26610** zurück, um die Erkennungsfunktion wiederaufzunehmen. Der scheue Roboter kann sich in manchen Fällen erst bewegen, wenn das Vorhandensein nicht mehr erkannt wird.

[1561] Falls in Block **26620** kein Vorhandensein erkannt wird, kann festgestellt werden, ob sich der scheue Roboter bewegen sollte. Die Bewegung kann eine Fortsetzung vorhergehender Bewegungen bei Nichtvorhandensein einer Erkennung oder eine neue Bewegung sein, die verwendet wird, um eine Funktion in der Art des Aufnehmens eines Stücks Abfall in der Nähe des scheuen Roboters auszuführen.

[1562] In Block **26622** kann der scheue Roboter die Geolokationstechniken zum Erhalten seines aktuellen Orts verwenden. In Block **26624** kann der scheue Roboter eine Karte unter Verwendung des aktuellen Orts, des Orts eines Ziels in der Art eines Stücks Abfall und einen Weg zum Erreichen des Ziels erzeugen. Der Roboter kann verschiedene Verfahren für die Kartenbildung verwenden. Er kann GPS verwenden, um einen Ort zu bestimmen und diesen mit den vom Internet heruntergeladenen Karten vergleichen. In Block **26626** kann das Antriebssystem verwendet werden, um den scheuen Roboter vom aktuellen Ort zum Zielort zu bewegen. Der scheue Roboter kann elektronischen Richtlinien in der Art von Drähten oder Magneten oder optischen Richtlinien in der Art gemalter Linien folgen. Ferner kann sich der scheue Roboter in einer Anzahl von Richtungen bewegen, die erforderlich sind, um eine Zielaufgabe in der Art des Aufnehmens eines Stücks Abfall auszuführen. In jedem Fall hat der Roboter gemäß einigen Ausführungsformen die Option, sich so lange zu bewegen, wie kein Vorhandensein erkannt wird.

[1563] **Fig. 267** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **26700**, das Code aufweist, um einen Prozessor **902** anzuweisen, Operationen scheuer Roboter zu behandeln. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige maschinenlesbare Medium **26700** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **26700** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1564] Das nichtflüchtige maschinenlesbare Medium **26700** kann einen Code **26702** aufweisen, um den Prozessor **902** anzuweisen, einen scheuen Roboter in eine vertrauenswürdige Ausführungsumgebung (TEE) zu booten. Ein Code **26704** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Sensoren in Zusammenhang mit dem scheuen Roboter, einschließlich in den scheuen Roboter eingebauter Sensoren, externer Sensoren, neu angebrachter Sensoren und Sensoren in Zusammenhang mit anderen Mitgliedern eines Roboterschwarms, zu entdecken.

[1565] Ein Code **26706** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den Roboter zu initialisieren, beispielsweise Richtlinien zu laden und eine Kontextmaschine zu starten. Ein Code **26708** kann aufgenommen sein, um den Prozessor **902** anzuweisen, das Vorhandensein von Personen, Wildtieren oder anderen Objekten, auf welche der scheue Roboter reagieren muss, wie unter anderem ein Fahrzeug oder ein Fahrrad, zu erkennen. Diese Objekte können in einer Objektliste in einer Richtlinie im maschinenlesbaren Medium gespeichert werden, wobei jedes Objekt in Zusammenhang mit einer Aktion steht, die der scheue Roboter ausführen soll. Die Aktion kann das Beenden einer Bewegung, das Bewegen aus dem Weg des Objekts, das Bewegen zum aufzusammelnden Objekt und dergleichen einschließen.

[1566] Ein Code **26710** kann aufgenommen sein, um den Prozessor **902** anzuweisen, konfigurierte Funktionen, beispielsweise das Aufsammeln von Abfall, das Bereitstellen von Internetkommunikationen, das autonome Anschließen an eine Ladeeinheit und dergleichen, auszuführen. Ein Code **26712** kann aufgenommen sein, um den Prozessor **902** anzuweisen, den scheuen Roboter zu einem neuen Ort zu bewegen, beispielsweise

um eine Funktion auszuführen, den scheuen Roboter aufzuladen, sich aus dem Weg sich nähernder Personen oder sich nähernden Verkehrs zu bewegen, und dergleichen.

[1567] Ein Code **26714** kann aufgenommen sein, um den Prozessor **902** anzuweisen, mit anderen Robotern in einem Schwarm zu kommunizieren. Dies kann ausgeführt werden, um andere Roboter über die Erkennung eines Vorhandenseins zu warnen, die Ausführung von Funktionen zu koordinieren und dergleichen.

[1568] Eine beliebige Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **26700** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Messeinrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1569] Bei anderen Anwendungen können angetriebene IoT-Vorrichtungen verwendet werden, um Orte zu sichern und Dienste während Ereignissen bereitustellen. Szenarien, bei denen eine unerwartete Koordination mobiler IoT-Vorrichtungen auftreten kann, können unter vielen anderen Notfälle, eine dynamische Verkehrsregelung, eine Karawanenbildung mit mehreren Parteien und Zielorten, eine Paketauslieferung und -verteilung, Fahrgemeinschaften oder eine Luftverkehrssteuerung einschließen.

[1570] Die Privatheit und Sicherheit können für jeden dieser Fälle Erwägungen sein. Falls eine Vorrichtung ohne Berechtigungsnachweise an einer Szene ankommt, kann es nicht bekannt sein, ob die Vorrichtung eine bösartige Vorrichtung sein kann. Ferner können die Ereignisse an einer Szene privatheitssensitiv sein. Demgemäß können Bedingungen und die Wahrnehmung erfasster Szenen der Sammlungs- und Verwendungssteuerung eines Fahrenden unterliegen. Wie hier verwendet, bezieht sich ein Fahrender auf die Interaktionen an einer Szene vorhersehende Entität.

[1571] Bei einem Beispiel sieht ein Fahrender die Verfolgung eines Wegs zu einem Ziel, der von einer Reihe von Wegpunkten geschnitten wird, vorher. Der Fahrtwegplaner berechnet die erwarteten Wegpunkte, die während einer Fahrt angetroffen werden. An jedem Wegpunkt wird eine Pre-first-Responder-Zuständigkeit berechnet. Wie mit Bezug auf die **Fig. 270** und **Fig. 271** beschrieben, kann eine Notfallbehandlungs(EM)-Baumstruktur verwendet werden, um Schlüssel zu erzeugen, die vom Fahrenden und Pre-first-Respondern unter anderem in der Art von Drohnen, Robotern oder selbst geleiteten Fahrzeugen verwendet werden können, um Kommunikationen zu koordinieren, die durch von den EM-Bäumen erzeugte Schlüssel geschützt sind.

[1572] **Fig. 268** ist ein schematisches Diagramm eines Verwendungsfalls **26800**, worin Drohnen als Pre-first-Responder-Vorrichtungen an einer Szene innerhalb eines Zuständigkeitsbereichs dargestellt sind, gemäß einigen Ausführungsformen. Beim Verwendungsfall **26800** können drei Zuständigkeitsszenarien betrachtet werden, wenn Pre-first-Responder-Vorrichtungen (PFRD) **26802** mit sicherheits- und privatheitsbewahrender Kryptographie kommissioniert werden. Bei einem ersten in **Fig. 268(A)** dargestellten beispielhaften Verwendungsfall ist ein einziger Zuständigkeitsbereich **26806** vorhanden und sind alle Notfall-Responder **26804** aus diesem Zuständigkeitsbereich. Die PFRD **26802** ist autorisiert, innerhalb des einzelnen Zuständigkeitsbereichs **26806** im Auftrag von Typen von Notfall-Respondern **26804** zu wirken. In **Fig. 268(B)** ist eine PFRD **26802** autorisiert, innerhalb überlappender Zuständigkeitsbereiche **26808** im Auftrag von N Typen von Notfall-Respondern **26804** zu wirken. In **Fig. 268(C)** ist eine PFRD **26802** autorisiert, innerhalb von drei getrennten und verschiedenen Zuständigkeitsbereichen **26810** im Auftrag von N Typen von Notfall-Respondern **26804** zu wirken.

[1573] **Fig. 269** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **26900** zur Ausführung eines Beitritts- und Registrierungsprozesses in Zusammenhang mit der einzelnen und den mehreren Zuständigkeitsbereichs-Kontrollzonen in **Fig. 268** gemäß einigen Ausführungsformen. Das Verfahren **26900** aus **Fig. 269** kann durch die mit Bezug auf **Fig. 274** beschriebene PFRD **27400** implementiert werden. Das Verfahren kann in Block **26902** beginnen, wenn eine Pre-first-Responder-Vorrichtung (PFRD) beispielsweise einen Befehl zum Teilnehmen am Ereignis in der Art einer Notsituation empfängt.

[1574] In Block **26904** prüft die PFRD, in welchen Zuständigkeitsbereichen sie teilnehmen kann. In Block **26906** wird festgestellt, dass die PFRD in einem einzigen Zuständigkeitsbereich teilnimmt, wie beispielsweise mit Bezug auf **Fig. 268(A)** beschrieben. In Block **26908** verhandelt die PFRD einen Schlüssel mit dem Zuständigkeitsbereich, um Kommunikationen zu erlauben. In Block **26910** registriert sich die PFRD beim einzelnen Zuständigkeitsbereich unter Verwendung des Schlüssels.

[1575] In Block **26912** wird festgestellt, ob die Registrierung erfolgreich war. Falls dies der Fall ist, kann die Kommunikationsaufgabe für die PFRD in Block **26914** beginnen. Falls die Registrierung nicht erfolgreich war, wird in Block **26916** ein Zuständigkeitsbereichs-Kontrollnetz alarmiert.

[1576] In Block **26918** wird festgestellt, ob die PFRD in überlappenden Zuständigkeitsbereichen teilnehmen kann. Falls dies der Fall ist, wird in Block **26920** eine Geteilter-Schlüssel-Verhandlung mit den überlappenden Zuständigkeitsbereichen ausgeführt. Weil die Zuständigkeitsbereiche einander überlappen, kann ein einziger Schlüssel für Kommunikationen in beide Zuständigkeitsbereiche erteilt werden.

[1577] In Block **26922** wird ein erteilter Schlüssel zur Registrierung beim Zuständigkeitsbereich 1 verwendet, bevor der Prozessfluss zu Block **26912** zurückkehrt, um festzustellen, ob die Registrierung erfolgreich war. In Block **26924** werden die erteilten Schlüssel zur Registrierung beim Zuständigkeitsbereich 2 verwendet, bevor der Prozessfluss zu Block **26912** zurückkehrt, um festzustellen, ob die Registrierung erfolgreich war. Wie in Block **26926** dargestellt, kann die Registrierung durch eine Anzahl anderer Zuständigkeitsbereiche hindurch fortgesetzt werden.

[1578] In Block **26928** wird festgestellt, ob es mehrere nicht überlappende Zuständigkeitsbereiche gibt, in denen die PFRD zur gleichen Zeit teilnehmen kann. Falls dies der Fall ist, werden in Block **26930** mit den jeweiligen einzelnen Zuständigkeitsbereichen einzelne Schlüsselverhandlungen ausgeführt. Sobald von jedem der einzelnen Zuständigkeitsbereiche Schlüssel erteilt wurden, wird der Prozessfluss in Block **26922** fortgesetzt, um mit der Registrierung bei jedem der Zuständigkeitsbereiche zu beginnen.

[1579] Falls in den Blöcken **26906**, **26918** oder **26928** keine Zuständigkeitsbereiche identifiziert werden, kann es der PFRD in Block **26932** nicht erlaubt werden, sich bei irgendwelchen der Zuständigkeitsbereiche zu registrieren. Falls der Registrierungsprozess für einen oder mehrere der Zuständigkeitsbereiche fehlschlägt, wird eine Warnung an das Kontrollnetz abgesendet.

[1580] Fig. **270** ist ein schematisches Diagramm einer beispielhaften Fahrtplanung **27000** für einen Notfall-Responder (ER) **27002** oder eine andere Entität, um einen Weg **27004** zu einem Ziel zu bestimmen, gemäß einigen Ausführungsformen. Die Fahrtwegplanung **27000** sieht die verschiedenen Szenen, Wegpunkte, Orte, Kreuzungen und Ziele entlang dem Weg **27002** vorher. Die Fahrtwegplanung **27004** kann alternative Wege und eine Planung in der letzten Minute oder nach Bedarf einschließen. An jedem Wegpunkt **27006** können eine Ortskoordinate und die vorhergesehene Ankunftszeit **27008** berechnet werden. Die Granularität der Zeit kann gewählt werden, um eine frühe oder eine späte Ankunft zu ermöglichen. Für jeden Wegpunkt **27006** wird ein Entropiemultiplexier(EM)-Baum **27010** gefunden, der Wegpunkte Zuständigkeitsbereichen zuordnet, worin eine erwartete PFRD-Gemeinschaft zum Antworten kommissioniert werden kann. In einem gegebenen Notfall-Responder-Verwendungsfallzusammenhang können die PFRD mit einem Zweig des vom Fahrenden ER **27008** entwickelten EM-Baums **27010** kommissioniert werden.

[1581] Fahrtinformationen von der Fahrtwegplanung **27000** können einen zufälligen Nonce-Wert aufweisen, der mit möglichen PFRD geteilt wird. Dies kann durch den ER **27008** an einer Szene validiert werden. Der Nonce-Wert gewährleistet, dass die PFRD den korrekten EM-Baum verwendet. Bei der Fahrtwegplanung können Responder in eine Fahrt eingeschrieben werden, indem jedem PFRD-Teilnehmer ein privater EPID-Schlüssel erteilt wird. Die EPID kann verwendet werden, um PFRD an der Szene zu authentifizieren.

[1582] Fig. **271** ist ein schematisches Diagramm der Verwendung von EM-Teilbäumen **27102** und **27104** an jedem Wegpunkt gemäß einigen Ausführungsformen. Bei diesem Beispiel repräsentiert ein erster EM-Teilbaum **27102** einen EM-Baum für einen gegebenen Ort (A) und eine gegebene Zeit (A-C). Ein zweiter EM-Teilbaum **27104** repräsentiert einen anderen Ort (N) und eine andere Zeit (A-C) weiter in den Weg hinein. Jeder EM-Teilbaum **27102** und **27104** codiert die Zeit **27106** mit abnehmender Granularität. PFRD werden mit dem EM-Teilbaum **27102** oder **27104** bereitgestellt, der unter Berücksichtigung der verschiedenen Permutationen der Zuständigkeitsbereichsabdeckung mit seinem Zuständigkeitsbereich ausgerichtet ist. Beispielsweise kann ein getrennter Teilbaumzweig **27108** verwendet werden, um die Überlappung zwischen mehreren Zuständigkeitsbereichen für eine erhöhte Privatheit und Sicherheit zu repräsentieren. Oder es kann eine PFRD mit mehreren Teilbaumzweigen **27110** versehen werden, so dass sie an diesen mehreren Zuständigkeitsbereichen eingesetzt werden kann.

[1583] Fig. **272** ist ein Prozessflussdiagramm eines beispielhaften Verfahrens **27200** zur dynamischen Konfiguration eines PFRD-Netzes an einer Szene gemäß einigen Ausführungsformen. Das Verfahren **27200** aus Fig. **272** kann durch die mit Bezug auf Fig. **274** beschriebene PFRD **27400** implementiert werden. Das Verfah-

ren **27200** kann in Block **27202** beginnen, wenn ein Notfall-Responder oder ein anderer Fahrer zu einem Zielort aufbricht. In Block **27204** wird festgestellt, ob PFRD entlang dem Weg ausgelöst werden. Falls dies der Fall ist, werden in Block **27206** Richtlinien abgerufen, beispielsweise aus einem sicheren Speicher in einer vertrauenswürdigen Ausführungsumgebung (TEE). In Block **27208** wird eine im sicheren TEE-Speicher bereitgestellte anfängliche Flugplanrichtlinie geladen und aktiviert. Die Flugplanrichtlinie kann spezifische Operationen aufweisen, welche die PFRD ausführen soll. Diese können solche Operationen wie das Lösen vom Ankerpunkt, das Schweben, das Erfassen von Bildern, das Ausführen einer Bilderkennung und dergleichen einschließen. In Fällen, in denen andere autonome Fahrzeuge abgesehen von Drohnen verwendet werden, wie autonome Oberflächenfahrzeuge, kann im sicheren TEE-Speicher eine Zuordnungsrichtlinie bereitgestellt werden, die andere Aktivitäten einschließen kann, wie das Vermeiden von Hindernissen, das Steuern der Geschwindigkeit und dergleichen.

[1584] In Block **27210** wird festgestellt, ob es eine schlechte Netzverbindbarkeit gibt. Falls dies der Fall ist, werden in Block **27212** Richtlinien maximaler Verbindbarkeit aus dem sicheren TEE-Speicher abgerufen und implementiert. Diese Richtlinien können auf dem aktuellen Ort beruhen, beispielsweise einschließlich Befehle, sich zu einem bestimmten Ort zu bewegen, um mit einem Mast zu kommunizieren.

[1585] In Block **27214** wird festgestellt, ob die PFRD den ER-Vertrag einhalten soll. Falls dies der Fall ist, werden in Block **27216** Fernbehandlungsrichtlinien aus dem sicheren TEE-Speicher abgerufen und implementiert. In Block **27218** kann eine Fernflugbehandlung zugelassen werden, so dass der ER weitere Informationen sammeln kann. Hierdurch kann die Vorbereitung auf Vorfälle verbessert werden, und dies kann zusätzlich zu anfänglich konfigurierten Richtlinien geschehen.

[1586] Das Verfahren **27200** endet in Block **27220**. Dies kann beispielsweise geschehen, falls eine PFRD in Block **27204** nicht ausgelöst wird oder falls ein ER-Vertrag in Block **27214** nicht eingehalten wird.

[1587] Fig. 273 ist ein Leiterdiagramm eines beispielhaften Verfahrens **27300** zum Versehen von Szeneninformationen mit Baken durch eine PFRD gemäß einigen Ausführungsformen. Das Verfahren **27300** aus Fig. 273 kann durch die mit Bezug auf Fig. 274 beschriebene PFRD **27400** implementiert werden. Wenn ein Fahrer in der Art eines Notfall-Responders (ER) **27302** an einer Szene ankommt, werden die zur Verschlüsselung von Nachrichten zwischen anderen Vorrichtungen, die autorisiert sind, an der Szene teilzunehmen, verwendeten Schlüssel bei dem gegebenen Szenen-Seed dynamisch erzeugt. Eine teilnehmende Vorrichtung kann ihre Autorisierung durch Signieren der Szeneninformationen vom Fahrtwegplaner **27304** mit einer EPID attestieren und sie dem Fahrer zur Überprüfung zuführen.

[1588] Wenn die Vorrichtungen **27302** an der Szene ankommen, kann eine Bereitstellung **27306** geschehen, um auf Richtlinien, Schlüssel und andere benötigte Informationen zuzugreifen. Dies kann durch einen Außerband(OOB)-Bereitstellungsprozess geschehen.

[1589] Ein Fahrplan **27308** kann einem ER **27302** durch eine vom Fahrtwegplaner **27304** bereitgestellte Fahrplannachricht **27310** bereitgestellt werden. Es kann festgestellt werden **27312**, ob eine Notfall-Responder-Unterstützung benötigt wird. Falls dies der Fall ist, kann der Fahrtwegplaner Notfall-Responder unter Verwendung des EM-Baums zum Identifizieren von Respondern identifizieren **27314** und jeden der Notfall-Responder zum Helfen auffordern. Bakenattestierungsnachrichten **27316** können zwischen dem Fahrtwegplaner **27304** und jedem der ER **27302** ausgetauscht werden. Ferner können die ER **27302** untereinander für die Ressourcenzuweisung verhandeln **27318**. Ein ER **27302** kann dann eine Nachricht **27320** zurückgeben, welche die Bake bestätigt, und den Fahrtwegplaner **27304** über die von diesem ER **27302** verwendeten Richtlinien informieren.

[1590] Fig. 274 ist ein Blockdiagramm eines Beispiels von Komponenten, die in einer Pre-first-Responder-Drohne (PFRD) **27400** vorhanden sein können, gemäß einigen Ausführungsformen. Gleich nummerierte Elemente entsprechen jenen, die mit Bezug auf die Fig. 8 und Fig. 263 beschrieben wurden. Es sei bemerkt, dass für die PFRD **27400** andere Komponenten ausgewählt und verwendet werden können als jene, die für den mit Bezug auf Fig. 263 erörterten scheuen Roboter **26300**, die mit Bezug auf Fig. 8 erörterte IoT-Vorrichtung **800** und andere hier erörterte IoT-Vorrichtungen ausgewählt wurden. Beispielsweise kann die PFRD **27400** eine Drohne oder ein anderes unbemanntes Luftfahrzeug sein. Dementsprechend kann das Antriebssystem **26308** für die PFRD **27400** ein Hexakopter- oder Quadroktersystem sein. Bei einigen Beispielen kann die PFRD **27400** ein aerodynamisches Flugfahrzeug in der Art eines unbemannten aerodynamischen Flugzeugs sein.

[1591] Die PFRD **27400** kann ein vertrauenswürdiges Plattformmodul (TPM) aufweisen, das beispielsweise mit der von der Trusted Computing Group **2009** als ISO/IEC **11889** herausgegebenen Spezifikation kompatibel

ist. Das TPM kann einen kryptographischen Prozessor (CP), einen nichtflüchtigen Speicher (NVM) und einen sicheren Speicher (SM) aufweisen. Der CP kann unter anderem einen Zufallszahlengenerator, einen RSA-Hash-Generator, einen SHA-1-Hash-Generator und eine Verschlüsselungs-Entschlüsselungs-Engine bereitstellen. Der NVM kann bei der Herstellung programmierte Schlüssel aufweisen, die beispielsweise unter anderem einen RSA-Schlüssel einschließen. Der SM kann an Software in Plattformkonfigurationsregistern genommene Messungen halten. Wie hier verwendet, ist eine Messung ein an einem Code oder Datenssegment, das im permanenten Speicher **808** oder im flüchtigen Speicher **804** vorhanden ist, berechneter Hash-Code. Ausgehend von einer Messung eines Boot-Codesegments können die Messungen verwendet werden, um eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) **26302** durch Erzeugen einer Vertrauenskette vom anfänglichen Booten einzurichten. Der SM kann einen sicheren Speicher bereitstellen.

[1592] Die TEE **26302** stellt einen Bereich bereit, worin Prozesse ausgeführt werden können, die hardware-gestützte Enklaven für den Schutz benötigen. Unter Verwendung des TPM und anderer hier beschriebener Verfahren kann eine Attestierung der Funktion und Identität der PFRD **27400** anderen Systemen bereitgestellt werden, einschließlich unter anderem Fahrtwegplanern, Notfall-Respondern, cloudbasierter Systeme und tragbarer Benutzervorrichtungen in der Art von Smartphones.

[1593] Ein Netzsystem **26318** kann aufgenommen sein, um Kommunikationen mit der PFRD **27400** bereitzustellen. Das Netzsystem **26318** kann drahtlose oder festverdrahtete Technologien aufweisen. Die PFRD **27400** kann mit der Cloud verbunden sein oder als Mitglied eines Peer-to-Peer-Netzes oder Schwarms wirken. Während sie als Schwarm wirken, können PFRD in einem ähnlichen geographischen Gebiet interessierende Informationen austauschen. In diesem Szenario können die PFRD Informationen über Notfall-Responder, Kommunikationsrichtlinien, Ereignisse und andere relevante Informationen zu anderen PFRD weiterleiten.

[1594] Der Massenspeicher **808** kann eine Anzahl von Modulen zur Implementation der Funktionen für die PFRD **27400**, wie hier beschrieben, aufweisen. Wenngleich sie im Massenspeicher **808** als Codeblöcke dargestellt sind, ist zu verstehen, dass jegliche der Module ganz oder teilweise durch festverdrahtete Schaltungen ersetzt werden können, die beispielsweise in eine anwendungsspezifische integrierte Schaltung (ASIC) eingebaut sind.

[1595] Der Massenspeicher **808** kann ein Betriebssystem und ein einfaches Ein-Ausgabe-System (OS/BIOS) **26318** aufweisen. Das OS/BIOS **26318** ist das Betriebssystem für die PFRD **27400**. Das OS/BIOS **26318** kann ein vollständiges OS sein, beispielsweise von Typen, die in höher entwickelten Systemen verwendet werden, oder es kann ein erweitertes BIOS sein.

[1596] Der Massenspeicher **808** kann Firmware und Treiber (FW/Treiber) **26320** für die PFRD **27400** aufweisen. Die FW/Treiber **26320** können die Treiber einschließen, die für den Betrieb der Sensoren **26306**, des Antriebs **26308**, der Geolokation **26310**, der Robotik **26312** und von Netzsystemen **26318** erforderlich sind. Ferner können die FW/Treiber **26320** Treiber für die Entdeckung anderer Vorrichtungen in der Art jener, die von der Open Connectivity Foundation (OCF) ausgegeben wurden, oder jener gemäß den AllJoyn-Spezifikationen, die von der AllSeen Alliance ausgegeben wurden, einschließen. Die FW/Treiber **26320** können Code zur Bildung von Fog-Vorrichtungen in der Art jener für das Open Fog Consortium (OFC), wie hier beschrieben, aufweisen.

[1597] Der Massenspeicher **808** kann eine Szenenbewertungs-Steuereinrichtung **27402** aufweisen. Die Szenenbewertungs-Steuereinrichtung kann dem System Funktionalität bereitstellen, wie das Steuern der Bewegung, Bilderfassung und -übertragung, Netzbereitstellung für andere Vorrichtungen, Kommunikationen zwischen Vorrichtungen, Batteriewarnungen und dergleichen. Wie hier beschrieben, kann die Szenenbewertungs-Steuereinrichtung **27402** aus dem Systemspeicher ausgeführt werden oder sich in einer anwendungsspezifischen integrierten Schaltung (ASIC) oder einem Gleitkomma-Gate-Array (FPGA) befinden.

[1598] Bei der PFRD **27400** kann Hardware das Vertrauen erhöhen, beispielsweise durch Segmentieren von EM-Seeds im sicheren Speicher und Verwenden einer Technologie einer vertrauenswürdigen Ausführungsumgebung in der Art von Intel SGX, Intel CSME, Intel VTx oder anderer Systeme. Die Verwendung von Intel DRNG kann die Qualität und Funktionsweise der Baumerzeugung verbessern, so dass beschränkte Vorrichtungen realistisch Schlüssel dynamisch anhand des gespeicherten Seed-Werts erzeugen können.

[1599] **Fig. 275** ist ein Blockdiagramm eines nichtflüchtigen maschinenlesbaren Mediums **27500**, dass Code aufweist, um einen Prozessor **902** anzuweisen, Operationen von Pre-first-Respondervorrichtungen zu behandeln, gemäß einigen Ausführungsformen. Der Prozessor **902** kann über einen Bus **904** auf das nichtflüchtige

maschinenlesbare Medium **27500** zugreifen. Der Prozessor **902** und der Bus **904** können ähnlich wie der Prozessor **902** und der Bus **904**, die mit Bezug auf **Fig. 9** beschrieben wurden, implementiert sein. Das nichtflüchtige maschinenlesbare Medium **27500** kann für den Massenspeicher **808** aus **Fig. 8** beschriebene Vorrichtungen oder optische Scheiben, Speichersticks oder eine Anzahl anderer Hardwarevorrichtungen aufweisen.

[1600] Das nichtflüchtige maschinenlesbare Medium **27500** kann einen Code **27502** aufweisen, um den Prozessor **902** anzuweisen, eine PFRD auszulösen, beispielsweise wenn ein Notfall-Responder in einen Zuständigkeitsbereich für die PFRD eindringt. Ein Code **27504** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Richtlinien für automatisierte Operationen abzurufen, wie Flugoperationen einschließlich Schweben, Erfassen von Bildern und dergleichen.

[1601] Ein Code **27506** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Fahrpläne für den Fahrtwegplaner entgegenzunehmen und die Fahrpläne Notfall-Respondern bereitzustellen. Ein Code **27508** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine schlechte Netzleitfähigkeit zu erkennen und Richtlinien für Kommunikationen zu implementieren.

[1602] Ein Code **27510** kann aufgenommen sein, um den Prozessor **902** anzuweisen, Fernbehandlungsfunktionen zu implementieren, wie es einem Notfall-Responder zu ermöglichen, die PFRD für die Datensammlung zu steuern. Ein Code **27512** kann aufgenommen sein, um den Prozessor **902** anzuweisen, eine Fernoperationsbehandlung, beispielsweise in einem Fernsteuermodus, zu ermöglichen. Ein Code **27514** kann aufgenommen sein, um den Prozessor **902** anzuweisen, einen Fernnetzzugriff für Benutzer bereitzustellen.

[1603] Eine beliebige Anzahl anderer Codeblöcke kann in das maschinenlesbare Medium **27500** aufgenommen werden, um Funktionen der IoT-Vorrichtungen zu implementieren. Diese Codeblöcke können einen Kommunikator zur Bildung und Übertragung von Paketen zwischen IoT-Vorrichtungen, eine sichere Messeinrichtung zur Ausführung von Messungen für das sichere Ausführen von Code, einen Schlüsselgenerator oder eine Anzahl anderer hier beschriebener Codeblöcke einschließen.

[1604] Beispiel 1 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein zusammengesetztes Objekt, das einen Vorrichtungseigentümer umfasst, welcher auch einen Nameserver zum Bereitstellen von Namen für Sub-Objekte, die das zusammengesetzte Objekt bilden, und eine Sub-Objekt-Liste der Sub-Objekte, die das zusammengesetzte Objekt bilden, umfasst, und mehrere Sub-Objekte, die das zusammengesetzte Objekt bilden, und eine Blockchain, welche die das zusammengefasste Objekt bildenden Sub-Objekte aufzeichnet.

[1605] Beispiel 2 umfasst den Gegenstand von Beispiel 1. In Beispiel 2 umfasst ein Sub-Objekt ein zusammengesetztes Objekt, das aus Sub-Objekten niedrigerer Ebene gebildet ist.

[1606] Beispiel 3 umfasst den Gegenstand von Beispiel 1 oder 2. In Beispiel 3 umfasst ein Sub-Objekt ein atomares Objekt.

[1607] Beispiel 4 umfasst den Gegenstand von einem der Beispiele 1 bis 3. In Beispiel 4 umfasst der Name des zusammengesetzten Objekts einen anhand der Namen der mehreren Sub-Objekte berechneten Hash.

[1608] Beispiel 5 umfasst den Gegenstand von einem der Beispiele 1 bis 4. In Beispiel 5 umfasst jedes Sub-Objekt einen Gruppenschlüssel, der es dem Sub-Objekt ermöglicht, im Auftrag der Gruppe zu wirken.

[1609] Beispiel 6 umfasst den Gegenstand von einem der Beispiele 1 bis 5. In Beispiel 6 umfasst der Vorrichtungseigentümer einen EPID-Server.

[1610] Beispiel 7 umfasst den Gegenstand von einem der Beispiele 1 bis 6. In Beispiel 7 umfasst der Vorrichtungseigentümer einen Proxy-Broker.

[1611] Beispiel 8 umfasst den Gegenstand von einem der Beispiele 1 bis 7. In Beispiel 8 umfasst der Vorrichtungseigentümer eine Blockchain.

[1612] Beispiel 9 umfasst den Gegenstand von einem der Beispiele 1 bis 8. In Beispiel 9 umfasst die Blockchain einen Datensatz des zusammengesetzten Objekts.

[1613] Beispiel 10 umfasst ein Verfahren zur Bildung eines zusammengesetzten Objekts in einem IoT-Netz. Das Verfahren zur Bildung eines zusammengesetzten Objekts in einem IoT-Netz umfasst das Bilden einer

Liste von Sub-Objekten in einem Vorrichtungseigentümer, das Erzeugen einer Sammlungsgruppenkennung, das Übergeben der Sammlungsgruppenkennung an eine Blockchain in einer Blockchain-Transaktion und das Erhalten eines Gruppennamens von der Blockchain im Nameserver.

[1614] Beispiel 11 umfasst den Gegenstand von Beispiel 10. In Beispiel 11 umfasst das Verfahren das Feststellen anhand der Blockchain, ob die Sammlungsgruppenkennung bereits in Gebrauch ist, und, falls dies der Fall ist, das Erzeugen einer neuen Sammlungsgruppenkennung.

[1615] Beispiel 12 umfasst den Gegenstand von Beispiel 10 oder 11. In Beispiel 12 umfasst das Verfahren das Akzeptieren einer Beitrittsanforderung von einem Sub-Objekt, das Bestätigen, dass das Sub-Objekt ein Gruppenmitglied ist, das Nachschlagen des Namens des Sub-Objekts in der Blockchain und das Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom Nameserver.

[1616] Beispiel 13 umfasst den Gegenstand von einem der Beispiele 10 bis 12. In Beispiel 13 umfasst das Verfahren das Feststellen, ob die Gruppenmitgliedschaft privat ist, und, falls dies der Fall ist, das Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom als Proxy für den Nameserver wirkenden Vorrichtungseigentümer.

[1617] Beispiel 14 umfasst den Gegenstand von einem der Beispiele 10 bis 13. In Beispiel 14 umfasst das Verfahren das Erzeugen der Sammlungsgruppenkennung durch Kombinieren der Namen des Sub-Objekts zur Bildung einer Kombination und das Berechnen eines Hash-Codes der Kombination.

[1618] Beispiel 15 umfasst den Gegenstand von einem der Beispiele 10 bis 14. In Beispiel 15 umfasst das Verfahren das Erzeugen eines Namens für ein Sub-Objekt durch Kombinieren der Namen aller Sub-Sub-Objekte, die das Sub-Objekt bilden, zur Bildung einer Kombination und das Berechnen eines Hash-Codes der Kombination.

[1619] Beispiel 16 umfasst den Gegenstand von einem der Beispiele 10 bis 15. In Beispiel 16 umfasst das Verfahren das Bestätigen, dass die Blockchain-Transaktion in einer Gruppe von Vorrichtungen in einem Mesh-Netz gültig ist, und das Umkehren der Blockchain-Transaktion, falls sie nicht gültig ist.

[1620] Beispiel 17 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, um einen Prozessor anzuweisen, eine Liste von Sub-Objekten für eine Gruppe zu speichern, eine Sammelgruppenidentität für die Gruppe zu berechnen und Gruppenidentitäts-Berechtigungsnachweise Sub-Objekten in der Gruppe bereitzustellen.

[1621] Beispiel 18 umfasst den Gegenstand von Beispiel 17. In Beispiel 18 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, als ein Proxy-Server für Sub-Objekte zu wirken.

[1622] Beispiel 19 umfasst den Gegenstand von Beispiel 17 oder 18. In Beispiel 19 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, eine Transaktion einschließlich einer Sammelgruppenidentität an eine Blockchain zu übergeben und die Blockchain zu anderen Vorrichtungen in einem Mesh zu migrieren.

[1623] Beispiel 20 umfasst den Gegenstand von einem der Beispiele 17 bis 19, einschließlich einer Blockchain, welche Transaktionsblöcke zu 20 umfasst. In Beispiel 20 umfasst ein Transaktionsblock eine Sammelgruppenidentität.

[1624] Beispiel 21 umfasst eine Vorrichtung, die ein zusammengesetztes Objekt umfasst. Diese umfasst einen Vorrichtungseigentümer, der einen Typnameserver zur Erzeugung von Typnamen für das zusammengesetzte Objekt umfasst, und eine Blockchain, die eine Transaktion umfasst, welche Typen von Sub-Objekten umfasst, die das zusammengesetzte Objekt bilden.

[1625] Beispiel 22 umfasst den Gegenstand von Beispiel 21. In Beispiel 22 umfasst die Vorrichtung einen Typinspektor zur Bestimmung der Typen von Sub-Objekten, die das zusammengesetzte Objekt umfassen.

[1626] Beispiel 23 umfasst den Gegenstand von Beispiel 21 und 22. In Beispiel 23 umfasst der Typinspektor ein Typintrospektionssystem.

- [1627]** Beispiel 24 umfasst den Gegenstand von einem der Beispiele 21 bis 23. In Beispiel 24 umfasst der Typinspektor ein Typattestierungssystem.
- [1628]** Beispiel 25 umfasst den Gegenstand von einem der Beispiele 21 bis 24. In Beispiel 25 umfasst die Vorrichtung einen Typgraphengenerator zur Erzeugung eines Typgraphen von Typen von Sub-Sub-Objekten, welche die Sub-Objekte bilden.
- [1629]** Beispiel 26 umfasst den Gegenstand von einem der Beispiele 21 bis 25. In Beispiel 26 umfasst die Vorrichtung einen Typnamensberechner zur Erzeugung eines Typnamens anhand eines Typgraphen.
- [1630]** Beispiel 27 umfasst den Gegenstand von einem der Beispiele 21 bis 26. In Beispiel 27 umfasst die Transaktion einen Typgraphen.
- [1631]** Beispiel 28 umfasst den Gegenstand von einem der Beispiele 21 bis 27. In Beispiel 28 umfasst ein Objekt einen Typberechtigungs nachweis.
- [1632]** Beispiel 29 umfasst den Gegenstand von einem der Beispiele 21 bis 28. In Beispiel 29 umfasst ein Typberechtigungs nachweis einen Herstellerschlüssel.
- [1633]** Beispiel 30 umfasst den Gegenstand von einem der Beispiele 21 bis 29. In Beispiel 30 wird der Typberechtigungs nachweis durch einen Nameserver bereitgestellt.
- [1634]** Beispiel 31 umfasst den Gegenstand von einem der Beispiele 21 bis 30. In Beispiel 31 umfasst ein Sub-Objekt Sub-Sub-Objekte und wird der Typname des Sub-Objekts anhand der Typen der Sub-Sub-Objekte bestimmt.
- [1635]** Beispiel 32 umfasst den Gegenstand von einem der Beispiele 21 bis 31. In Beispiel 32 umfasst die Vorrichtung einen durch ein Sub-Objekt, das die Typen der Sub-Sub-Objekte umfasst, erzeugten Typgraphen.
- [1636]** Beispiel 33 umfasst ein Verfahren zur Erzeugung eines Objekttyps in einem IoT-Netz. Das Verfahren zur Erzeugung eines Objekttyps in einem IoT-Netz umfasst das Anfordern einer Erzeugung einer Typgruppe durch einen Nameserver, das Ausführen einer Typinspektion von Sub-Objekten, die ein zusammengesetztes Objekt bilden, um einen Typgraphen von Objekten, die das zusammengesetzte Objekt bilden, zu erzeugen, das Berechnen eines Typgruppennamens anhand des Typgraphen und das Zugreifen auf eine Blockchain zur Feststellung, ob der Typgruppename bereits erzeugt wurde.
- [1637]** Beispiel 34 umfasst den Gegenstand von Beispiel 33. In Beispiel 34 umfasst das Verfahren das Erzeugen des Typgruppennamens durch Schreiben einer Transaktion, die den Typgraphen umfasst, in die Blockchain.
- [1638]** Beispiel 35 umfasst den Gegenstand von Beispiel 33 oder 34. In Beispiel 35 umfasst das Verfahren das Ausgeben einer EPID-Beitrittsanforderung vom Nameserver an das zusammengesetzte Objekt.
- [1639]** Beispiel 36 umfasst den Gegenstand von einem der Beispiele 33 bis 35. In Beispiel 36 umfasst das Verfahren das Ausgeben von Typberechtigungs nachweisen an die Sub-Objekte, die das zusammengesetzte Objekt bilden.
- [1640]** Beispiel 37 umfasst den Gegenstand von einem der Beispiele 33 bis 36. In Beispiel 37 fordert der Nameserver das zusammengesetzte Objekt auf, die Typinspektion auszuführen.
- [1641]** Beispiel 38 umfasst den Gegenstand von einem der Beispiele 33 bis 37. In Beispiel 38 umfasst die Typinspektion eine rekursive Introspektion der das zusammengesetzte Objekt bildenden Sub-Objekte.
- [1642]** Beispiel 39 umfasst den Gegenstand von einem der Beispiele 33 bis 38. In Beispiel 39 umfasst die rekursive Introspektion das Senden einer Typintrospektionsanforderung zu jedem der das zusammengesetzte Objekt bildenden Sub-Objekte, das Ausführen einer Typintrospektion zur Bestimmung eines Typs für jedes der das Sub-Objekt bildenden Sub-Sub-Objekte, das Bilden eines Typgraphen an jedem der Sub-Objekte, die anhand der Sub-Sub-Objekte gebildet wurden, das Zurückgeben der Typgraphen zum zusammengesetzten Objekt und das Verifizieren von Signaturen an den Typgraphen.

[1643] Beispiel 40 umfasst den Gegenstand von einem der Beispiele 33 bis 39. In Beispiel 40 umfasst das Verfahren das Ausführen einer rekursiven Typintrospektion von jedem Sub-Sub-Objekt zu Objekten niedrigerer Ebene in einer Hierarchie, das Bilden eines Typgraphen für Objekte auf jeder Ebene der Hierarchie und das Zurückgeben des Typgraphen an eine nächsthöhere Ebene der Hierarchie.

[1644] Beispiel 41 umfasst den Gegenstand von einem der Beispiele 33 bis 40. In Beispiel 41 umfasst die Typinspektion das Ausführen einer rekursiven Attestierung der das zusammengesetzte Objekt bildenden Sub-Objekte.

[1645] Beispiel 42 umfasst den Gegenstand von einem der Beispiele 33 bis 41. In Beispiel 42 umfasst die rekursive Attestierung das Senden einer Typattestierungsanforderung von jeder Ebene zu Objekten auf einer nächstniedrigeren Ebene, das Zurückgeben eines Typgraphen aller Objekte, welche die Objekte auf einer bestimmten Ebene einer Hierarchie ausmachen, an eine nächsthöhere Ebene und das Bilden eines Gesamttypgraphen im zusammengesetzten Objekt.

[1646] Beispiel 43 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, um einen Prozessor anzuweisen, einen Typgraphen von ein zusammengesetztes Objekt bildenden Objekten zu erzeugen, einen Typnamen für das zusammengesetzte Objekt zu berechnen und den Typnamen und Typgraphen in einer Blockchain aufzuzeichnen.

[1647] Beispiel 44 umfasst den Gegenstand von Beispiel 43. In Beispiel 44 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, eine rekursive Typintrospektion der das zusammengesetzte Objekt bildenden Objekte auszuführen.

[1648] Beispiel 45 umfasst den Gegenstand von Beispiel 43 oder 44. In Beispiel 45 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, eine rekursive Typattestierung der das zusammengesetzte Objekt bildenden Objekte auszuführen.

[1649] Beispiel 46 umfasst den Gegenstand von einem der Beispiele 43 bis 45. In Beispiel 46 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, den Typnamen zu erzeugen, falls dieser in der Blockchain nicht vorhanden ist.

[1650] Beispiel 47 umfasst den Gegenstand von einem der Beispiele 43 bis 46. In Beispiel 47 umfasst das nichtflüchtige maschinenlesbare Medium das Senden einer EPID-Beitrittsanforderung zu einem Sub-Objekt mit einem Typberechtigungs nachweis.

[1651] Beispiel 48 umfasst eine Vorrichtung. Die Vorrichtung umfasst eine Koalitionsgruppe, einschließlich eines Koalitionsgruppen-Nameservers zum Bereitstellen von Namen für die Koalitionsgruppe bildende Objekte, eine Koalitionsgruppenmitgliedsliste der zur Koalitionsgruppe gehörenden Objekte und eine Blockchain, welche die Namen der die Koalitionsgruppe bildenden Objekte aufzeichnet.

[1652] Beispiel 49 umfasst den Gegenstand von Beispiel 48. In Beispiel 49 umfasst die Vorrichtung einen Veröffentlichender zur Ausstrahlung einer Koalitionsgruppenexistenz.

[1653] Beispiel 50 umfasst den Gegenstand von Beispiel 48 oder 49. In Beispiel 50 umfasst die Vorrichtung einen Berechtigungsnachweisprüfer zur Bestätigung eines von einem Objekt empfangenen Identitätsberechtigungs nachweis.

[1654] Beispiel 51 umfasst den Gegenstand von einem der Beispiele 48 bis 50. In Beispiel 51 umfasst die Vorrichtung einen EPID-Server, um einem Objekt einen Berechtigungsnachweis für den Beitritt zur Koalitionsgruppe bereitzustellen.

[1655] Beispiel 52 umfasst den Gegenstand von einem der Beispiele 48 bis 51. In Beispiel 52 umfasst die Vorrichtung einen Vorrichtungseigentümer zum Verifizieren eines Identitätsberechtigungs nachweises von einem Objekt und zum Bereitstellen eines Koalitionsgruppen-Berechtigungsnachweises für das Objekt und mehrere Objekte, die jeweils einen Koalitionsgruppen-Berechtigungsnachweis aufweisen, der die Mitgliedschaft in der Koalitionsgruppe angibt.

[1656] Beispiel 53 umfasst den Gegenstand von einem der Beispiele 48 bis 52. In Beispiel 53 werden Objekte in der Koalitionsgruppe nach dem Ort gruppiert.

- [1657]** Beispiel 54 umfasst den Gegenstand von einem der Beispiele 48 bis 53. In Beispiel 54 werden Objekte in der Koalitionsgruppe nach der Funktion gruppiert.
- [1658]** Beispiel 55 umfasst ein Verfahren zur Bildung einer Koalitionsgruppe in einem IoT-Netz. Das Verfahren zur Bildung einer Koalitionsgruppe in einem IoT-Netz umfasst das Definieren einer Koalitionsgruppe, das Empfangen einer Anforderung von einem Objekt, der Koalitionsgruppe beizutreten, und das Ausgeben von Koalitionsgruppen-Berechtigungsnachweisen an das Objekt.
- [1659]** Beispiel 56 umfasst den Gegenstand von Beispiel 55. In Beispiel 56 umfasst das Definieren der Koalitionsgruppe das Gruppieren von Vorrichtungen nach dem Ort.
- [1660]** Beispiel 57 umfasst den Gegenstand von Beispiel 55 oder 56. In Beispiel 57 umfasst das Definieren der Koalitionsgruppe das Gruppieren von Vorrichtungen nach der Funktion.
- [1661]** Beispiel 58 umfasst den Gegenstand von einem der Beispiele 55 bis 57. In Beispiel 58 umfasst das Verfahren das Veröffentlichen der Koalitionsgruppe für eine Blockchain, falls die Koalitionsgruppe nicht entdeckbar ist.
- [1662]** Beispiel 59 umfasst den Gegenstand von einem der Beispiele 55 bis 58. In Beispiel 59 umfasst das Verfahren das Verifizieren der Anforderung vor dem Ausgeben der Koalitionsgruppen-Berechtigungsnachweise.
- [1663]** Beispiel 60 umfasst den Gegenstand von einem der Beispiele 59 bis 59. In Beispiel 60 umfasst das Verfahren das Verifizieren der Anforderung durch Bestätigen, dass die Anforderung einen gültigen Identitäts-berechtigungsnachweis umfasst.
- [1664]** Beispiel 61 umfasst den Gegenstand von einem der Beispiele 59 bis 60. In Beispiel 61 umfasst das Verfahren das Verifizieren der Anforderung durch Bestätigen, dass die Anforderung einen gültigen Instanz-berechtigungsnachweis umfasst.
- [1665]** Beispiel 62 umfasst den Gegenstand von einem der Beispiele 59 bis 61. In Beispiel 62 umfasst das Verfahren das Verifizieren der Anforderung durch Bestätigen, dass die Anforderung einen gültigen Typ-berechtigungsnachweis umfasst.
- [1666]** Beispiel 63 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, um einen Prozessor anzuweisen, eine Koalitionsgruppe zu definieren, die Koalitionsgruppe für eine Blockchain zu veröffentlichen und eine Beitrittsanforderung von einem Objekt zu akzeptieren.
- [1667]** Beispiel 64 umfasst den Gegenstand von Beispiel 63. In Beispiel 64 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, zu bestätigen, dass die Koalitionsgruppe entdeckbar ist.
- [1668]** Beispiel 65 umfasst den Gegenstand von Beispiel 63 oder 64. In Beispiel 65 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, festzustellen, ob die Beitrittsanforderung vom Objekt gültig ist.
- [1669]** Beispiel 66 umfasst den Gegenstand von einem der Beispiele 63 bis 65. In Beispiel 66 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, Berechtigungsnachweise ansprechend auf eine gültige Beitrittsanforderung zu erteilen.
- [1670]** Beispiel 67 umfasst den Gegenstand von einem der Beispiele 63 bis 66. In Beispiel 67 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, EPID-Berechtigungsnachweise zu erzeugen.
- [1671]** Beispiel 68 umfasst eine Vorrichtung. Die Vorrichtung umfasst eine vertrauenswürdige Kommunikationsumgebung, einschließlich eines primären Teilnehmers mit einem Gruppenerzeuger zum Einleiten der Erzeugung einer vertrauenswürdigen Gruppe und eines verteilten Ledgers zum Speichern von Identitäten und Berechtigungsnachweisen für Gruppenmitglieder. Die Vorrichtung umfasst auch einen sekundären Teilnehmer mit Kommunikationsberechtigungsnachweisen für die vom primären Teilnehmer bereitgestellte vertrauenswürdige Gruppe.

[1672] Beispiel 69 umfasst den Gegenstand von Beispiel 68. In Beispiel 69 umfassen die Kommunikationsberechtigungs-nachweise einen privaten Schlüssel für die vertrauenswürdige Gruppe und einen vom verteilten Ledger erhaltenen Transaktionsschlüssel.

[1673] Beispiel 70 umfasst den Gegenstand von Beispiel 68 oder 69. In Beispiel 70 umfasst der primäre Teilnehmer eine Beitrittsanforderung für eine Verteilte-Ledger-Spezifikationsstelle bzw. Distributed Ledger Enumeration Authority (DLEA), wobei die Beitrittsanforderung einen mit einem privaten Schlüssel für den primären Teilnehmer signierten vertrauenswürdigen Gruppennamen umfasst.

[1674] Beispiel 71 umfasst den Gegenstand von einem der Beispiele 68 bis 70. In Beispiel 71 umfasst die Vorrichtung einen Verteilte-Ledger-Spezifikationsstellen-(DLEA)-Zugreifer zur Feststellung, ob ein vertrauenswürdiger Gruppenname erzeugt wurde.

[1675] Beispiel 72 umfasst den Gegenstand von einem der Beispiele 68 bis 71, einschließlich des verteilten Ledgers zu 72. In Beispiel 72 umfasst der verteilte Ledger einen öffentlichen Schlüssel für die vertrauenswürdige Gruppe und eine Permissionierungsrichtlinie.

[1676] Beispiel 73 umfasst den Gegenstand von einem der Beispiele 68 bis 72. In Beispiel 73 umfasst der primäre Teilnehmer einen Schlüsselerzeuger zur Erzeugung eines Schlüssels zumindest teilweise auf der Grundlage eines vertrauenswürdigen Gruppennamens.

[1677] Beispiel 74 umfasst den Gegenstand von einem der Beispiele 68 bis 73. In Beispiel 74 umfasst die Vorrichtung einen Attestierungsvalidierer zum Validieren einer Beitrittsanforderung vom sekundären Teilnehmer.

[1678] Beispiel 75 umfasst den Gegenstand von einem der Beispiele 68 bis 74. In Beispiel 75 umfasst die Vorrichtung eine Gruppenbeitrittseinrichtung zum Ausgeben der Kommunikationsberechtigungs-nachweise an den sekundären Teilnehmer.

[1679] Beispiel 76 umfasst den Gegenstand von einem der Beispiele 68 bis 75. In Beispiel 76 umfasst die Vorrichtung einen tertiären Teilnehmer, der sekundäre Kommunikationsberechtigungs-nachweise für die vom sekundären Teilnehmer bereitgestellte vertrauenswürdige Gruppe umfasst.

[1680] Beispiel 77 umfasst den Gegenstand von einem der Beispiele 68 bis 76. In Beispiel 77 umfassen die sekundären Kommunikationsberechtigungs-nachweise einen privaten Schlüssel für die Gruppe und einen sekundären Transaktionsschlüssel.

[1681] Beispiel 78 umfasst den Gegenstand von einem der Beispiele 68 bis 77. In Beispiel 78 umfasst die Vorrichtung mehrere sekundäre Teilnehmer, die vom primären Teilnehmer ausgegebene Kommunikationsberechtigungs-nachweise umfassen.

[1682] Beispiel 79 umfasst den Gegenstand von einem der Beispiele 68 bis 78. In Beispiel 79 umfasst die Vorrichtung mehrere tertiäre Teilnehmer, die jeweils vom primären Teilnehmer ausgegebene sekundäre Kommunikationsberechtigungs-nachweise umfassen.

[1683] Beispiel 80 umfasst den Gegenstand von einem der Beispiele 68 bis 79. In Beispiel 80 umfasst der verteilte Ledger mit einem Gruppenschlüssel und einem privaten Schlüssel für einen Teilnehmer signierte Transaktionsdaten.

[1684] Beispiel 81 umfasst ein Verfahren zum Sichern von Kommunikationstransaktionen in einem IoT-Netz. Das Verfahren zum Sichern von Kommunikationstransaktionen in einem IoT-Netz umfasst das Feststellen durch einen ersten Teilnehmer, dass eine Gruppe von Teilnehmern mit Integritätsgarantien kommunizieren kann, das Reservieren eines Namens für die Gruppe von einer Verteilte-Ledger-Spezifikationsstelle (DLEA), das Einrichten eines verteilten Ledgers für die Gruppe unter Verwendung des Namens und das Bereitstellen eines privaten Schlüssels für die Gruppe an einen zweiten Teilnehmer.

[1685] Beispiel 82 umfasst den Gegenstand von Beispiel 81. In Beispiel 82 umfasst das Reservieren des Namens das Senden des Namens und eines öffentlichen Schlüssels für den ersten Teilnehmer zur DLEA in einer unter Verwendung eines privaten Schlüssels für den ersten Teilnehmer signierten Nachricht, das Feststellen, dass die Gruppe erzeugt wurde, wenn die DLEA den Namen an einen öffentlichen verteilten Ledger übergibt,

und das Einrichten eines öffentlichen Gruppenschlüssels unter Verwendung eines Enhanced-Privacy-Identification(EPID)-Systems.

[1686] Beispiel 83 umfasst den Gegenstand von Beispiel 81 oder 82. In Beispiel 83 umfasst das Einrichten des verteilten Ledgers für die Gruppe das Übergeben einer Transaktion vom ersten Teilnehmer an den verteilten Gruppen-Ledger, wobei die Transaktion einen öffentlichen Gruppenschlüssel und eine durch einen Transaktionschlüssel für den ersten Teilnehmer signierte Permissionierungsrichtlinie umfasst.

[1687] Beispiel 84 umfasst den Gegenstand von einem der Beispiele 81 bis 83. In Beispiel 84 umfasst das Bereitstellen eines privaten Schlüssels das Empfangen einer Beitrittsanforderung vom zweiten Teilnehmer, der eine Erlaubnis fordert, der Gruppe beizutreten, und das Validieren der Vertrauenswürdigkeit des zweiten Teilnehmers.

[1688] Beispiel 85 umfasst den Gegenstand von einem der Beispiele 81 bis 84. In Beispiel 85 umfasst das Validieren der Vertrauenswürdigkeit das Verifizieren eines zum Signieren der Beitrittsanforderung verwendeten Herstellerschlüssels.

[1689] Beispiel 86 umfasst den Gegenstand von einem der Beispiele 81 bis 85. In Beispiel 86 umfasst das Verfahren das Erzeugen eines zweiten privaten Schlüssels für die Gruppe im zweiten Teilnehmer, wobei der zweite private Schlüssel unter einem öffentlichen Gruppenschlüssel liegt, das Senden einer Nachricht zum ersten Teilnehmer, wobei die Nachricht ein mit dem zweiten privaten Schlüssel signierter öffentlicher Schlüssel für den zweiten Teilnehmer ist, und das Übergeben einer Transaktion an den verteilten Gruppen-Ledger, wobei die Transaktion den mit dem privaten Schlüssel signierten öffentlichen Schlüssel des zweiten Teilnehmers umfasst.

[1690] Beispiel 87 umfasst den Gegenstand von einem der Beispiele 81 bis 86. In Beispiel 87 umfasst das Verfahren das Erzeugen einer Beitrittsanforderung in einem dritten Teilnehmer, wobei die Beitrittsanforderung einen mit einem privaten Schlüssel für den dritten Teilnehmer signierten Transaktionschlüssel des dritten Teilnehmers umfasst, das Senden der Beitrittsanforderung zum zweiten Teilnehmer, das Signieren der Beitrittsanforderung durch den zweiten Teilnehmer mit einem öffentlichen Schlüssel für den dritten Teilnehmer, einem Transaktionschlüssel für den zweiten Teilnehmer und dem Gruppenschlüssel zur Erzeugung einer signierten Transaktion und das Senden der signierten Transaktion zum dritten Teilnehmer zurück.

[1691] Beispiel 88 umfasst den Gegenstand von einem der Beispiele 81 bis 87. In Beispiel 88 umfasst das Verfahren das Aufnehmen von Transaktionsdaten vom zweiten Teilnehmer in die signierte Transaktion.

[1692] Beispiel 89 umfasst den Gegenstand von einem der Beispiele 81 bis 88. In Beispiel 89 umfasst das Verfahren das Signieren der signierten Transaktion mit einem privaten Gruppenschlüssel für den dritten Teilnehmer und das Übergeben der signierten Transaktion an den verteilten Gruppen-Ledger.

[1693] Beispiel 90 umfasst den Gegenstand von einem der Beispiele 81 bis 89. In Beispiel 90 umfasst das Verfahren das Signieren von Transaktionsdaten beim zweiten Teilnehmer unter Verwendung des privaten Gruppenschlüssels für den zweiten Teilnehmer und das Übermitteln der Transaktionsdaten zum verteilten Gruppen-Ledger.

[1694] Beispiel 91 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, um einen Prozessor anzuweisen, festzustellen, dass eine Gruppe Integritäts-garantien hat, einen Gruppennamen mit einer Verteilte-Ledger-Spezifizierungsstelle (DLEA) zu reservieren, einen öffentlichen Gruppenschlüssel und eine Permissionierungsrichtlinie zu erzeugen und den Gruppennamen und den öffentlichen Gruppenschlüssel an einen verteilten Gruppen-Ledger zu übergeben.

[1695] Beispiel 92 umfasst den Gegenstand von Beispiel 91. In Beispiel 92 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, eine Beitrittsanforderung von einem zweiten Teilnehmer zu validieren und eine Beitrittsnachricht zum zweiten Teilnehmer zu senden, wobei die Beitrittsanforderung einen privaten Gruppenschlüssel umfasst.

[1696] Beispiel 93 umfasst den Gegenstand von Beispiel 91 oder 92. In Beispiel 93 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, Transaktionsdaten mit einem privaten Gruppenschlüssel zu signieren und die signierten Transaktionsdaten an den verteilten Gruppen-Ledger zu übergeben.

[1697] Beispiel 94 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein IoT-Netz, wobei das IoT-Netz eine vertrauenswürdige Ausführungsumgebung bzw. Trusted Execution Environment (TEE) umfasst. Diese umfasst auch eine Chain-Geschichte für eine Blockchain, wobei die Chain-Geschichte eine Whitelist von Hash-Signaturen, eine Vertrauens-Root für die Verkettung bzw. Root-of-Trust for Chaining (RTC) zum Bereitstellen der Chain-Geschichte für lokale Rechen-Vertrauens-Roots und eine Vertrauens-Root-für-die-Archivierung bzw. Root-of-Trust for Archives (RTA) zum Bereitstellen einer Archivierungsfunktion für beschränkte Vorrichtungen im IoT-Netz umfasst.

[1698] Beispiel 95 umfasst den Gegenstand von Beispiel 94. In Beispiel 95 umfasst die TEE einen Vertrauens-Root-Messer bzw. Root-of-Trust Measurer (RTM) zum Verifizieren eines ersten ladbaren Objekts in einem System.

[1699] Beispiel 96 umfasst den Gegenstand von Beispiel 94 oder 95. In Beispiel 96 umfasst die TEE eine Vertrauens-Root für die Mitteilung bzw. Root of Trust for Reporting (RTR) zum Attestieren der Werte in einer Vertrauens-Root für die Speicherung und die Vertrauens-Root für die Speicherung bzw. Root of Trust for Storage (RTS) zum Speichern von Werten für Vertrauens-Root-Vorrichtungen.

[1700] Beispiel 97 umfasst den Gegenstand von einem der Beispiele 94 bis 96. In Beispiel 97 umfasst die TEE eine Blockchain-Logik zum Migrieren der Blockchain-Geschichte zu anderen Vorrichtungen und zum Verifizieren von Chain-Geschichten von anderen Vorrichtungen.

[1701] Beispiel 98 umfasst den Gegenstand von einem der Beispiele 94 bis 97. In Beispiel 98 umfasst die TEE eine Whitelist-Geschichte, die aktuelle Konfigurationen, frühere Konfigurationen oder vorhergesehene Konfigurationen oder Kombinationen davon umfasst.

[1702] Beispiel 99 umfasst den Gegenstand von einem der Beispiele 94 bis 98. In Beispiel 99 umfasst die TEE eine Messgeschichte zum Aufzeichnen während eines Boot-Prozesses vorgenommener Messungen.

[1703] Beispiel 100 umfasst den Gegenstand von einem der Beispiele 94 bis 99. In Beispiel 100 umfasst die Messgeschichte Messprotokolle von mehreren Boot-Sequenzen.

[1704] Beispiel 101 umfasst den Gegenstand von einem der Beispiele 94 bis 100. In Beispiel 101 umfasst die Vorrichtung ein Mesh-Netz von Vorrichtungen, die in eine vertrauenswürdige Umgebung booten.

[1705] Beispiel 102 umfasst den Gegenstand von einem der Beispiele 94 bis 101. In Beispiel 102 umfasst die Chain-Geschichte einen Blockchain-Block, der mehrere Werte von Plattformsteuerregistern bzw. Platform Control Registers (PCR) umfasst, die mit einem Attestierungsschlüssel signiert sind.

[1706] Beispiel 103 umfasst den Gegenstand von einem der Beispiele 94 bis 102. In Beispiel 103 umfasst der Blockchain-Block einen Block-Signaturschlüssel.

[1707] Beispiel 104 umfasst den Gegenstand von einem der Beispiele 94 bis 103. In Beispiel 104 umfasst die Vorrichtung einen Bildablagespeicher, in dem Whitelist-Werte gespeichert werden.

[1708] Beispiel 105 umfasst den Gegenstand von einem der Beispiele 94 bis 104. In Beispiel 105 umfasst die Chain-Geschichte einen Blockchain-Block, der mehrere Manifeste von Whitelist-Bildern umfasst, die jeweils mit einem Herstellerattestierungsschlüssel signiert sind.

[1709] Beispiel 106 umfasst den Gegenstand von einem der Beispiele 94 bis 105. In Beispiel 106 umfasst der Blockchain-Block einen Block-Signaturschlüssel.

[1710] Beispiel 107 umfasst ein Verfahren zum sicheren Booten einer Vorrichtung in einem IoT-Netz. Das Verfahren zum sicheren Booten einer Vorrichtung in einem IoT-Netz umfasst das Messen eines Codeobjekts, bevor das Codeobjekt ausgeführt wird, das Vergleichen der Messung mit einem aus einer Blockchain abgerufenen bekannten guten Bild und das Ausführen des Codeobjekts, falls die Messung mit dem bekannten guten Bild übereinstimmt.

[1711] Beispiel 108 umfasst den Gegenstand von Beispiel 107. In Beispiel 108 umfasst das Verfahren das Vergleichen der Messung mit einem aus einer Blockchain abgerufenen bekannten schlechten Bild, das Setzen

der Vorrichtung unter Quarantäne, falls die Messung mit dem bekannten schlechten Bild übereinstimmt, und das Reparieren des Codes, falls die Messung mit dem bekannten schlechten Bild übereinstimmt.

[1712] Beispiel 109 umfasst den Gegenstand von Beispiel 107 oder 108. In Beispiel 109 umfasst das Verfahren das Verfolgen einer vordefinierten Richtlinie, falls die Messung nicht mit einem bekannten guten Bild oder einem bekannten schlechten Bild übereinstimmt.

[1713] Beispiel 110 umfasst den Gegenstand von einem der Beispiele 107 bis 109. In Beispiel 110 weist die vordefinierte Richtlinie die Vorrichtung an, in einen nicht vertrauenswürdigen Zustand zu booten und nicht mit vertrauenswürdigen Vorrichtungen zu kommunizieren.

[1714] Beispiel 111 umfasst den Gegenstand von einem der Beispiele 107 bis 110. In Beispiel 111 umfasst das Verfahren das Erhalten des Bilds für eine Messung von einem Cloud-Ablagespeicher, das Bestätigen, dass die Signatur für das Bild gültig ist, das Bestätigen, dass das Bild ein Hash einer Bootchain ist, das Signieren eines Manifests für das Bild, das Hinzufügen des Bilds zu einer Whitelist und das Übergeben der Whitelist an eine Blockchain.

[1715] Beispiel 112 umfasst den Gegenstand von einem der Beispiele 107 bis 111. In Beispiel 112 umfasst das Verfahren das Feststellen, dass ein Bild nicht von einer Bootchain ist, das Feststellen, dass das Bild ein Angriffsbild ist, das Hinzufügen des Angriffsbilds zu einer Blacklist und das Übergeben der Blacklist an eine Blockchain.

[1716] Beispiel 113 umfasst den Gegenstand von einem der Beispiele 107 bis 112. In Beispiel 113 umfasst das Verfahren das Feststellen, dass ein Bild nicht von einer Bootchain ist, das Feststellen, dass das Bild ein unbekanntes Bild ist, das Hinzufügen des unbekannten Bilds seiner nicht klassifizierten Liste und das Übergeben der nicht klassifizierten Liste an eine Blockchain.

[1717] Beispiel 114 umfasst den Gegenstand von einem der Beispiele 107 bis 113. In Beispiel 114 umfasst das Verfahren das Erzeugen eines Blocks, der ein Bild eines erfolgreich ausgeführten Codeblocks umfasst, und das Übergeben des Blocks an die Blockchain.

[1718] Beispiel 115 umfasst den Gegenstand von einem der Beispiele 107 bis 114. In Beispiel 115 umfasst das Verfahren das Erzeugen eines Blocks, der ein Bild eines abgelehnten Codeblocks umfasst, und das Übergeben des Blocks an die Blockchain.

[1719] Beispiel 116 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, um einen Prozessor anzuweisen, ein Codeobjekt zu messen, bevor das Codeobjekt ausgeführt wird, um eine Messung zu erhalten, das Vergleichen der Messung mit einem von einer Blockchain abgerufenen bekannten guten Bild und das Klassifizieren des Codeobjekts als gut, falls es eine Übereinstimmung zwischen der Messung und dem bekannten guten Bild gibt.

[1720] Beispiel 117 umfasst den Gegenstand von Beispiel 116. In Beispiel 117 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, die Messung mit einem bekannten schlechten Bild zu vergleichen, und das Verhindern der Ausführung des Codeobjekts, falls die Messung mit dem bekannten schlechten Bild übereinstimmt.

[1721] Beispiel 118 umfasst den Gegenstand von Beispiel 116 oder 117. In Beispiel 118 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, um den Prozessor anzuweisen, eine Blockchain, die eine Chain-Geschichte umfasst, zu unterhalten, und Vertrauens-Root-Messungen in der Blockchain zu halten.

[1722] Beispiel 119 umfasst eine Vorrichtung. Die Vorrichtung umfasst eine IoT-Vorrichtung, wobei die IoT-Vorrichtung ein Dienstbarkeitssystem zum Steuern einer Netzschicht, die verwendet wird, um Zugriffsrechte zum Ermöglichen, dass Verkehr durch die IoT-Vorrichtung fließt, bereitzustellen, und eine Dienstbarkeitsschicht in einem Netzstapel, wobei die Dienstbarkeitsschicht dafür ausgelegt ist, Pakete unterhalb der Anwendungsschicht durch die IoT-Vorrichtung zu leiten.

[1723] Beispiel 120 umfasst den Gegenstand von Beispiel 119. In Beispiel 120 beruhen die Zugriffsrechte zumindest teilweise auf Identität, Autorisierung, Bezahlung oder Zeit.

[1724] Beispiel 121 umfasst den Gegenstand von Beispiel 119 oder 120. In Beispiel 121 umfasst die Vorrichtung ein Mesh-Netz von Vorrichtungen, die über zwei Domains koppeln, wobei die beiden Domains jeweils ein anderes Kommunikationsprotokoll verwenden.

[1725] Beispiel 122 umfasst den Gegenstand von einem der Beispiele 119 bis 121. In Beispiel 122 umfasst die IoT-Vorrichtung eine Edge-Vorrichtung zwischen den beiden Domains.

[1726] Beispiel 123 umfasst den Gegenstand von einem der Beispiele 119 bis 122. In Beispiel 123 übersetzt das Dienstbarkeitssystem ein Paket von einem durch eine erste Domain verwendeten Protokoll in ein durch eine zweite Domain verwendetes Protokoll.

[1727] Beispiel 124 umfasst den Gegenstand von einem der Beispiele 119 bis 123. In Beispiel 124 umfasst die IoT-Vorrichtung einen Anwendungs-/Dienstmanager zum Austausch mit Anwendungen und zur Behandlung des Netzstapels für Kommunikationen.

[1728] Beispiel 125 umfasst den Gegenstand von einem der Beispiele 119 bis 124. In Beispiel 125 dient der Anwendungs-/Dienstmanager dazu, eine Anwendungsbetriebsumgebung in der Art eines Betriebssystems und einer Programmierschnittstelle für die Übermittlung von Paketen zu anderen Mesh-Vorrichtungen bereitzustellen.

[1729] Beispiel 126 umfasst den Gegenstand von einem der Beispiele 119 bis 125. In Beispiel 126 umfasst die IoT-Vorrichtung einen Zahlungs-/Kreditmanager zur Behandlung von Mikrozahlungen für Kommunikationen.

[1730] Beispiel 127 umfasst den Gegenstand von einem der Beispiele 119 bis 126. In Beispiel 127 umfasst die IoT-Vorrichtung ein Zeitsteuerungs-Untersystem für die Zeitsteuerung von Kommunikationen.

[1731] Beispiel 128 umfasst den Gegenstand von einem der Beispiele 119 bis 127. In Beispiel 128 umfasst die IoT-Vorrichtung eine Blockchain-Logik zum Zugreifen auf eine Blockchain von Kommunikationserlaubnissen, Zahlungen oder Guthaben oder eine Kommunikation davon und zum Unterhalten von dieser.

[1732] Beispiel 129 umfasst den Gegenstand von einem der Beispiele 119 bis 128. In Beispiel 129 umfasst die IoT-Vorrichtung eine Blockchain-Logik zur Migration einer Blockchain zu anderen Vorrichtungen und zum Verifizieren einer von einer anderen Vorrichtung übertragenen Blockchain.

[1733] Beispiel 130 umfasst ein Verfahren zur Verwendung eines Dienstbarkeitssystems für Kommunikationen in einem IoT-Netz. Das Verfahren zur Verwendung eines Dienstbarkeitssystems für Kommunikationen in einem IoT-Netz umfasst das Übertragen eines Pakets von einer ersten Vorrichtung zu einer zweiten Vorrichtung durch eine Dienstbarkeitsschicht, wobei sich die Dienstbarkeitsschicht in einem Netzstapel unterhalb einer Anwendungsschicht befindet.

[1734] Beispiel 131 umfasst den Gegenstand von einem der Beispiele 130 bis 131. In Beispiel 133 umfasst das Verfahren das Senden einer Weiterleitungsanforderung zu einem Vorrichtungsregistrator, das Empfangen einer Erlaubnis vom Vorrichtungsregistrator und das Leiten eines Pakets durch die Dienstbarkeitsschicht.

[1735] Beispiel 132 umfasst den Gegenstand von Beispiel 130. In Beispiel 132 umfasst das Verfahren das Senden einer Benachrichtigung, deren Weiterleitung abgeschlossen wurde, zum Vorrichtungsregistrator.

[1736] Beispiel 133 umfasst den Gegenstand von Beispiel 130 oder 132. In Beispiel 133 umfasst das Verfahren das Akzeptieren einer Zahlung vom Vorrichtungsregistrator.

[1737] Beispiel 134 umfasst den Gegenstand von einem der Beispiele 130 bis 133. In Beispiel 134 umfasst das Verfahren das Senden einer Weiterleitungsanforderung zu einem Vorrichtungsregistrator, das Empfangen einer Ablehnungsentscheidung vom Vorrichtungsregistrator und das Löschen des Pakets.

[1738] Beispiel 135 umfasst den Gegenstand von einem der Beispiele 130 bis 134. In Beispiel 135 umfasst das Verfahren das Nachschlagen des Antragstellers zur Feststellung, ob die Anforderung autorisiert ist, das Prüfen eines Datensatzes zur Feststellung, ob für die Datenübertragung genügend Guthaben existiert, und das Senden einer Nachricht zum Autorisieren der Übertragung des Pakets.

[1739] Beispiel 136 umfasst den Gegenstand von einem der Beispiele 130 bis 135. In Beispiel 136 umfasst das Verfahren das Nachschlagen des Antragstellers zum Feststellen, ob die Anforderung autorisiert ist, das Prüfen eines Datensatzes zur Berechnung einer Mietzeit und das Senden einer Nachricht, um eine zeitlich begrenzte Miete für Kommunikationen zu gewähren.

[1740] Beispiel 137 umfasst den Gegenstand von einem der Beispiele 130 bis 136. In Beispiel 137 umfasst das Verfahren das Freigeben einer treuhänderischen Zahlung für die Übertragung des Pakets.

[1741] Beispiel 138 umfasst den Gegenstand von einem der Beispiele 130 bis 137. In Beispiel 138 umfasst das Verfahren das Ungültigmachen einer zeitlich begrenzten Miete für Kommunikationen und das Senden einer Nachricht zur Benachrichtigung einer Vorrichtung über das Ablaufen.

[1742] Beispiel 139 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, ein Paket zumindest teilweise auf der Grundlage einer Autorisierung für das Paket durch eine Dienstbarkeitsschicht in einer Internetder-Dinge bzw. Internet-of-Things(IoT)-Vorrichtung zu leiten.

[1743] Beispiel 140 umfasst den Gegenstand von Beispiel 139. In Beispiel 140 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Weiterleitungsanforderung zu einem Vorrichtungsregistrator zu senden, um die Autorisierung zum Weiterleiten des Pakets zu erhalten.

[1744] Beispiel 141 umfasst den Gegenstand von Beispiel 139 oder 140. In Beispiel 141 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Abschlussbenachrichtigung zu einem Vorrichtungsregistrator zu senden, nachdem eine das Paket aufweisende Kommunikation abgeschlossen wurde.

[1745] Beispiel 142 umfasst den Gegenstand von einem der Beispiele 139 bis 141. In Beispiel 142 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Zahlung zur Übertragung des Pakets von einem Vorrichtungsregistrator zu akzeptieren.

[1746] Beispiel 143 umfasst den Gegenstand von einem der Beispiele 139 bis 142. In Beispiel 143 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Antragsteller zur Feststellung einer Autorisierung nachzuschlagen.

[1747] Beispiel 144 umfasst den Gegenstand von einem der Beispiele 139 bis 143. In Beispiel 144 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, ob genügend Guthaben vorhanden ist, um für die Übertragung des Pakets zu bezahlen.

[1748] Beispiel 145 umfasst den Gegenstand von einem der Beispiele 139 bis 144. In Beispiel 145 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Mietzeit für eine das Paket aufweisende Kommunikation zu berechnen.

[1749] Beispiel 146 umfasst den Gegenstand von einem der Beispiele 139 bis 145. In Beispiel 146 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Autorisierung, eine Mietzeit oder beide zur IoT-Vorrichtung zu übermitteln.

[1750] Beispiel 147 umfasst den Gegenstand von einem der Beispiele 139 bis 146. In Beispiel 147 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Miete zu beenden und die IoT-Vorrichtung über die Beendigung zu informieren.

[1751] Beispiel 148 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung, einschließlich eines Herkunftsnachweis- bzw. Proof-of-Provenance(POP)-Schlüsselberechners zum Berechnen eines PoP-Schlüssels, der identifiziert, dass ein Paket von der IoT-Vorrichtung übertragen wurde, eines Paketerzeugers zum Anhängen des PoP-Schlüssels an das Paket und eines Kommunikators zum Senden des Pakets zu einer anderen Vorrichtung umfasst.

[1752] Beispiel 149 umfasst den Gegenstand von Beispiel 148. In Beispiel 149 umfasst die IoT-Vorrichtung eine Eingangsschlüssel-Berechnungseinrichtung zum Bestimmen eines Eingangsschlüssels.

[1753] Beispiel 150 umfasst den Gegenstand von Beispiel 148 oder 149. In Beispiel 150 umfasst die Eingangsschlüssel-Berechnungseinrichtung einen Paketleser zum Identifizieren eines vorhergehenden PoP-Schlüssels im Paket und zum Festlegen des Eingangsschlüssels für den vorhergehenden PoP-Schlüssel.

[1754] Beispiel 151 umfasst den Gegenstand von einem der Beispiele 148 bis 150. In Beispiel 151 umfasst die Eingangsschlüssel-Berechnungseinrichtung einen Zufallszahlengenerator zur Berechnung des Eingangsschlüssels.

[1755] Beispiel 152 umfasst den Gegenstand von einem der Beispiele 148 bis 151. In Beispiel 152 umfasst die IoT-Vorrichtung einen Vorrichtungsschlüssel.

[1756] Beispiel 153 umfasst den Gegenstand von einem der Beispiele 148 bis 152. In Beispiel 153 ist der Vorrichtungsschlüssel ein in die Vorrichtung programmierter Herstellerschlüssel.

[1757] Beispiel 154 umfasst den Gegenstand von einem der Beispiele 148 bis 153. In Beispiel 154 wird der Vorrichtungsschlüssel von einer Blockchain erhalten.

[1758] Beispiel 155 umfasst den Gegenstand von einem der Beispiele 148 bis 154. In Beispiel 155 umfasst die PoP-Schlüsselberechnungseinrichtung eine Funktion zum Kombinieren eines Eingangsschlüssels und eines Vorrichtungsschlüssels zur Bildung des PoP-Schlüssels.

[1759] Beispiel 156 umfasst den Gegenstand von einem der Beispiele 148 bis 155. In Beispiel 156 umfasst die IoT-Vorrichtung einen PoP-Sequenzverifizierer zum Bestätigen, dass eine Sequenz von PoP-Schlüsseln in einem Paket autorisiert ist.

[1760] Beispiel 157 umfasst den Gegenstand von einem der Beispiele 148 bis 156. In Beispiel 157 umfasst die Vorrichtung mehrere Vorrichtungen, die jeweils eine PoP-Schlüsselberechnungseinrichtung zur Berechnung eines PoP-Schlüssels für ein empfangenes Paket, einen Paketerzeuger zum Anhängen des PoP-Schlüssels an das Paket und einen Kommunikator zum Senden des Pakets zu einer anderen Vorrichtung umfassen.

[1761] Beispiel 158 umfasst den Gegenstand von Beispiel 157. In Beispiel 158 umfasst der Kommunikator einen Protokollübersetzer zur Übersetzung des Pakets in ein neues Protokoll.

[1762] Beispiel 159 umfasst den Gegenstand von einem der Beispiele 157 und 158, einschließlich eines Pakets für 159. In Beispiel 159 umfasst das Paket mehrere Platzhalter für PoP-Schlüssel.

[1763] Beispiel 160 umfasst ein Verfahren zur Übertragung eines Pakets zwischen Vorrichtungen mit einem Herkunftsnachweis (PoP) für das Paket. Das Verfahren zur Übertragung eines Pakets zwischen Vorrichtungen mit einem Herkunftsnachweis (PoP) für das Paket umfasst das Empfangen des Pakets von einer Vorrichtung, das Bestimmen eines Eingangsschlüssels vom Paket, das Berechnen eines PoP-Schlüssels für das Paket, das Anhängen des PoP-Schlüssels an das Paket und das Senden des Pakets zu einer anderen Vorrichtung.

[1764] Beispiel 161 umfasst den Gegenstand von Beispiel 160. In Beispiel 161 umfasst das Verfahren das Erhalten eines Vorrichtungsschlüssels von einer Blockchain.

[1765] Beispiel 162 umfasst den Gegenstand von Beispiel 160 oder 161. In Beispiel 162 umfasst das Verfahren das Berechnen des PoP-Schlüssels als Funktion des Eingangsschlüssels und eines Vorrichtungsschlüssels.

[1766] Beispiel 163 umfasst den Gegenstand von einem der Beispiele 160 bis 162. In Beispiel 163 umfasst das Verfahren das Extrahieren eines PoP-Schlüssels aus einem Paket, die Tokenbildung des Schlüssels, das Verifizieren des PoP-Tokens und das Mitteilen eines Ergebnisses für die Verifizierung.

[1767] Beispiel 164 umfasst den Gegenstand von einem der Beispiele 160 bis 163. In Beispiel 164 umfasst das Verifizieren des Ergebnisses das Ausführen einer Funktion zum Kombinieren des PoP-Tokens mit einem gemeinsamen geheimen Schlüssel zur Bestätigung einer Identität einer sendenden Vorrichtung.

[1768] Beispiel 165 umfasst den Gegenstand von einem der Beispiele 160 bis 164. In Beispiel 165 umfasst das Mitteilen des Ergebnisses das Informieren anderer Vorrichtungen über den Fehler.

[1769] Beispiel 166 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, ein empfangenes Paket zu lesen, um einen Eingangsschlüssel zu identifizieren, einen Herkunftsnachweis(PoP)-Schlüssel zumindest teilweise auf der Grundlage des Eingangsschlüssels zu berechnen, den PoP-Schlüssel an das Paket anzuhängen und das Paket zu einer Empfangsvorrichtung zu senden.

[1770] Beispiel 167 umfasst den Gegenstand von Beispiel 166. In Beispiel 167 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Vorrichtungsschlüssel von einer Blockchain zu erhalten.

[1771] Beispiel 168 umfasst den Gegenstand von Beispiel 166 oder 167. In Beispiel 168 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Vorrichtungsschlüssel zu erzeugen.

[1772] Beispiel 169 umfasst den Gegenstand von einem der Beispiele 166 bis 168. In Beispiel 169 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Funktion am Eingangsschlüssel und einem Vorrichtungsschlüssel auszuführen, um den PoP-Schlüssel zu berechnen.

[1773] Beispiel 170 umfasst den Gegenstand von einem der Beispiele 166 bis 169. In Beispiel 170 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Sequenz von PoP-Schlüsseln im Paket zu verifizieren.

[1774] Beispiel 171 umfasst den Gegenstand von einem der Beispiele 166 bis 170. In Beispiel 171 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Ergebnis für die Verifizierung einer anderen Vorrichtung mitzuteilen.

[1775] Beispiel 172 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Kommunikator zum Akzeptieren einer Kommunikation und zum Senden der Kommunikation, einen Datenparser zum Parsen von Metadaten in einer Kommunikation, um ein Token-Bucket zu identifizieren, eine Zahlungsberechnungseinrichtung zum Berechnen einer für das Senden der Kommunikation erforderlichen Zahlung, eine Zahlungsextrahiereinrichtung zum Abziehen der Zahlung vom Token-Bucket und zum Aktualisieren der Metadaten und einen Frame-Erzeuger zur Neubildung der Kommunikation mit den aktualisierten Metadaten.

[1776] Beispiel 173 umfasst den Gegenstand von Beispiel 172. In Beispiel 173 soll der Kommunikator die Kommunikation zwischen Protokollen übersetzen oder einen Herkunftsnachweis zur Kommunikation hinzufügen oder beides ausführen.

[1777] Beispiel 174 umfasst den Gegenstand von Beispiel 172 oder 173. In Beispiel 174 umfasst die Vorrichtung ein Dienstbarkeitssystem.

[1778] Beispiel 175 umfasst den Gegenstand von einem der Beispiele 172 bis 174. In Beispiel 175 soll der Datenparser das Token-Bucket entschlüsseln.

[1779] Beispiel 176 umfasst den Gegenstand von einem der Beispiele 172 bis 175. In Beispiel 176 soll die Zahlungsberechnungseinrichtung die Zahlung zumindest teilweise auf der Grundlage der mit einem Kostenbetrag pro Byte für die Übertragung des Nutzinhalts multiplizierten Nutzinhaltsgröße berechnen.

[1780] Beispiel 177 umfasst den Gegenstand von einem der Beispiele 172 bis 176. In Beispiel 177 soll die Zahlungsberechnungseinrichtung die Zahlung zumindest teilweise auf der Grundlage eines Prioritätsniveaus für die Kommunikation berechnen.

[1781] Beispiel 178 umfasst den Gegenstand von einem der Beispiele 172 bis 177. In Beispiel 178 soll die Zahlungsextrahiereinrichtung die Zahlung von einem im Token-Bucket aufgezeichneten Guthabenbetrag abziehen.

[1782] Beispiel 179 umfasst den Gegenstand von einem der Beispiele 172 bis 178. In Beispiel 179 soll die Zahlungsextrahiereinrichtung ein Token aus dem Token-Bucket entfernen.

[1783] Beispiel 180 umfasst den Gegenstand von einem der Beispiele 172 bis 179. In Beispiel 180 soll der Frame-Erzeuger das Token-Bucket verschlüsseln und den Paketnutzinhalt und Metadaten zusammenstellen, um einen Frame zu bilden, der zumindest einen Teil der Kommunikation umfasst.

[1784] Beispiel 181 umfasst den Gegenstand von einem der Beispiele 172 bis 180. In Beispiel 181 umfasst die Vorrichtung eine Zahlungsakzeptiereinrichtung zum Akzeptieren einer Zahlung von einer Quelle, auf die im Token-Bucket Bezug genommen wird.

[1785] Beispiel 182 umfasst den Gegenstand von einem der Beispiele 172 bis 181. In Beispiel 182 umfasst die Quelle eine durch eine Bitsequenz im Token-Bucket identifizierte Blockchain.

[1786] Beispiel 183 umfasst ein Verfahren zur Verwendung eines Token-Buckets zur Bezahlung einer Weiterleitungsvorrichtung für die Übertragung einer Kommunikation. Das Verfahren zur Verwendung eines Token-Buckets zur Bezahlung einer Weiterleitungsvorrichtung für die Übertragung einer Kommunikation umfasst das Empfangen eines Frames im sendenden System, das Parsen der Paket-Frame-Metadaten zur Identifikation des Token-Buckets, das Berechnen der Zahlung für die Übertragung der Kommunikation, das Extrahieren der Zahlung aus dem Token-Bucket, wodurch ein neues Token-Bucket bereitgestellt wird, das Aktualisieren der Frame-Metadaten unter Verwendung des neuen Token-Buckets und das Weiterleiten des Frames von der Weiterleitungsvorrichtung.

[1787] Beispiel 184 umfasst den Gegenstand von Beispiel 183. In Beispiel 184 umfasst die Vorrichtung das Warnen eines Absenders, falls festgestellt wird, dass die Frame-Metadaten inkorrekt sind.

[1788] Beispiel 185 umfasst den Gegenstand von Beispiel 183 oder 184. In Beispiel 185 umfasst das Berechnen der Zahlung das Multiplizieren der Nutzinhaltsgröße mit einer Gebühr für jedes weitergeleitete Byte.

[1789] Beispiel 186 umfasst den Gegenstand von einem der Beispiele 183 bis 185. In Beispiel 186 umfasst das Berechnen der Zahlung das Belasten mit einer höheren Rate für eine Prioritäts-Weiterleitung.

[1790] Beispiel 187 umfasst den Gegenstand von einem der Beispiele 183 bis 186. In Beispiel 187 umfasst das Extrahieren der Zahlung aus dem Token-Bucket das Dekrementieren des Token-Felds um den Betrag für die Übertragung der Kommunikation.

[1791] Beispiel 188 umfasst den Gegenstand von einem der Beispiele 183 bis 187. In Beispiel 188 umfasst das Verfahren das Inkrementieren eines lokalen Zahlungsteilbetrags.

[1792] Beispiel 189 umfasst den Gegenstand von einem der Beispiele 183 bis 188. In Beispiel 189 umfasst das Verfahren das Aktualisieren eines Zahlungsspeichers mit dem lokalen Zahlungsteilbetrag.

[1793] Beispiel 190 umfasst den Gegenstand von einem der Beispiele 183 bis 189. In Beispiel 190 ist der Zahlungsspeicher eine Blockchain, welche Zahlungstransaktionen umfasst.

[1794] Beispiel 191 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Metadaten für einen Frame zu parsen, um ein Token-Bucket zu identifizieren, das Token-Bucket aus den Metadaten zu extrahieren, eine Zahlung für das Senden einer Nachricht zu einem Bestimmungsort zu berechnen, die Zahlung aus dem Token-Bucket zu extrahieren, den Frame zu aktualisieren, so dass er die Änderung im Token-Bucket aufweist, und den Frame zu einem Bestimmungsort weiterzuleiten.

[1795] Beispiel 192 umfasst den Gegenstand von Beispiel 191. In Beispiel 192 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Frame zu einer nachfolgenden Vorrichtung mit einer den endgültigen Bestimmungsort angegebenden Adresse zu senden.

[1796] Beispiel 193 umfasst den Gegenstand von Beispiel 191 oder 192. In Beispiel 193 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Zahlung zumindest teilweise auf der Grundlage der Größe des Frames und der Kosten pro übertragenem Byte zu berechnen.

[1797] Beispiel 194 umfasst den Gegenstand von einem der Beispiele 191 bis 193. In Beispiel 194 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Zahlung zumindest teilweise auf der Grundlage der Priorität für die Nachrichtenübertragung zu berechnen.

[1798] Beispiel 195 umfasst den Gegenstand von einem der Beispiele 191 bis 194. In Beispiel 195 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen im Token-Bucket aufgezeichneten Guthabenbetrag zu verringern.

[1799] Beispiel 196 umfasst den Gegenstand von einem der Beispiele 191 bis 195. In Beispiel 196 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Token aus dem Token-Bucket zu entfernen, wobei das Token eine verschlüsselte Bitsequenz umfasst.

[1800] Beispiel 197 umfasst den Gegenstand von einem der Beispiele 191 bis 196. In Beispiel 197 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, das Token-Bucket vor der Aktualisierung des Frames zu verschlüsseln.

[1801] Beispiel 198 umfasst den Gegenstand von einem der Beispiele 191 bis 197. In Beispiel 198 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Zahlung zu akzeptieren, um den Guthabenbetrag im Token-Bucket zu erhöhen.

[1802] Beispiel 199 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Diese umfasst einen Kommunikator zum Senden einer Kommunikation, die einen Ausgangs-Frame umfasst, einen Protokollbibliothekserzeuger zum Feststellen, welche Protokolle verfügbar sind, einen Frame-Analysator zum Analysieren eines Eingangs-Frames und einen Frame-Erzeuger zum Bilden des Ausgangs-Frames aus dem Eingangs-Frame.

[1803] Beispiel 200 umfasst den Gegenstand von Beispiel 199. In Beispiel 200 soll der Frame-Analysator die Frame-Länge für den Eingangs-Frame, ein Protokoll für einen Ausgangs-Frame oder beides bestimmen.

[1804] Beispiel 201 umfasst den Gegenstand von Beispiel 199 oder 200. In Beispiel 201 umfasst die Vorrichtung ein Dienstbarkeitssystem.

[1805] Beispiel 202 umfasst den Gegenstand von einem der Beispiele 199 bis 201. In Beispiel 202 umfasst die Vorrichtung eine Protokollbibliothek zum Speichern verfügbarer Protokolle.

[1806] Beispiel 203 umfasst den Gegenstand von einem der Beispiele 199 bis 202. In Beispiel 203 umfasst die Protokollbibliothek Formatierdaten für Protokolle, einschließlich Nutzinhaltsgröße, Feldlänge, Frame-Header-Format oder Frame-Footer-Format oder jeglicher Kombinationen davon.

[1807] Beispiel 204 umfasst den Gegenstand von einem der Beispiele 199 bis 203. In Beispiel 204 umfasst die Vorrichtung den Eingangs-Frame in einem ersten Protokoll und den Ausgangs-Frame in einem zweiten Protokoll, wobei der Ausgangs-Frame ein Nutzinhaltsfeld umfasst, das zumindest einen Teil des Eingangs-Frames umfasst.

[1808] Beispiel 205 umfasst den Gegenstand von einem der Beispiele 199 bis 204. In Beispiel 205 umfasst die Vorrichtung den Eingangs-Frame in einem ersten Protokoll und mehrere Ausgangs-Frames jeweils in einem zweiten Protokoll, wobei jeder der mehreren Ausgangs-Frames ein Nutzinhaltsfeld umfasst, das zumindest einen Teil des Eingangs-Frames umfasst.

[1809] Beispiel 206 umfasst ein Verfahren zum Packen eines Frames in einem ersten Protokoll in ein Nutzinhaltsfeld eines zweiten Protokolls. Das Verfahren zum Packen eines Frames in einem ersten Protokoll in ein Nutzinhaltsfeld eines zweiten Protokolls umfasst das Identifizieren eines Protokolls für einen Eingangs-Frame und eines Protokolls für einen Ausgangs-Frame, das Schreiben zumindest eines Teils des Eingangs-Frames in das Nutzinhaltsfeld des Ausgangs-Frames und das Absenden des Ausgangs-Frames zum Bestimmungsort.

[1810] Beispiel 207 umfasst den Gegenstand von Beispiel 206. In Beispiel 207 umfasst das Verfahren das Erzeugen einer Bibliothek des verfügbaren Eingangs- und Ausgangsprotokolls, wobei die Bibliothek die Größe eines Frames in jedem Protokoll, die Nutzinhaltsfeldgröße eines Frames in jedem Protokoll, die Adressen für jedes Protokoll oder Kombinationen davon umfasst.

[1811] Beispiel 208 umfasst den Gegenstand von Beispiel 206 oder 207. In Beispiel 208 umfasst das Verfahren das Bestimmen der Größe des Nutzinhaltsfelds im Ausgangs-Frame und das Feststellen, ob der Eingangs-Frame in den Nutzinhalt passt.

[1812] Beispiel 209 umfasst den Gegenstand von einem der Beispiele 206 bis 208. In Beispiel 209 umfasst das Verfahren das Fragmentieren eines Eingangs-Frames, der zu groß ist, um in ein Nutzinhaltsfeld eines Ausgangs-Frames zu passen.

[1813] Beispiel 210 umfasst den Gegenstand von einem der Beispiele 206 bis 209. In Beispiel 210 umfasst das Verfahren das Zerlegen des Eingangs-Frames in mehrere Bytesequenzen, wobei jede Bytesequenz in ein Nutzinhaltsfeld eines Ausgangs-Frames passt.

[1814] Beispiel 211 umfasst den Gegenstand von einem der Beispiele 206 bis 210. In Beispiel 211 umfasst das Verfahren das Schreiben von jeder der mehreren Bytesequenzen in das Datenfeld eines getrennten Ausgangs-Frames und das Schreiben einer Sequenznummer für jede der mehreren Bytesequenzen in das Datenfeld des getrennten Ausgangs-Frames.

[1815] Beispiel 212 umfasst den Gegenstand von einem der Beispiele 209 bis 211. In Beispiel 212 umfasst das Verfahren das Feststellen, ob alle Fragmente des Eingangs-Frames verarbeitet wurden, und, falls dies nicht der Fall ist, das Fortsetzen des Schreibens von Fragmenten in Ausgangs-Frames.

[1816] Beispiel 213 umfasst den Gegenstand von einem der Beispiele 206 bis 212. In Beispiel 213 umfasst das Verfahren das Feststellen, ob ein neuer Eingangs-Frame empfangen wurde, und das Verarbeiten des neuen Eingangs-Frames, um ihn in einen Ausgangs-Frame zu packen.

[1817] Beispiel 214 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Eingangs-Frame zu analysieren, um die Größe und andere Randbedingungen zu bestimmen, ein Protokoll für einen Ausgangs-Frame zu bestimmen, zumindest einen Teil des Eingangs-Frames in das Datenfeld des Ausgangs-Frames zu schreiben und den Ausgangs-Frame zu einem Bestimmungsort zu leiten.

[1818] Beispiel 215 umfasst den Gegenstand von Beispiel 214. In Beispiel 215 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Inventar verfügbarer Eingangsprotokolle und Ausgangsprotokolle zu erzeugen.

[1819] Beispiel 216 umfasst den Gegenstand von Beispiel 214 oder 215. In Beispiel 216 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Eingangs-Frame zu fragmentieren, falls er zu groß ist, um in das Datenfeld des Ausgangs-Frames zu passen.

[1820] Beispiel 217 umfasst den Gegenstand von einem der Beispiele 214 bis 216. In Beispiel 217 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, jedes Fragment mit einer Sequenznummer für die korrekte Neuzusammensetzung an einem Bestimmungsort zu bezeichnen.

[1821] Beispiel 218 umfasst den Gegenstand von einem der Beispiele 214 bis 217. In Beispiel 218 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Eingangs-Frame oder ein Fragment des Eingangs-Frames in das Nutzinhaltsfeld des Ausgangs-Frames einzubringen.

[1822] Beispiel 219 umfasst den Gegenstand von einem der Beispiele 214 bis 218. In Beispiel 219 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Frame zu einer nachfolgenden Vorrichtung mit einer den endgültigen Bestimmungsort angegebenden Adresse zu senden.

[1823] Beispiel 220 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Netzentdecker zur Identifikation verfügbarer paralleler Kommunikationskanäle zwischen der IoT-Vorrichtung und einer Zielvorrichtung, einen Nutzinhalt, einen Nutzinhaltsfragmentierer/-verpacker zum Fragmentieren des Nutzinhalts in Sub-Objekte für die Übertragung und einen Paketkommunikator zum Senden der Sub-Objekte zur Zielvorrichtung über die parallelen Kommunikationskanäle.

- [1824]** Beispiel 221 umfasst den Gegenstand von Beispiel 220. In Beispiel 221 soll der Netzentdecker eine Kombination der Kommunikationskanäle bestimmen.
- [1825]** Beispiel 222 umfasst den Gegenstand von Beispiel 220 oder 221. In Beispiel 222 soll der Netzentdecker die Bestimmung der Kommunikationskanäle zumindest teilweise auf der Geschwindigkeit von Kommunikationen, der Zuverlässigkeit oder der Verfügbarkeit einer Stromversorgung oder Kombinationen davon vornehmen.
- [1826]** Beispiel 223 umfasst den Gegenstand von einem der Beispiele 220 bis 222. In Beispiel 223 umfasst der Nutzinhalt von einem Sensor erhaltene Messungen.
- [1827]** Beispiel 224 umfasst den Gegenstand von einem der Beispiele 220 bis 223. In Beispiel 224 umfasst der Nutzinhalt eine Nachricht von einer anderen IoT-Vorrichtung.
- [1828]** Beispiel 225 umfasst den Gegenstand von einem der Beispiele 220 bis 224. In Beispiel 225 soll der Netzentdecker eine Liste verfügbarer Netzkommunikationswege und zugeordneter Protokolle, die für parallele Kommunikationen zu verwenden sind, aufrechterhalten.
- [1829]** Beispiel 226 umfasst den Gegenstand von einem der Beispiele 220 bis 225. In Beispiel 226 soll der Nutzinhaltsfragmentierer/-verpacker jedes der Sub-Objekte in einen Frame packen, wobei der Frame in einem Protokoll vorliegt, das einem Kommunikationskanal zugeordnet ist.
- [1830]** Beispiel 227 umfasst den Gegenstand von einem der Beispiele 220 bis 226. In Beispiel 227 umfasst die Vorrichtung einen Nutzinhaltsempfänger/-analysator zum Empfangen von Frames von anderen Vorrichtungen, zum Entfernen der Protokollverpackung und zum Analysieren der Frames, um Nachrichten- und Sequenzinformationen zu identifizieren.
- [1831]** Beispiel 228 umfasst den Gegenstand von einem der Beispiele 220 bis 227. In Beispiel 228 soll der Nutzinhaltsempfänger/-analysator feststellen, dass die Datenfelder empfangener Frames Sub-Objekte von Nutzinhalt sind.
- [1832]** Beispiel 229 umfasst den Gegenstand von einem der Beispiele 220 bis 228. In Beispiel 229 soll der Nutzinhaltsempfänger/-analysator die Sub-Objekte und Sequenznummern, die alle Sub-Objekte empfangen haben, speichern und dann die Sequenznummern und Speicherinformationen zu einem Nutzinhaltsdefragmentierer weiterleiten.
- [1833]** Beispiel 230 umfasst den Gegenstand von einem der Beispiele 220 bis 229. In Beispiel 230 umfasst die Vorrichtung einen Nutzinhaltsdefragmentierer zum Neuzusammenstellen der Nutzinhalt zum endgültigen Nutzinhaltsobjekt.
- [1834]** Beispiel 231 umfasst ein Verfahren zum Fragmentieren und Übertragen eines Nutzinhalt über mehrere parallele Kommunikationskanäle. Das Verfahren zum Fragmentieren und Übertragen eines Nutzinhalt über mehrere parallele Kommunikationskanäle umfasst das Fragmentieren des Nutzinhalt in Fragmente zumindest teilweise auf der Grundlage verfügbarer Kommunikationskanäle und Nutzinhaltsgößen, die von Protokoll-Frames in Zusammenhang mit den Kommunikationskanälen unterstützt werden. Das Verfahren umfasst auch das Zuweisen von Sequenznummern zu den Fragmenten, das Anhängen von Headern, welche die Sequenznummern aufweisen, an die Fragmente, um Teilblöcke zu bilden, das Einkapseln der Teilblöcke in die Protokoll-Frames zur Übertragung über die verschiedenen Wege und das Übertragen der Teilblöcke über die verschiedenen Kommunikationskanäle.
- [1835]** Beispiel 232 umfasst den Gegenstand von Beispiel 231. In Beispiel 232 erfolgt das Fragmentieren des Nutzinhalt in Fragmente zumindest teilweise auf der Grundlage von Übertragungsraten über Kommunikationskanäle oder der Priorität über Kommunikationskanäle oder auf der Grundlage beider.
- [1836]** Beispiel 233 umfasst den Gegenstand von Beispiel 231 oder 232. In Beispiel 233 umfasst das Verfahren das Entdecken der verfügbaren Kommunikationskanäle zu einem Ziel und das Speichern der Kommunikationskanäle und der zugeordneten Protokolle in einer Bibliothek.
- [1837]** Beispiel 234 umfasst den Gegenstand von einem der Beispiele 231 bis 233. In Beispiel 234 umfasst das Verfahren das Testen der Kommunikationskanäle auf einer periodischen Basis, um zu bestätigen, dass die Verbindbarkeit noch vorhanden ist.

[1838] Beispiel 235 umfasst ein Verfahren zum Empfangen und Rekombinieren über mehrere parallele Kommunikationskanäle gesendeter Pakete. Das Verfahren zum Empfangen und Rekombinieren über mehrere parallele Kommunikationskanäle gesendeter Pakete umfasst das Empfangen von Frames von einer sendenden Vorrichtung über eine Anzahl verschiedener Kommunikationskanäle, das Parsen der Frames, um Fragmente eines Nutzinhalts zu erhalten, das Bestimmen, wann der Nutzinhalt von den Fragmenten neu zusammenzustellen ist, und das Kombinieren der Fragmente, um den Nutzinhalt neu zusammenzustellen.

[1839] Beispiel 236 umfasst den Gegenstand von Beispiel 235. In Beispiel 236 umfasst das Verfahren das Erkennen des Empfangs von Frames auf einer Anzahl verschiedener Kommunikationskanäle, das Entfernen der Nutzinhalt aus den Frames, das Analysieren der Nutzinhalt, um Nutzinhaltsfragmente und zugeordnete Sequenznummern zu identifizieren, und das Speichern der Nutzinhaltsfragmente und das Aufzeichnen der zugeordneten Sequenznummern.

[1840] Beispiel 237 umfasst den Gegenstand von Beispiel 235 oder 236. In Beispiel 237 umfasst das Verfahren das Kombinieren der Fragmente, bevor alle Frames empfangen wurden, um einen Teilnutzinhalt zu erhalten.

[1841] Beispiel 238 umfasst den Gegenstand von einem der Beispiele 235 bis 237. In Beispiel 238 umfasst das Verfahren das Bereitstellen des Nutzinhalt für einen Verbraucherprozess.

[1842] Beispiel 239 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Nutzinhalt zu fragmentieren, so dass er in die Datenfelder von Frames für die für die Kommunikationen ausgewählten Protokolle passt, einen Header an jedes Fragment anzuhängen, der die Sequenznummer des Fragments zur Bildung eines Teilblocks umfasst, jeden Teilblock in einen Protokoll-Frame, abhängig vom Kommunikationskanal, zu packen, und jeden der Protokoll-Frames über den Kommunikationskanal in Zusammenhang mit dem Protokoll-Frame zur Zielvorrichtung zu übertragen.

[1843] Beispiel 240 umfasst den Gegenstand von Beispiel 239. In Beispiel 240 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, verfügbare Kommunikationskanäle zu einer Empfangsvorrichtung zu entdecken.

[1844] Beispiel 241 umfasst den Gegenstand von Beispiel 239 oder 240. In Beispiel 241 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Protokoll-Frames zu empfangen, die Teilblöcke aus den Protokoll-Frames zu entpacken, den Header in jedem der Teilblöcke zu parsen, um das Fragment und die Sequenznummer des Fragments zu identifizieren, und den Prozessor anzuweisen, den Nutzinhalt aus den Fragmenten in der Reihenfolge der Sequenznummern neu zusammenzustellen.

[1845] Beispiel 242 umfasst den Gegenstand von einem der Beispiele 239 bis 241. In Beispiel 242 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass der Nutzinhalt neu zusammengestellt werden sollte, bevor alle Fragmente empfangen wurden.

[1846] Beispiel 243 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz einen Bakenknoten umfasst. Der Bakenknoten umfasst ein satellitengestütztes Navigationsmodul zum Empfangen und Verarbeiten von Satellitenpositionsdaten, einen satellitengestützten Navigationslokalisierer zum Steuern des satellitengestützten Navigationsmoduls und zum Erhalten von Ortsinformationen der satellitengestützten Navigation, einen Datenparser zum Parsen der Ortsinformationen der satellitengestützten Navigation, einen Nutzinhaltserzeuger zur Bildung eines Ortspakets und einen Kommunikator zum Senden des Pakets zu einer Mesh-Vorrichtung über eine Kommunikationsstrecke.

[1847] Beispiel 244 umfasst den Gegenstand von Beispiel 243. In Beispiel 244 umfassen die Ortsinformationen der satellitengestützten Navigation Breite, Länge, Zeit oder Höhe oder Kombinationen davon.

[1848] Beispiel 245 umfasst den Gegenstand von Beispiel 243 oder 244. In Beispiel 245 umfasst die Vorrichtung einen Mesh-Sendeempfänger, einen Uplink-Sendeempfänger oder eine Netzschnittstellen-Steuerereinrichtung bzw. Network Interface Controller (NIC) oder Kombinationen davon, um die Kommunikationsstrecke bereitzustellen.

[1849] Beispiel 246 umfasst den Gegenstand von einem der Beispiele 243 bis 245. In Beispiel 246 soll der Nutzinhaltserzeuger das Paket in einen Frame mit einem der Kommunikationsstrecke zugeordneten Protokoll einbringen.

[1850] Beispiel 247 umfasst den Gegenstand von einem der Beispiele 243 bis 246. In Beispiel 247 soll der satellitengestützte Navigationslokalisierer das GPS-Modul zumindest teilweise auf der Grundlage der Batteriekapazität deaktivieren.

[1851] Beispiel 248 umfasst den Gegenstand von einem der Beispiele 243 bis 247. In Beispiel 248 soll der satellitengestützte Navigationslokalisierer das satellitengestützte Navigationsmodul aktivieren, wenn ein anderer Bakenknoten ein satellitengestütztes Navigationsmodul deaktiviert.

[1852] Beispiel 249 umfasst den Gegenstand von einem der Beispiele 243 bis 248, einschließlich einer IoT-Vorrichtung zu 249. In Beispiel 249 umfasst die IoT-Vorrichtung einen Paketvalidierer zum Validieren von einem Bakenknoten empfangener Pakete, um festzustellen, ob die Pakete mit einer Bakenkennung übereinstimmen, gültige Daten aufweisen oder beides.

[1853] Beispiel 250 umfasst den Gegenstand von einem der Beispiele 243 bis 249. In Beispiel 250 soll der Paketvalidierer ungültige Frames ignorieren.

[1854] Beispiel 251 umfasst den Gegenstand von einem der Beispiele 243 bis 250. In Beispiel 251 soll der Paketvalidierer eine Rücksendeanforderung senden.

[1855] Beispiel 252 umfasst den Gegenstand von einem der Beispiele 243 bis 251. In Beispiel 252 umfasst die IoT-Vorrichtung eine Paketextrahiereinrichtung zum Analysieren eines empfangenen Frames, um einen Orts-Nutzinhalt zu extrahieren.

[1856] Beispiel 253 umfasst ein Verfahren zum Erzeugen eines Orts-Nutzinhalts. Das Verfahren zum Erzeugen eines Orts-Nutzinhalts umfasst das Erhalten von Ortsdaten von einem satellitengestützten Navigationsmodul, das Parsen der Ortsdaten, das Erzeugen eines Orts-Nutzinhalts anhand der Ortsdaten und das Ausstrahlen des Orts-Nutzinhalts zu umgebenden IoT-Vorrichtungen.

[1857] Beispiel 254 umfasst den Gegenstand von Beispiel 253. In Beispiel 254 umfasst das Verfahren das Erhalten einer Positionsfestlegung, einschließlich des Sendens eines Befehls zum satellitengestützten Navigationsmodul, um eine Positionsfestlegung zu erhalten, des Wartens während eines vorgegebenen Zeitraums, des Feststellens, ob eine Festlegung erhalten wurde, und des Bereitstellens von Ortsdaten, falls eine Festlegung erhalten wurde.

[1858] Beispiel 255 umfasst den Gegenstand von Beispiel 253 oder 254. In Beispiel 255 umfasst das Parsen der Ortsdaten das Extrahieren einer Länge, einer Breite oder einer Zeit oder Kombinationen davon und das Speichern der Ortsdaten in einem lokalen Speicher.

[1859] Beispiel 256 umfasst den Gegenstand von einem der Beispiele 253 bis 255. In Beispiel 256 umfasst das Erzeugen eines Orts-Nutzinhalts anhand der Ortsdaten das Schreiben der Breite, der Länge und der Zeit in einer sequenziellen Zeichenkette von Bits, das Schreiben einer Bakenkennung in die sequenzielle Zeichenkette von Bits und das Schreiben eines Zeitstempels in die sequenzielle Zeichenkette von Bits.

[1860] Beispiel 257 umfasst den Gegenstand von einem der Beispiele 253 bis 256. In Beispiel 257 umfasst der Zeitstempel aus Satellitendaten extrahierte Zeitdaten.

[1861] Beispiel 258 umfasst den Gegenstand von einem der Beispiele 253 bis 257. In Beispiel 258 umfasst das Verfahren das Packen des Nutzinhalts in einen Frame für die Übertragung.

[1862] Beispiel 259 umfasst den Gegenstand von einem der Beispiele 253 bis 258. In Beispiel 259 ist der Frame ein Ethernet-Frame, ein LoRaWAN-Frame, ein 4G-Frame, ein 3G-Frame oder Kombinationen davon.

[1863] Beispiel 260 umfasst den Gegenstand von einem der Beispiele 253 bis 259. In Beispiel 260 umfasst das Verfahren das Aktivieren eines Senders und das Übertragen des Frames als Nachricht über eine Funkübertragung, ein festverdrahtetes Netz oder beide.

[1864] Beispiel 261 umfasst den Gegenstand von einem der Beispiele 253 bis 260. In Beispiel 261 umfasst das Verfahren das Warten während eines vorgegebenen Intervalls.

[1865] Beispiel 262 umfasst den Gegenstand von einem der Beispiele 253 bis 261. In Beispiel 262 umfasst das Verfahren das Empfangen eines Baken-Frames, der den Orts-Nutzinhalt umfasst, das Bestätigen einer Identität eines Bakenknotens, das Bestätigen, dass der Frame gültig ist, das Extrahieren des Orts-Nutzinhalts aus dem Baken-Frame und das Parsen des Orts-Nutzinhalts zum Extrahieren von Breiten- und Längeninformati-
onem.

[1866] Beispiel 263 umfasst den Gegenstand von einem der Beispiele 253 bis 262. In Beispiel 263 umfasst das Verfahren das Parsen des Orts-Nutzinhalts zum Extrahieren eines Zeitstempels.

[1867] Beispiel 264 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Ortsdaten von einem satellitengestützten Navigationsmodul zu erhalten, die Ortsdaten vom satellitengestützten Navigationsmodul zu parsen, um getrennte Werte für Breite und Länge zu erhalten, einen Orts-Nutzinhalt einschließlich der Breite und der Länge zu bilden, den Orts-Nutzinhalt in einen Frame eines bestimmten Protokolls zu packen und den Orts-Nutzinhalt abzusenden.

[1868] Beispiel 265 umfasst den Gegenstand von Beispiel 264. In Beispiel 265 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Zeitstempel zum Orts-Nutzinhalt hinzuzufügen.

[1869] Beispiel 266 umfasst den Gegenstand von Beispiel 264 oder 265. In Beispiel 266 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Orts-Nutzinhalt aus einem von einem Bakenknoten empfangenen Frame zu extrahieren.

[1870] Beispiel 267 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Datenmanager zur Behandlung von Daten an der IoT-Vorrichtung, einen Datenklassifizierer zum Klassifizieren jedes Fragments der durch die IoT-Vorrichtung laufenden Daten als einwärts gerichtet, auswärts gerichtet oder Cache und einen Daten-Mapper zum Zuordnen der klassifizierten Daten zu einem physikalischen Ort auf der IoT-Vorrichtung.

[1871] Beispiel 268 umfasst den Gegenstand von Beispiel 267. In Beispiel 268 umfasst die IoT-Vorrichtung einen Datengeschichtsaufzeichner zum Verfolgen von Daten, die sich in die IoT-Vorrichtung und aus dieser heraus bewegen.

[1872] Beispiel 269 umfasst den Gegenstand von einem der Beispiele 267 bis 268. In Beispiel 269 umfasst die IoT-Vorrichtung einen Protokollmanager zur Behandlung von Protokollen, die für Frames für Kommunikationskanäle verwendet werden.

[1873] Beispiel 270 umfasst den Gegenstand von einem der Beispiele 267 bis 269. In Beispiel 270 umfasst die IoT-Vorrichtung einen Netzmanager zur Behandlung von Netzkommunikationen auf einer Anzahl von Kommunikationskanälen.

[1874] Beispiel 271 umfasst den Gegenstand von einem der Beispiele 267 bis 270. In Beispiel 271 umfasst die IoT-Vorrichtung einen Kommunikationsmanager zur Behandlung eines Mesh-Sendeempfängers, eines Uplink-Sendeempfängers, einer Ethernet-Verbindung oder einer Kombination davon.

[1875] Beispiel 272 umfasst den Gegenstand von einem der Beispiele 267 bis 271. In Beispiel 272 umfasst die IoT-Vorrichtung ein Eingangsfach zum Speichern einwärts gerichteter Daten für die IoT-Vorrichtung selbst.

[1876] Beispiel 273 umfasst den Gegenstand von einem der Beispiele 267 bis 272. In Beispiel 273 umfasst die IoT-Vorrichtung ein Ausgangsfach zum Speichern auswärts gerichteter Daten, die zu einer anderen Mesh-Vorrichtung zu senden sind.

[1877] Beispiel 274 umfasst den Gegenstand von einem der Beispiele 267 bis 273. In Beispiel 274 umfasst die IoT-Vorrichtung einen Cache zum Speichern von Datenanforderungen .

[1878] Beispiel 275 umfasst den Gegenstand von einem der Beispiele 267 bis 275. In Beispiel 275 umfasst die IoT-Vorrichtung mehrere Mesh-Vorrichtungen, die in Kommunikation miteinander stehen, um Daten zu speichern und zu verteilen.

[1879] Beispiel 276 umfasst den Gegenstand von einem der Beispiele 267 bis 276. In Beispiel 276 werden die Daten zustandslos verteilt.

[1880] Beispiel 277 umfasst ein Verfahren zur verteilten Inhaltsverteilung. Das Verfahren zur verteilten Inhaltsverteilung umfasst das Klassifizieren eines Datenfragments als einwärts gerichtete Daten, auswärts gerichtete Daten oder Cache-Daten und das Zuordnen des Datenfragments zu einem physikalischen Ort in einem Datenspeicher.

[1881] Beispiel 278 umfasst den Gegenstand von Beispiel 277. In Beispiel 278 umfasst das Verfahren das Berechnen eines Hash-Schlüssels für ein einwärts gerichtetes Datenfragment und das Feststellen, ob der Hash-Schlüssel im Datenspeicher ist, und, falls dies nicht der Fall ist, das Speichern des Datenfragments.

[1882] Beispiel 279 umfasst den Gegenstand von Beispiel 277 oder 278. In Beispiel 279 umfasst das Verfahren das Berechnen einer Lebenszeit bzw. Time-to-Live (TTL) für die Daten, falls festgestellt wird, dass das Datenfragment aus auswärts gerichteten Daten besteht.

[1883] Beispiel 280 umfasst den Gegenstand von einem der Beispiele 277 bis 279. In Beispiel 280 wird die TTL als Anzahl der Hops berechnet, über welche das Datenfragment vor dem Löschen übertragen werden kann.

[1884] Beispiel 281 umfasst den Gegenstand von einem der Beispiele 277 bis 280. In Beispiel 281 wird die TTL als Zeitraum vor dem Löschen berechnet.

[1885] Beispiel 282 umfasst den Gegenstand von einem der Beispiele 277 bis 281. In Beispiel 282 wird die TTL als Kombination einer Anzahl von Hops, über die das Datenfragment übertragen werden kann, und eines Zeitraums berechnet.

[1886] Beispiel 283 umfasst den Gegenstand von einem der Beispiele 277 bis 282. In Beispiel 283 umfasst das Verfahren das Verfolgen einer einwärts gerichteten Datenanforderung, auswärts gerichteter Datenanforderungen oder beider in einem Datengeschichtsaufzeichner.

[1887] Beispiel 284 umfasst den Gegenstand von einem der Beispiele 277 bis 283. In Beispiel 284 umfasst das Verfahren das Anpassen der Größe eines Caches zumindest teilweise auf der Grundlage der Häufigkeit von Datenzugriffen auf den Cache.

[1888] Beispiel 285 umfasst den Gegenstand von einem der Beispiele 277 bis 284. In Beispiel 285 umfasst das Verfahren das Auswählen eines Speichertyps zumindest teilweise ausgewählt wird der Häufigkeit von Datenzugriffen auf den Cache.

[1889] Beispiel 286 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, ein Datenfragment, das eine Internet-der-Dinge(IoT)-Vorrichtung durchläuft, als einwärts gerichtete Daten, auswärts gerichtete Daten oder Cache-Daten zu klassifizieren und das klassifizierte Datenfragment einem physikalischen Ort auf der IoT-Vorrichtung zuzuordnen.

[1890] Beispiel 287 umfasst den Gegenstand von Beispiel 286. In Beispiel 287 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Hash-Schlüssel für ein Datenfragment einwärts gerichteter Daten zu berechnen und festzustellen, ob der Hash-Schlüssel in einem lokalen Speicher ist, und, falls dies nicht der Fall ist, das Datenfragment im lokalen Speicher zu speichern.

[1891] Beispiel 288 umfasst den Gegenstand von Beispiel 286 oder 287. In Beispiel 288 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Datenfragment, das sich in einem lokalen Speicher befindet, zu aktualisieren.

[1892] Beispiel 289 umfasst den Gegenstand von einem der Beispiele 286 bis 288. In Beispiel 289 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Lebenszeit für ein Datenfragment zu berechnen.

[1893] Beispiel 290 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz ein Drahtlosspeichersystem umfasst, das wenigstens zwei IoT-Vorrichtungen umfasst. Das Drahtlosspeichersystem umfasst einen Ursprungsknoten und einen Empfangsknoten. Der Ursprungsknoten umfasst einen Nutzinhaltsfragmentierer zum Zerlegen zu speichernder Daten in Fragmente, eine Einkapselungseinrichtung zur Bildung von Paketen aus den Fragmenten, einen Kommunikator zum Senden der Pakete zum Empfangsknoten und einen Router zum Zurücksenden vom Empfangsknoten empfangener Pakete zum Ursprungsknoten. Der Empfangsknoten umfasst einen Router zum Zurücksenden vom Ursprungsknoten empfangener Pakete zum Empfangsknoten, wobei die Pakete bei der Übertragung zwischen dem Ursprungsknoten und dem Empfangsknoten gespeichert werden.

[1894] Beispiel 291 umfasst den Gegenstand von Beispiel 290. In Beispiel 291 umfasst der Ursprungsknoten eine Nutzinhaltsextrahiereinrichtung zum Entfernen der Fragmente aus vom Empfangsknoten empfangenen Paketen und einen Datenzusammensteller zum Rekonstruieren der Daten aus den Fragmenten.

[1895] Beispiel 292 umfasst den Gegenstand von Beispiel 290 oder 291. In Beispiel 292 umfassen der Ursprungsknoten und der Empfangsknoten jeweils einen kompatiblen Mesh-Sendeempfänger.

[1896] Beispiel 293 umfasst den Gegenstand von einem der Beispiele 290 bis 292. In Beispiel 293 umfasst die Vorrichtung mehrere Knoten, die eine Kette zwischen dem Ursprungsknoten und dem Empfangsknoten bilden.

[1897] Beispiel 294 umfasst den Gegenstand von einem der Beispiele 290 bis 293. In Beispiel 294 umfassen der Ursprungsknoten und der Empfangsknoten einen Kommunikationsstapel, der eine Netz-/Weiterleitungsschicht umfasst.

[1898] Beispiel 295 umfasst den Gegenstand von einem der Beispiele 290 bis 294. In Beispiel 295 soll die Netz-/Weiterleitungsschicht ein an der Netz-/Weiterleitungsschicht empfangenes Paket zur physikalischen Schicht zur Übertragung zurücksenden.

[1899] Beispiel 296 umfasst den Gegenstand von einem der Beispiele 290 bis 295. In Beispiel 296 soll die Netz-/Weiterleitungsschicht einen Datenzugriffsbefehl akzeptieren und die Pakete zur Anwendungsschicht umleiten.

[1900] Beispiel 297 umfasst ein Verfahren zum Fragmentieren und Speichern von Daten in einer Übertragungsschleife zwischen Vorrichtungen. Das Verfahren zum Fragmentieren und Speichern von Daten in einer Übertragungsschleife zwischen Vorrichtungen umfasst das Fragmentieren zu speichernder Daten in Datenfragmente, das Einkapseln der Datenfragmente in Speicherpakete, das Senden der Speicherpakete zu einer anderen Vorrichtung, das Empfangen der Speicherpakete von der anderen Vorrichtung und das Zurücksenden der Speicherpakete zur anderen Vorrichtung.

[1901] Beispiel 298 umfasst den Gegenstand von Beispiel 297. In Beispiel 298 umfasst das Verfahren das Zuweisen einer Sequenznummer zu jedem der Fragmente.

[1902] Beispiel 299 umfasst den Gegenstand von Beispiel 297 oder 298. In Beispiel 299 umfasst das Verfahren das Anhängen eines die Sequenznummer umfassenden Headers an ein Fragment zur Bildung eines Speicherpakets.

[1903] Beispiel 300 umfasst den Gegenstand von einem der Beispiele 297 bis 299. In Beispiel 300 umfasst der Header eine Angabe, dass das Speicherpaket aus gespeicherten Daten besteht.

[1904] Beispiel 301 umfasst den Gegenstand von einem der Beispiele 297 bis 300. In Beispiel 301 umfasst das Verfahren das Zurücksenden der Speicherpakete zur anderen Vorrichtung auf einer Stapelschicht unterhalb der Anwendungsschicht.

[1905] Beispiel 302 umfasst den Gegenstand von einem der Beispiele 297 bis 301. In Beispiel 302 umfasst das Verfahren das Feststellen, ob ein empfangenes Speicherpaket weiter gespeichert werden sollte, und, falls dies der Fall ist, das Übertragen des Pakets zur anderen Vorrichtung.

- [1906]** Beispiel 303 umfasst den Gegenstand von einem der Beispiele 297 bis 302. In Beispiel 303 umfasst das Verfahren das Feststellen, dass Daten von einem Speicherpaket empfangen werden sollten.
- [1907]** Beispiel 304 umfasst den Gegenstand von einem der Beispiele 297 bis 303. In Beispiel 304 umfasst das Verfahren das Entnehmen eines Datenfragments aus dem Speicherpaket.
- [1908]** Beispiel 305 umfasst den Gegenstand von einem der Beispiele 297 bis 304. In Beispiel 305 umfasst das Verfahren das Entnehmen einer Sequenznummer aus dem Speicherpaket.
- [1909]** Beispiel 306 umfasst den Gegenstand von einem der Beispiele 297 bis 305. In Beispiel 306 umfasst das Verfahren das Feststellen, ob alle Datenfragmente empfangen wurden, und, falls dies der Fall ist, das Zusammenstellen der Datenfragmente in der Reihenfolge der Sequenznummer, um die Daten zu erhalten.
- [1910]** Beispiel 307 umfasst den Gegenstand von einem der Beispiele 297 bis 306. In Beispiel 307 umfasst das Verfahren das Übergeben der Daten an eine Anwendungsschicht zur Verwendung bei einer Datenoperation.
- [1911]** Beispiel 308 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, festzustellen, ob eine Weiterleitungsanforderung eine Datenspeicheranforderung ist, Daten in Nutzinhalte zu fragmentieren, die Nutzinhalte in Speicherpakete einzukapseln, die Speicherpakete über einen Kommunikationskanal weiterzuleiten und festzustellen, ob empfangene Speicherpakete zu einer anderen Vorrichtung gesendet oder für die Verwendung abgefangen werden sollten.
- [1912]** Beispiel 309 umfasst den Gegenstand von Beispiel 308. In Beispiel 309 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Nutzinhalte aus den Speicherpaketen zu entpacken.
- [1913]** Beispiel 310 umfasst den Gegenstand von Beispiel 308 oder 319. In Beispiel 310 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Nutzinhalte in der Reihenfolge der Sequenznummer neu zusammenzustellen, um die Daten zu bilden.
- [1914]** Beispiel 311 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung einschließlich eines Funksenders zum Senden einer mehrere überlappende Frames umfassenden modulierten Wellenform umfasst, wobei die modulierte Wellenform eine Zadoff-Chu(ZC)-Sequenz ist.
- [1915]** Beispiel 312 umfasst den Gegenstand von Beispiel 311. In Beispiel 312 umfasst die Vorrichtung einen Kanalidentifizierer zur Bestimmung der Anzahl der verfügbaren Kanäle, die bei einer Funkkommunikation mit einer anderen Vorrichtung vorhanden sein können.
- [1916]** Beispiel 313 umfasst den Gegenstand von Beispiel 311 oder 312. In Beispiel 313 umfasst die Vorrichtung einen Sequenzgenerator zur Erzeugung einer ZC-Sequenz für jeden der verfügbaren Kanäle.
- [1917]** Beispiel 314 umfasst den Gegenstand von einem der Beispiele 311 bis 313. In Beispiel 314 umfasst die Vorrichtung einen Präambelgenerator zur Erzeugung einer Präambel-Wellenform, welche die Anzahl der für eine Übertragung verwendeten Kanäle codiert.
- [1918]** Beispiel 315 umfasst den Gegenstand von einem der Beispiele 311 bis 314. In Beispiel 315 umfasst die Vorrichtung einen Kommunikator zur Erzeugung einer modulierten Wellenform, die einen ZC-modulierten Frame für jeden von der Anzahl von Kanälen und eine Präambel-Wellenform, welche die Anzahl der Kanäle angibt, umfasst.
- [1919]** Beispiel 316 umfasst den Gegenstand von einem der Beispiele 311 bis 315. In Beispiel 316 umfasst die Vorrichtung einen Indexidentifizierer zur Ausführung einer Kreuzkorrelation an einer empfangenen Wellenform, um festzustellen, ob eine Präambel-Wellenform vorhanden ist, und, falls dies der Fall ist, die Anzahl der Kanäle in der empfangenen Wellenform zu bestimmen.
- [1920]** Beispiel 317 umfasst den Gegenstand von einem der Beispiele 311 bis 316. In Beispiel 317 umfasst die Vorrichtung einen Kanaldemodulator zum Demodulieren der modulierten Wellenform, um die mehreren überlappenden Frames zu extrahieren.

[1921] Beispiel 318 umfasst ein Verfahren zur Übertragung von Daten auf mehreren Kanälen unter Verwendung von Zadoff-Chu(ZC)-verschobenen Sequenzen. Das Verfahren zur Übertragung von Daten auf mehreren Kanälen unter Verwendung von Zadoff-Chu(ZC)-verschobenen Sequenzen umfasst das Bestimmen der Anzahl der für eine Kommunikation zu verwendenden Kanäle, das Erzeugen einer Präambel-Wellenform einschließlich der Anzahl der Kanäle in einer ZC-verschobenen Sequenz, das Hängen der Präambel-Wellenform vor eine modulierte Wellenform, um eine Nachricht zu erzeugen, und das Senden der Nachricht zur Empfangsvorrichtung.

[1922] Beispiel 319 umfasst den Gegenstand von Beispiel 318. In Beispiel 319 umfasst das Verfahren das Bestimmen der Anzahl verfügbarer Kanäle.

[1923] Beispiel 320 umfasst den Gegenstand von Beispiel 318 oder 319. In Beispiel 320 wird die Anzahl der verfügbaren Kanäle unter Verwendung der Informationstheorie bestimmt, um die maximale Informationsmenge, die in einer Übertragung übermittelt werden kann, zu bestimmen.

[1924] Beispiel 321 umfasst den Gegenstand von einem der Beispiele 318 bis 320. In Beispiel 321 umfasst das Verfahren das Senden der Anzahl der verfügbaren Kanäle zu einer Empfangsvorrichtung.

[1925] Beispiel 322 umfasst den Gegenstand von einem der Beispiele 318 bis 321. In Beispiel 322 umfasst das Verfahren das Erzeugen ZC-verschobener Sequenzen entsprechend jedem der verfügbaren Kanäle.

[1926] Beispiel 323 umfasst den Gegenstand von einem der Beispiele 318 bis 322. In Beispiel 323 umfasst das Verfahren das Modulieren eines von jedem von der Anzahl von Kanälen bei getrennten ZC-verschobenen Sequenzen zu übertragenden Frames und das Kombinieren der modulierten Frames zur Erzeugung der modulierten Wellenform.

[1927] Beispiel 324 umfasst den Gegenstand von einem der Beispiele 318 bis 323. In Beispiel 324 umfasst das Verfahren das Demodulieren der modulierten Wellenform in einer Nachricht bei jeder von der Anzahl ZC-verschobener Sequenzen, um einen von einem entsprechenden Kanal übertragenen Frame zu erhalten.

[1928] Beispiel 325 umfasst den Gegenstand von einem der Beispiele 318 bis 324. In Beispiel 325 umfasst das Verfahren das Ausführen einer Autokorrelation an der Nachricht, um festzustellen, ob die Präambel-Wellenform vorhanden ist.

[1929] Beispiel 326 umfasst den Gegenstand von einem der Beispiele 318 bis 325. In Beispiel 326 umfasst das Verfahren das Ausführen einer Kreuzkorrelation an der Präambel-Wellenform an jeder der mehreren ZC-verschobenen Sequenzen, um die Anzahl der Kanäle in der modulierten Wellenform zu bestimmen.

[1930] Beispiel 327 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Anzahl der für eine Kommunikation zu verwendenden Kanäle zu bestimmen, eine Präambel-Wellenform, einschließlich der Anzahl der Kanäle, zu erzeugen, die Präambel-Wellenform vor eine modulierte Wellenform zu hängen, um eine Nachricht zu erzeugen, und die Nachricht zur Empfangsvorrichtung zu senden.

[1931] Beispiel 328 umfasst den Gegenstand von Beispiel 327. In Beispiel 328 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Anzahl der verfügbaren Kanäle zu bestimmen.

[1932] Beispiel 329 umfasst den Gegenstand von Beispiel 327 oder 328. In Beispiel 329 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Anzahl der verfügbaren Kanäle zu einer Empfangsvorrichtung zu senden.

[1933] Beispiel 330 umfasst den Gegenstand von einem der Beispiele 327 bis 329. In Beispiel 330 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, getrennte ZC-verschobene Sequenzen entsprechend jedem verfügbaren Kanal zu erzeugen.

[1934] Beispiel 331 umfasst den Gegenstand von einem der Beispiele 327 bis 330. In Beispiel 331 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, von jeder der Anzahl von Kanälen zu übertragende Frames bei einer getrennten ZC-verschobenen Sequenz zu modulieren und die modulierten Frames zu kombinieren, um die modulierte Wellenform zu erzeugen.

[1935] Beispiel 332 umfasst den Gegenstand von einem der Beispiele 327 bis 331. In Beispiel 332 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die modulierte Wellenform bei jeder der mehreren ZC-verschobenen Sequenzen zu demodulieren, um einen von einem entsprechenden Kanal übertragenen Frame zu erhalten.

[1936] Beispiel 333 umfasst den Gegenstand von einem der Beispiele 327 bis 332. In Beispiel 333 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Autokorrelation an der Nachricht auszuführen, um festzustellen, ob die Präambel-Wellenform vorhanden ist.

[1937] Beispiel 334 umfasst den Gegenstand von einem der Beispiele 327 bis 333. In Beispiel 334 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Kreuzkorrelation an der Präambel-Wellenform bei jeder von mehreren ZC-verschobenen Sequenzen auszuführen, um die Anzahl der Kanäle in der modulierten Wellenform zu bestimmen.

[1938] Beispiel 335 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst Funksendeempfänger, die eine Anzahl von Frequenzen, Protokollen oder beides umfassen, einen Koexistenzmanager zur Verfolgung der für eine Kommunikation verwendeten Funksendeempfänger, eine universelle Koexistenzschnittstelle zur Identifikation, welche Funksendeempfänger für die Kommunikation verwendet werden können, eine Protokollabstraktions-Anwendungsprogrammierschnittstelle (API) zur Bildung von Frames für die Kommunikation und einen Kommunikator zur Übertragung der Frames von der Protokollübersetzungs-API zu einer anderen Vorrichtung unter Verwendung eines Funksendeempfängers.

[1939] Beispiel 336 umfasst den Gegenstand von Beispiel 335. In Beispiel 336 soll die universelle Koexistenzschnittstelle konfigurierbare Parameter für die Funksendeempfänger modifizieren.

[1940] Beispiel 337 umfasst den Gegenstand von einem der Beispiele 335 bis 336. In Beispiel 337 umfassen die konfigurierbaren Parameter Codierschema, Modulation oder beide.

[1941] Beispiel 338 umfasst den Gegenstand von einem der Beispiele 335 bis 337. In Beispiel 338 umfassen die konfigurierbaren Parameter die Sendeleistung für einzelne Funksendeempfänger.

[1942] Beispiel 339 umfasst den Gegenstand von einem der Beispiele 335 bis 338. In Beispiel 339 soll die universelle Koexistenzschnittstelle feststellen, wann Funksendeempfänger für die Kommunikation zu verwenden sind.

[1943] Beispiel 340 umfasst den Gegenstand von einem der Beispiele 335 bis 339. In Beispiel 340 umfasst die IoT-Vorrichtung einen Protokolldatenspeicher zum Erhalten der Protokolle, die für Kommunikationen über die ausgewählten Funksendeempfänger verwendet werden können.

[1944] Beispiel 341 umfasst den Gegenstand von einem der Beispiele 335 bis 340. In Beispiel 341 umfasst die IoT-Vorrichtung einen Kanaldatenspeicher, der einen aktiven Kommunikationsplan und Funkeinrichtungen umfasst, die durch den Koexistenzmanager bestimmt wurden.

[1945] Beispiel 342 umfasst den Gegenstand von einem der Beispiele 335 bis 341. In Beispiel 342 entsprechen die Funksendeempfänger IEEE-802.11x-Standards, IEEE-802.15.24-Standards oder LoRa-Standards oder Kombinationen davon.

[1946] Beispiel 343 umfasst ein Verfahren zum Steuern und zur Behandlung mehrerer koexistierender Funkeinrichtungen. Das Verfahren zum Steuern und zur Behandlung mehrerer koexistierender Funkeinrichtungen umfasst das Spezifizieren in einem Funksystem in einer Internet-der-Dinge(IoT)-Vorrichtung verfügbarer Funkeinrichtungen, das Bestimmen eines aktiven Koexistenzmodells, das sowohl Nachbarn als auch zugeordnete Funkeinrichtungen identifiziert, und das Kommunizieren mit Nachbarn durch zugeordnete Funkeinrichtungen auf verfügbaren Bändern.

[1947] Beispiel 344 umfasst den Gegenstand von Beispiel 343. In Beispiel 344 umfasst das Verfahren das Identifizieren der verfügbaren Bänder für Kommunikationen von einem lokalen Sicherheitsstellen-Domain-Broker bzw. Local Security Authority Domain Broker (LSADB), das Bestimmen eines Bandplans zur Verwendung der Bänder, das Bilden einer Nachbarkarte, die eine Identität für jeden Nachbarn umfasst, das Identifizieren ei-

nes zugeordneten Bands, das für die Kommunikation mit jedem Nachbarn verwendet werden kann, anhand eines Über-Domains-Informationsteilungssystems bzw. Cross-Domain Information Sharing System (CDIS) und das Bestimmen eines anfänglichen Koexistenzmodells, das sowohl Nachbarn als auch zugeordnete Bänder identifiziert.

[1948] Beispiel 345 umfasst den Gegenstand von Beispiel 343 oder 344. In Beispiel 345 umfasst das Verfahren das Senden einer Anforderung einer Subskription für eine Funkeinrichtung zum CDIS, falls ein Band der Funkeinrichtung im anfänglichen Koexistenzmodell nicht vorhanden war.

[1949] Beispiel 346 umfasst den Gegenstand von einem der Beispiele 343 bis 345. In Beispiel 346 umfasst das Verfahren das Verifizieren, dass für die Funkeinrichtungen in der IoT-Vorrichtung verwendete Standards verfügbar sind.

[1950] Beispiel 347 umfasst den Gegenstand von einem der Beispiele 343 bis 346. In Beispiel 347 umfasst das Verfahren das Herunterladen der für die Funkeinrichtungen in der IoT-Vorrichtung verwendeten Standards von einem Cloud-Rechnetz.

[1951] Beispiel 348 umfasst den Gegenstand von einem der Beispiele 343 bis 347. In Beispiel 348 umfasst das Verfahren das Initialisieren jeder der Funkeinrichtungen in einer Vorrichtung, das Feststellen, ob irgendwelche der Funkeinrichtungen den Standards nicht folgen, das Senden einer Nachricht zu einem Funksystem, um konfigurierbare Parameter für jede der Funkeinrichtungen festzulegen, das Erzeugen eines Parameterzuordnungssatzes für die verwendeten Funkeinrichtungen und das Senden einer Nachricht zum Funksystem, um anzugeben, dass die Kommunikationen mit dem Funksystem initialisiert werden.

[1952] Beispiel 349 umfasst den Gegenstand von einem der Beispiele 343 bis 348. In Beispiel 349 umfasst das Verfahren das Erkennen einer Koexistenzverletzung und das Rekonfigurieren konfigurierbarer Parameter.

[1953] Beispiel 350 umfasst den Gegenstand von einem der Beispiele 343 bis 349. In Beispiel 350 umfasst das Verfahren das Erkennen einer Koexistenzverletzung durch Feststellen, dass ein lizenzierter Benutzer ein Band belegt.

[1954] Beispiel 351 umfasst den Gegenstand von einem der Beispiele 343 bis 350. In Beispiel 351 umfasst das Rekonfigurieren der konfigurierbaren Parameter das Senden einer Nachricht zum Funksystem mit einem neuen Parametersatz und das Reinitialisieren der Kommunikationen mit den Funkeinrichtungen.

[1955] Beispiel 352 umfasst den Gegenstand von einem der Beispiele 343 bis 351. In Beispiel 352 umfasst das Rekonfigurieren der konfigurierbaren Parameter das vorübergehende Deaktivieren einer Funkeinrichtung.

[1956] Beispiel 353 umfasst den Gegenstand von einem der Beispiele 343 bis 352. In Beispiel 353 umfasst das Rekonfigurieren der konfigurierbaren Parameter das Übergehen zu einem anderen Band.

[1957] Beispiel 354 umfasst den Gegenstand von einem der Beispiele 343 bis 353. In Beispiel 354 umfasst das Verfahren das Feststellen, welche anderen Knoten mit der IoT-Vorrichtung kommunizieren, das Feststellen einer Nachbarbefehlslistenänderung und das Senden einer Rekonfigurationsanforderung zum Funksystem mit einem neuen Parametersatz.

[1958] Beispiel 355 umfasst den Gegenstand von einem der Beispiele 343 bis 354. In Beispiel 355 umfasst das Verfahren das Empfangen einer Anforderung für eine Änderung von einem Nachbarn mit vorgeschlagenen Parametern und das Senden einer Rekonfigurationsanforderung mit den vorgeschlagenen Parametern zum Funksystem.

[1959] Beispiel 356 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Bandplan zu bestimmen, eine Nachbarkarte zu bilden, welche eine Identität für jeden Nachbarn und diesem Nachbarn zugeordnete Bänder umfasst, und ein Koexistenzmodell zu bestimmen, das sowohl Nachbarn als auch zugeordnete Kommunikationskanäle identifiziert.

[1960] Beispiel 357 umfasst den Gegenstand von Beispiel 356. In Beispiel 357 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Bandplan nach der Bestimmung verfügbarer Funkeinrichtungen zu revidieren.

[1961] Beispiel 358 umfasst den Gegenstand von Beispiel 356 oder 357. In Beispiel 358 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, das Koexistenzmodell nach dem Bestimmen verfügbarer Funkeinrichtungen zu revidieren.

[1962] Beispiel 359 umfasst den Gegenstand von einem der Beispiele 356 bis 358. In Beispiel 358 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Kommunikationen zu rekonfigurieren, falls eine Koexistenzverletzung erkannt wird, falls eine wiederholte Prüfung von benachbarten Einheiten angibt, dass konfigurierbare Parameter eine Anpassung benötigen, oder falls dies von einer benachbarten Einheit gefordert wird, oder Kombinationen davon.

[1963] Beispiel 360 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz Vorrichtungen umfasst, die einen Orchestrator zum Ausgeben von Dienstbehandlungsanforderungen an einen Dienstkoordinator zur Bildung eines Dienstes, wobei der Dienstkoordinator mehrere Komponenten zur Teilnahme am Dienst identifizieren soll, und eine Komponente zur Ausführung eines Netzdienstelements für den Dienst umfassen.

[1964] Beispiel 361 umfasst den Gegenstand von Beispiel 360. In Beispiel 361 behandelt der Orchestrator mehrere Netzdienst-Overlays zur Ausführung von Aufgaben.

[1965] Beispiel 362 umfasst den Gegenstand von einem der Beispiele 360 bis 361. In Beispiel 362 umfasst die Vorrichtung einen geteilten Ablagespeicher, der die mehrere Netzdienst-Overlays umfasst.

[1966] Beispiel 363 umfasst den Gegenstand von einem der Beispiele 360 bis 362. In Beispiel 363 umfasst ein Netzdienst-Overlay ein Codesegment zum Ermöglichen, dass die Komponente das Netzdienstelement ausführt.

[1967] Beispiel 364 umfasst den Gegenstand von einem der Beispiele 360 bis 363. In Beispiel 364 umfasst der Dienstkoordinator eine Datenbank zum Speichern von Daten oder Metadaten oder beider von einer Komponente, einen geteilten virtuellen Ablagespeicher zum Halten eines Netzdienstelements, das vervollständigt werden muss, und eine Maschinenlernengine zur Auswahl der Komponente, um das Netzdienstelement zu vervollständigen.

[1968] Beispiel 365 umfasst den Gegenstand von einem der Beispiele 360 bis 364. In Beispiel 365 speichert der geteilte virtuelle Ablagespeicher die Identität der dem Netzdienstelement zugewiesenen Komponente.

[1969] Beispiel 366 umfasst den Gegenstand von einem der Beispiele 360 bis 365. In Beispiel 366 umfasst der Dienst mehrere Netzdienstelemente, wobei die Netzdienstelemente durch die mehreren Komponenten vervollständigt werden.

[1970] Beispiel 367 umfasst den Gegenstand von einem der Beispiele 360 bis 366. In Beispiel 367 umfasst der Dienst eine Fog-Vorrichtung, die mehrere Internet-der-Dinge(IoT)-Vorrichtungen umfasst.

[1971] Beispiel 368 umfasst den Gegenstand von einem der Beispiele 360 bis 367. In Beispiel 368 umfasst der Dienstkoordinator eine Netzdomain-Steuereinrichtung.

[1972] Beispiel 369 umfasst den Gegenstand von einem der Beispiele 360 bis 368. In Beispiel 369 ist die Komponente eine Vorrichtung, die einen Client umfasst, wobei der Client die Vorrichtung beim Dienstkoordinator registriert.

[1973] Beispiel 370 umfasst den Gegenstand von einem der Beispiele 360 bis 369. In Beispiel 370 sendet der Client eine Nachricht, die angeschlossene Sensoren, Betätigungselemente oder Vorrichtungen oder Kombinationen davon umfasst, zum Dienstkoordinator.

[1974] Beispiel 371 umfasst den Gegenstand von einem der Beispiele 360 bis 370. In Beispiel 371 werden die mehreren Komponenten aus mehreren Domains ausgewählt.

[1975] Beispiel 372 umfasst ein Verfahren zur Vervollständigung von Dienstanforderungen. Das Verfahren zur Vervollständigung von Dienstanforderungen umfasst das Empfangen einer Orchestrierungsanforderung an einer Netzdomain-Steuereinrichtung, das Feststellen, ob die Orchestrierungsanforderung für einen existie-

renden Dienst vorgesehen ist, und, falls die Orchestrierungsanforderung für einen existierenden Dienst vorgesehen ist, das Senden der Orchestrierungsanforderung zu einem Dienstkoordinator.

[1976] Beispiel 373 umfasst den Gegenstand von Beispiel 372. In Beispiel 373 umfasst das Verfahren, falls die Orchestrierungsanforderung eine neue Anforderung zur Präparation eines Dienstmodells einschließlich eines Netzdienstelements ist, das Präparieren des Netzdienstelements, das Identifizieren einer Dienstkomponente zur Ausführung des Netzdienstelements und das Absenden einer Subskriptionsanforderung zur Dienstkomponente, um eine Aktion für das Netzdienstelement auszuführen.

[1977] Beispiel 374 umfasst den Gegenstand von Beispiel 372 oder 373. In Beispiel 374 umfasst das Verfahren das Identifizieren eines Dienstkoordinators.

[1978] Beispiel 375 umfasst den Gegenstand von einem der Beispiele 372 bis 374. In Beispiel 375 umfasst das Identifizieren einer Dienstkomponente das Zugreifen auf Daten über die historische Funktionsweise einer Anzahl von Dienstkomponenten und das Verwenden einer Maschinenlerntechnik zur Auswahl der Dienstkomponente.

[1979] Beispiel 376 umfasst den Gegenstand von einem der Beispiele 372 bis 375. In Beispiel 376 umfasst das Verfahren das Validieren der Subskriptionsanforderung an der Dienstkomponente und das Senden einer Bestätigung zum Dienstkoordinator, falls die Subskriptionsanforderung gültig ist.

[1980] Beispiel 377 umfasst den Gegenstand von einem der Beispiele 372 bis 376. In Beispiel 377 umfasst das Verfahren das Senden einer Ablehnung zum Dienstkoordinator, falls die Subskriptionsanforderung nicht gültig ist.

[1981] Beispiel 378 umfasst den Gegenstand von einem der Beispiele 372 bis 377. In Beispiel 378 ist eine Subskriptionsanforderung gültig, falls sie von der Dienstkomponente unterstützt wird.

[1982] Beispiel 379 umfasst den Gegenstand von einem der Beispiele 372 bis 378. In Beispiel 379 umfasst das Verfahren das Ausführen des Netzdienstelements in der Dienstkomponente und das Zurückgeben von Daten von der Dienstkomponente zum Dienstkoordinator.

[1983] Beispiel 380 umfasst den Gegenstand von einem der Beispiele 372 bis 379. In Beispiel 380 lädt die Dienstkomponente ein Netzdienst-Overlay von einem virtuellen geteilten Ablagespeicher herunter, um das Netzdienstelement auszuführen.

[1984] Beispiel 381 umfasst den Gegenstand von einem der Beispiele 372 bis 380. In Beispiel 381 lädt die Dienstkomponente ein Netzdienst-Overlay von einem geteilten Ablagespeicher in einer Cloud herunter.

[1985] Beispiel 382 umfasst den Gegenstand von einem der Beispiele 372 bis 381. In Beispiel 382 umfasst das Verfahren das Senden einer Fähigkeiten einer Dienstkomponente aufweisenden Nachricht zu einem Dienstkoordinator zur Registrierung der Dienstkomponente.

[1986] Beispiel 383 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen oder mehrere Prozessoren anweisen, einen Dienstkoordinator zu identifizieren, Netzelemente zu präparieren, Dienstkomponenten zu identifizieren und Subskriptionsanforderungen zu Dienstkomponenten zu senden.

[1987] Beispiel 384 umfasst den Gegenstand von Beispiel 383. In Beispiel 384 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, eine Subskriptionsanforderung zu validieren, eine Aktion für ein Netzdienstelement auszuführen und Daten zum Dienstkoordinator zu senden.

[1988] Beispiel 385 umfasst den Gegenstand von Beispiel 383 oder 384. In Beispiel 385 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, eine Verbindungsanforderung zum Dienstkoordinator zu senden und Vorrichtungssperriehedaten zum Dienstkoordinator zu senden.

[1989] Beispiel 386 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz mehrere IoT-Vorrichtungen umfasst. Jede IoT-Vorrichtung umfasst einen Kommunikationskanal

zu einer vorgeschalteten Vorrichtung, eine Netzverbindung zu einer anderen der mehreren IoT-Vorrichtungen, eine Hash-Berechnungseinrichtung zur Identifikation einer Nachbar-IoT-Vorrichtung und einen Kommunikator zum Aussenden einer Nachricht zur Nachbar-IoT-Vorrichtung.

[1990] Beispiel 387 umfasst den Gegenstand von Beispiel 386. In Beispiel 387 umfasst die Vorrichtung einen Datenfragmentierer zum Fragmentieren einer Datendatei in mehrere Nutzinhalte.

[1991] Beispiel 388 umfasst den Gegenstand von Beispiel 386 oder 387. In Beispiel 388 umfasst die Vorrichtung einen Nachrichtengenerator zum Packen eines Nutzinhalts in einen Frame zur Erzeugung der Nachricht.

[1992] Beispiel 389 umfasst den Gegenstand von einem der Beispiele 386 bis 388. In Beispiel 389 empfängt ein Kommunikator in der Nachbar-IoT-Vorrichtung die Nachricht und sendet die Nachricht zur vorgeschalteten Vorrichtung.

[1993] Beispiel 390 umfasst den Gegenstand von einem der Beispiele 386 bis 389. In Beispiel 390 analysiert ein Nachrichtengenerator in der Nachbar-IoT-Vorrichtung die Nachricht, um einen Nutzinhalt zu erhalten, und speichert den Nutzinhalt in einem Datenspeicher.

[1994] Beispiel 391 umfasst den Gegenstand von einem der Beispiele 386 bis 390. In Beispiel 391 umfasst die Vorrichtung einen Daten-Tracker zum Speichern einer Identifikation für jede zu einer Nachbar-IoT-Vorrichtung gesendete Nachricht in einem Datenspeicher.

[1995] Beispiel 392 umfasst den Gegenstand von einem der Beispiele 386 bis 391. In Beispiel 392 umfasst die Identifikation einen Hash-Code eines Nutzinhalts in der Nachricht und eine Knotenkennung für die Nachbar-IoT-Vorrichtung.

[1996] Beispiel 393 umfasst den Gegenstand von einem der Beispiele 386 bis 392. In Beispiel 393 soll der Daten-Tracker eine von einer Nachbar-IoT-Vorrichtung empfangene Bestätigung speichern, dass die Nachricht von der vorgeschalteten Vorrichtung empfangen wurde.

[1997] Beispiel 394 umfasst ein Verfahren zum Senden von Daten von einer Internet-der-Dinge(IoT)-Vorrichtung zu einer Nachbar-IoT-Vorrichtung zur Übertragung oder zum Speichern. Das Verfahren zum Senden von Daten von einer Internet-der-Dinge(IoT)-Vorrichtung zu einer Nachbar-IoT-Vorrichtung zur Übertragung oder Speicherung umfasst das Segmentieren einer Datei in mehrere Fragmente, das Berechnen eines Hash-Codes für jedes Fragment, das Identifizieren eines Zielknotens anhand des Hash-Codes und das Senden des Fragments zum Zielknoten.

[1998] Beispiel 395 umfasst den Gegenstand von Beispiel 394. In Beispiel 395 umfasst das Identifizieren des Zielknotens das Extrahieren einer ausgewählten Anzahl von Bytes aus dem Hash-Code und das Vergleichen der ausgewählten Anzahl von Bytes mit einer Knotenkennung für jede von mehreren IoT-Vorrichtungen und das Identifizieren des Zielknotens durch eine nächste Übereinstimmung zwischen der ausgewählten Anzahl von Bytes und einer Knotenkennung in den mehreren IoT-Vorrichtungen.

[1999] Beispiel 396 umfasst den Gegenstand von Beispiel 394 oder 395. In Beispiel 396 umfasst das Senden des Fragments zum Zielknoten das Packen des Fragments in ein Nutzinhaltsfeld in einem Frame und das Senden des Frames zum Zielknoten.

[2000] Beispiel 397 umfasst den Gegenstand von einem der Beispiele 394 bis 396. In Beispiel 397 umfasst das Verfahren das Berechnen einer Knotenkennung und das Teilen der Knotenkennung mit Nachbarknoten in mehreren IoT-Vorrichtungen.

[2001] Beispiel 398 umfasst den Gegenstand von einem der Beispiele 394 bis 397. In Beispiel 398 umfasst das Verfahren das Empfangen eines Frames von einer Nachbar-IoT-Vorrichtung, die Feststellung, dass ein Nutzinhalt im Frame das zu speichernde Fragment umfasst, das Erzeugen eines Schlüssels für den Nutzinhalt unter Verwendung einer Hash-Funktion und das Speichern von Daten im Knoten, das Hängen einer Knotenkennung vor den Schlüssel zur Erzeugung eines Datenidentifizierers und das Speichern des Datenidentifizierers in einem lokalen Speicher.

[2002] Beispiel 399 umfasst den Gegenstand von einem der Beispiele 394 bis 398. In Beispiel 399 umfasst das Verfahren das Empfangen eines Frames von einer IoT-Vorrichtung, die Feststellung, dass ein Nutzinhalt

im Frame das zu einer vorgeschalteten Vorrichtung zu übertragende Fragment umfasst, und das Senden des Frames zur vorgeschalteten Vorrichtung.

[2003] Beispiel 400 umfasst den Gegenstand von einem der Beispiele 394 bis 399. In Beispiel 400 umfasst das Verfahren das Extrahieren des Nutzinhalts aus dem Frame, das Packen des Nutzinhalts in einen neuen Frame und das Übertragen des neuen Frames zur vorgeschalteten Vorrichtung.

[2004] Beispiel 401 umfasst den Gegenstand von einem der Beispiele 394 bis 400. In Beispiel 401 umfasst das Verfahren das Fragmentieren des Frames in Segmente, das Packen der Segmente in ein Nutzinhaltsfeld jedes von mehreren neuen Frames und das Übertragen der mehreren neuen Frames zur vorgeschalteten Vorrichtung.

[2005] Beispiel 402 umfasst den Gegenstand von einem der Beispiele 394 bis 401. In Beispiel 402 umfasst das Verfahren das Empfangen einer Bestätigung der Übertragung des Frames von der vorgeschalteten Vorrichtung und das Weiterleiten der Bestätigung zur IoT-Vorrichtung.

[2006] Beispiel 403 umfasst den Gegenstand von einem der Beispiele 394 bis 402. In Beispiel 403 umfasst das Verfahren das Überwachen der Rate an der IoT-Vorrichtung von Nachbar-IoT-Vorrichtungen empfangener Bestätigungen und das Steuern der Rate des Sendens von Frames zumindest teilweise auf der Grundlage der Bestätigungsrate.

[2007] Beispiel 404 umfasst den Gegenstand von einem der Beispiele 394 bis 403. In Beispiel 404 umfasst das Verfahren die Feststellung, dass eine Bestätigung für einen zu einer Nachbar-IoT-Vorrichtung gesendeten Frame nicht empfangen wurde, das Neusenden des Frames zu einer anderen Nachbar-IoT-Vorrichtung und das Markieren der Nachbar-IoT-Vorrichtung als möglicherweise schlecht.

[2008] Beispiel 405 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, Daten in Fragmente zu segmentieren, einen Hash-Code für jedes Fragment zu berechnen, einen Zielknoten für ein Fragment zu identifizieren und das Fragment zum Zielknoten zu senden.

[2009] Beispiel 406 umfasst den Gegenstand von Beispiel 405. In Beispiel 406 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, den Hash-Code mit einer Knotenkennung für mehrere Knoten zu vergleichen, um den Zielknoten zu identifizieren.

[2010] Beispiel 407 umfasst den Gegenstand von Beispiel 405 oder 406. In Beispiel 407 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, eine Knotenkennung für einen Knoten zu berechnen und die Knotenkennung mit mehreren Knoten zu teilen.

[2011] Beispiel 408 umfasst den Gegenstand von einem der Beispiele 405 bis 407. In Beispiel 408 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, das Fragment zu empfangen, einen Schlüssel für das Fragment zu berechnen und den Schlüssel und das Fragment zu speichern.

[2012] Beispiel 409 umfasst den Gegenstand von einem der Beispiele 405 bis 408. In Beispiel 409 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, das Fragment zu empfangen und es zu einer vorgeschalteten Vorrichtung zu senden.

[2013] Beispiel 410 umfasst den Gegenstand von einem der Beispiele 405 bis 409. In Beispiel 410 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den einen oder die mehreren Prozessoren anweisen, eine Bestätigung von der vorgeschalteten Vorrichtung zu empfangen und die Bestätigung zu einer Vorrichtung, die das Fragment gesendet hat, weiterzuleiten.

[2014] Beispiel 411 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst mehrere Kommunikationsvorrichtungen, die mehrere Kommunikationskanäle bereitstellen, eine Wegentdeckungseinrichtung zur Identifikation, welche der mehreren Kommunikationskanäle in Kommunikation mit einem Zielpunkt stehen, eine Weg-

Rangbewertungseinrichtung zur Rangbewertung der mehreren Kommunikationskanäle, die in Kommunikation mit dem Zielendpunkt stehen, und eine Wegdatenbank zum Speichern der Rangbewertungen.

[2015] Beispiel 412 umfasst den Gegenstand von Beispiel 411. In Beispiel 412 umfasst die Vorrichtung eine Wegberechnungseinrichtung zur Auswahl eines Wegs von der IoT-Vorrichtung zum Zielendpunkt, wobei der Weg einen Kommunikationskanal umfasst.

[2016] Beispiel 413 umfasst den Gegenstand von Beispiel 411 oder 412. In Beispiel 413 wird der Weg zumindest teilweise auf der Grundlage der Rangbewertungen ausgewählt.

[2017] Beispiel 414 umfasst den Gegenstand von einem der Beispiele 411 bis 413. In Beispiel 414 wird der Weg zumindest teilweise auf der Grundlage einer Anwendung ausgewählt.

[2018] Beispiel 415 umfasst den Gegenstand von einem der Beispiele 411 bis 414. In Beispiel 415 umfasst die Vorrichtung einen Paketpräparierer zur Präparation von Kommunikationspaketen für die Übertragung über die Kommunikationskanäle.

[2019] Beispiel 416 umfasst den Gegenstand von einem der Beispiele 411 bis 415. In Beispiel 416 soll der Paketpräparierer über den Kommunikationskanal zu übertragende Daten fragmentieren.

[2020] Beispiel 417 umfasst den Gegenstand von einem der Beispiele 411 bis 416. In Beispiel 417 soll der Paketpräparierer zu übertragende Daten in einen Frame für das Protokoll eines Kommunikationskanals packen.

[2021] Beispiel 418 umfasst den Gegenstand von einem der Beispiele 411 bis 417. In Beispiel 418 umfasst die Vorrichtung einen Kommunikator zur Übertragung von Daten über einen Kommunikationskanal.

[2022] Beispiel 419 umfasst den Gegenstand von einem der Beispiele 411 bis 418. In Beispiel 419 sendet der Kommunikator einen Teil der Daten parallel über zwei oder mehr Kommunikationskanäle.

[2023] Beispiel 420 umfasst den Gegenstand von einem der Beispiele 411 bis 419. In Beispiel 420 umfasst die Vorrichtung eine Funktionsweise-Überwachungseinrichtung zur Verfolgung einer Funktionsweisestatistik für jeden der mehreren Kommunikationskanäle, die in Kommunikation mit dem Zielendpunkt stehen, wobei die Funktionsweise-Überwachungseinrichtung die Rangbewertungen zumindest teilweise auf der Grundlage der Funktionsweisestatistik aktualisiert.

[2024] Beispiel 421 umfasst ein Verfahren zur Auswahl eines Kommunikationskanals von einer Internet-der-Dinge(IoT)-Vorrichtung zu einem Zielendpunkt. Das Verfahren zur Auswahl eines Kommunikationskanals von einer Internet-der-Dinge(IoT)-Vorrichtung zu einem Zielendpunkt umfasst das Berechnen einer Weiterleitungsstrategie, einschließlich eines oder mehrerer verfügbarer Wege zum Zielendpunkt, das Präparieren von Daten zur Übertragung über die in der Weiterleitungsstrategie identifizierten verfügbaren Wege, das Übertragen der Daten, das Sammeln von Funktionsweisestatistiken über die für die Übertragung verwendeten verfügbaren Wege und das Aktualisieren einer Rangbewertung der verfügbaren Wege zumindest teilweise auf der Grundlage der Funktionsweisestatistiken.

[2025] Beispiel 422 umfasst den Gegenstand von Beispiel 421. In Beispiel 422 umfasst das Berechnen der Weiterleitungsstrategie das Feststellen, dass ein einziger Weg verwendet werden sollte, und das Auswählen des zu verwendenden einzelnen Wegs.

[2026] Beispiel 423 umfasst den Gegenstand von Beispiel 421 oder 422. In Beispiel 423 umfasst das Berechnen der Weiterleitungsstrategie das Feststellen, dass mehrere Wege verwendet werden sollten, und das Auswählen der mehreren zu verwendenden Wege.

[2027] Beispiel 424 umfasst den Gegenstand von einem der Beispiele 421 bis 423. In Beispiel 424 umfasst das Präparieren der Daten für die Übertragung das Fragmentieren der Daten in Nutzinhalte, das Packen der Nutzinhalte in Pakete und das Zuweisen der Pakete zu einem Weg.

[2028] Beispiel 425 umfasst den Gegenstand von einem der Beispiele 421 bis 424. In Beispiel 425 umfasst das Verfahren das Packen der Nutzinhalte in Frames auf der Grundlage des Wegs.

- [2029]** Beispiel 426 umfasst den Gegenstand von einem der Beispiele 421 bis 425. In Beispiel 426 umfasst das Übertragen der Daten das Senden von Paketen über einen einzigen verfügbaren Weg.
- [2030]** Beispiel 427 umfasst den Gegenstand von einem der Beispiele 421 bis 426. In Beispiel 427 umfasst das Übertragen der Daten das Senden von Paketen über mehrere Wege.
- [2031]** Beispiel 428 umfasst den Gegenstand von einem der Beispiele 421 bis 427. In Beispiel 428 umfasst das Sammeln von Funktionsweisestatistiken das Überwachen des Datenauslieferungsstatus, das Bestimmen der Wegverfügbarkeit oder das Messen der pro Weg übertragenen Gesamtdaten oder Kombinationen davon.
- [2032]** Beispiel 429 umfasst den Gegenstand von einem der Beispiele 421 bis 428. In Beispiel 429 umfasst das Verfahren das Entdecken von Netzschnittstellen in der IoT-Vorrichtung, das Entdecken der verfügbaren Wege zum Zielpunkt über die Netzschnittstellen und die Rangbewertung der verfügbaren Wege zum Zielpunkt.
- [2033]** Beispiel 430 umfasst den Gegenstand von einem der Beispiele 421 bis 429. In Beispiel 430 umfasst das Entdecken von Netzschnittstellen das Zugreifen auf eine Liste von Schnittstellen durch einen Konfigurationsbefehl.
- [2034]** Beispiel 431 umfasst den Gegenstand von einem der Beispiele 421 bis 430. In Beispiel 431 umfasst das Entdecken der verfügbaren Wege das Entdecken logischer Kanäle zum Zielpunkt.
- [2035]** Beispiel 432 umfasst den Gegenstand von einem der Beispiele 421 bis 431. In Beispiel 432 umfasst das Entdecken der verfügbaren Wege das Erhalten vorhergehender aktiver Wege zum Zielpunkt.
- [2036]** Beispiel 433 umfasst den Gegenstand von einem der Beispiele 421 bis 432. In Beispiel 433 umfasst das Entdecken der verfügbaren Wege den Versuch, logische Kanäle zum Zielpunkt über die Netzschnittstellen einzurichten.
- [2037]** Beispiel 434 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Netzschnittstellen zu entdecken, verfügbare Wege zu entdecken, eine Weiterleitungsstrategie zu berechnen und Pakete entsprechend der Weiterleitungsstrategie zu übertragen.
- [2038]** Beispiel 435 umfasst den Gegenstand von Beispiel 434. In Beispiel 435 umfasst das nichtflüchtige maschinenlesbare Medium das Präparieren der Pakete zur Übertragung durch Fragmentieren von Daten zur Bildung von Nutzinhalt für die Pakete.
- [2039]** Beispiel 436 umfasst den Gegenstand von Beispiel 434 oder 435. In Beispiel 436 umfasst das nichtflüchtige maschinenlesbare Medium das Sammeln von Funktionsweisestatistiken über in der Weiterleitungsstrategie verwendete Wege.
- [2040]** Beispiel 437 umfasst den Gegenstand von einem der Beispiele 436 bis 436. In Beispiel 437 umfasst das nichtflüchtige maschinenlesbare Medium das Aktualisieren von Rangbewertungen verfügbarer Wege zumindest teilweise auf der Grundlage der Funktionsweisestatistiken.
- [2041]** Beispiel 438 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz ein IoT-Gateway umfasst. Der IoT-Gateway umfasst eine Kommunikationsschnittstelle zu einer ersten Domain, einschließlich Kommunikationen in einem ersten Protokoll und auf einem ersten Sicherheitsniveau, eine Kommunikationsschnittstelle zu einer zweiten Domain, einschließlich Kommunikationen in einem zweiten Protokoll und auf einem zweiten Sicherheitsniveau, einen Protokollübersetzer zum Übersetzen von Nutzinhalt zwischen dem ersten Protokoll und dem zweiten Protokoll und einen Sicherheitsübersetzer zum Übersetzen von Nutzinhalt zwischen dem ersten Protokoll und dem zweiten Protokoll.
- [2042]** Beispiel 439 umfasst den Gegenstand von Beispiel 438. In Beispiel 439 umfasst die Vorrichtung einen Semantikübersetzer zum Übersetzen semantischer Repräsentationen von Eingangsdaten in semantische Repräsentationen, die in Ausgangsdaten verwendet werden.
- [2043]** Beispiel 440 umfasst den Gegenstand von Beispiel 438 oder 439. In Beispiel 440 umfasst die Vorrichtung ein Semantikübersetzer-Plugin, um den Semantikübersetzer in die Lage zu versetzen, die semantischen

Repräsentationen eines Nutzinhalts in den Eingangsdaten in die in den Ausgangsdaten verwendeten semantischen Repräsentationen zu übersetzen.

[2044] Beispiel 441 umfasst den Gegenstand von einem der Beispiele 438 bis 440. In Beispiel 441 umfasst die Vorrichtung ein erstes semantisches Plugin, um den Semantikübersetzer in die Lage zu versetzen, die semantischen Repräsentationen eines Nutzinhalts in den Eingangsdaten in eine semantische Zwischenrepräsentation zu übersetzen.

[2045] Beispiel 442 umfasst den Gegenstand von einem der Beispiele 438 bis 441. In Beispiel 442 umfasst die Vorrichtung ein zweites semantisches Plugin, um den Semantikübersetzer in die Lage zu versetzen, die semantischen Repräsentationen eines Nutzinhalts von einer semantischen Zwischenrepräsentation in die in den Ausgangsdaten verwendeten semantischen Repräsentationen zu übersetzen.

[2046] Beispiel 443 umfasst den Gegenstand von einem der Beispiele 438 bis 442. In Beispiel 443 umfasst die Vorrichtung eine vertrauenswürdige Ausführungsumgebung, die mehrere Übersetzer zur Umwandlung eines Eingangsformats in ein Ausgangsformat umfasst.

[2047] Beispiel 444 umfasst den Gegenstand von einem der Beispiele 438 bis 443. In Beispiel 444 umfasst die Vorrichtung ein vertrauenswürdiges Plattformmodul zur Unterstützung der vertrauenswürdigen Ausführungsumgebung.

[2048] Beispiel 445 umfasst den Gegenstand von einem der Beispiele 438 bis 444. In Beispiel 445 umfasst die Vorrichtung ein erstes Protokoll-Plugin, um den Protokollübersetzer in die Lage zu versetzen, ein Eingangsprotokoll aus einem Nutzinhalt zu entfernen.

[2049] Beispiel 446 umfasst den Gegenstand von einem der Beispiele 438 bis 445. In Beispiel 446 umfasst die Vorrichtung ein zweites Protokoll-Plugin, um den Protokollübersetzer in die Lage zu versetzen, einen Nutzinhalt in ein Ausgangsprotokoll zu packen.

[2050] Beispiel 447 umfasst den Gegenstand von einem der Beispiele 438 bis 446. In Beispiel 447 umfasst die Vorrichtung ein erstes Sicherheits-Plugin, um den Sicherheitsübersetzer in die Lage zu versetzen, einen auf einem Eingangssicherheitsniveau verschlüsselten Nutzinhalt zu entschlüsseln.

[2051] Beispiel 448 umfasst den Gegenstand von einem der Beispiele 438 bis 447. In Beispiel 448 umfasst die Vorrichtung ein zweites Sicherheits-Plugin, um den Sicherheitsübersetzer in die Lage zu versetzen, einen Nutzinhalt auf einem Ausgangssicherheitsniveau zu verschlüsseln.

[2052] Beispiel 449 umfasst den Gegenstand von einem der Beispiele 438 bis 448. In Beispiel 449 umfasst die Vorrichtung einen sicheren Datenspeicher, der eine Sicherheitsübersetzungsrichtlinie zum Steuern einer zwischen einem Eingangs-Nutzinhalt und einem Ausgangs-Nutzinhalt erlaubten Sicherheitsniveaudifferenz umfasst.

[2053] Beispiel 450 umfasst den Gegenstand von einem der Beispiele 438 bis 449. In Beispiel 450 umfasst die Vorrichtung einen sicheren Datenspeicher, der einen Skript-Compiler zum Kompilieren eines Plugins umfasst.

[2054] Beispiel 451 umfasst den Gegenstand von einem der Beispiele 438 bis 450. In Beispiel 451 umfasst die Vorrichtung einen sicheren Datenspeicher, der einen Plugin-Installer umfasst.

[2055] Beispiel 452 umfasst ein Verfahren zur Übersetzung einer Kommunikation zwischen Domains in einem Internet-der-Dinge(IoT)-Gateway. Das Verfahren zur Übersetzung der Kommunikation zwischen Domains in einem Internet-der-Dinge(IoT)-Gateway umfasst das Empfangen eines Nutzinhalts in einem Eingangsprotokoll von einer ersten Domain, das Entfernen des Nutzinhalts aus dem Eingangsprotokoll, das Entschlüsseln des Nutzinhalts von einem Eingangssicherheitsniveau, falls der Nutzinhalt verschlüsselt ist, und das Übersetzen des Nutzinhalts von einer semantischen Eingangsrepräsentation in eine semantische Ausgangsrepräsentation, falls sie bei der semantischen Eingangsrepräsentation verschieden ist. Das Verfahren umfasst auch das Verschlüsseln des Nutzinhalts auf einem Ausgangssicherheitsniveau, falls dies von einer Sicherheitsrichtlinie gefordert wird, das Packen des Nutzinhalts in ein Ausgangsprotokoll und das Übertragen des Nutzinhalts in einer zweiten Domain.

[2056] Beispiel 453 umfasst den Gegenstand von Beispiel 452. In Beispiel 453 umfasst das Verfahren nach dem Entfernen des Nutzinhalts aus dem Eingangsprotokoll die Feststellung, ob der Nutzinhalt durch eine Biometrik geschützt ist, und, falls dies der Fall ist, das Analysieren der Biometrik auf Inhalt.

[2057] Beispiel 454 umfasst den Gegenstand von Beispiel 452 oder 453. In Beispiel 454 umfasst das Verfahren nach dem Verschlüsseln des Nutzinhalts das Schützen des Nutzinhalts mit einer Simulation der Biometrik.

[2058] Beispiel 455 umfasst den Gegenstand von einem der Beispiele 452 bis 454. In Beispiel 455 umfasst das Verfahren nach dem Entfernen des Nutzinhalts aus dem Eingangsprotokoll die Feststellung, ob der Nutzinhalt datenschutzsensibel ist, und falls dies der Fall ist, das Anwenden einer Datenschutzrichtlinie zur Analyse des Nutzinhalts auf Inhalt.

[2059] Beispiel 456 umfasst den Gegenstand von einem der Beispiele 452 bis 455. In Beispiel 456 umfasst das Verfahren nach dem Verschlüsseln des Nutzinhalts das Anwenden der Datenschutzrichtlinie, um den Nutzinhalt zu schützen.

[2060] Beispiel 457 umfasst den Gegenstand von einem der Beispiele 452 bis 456. In Beispiel 457 umfasst das Verfahren das Identifizieren mehrerer Plugins zum Ermöglichen einer Übersetzung von Kommunikationen zwischen einer ersten Domain und einer zweiten Domain, das Validieren von jedem der mehreren Plugins, um festzustellen, dass es sich dabei um ein gültiges Plugin handelt, die Feststellung, ob jedes gültige Plugin mit dem IoT-Gateway kompatibel ist, und, falls dies der Fall ist, die Installation jedes gültigen Plugins.

[2061] Beispiel 458 umfasst den Gegenstand von einem der Beispiele 452 bis 457. In Beispiel 458 umfasst das Verfahren das Kompilieren eines Plugins, um die Übersetzung, welche das Plugin ausführt, zu beschleunigen.

[2062] Beispiel 459 umfasst den Gegenstand von einem der Beispiele 452 bis 458. In Beispiel 459 umfasst das Verfahren das Konfigurieren jedes gültigen Plugins für die Übersetzung.

[2063] Beispiel 460 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Eingangsdaten von einer ersten Domain zu empfangen, die Eingangsdaten nach einem Eingangsprotokoll zu verarbeiten, um einen Nutzinhalt zu extrahieren, den Nutzinhalt nach der Eingangs-Sicherheitsrichtlinie zu verarbeiten, den Nutzinhalt aus einem semantischen Eingangsformat in ein semantisches Ausgangsformat zu übersetzen, den Nutzinhalt nach einer Ausgangs-Sicherheitsrichtlinie zu verarbeiten, den Nutzinhalt nach einem Ausgangsprotokoll zu verarbeiten, um ihn in Ausgangsdaten zu packen, und die Ausgangsdaten zu einer zweiten Domain zu senden.

[2064] Beispiel 461 umfasst den Gegenstand von Beispiel 460. In Beispiel 461 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Nutzinhalt nach einer biometrischen Eingangsrichtlinie zu verarbeiten, um Inhalt zu analysieren, und den Nutzinhalt nach einer biometrischen Ausgangsrichtlinie zu verarbeiten, um den Inhalt zu schützen.

[2065] Beispiel 462 umfasst den Gegenstand von Beispiel 460 oder 461. In Beispiel 462 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Nutzinhalt nach einer Eingangs-Datenschutzrichtlinie zu verarbeiten, um Inhalt zu analysieren, und den Nutzinhalt nach einer Ausgangs-Datenschutzrichtlinie zu verarbeiten, um den Inhalt zu schützen.

[2066] Beispiel 463 umfasst den Gegenstand von einem der Beispiele 460 bis 462. In Beispiel 463 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Softwaremessung auszuführen und ein Ergebnis für die Softwaremessung mit einem akzeptierten Wert zu vergleichen.

[2067] Beispiel 464 umfasst den Gegenstand von einem der Beispiele 460 bis 463. In Beispiel 464 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Plugin zu validieren und das Plugin zu installieren.

[2068] Beispiel 465 umfasst den Gegenstand von einem der Beispiele 460 bis 464. In Beispiel 465 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Plugin zu kompilieren.

[2069] Beispiel 466 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst ein Kommunikationssystem zur Kommunikation mit anderen IoT-Vorrichtungen in einer Domain, ein Eingliederungswerkzeug zur Entdeckung einer Vorrichtung in der Domain und der Erzeugung von Ressourcen für die Vorrichtung, einen Vorrichtungsentdecker zur Entdeckung einer fernen Vorrichtung, die von einem fernen Eingliederungswerkzeug bedient wird, das sich in einer fernen Domain befindet, einen Vertrauensbilder zum Herstellen von Vertrauen mit dem fernen Eingliederungswerkzeug, einen Geteilte-Domain-Erzeuger zur Bildung einer geteilten Domain mit dem fernen Eingliederungswerkzeug und ein geteiltes Ressourcenverzeichnis, das Ressourcen sowohl für die Vorrichtung als auch für die ferne Vorrichtung speichert.

[2070] Beispiel 467 umfasst den Gegenstand von Beispiel 466. In Beispiel 467 wird die Vorrichtung in der Domain durch Ressourcen in einem ersten Ressourcenblock repräsentiert, wird die ferne Vorrichtung in der fernen Domain durch Ressourcen in einem zweiten Ressourcenblock repräsentiert und speichert eine virtuelle Domain einen dritten Ressourcenblock, wobei der dritte Ressourcenblock die Ressourcen vom ersten Ressourcenblock und vom zweitem Ressourcenblock umfasst.

[2071] Beispiel 468 umfasst den Gegenstand von Beispiel 466 oder 467. In Beispiel 468 umfasst das Kommunikationssystem einen Mesh-Sendeempfänger, einen Uplink-Sendeempfänger oder eine Netzschnittstellen-Steuereinrichtung oder Kombinationen davon.

[2072] Beispiel 469 umfasst den Gegenstand von einem der Beispiele 466 bis 468. In Beispiel 469 umfasst die Vorrichtung mehrere Vorrichtungen in mehreren Domains, die eine Fog-Vorrichtung bilden, wobei ein gemeinsamer Ressourcenblock in einer virtuellen Domain Ressourcen für alle der mehreren Vorrichtungen speichert.

[2073] Beispiel 470 umfasst den Gegenstand von einem der Beispiele 466 bis 469. In Beispiel 470 umfasst die Vorrichtung einen Orchestrator zur Bereitstellung von Informationen über mehrere ferne Vorrichtungen für das Eingliederungswerkzeug.

[2074] Beispiel 471 umfasst den Gegenstand von einem der Beispiele 466 bis 470. In Beispiel 471 umfasst der Vertrauensbilder einen Attestierungsschlüssel, einen Identifikationsschlüssel oder ein zugewiesenes Vertrauen von einem Administrator oder Kombinationen davon.

[2075] Beispiel 472 umfasst den Gegenstand von einem der Beispiele 466 bis 471. In Beispiel 472 umfasst der Vertrauensbilder ein Blockchain-System zur Bildung einer Blockchain-Vertrauens-Root.

[2076] Beispiel 473 umfasst ein Verfahren zum Teilen von Ressourcen über Domains. Das Verfahren zum Teilen von Ressourcen über Domains umfasst das Beitreten einer Vorrichtung zu einem IoT-Netz in einer ersten Domain, das Hinzufügen von Ressourcen für die Vorrichtung in der ersten Domain zu einem lokalen Ressourcenblock, das Entdecken einer fernen Vorrichtung in einer fernen Domain, das Erzeugen von Vertrauen mit der fernen Domain und das Erzeugen eines geteilten Ressourcenblocks, der Ressourcen für die Vorrichtung und die ferne Vorrichtung umfasst.

[2077] Beispiel 474 umfasst den Gegenstand von Beispiel 473. In Beispiel 474 umfasst das Verfahren das Erzeugen eines lokalen Ressourcenblocks, falls er nicht bereits existiert.

[2078] Beispiel 475 umfasst den Gegenstand von Beispiel 473 oder 474. In Beispiel 475 umfasst das Entdecken der fernen Vorrichtung das Empfangen von Informationen von einem Orchestrator über die ferne Vorrichtung.

[2079] Beispiel 476 umfasst den Gegenstand von einem der Beispiele 473 bis 475. In Beispiel 476 umfasst das Entdecken der fernen Vorrichtung das Entdecken eines Eingliederungswerkzeugs in der fernen Domain und das Austauschen von Vorrichtungsinformationen mit dem Eingliederungswerkzeug in der fernen Domain.

[2080] Beispiel 477 umfasst den Gegenstand von einem der Beispiele 473 bis 476. In Beispiel 477 umfasst das Erzeugen von Vertrauen mit der fernen Vorrichtung das Austauschen von Attestierungsinformationen mit der fernen Vorrichtung.

[2081] Beispiel 478 umfasst den Gegenstand von einem der Beispiele 473 bis 477. In Beispiel 478 umfasst das Erzeugen von Vertrauen mit der fernen Vorrichtung das Nachschlagen der fernen Vorrichtung in einer Blockchain.

[2082] Beispiel 479 umfasst den Gegenstand von einem der Beispiele 473 bis 478. In Beispiel 479 umfasst das Erzeugen von Vertrauen mit der fernen Vorrichtung das Akzeptieren einer zugewiesenen Vertrauenseinstellung von einem Administrator.

[2083] Beispiel 480 umfasst den Gegenstand von einem der Beispiele 473 bis 479. In Beispiel 480 umfasst das Verfahren das Umbenennen einer Subdomain-Kennung, falls die Subdomain-Kennung mit einer vorhergehenden Subdomain-Kennung im geteilten Ressourcenblock übereinstimmt, und das Übermitteln der neuen Subdomain-Kennung zu allen Vorrichtungen, welche die Subdomain-Kennung verwenden.

[2084] Beispiel 481 umfasst den Gegenstand von einem der Beispiele 473 bis 480. In Beispiel 481 umfasst das Verfahren das Umbenennen einer Objektkennung, falls die Objektkennung mit einer vorhergehenden Objektkennung im geteilten Ressourcenblock übereinstimmt, und das Übermitteln der neuen Objektkennung zu allen Vorrichtungen, welche die Objektkennung verwenden.

[2085] Beispiel 482 umfasst den Gegenstand von einem der Beispiele 473 bis 481. In Beispiel 482 umfasst das Verfahren das Umbenennen einer Vorrichtungskennung, falls die Vorrichtungskennung mit einer vorhergehenden Vorrichtungskennung im geteilten Ressourcenblock übereinstimmt, und das Übermitteln der neuen Vorrichtungskennung zu allen Vorrichtungen, welche die Vorrichtungskennung verwenden.

[2086] Beispiel 483 umfasst den Gegenstand von einem der Beispiele 473 bis 482. In Beispiel 483 umfasst das Verfahren das Zugreifen auf die Ressourcen der fernen Vorrichtung von der ersten Domain.

[2087] Beispiel 484 umfasst den Gegenstand von einem der Beispiele 473 bis 483. In Beispiel 484 umfasst das Erzeugen des geteilten Ressourcenblocks das Bilden einer Vereinigung zwischen dem lokalen Ressourcenblock und einem fernen Ressourcenblock.

[2088] Beispiel 485 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Vorrichtung zu einem IoT-Netz hinzuzufügen, eine lokale Ressource für die Vorrichtung in einem lokalen Ressourcenblock zu erzeugen, Vorrichtungen in anderen Domains zu entdecken, eine geteilte Domain zu erzeugen, einen geteilten Ressourcenblock in der geteilten Domain zu erzeugen und die lokale Ressource und ferne Ressourcen für Vorrichtungen in den anderen Domains im geteilten Ressourcenblock zu vereinigen.

[2089] Beispiel 486 umfasst den Gegenstand von Beispiel 485. In Beispiel 486 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Eingliederungswerkzeug in einer fernen Domain zu entdecken, Vertrauen mit dem Eingliederungswerkzeug in der fernen Domain zu erzeugen und Informationen über mehrere Vorrichtungen in einer lokalen Domain und in der fernen Domain auszutauschen.

[2090] Beispiel 487 umfasst den Gegenstand von Beispiel 485 oder 486. In Beispiel 487 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Eingliederungswerkzeug in einer fernen Domain zu entdecken und auf eine Blockchain zuzugreifen, um das Eingliederungswerkzeug in der fernen Domain zu validieren.

[2091] Beispiel 488 umfasst den Gegenstand von einem der Beispiele 485 bis 487. In Beispiel 488 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Namensüberlappung im geteilten Ressourcenblock zu erkennen, die Namensüberlappung durch Ändern eines überlappenden Eintrags zu einem neuen Namen zu korrigieren, und den neuen Namen an alle Vorrichtungen zu übermitteln, die diesen Namen verwenden.

[2092] Beispiel 489 umfasst den Gegenstand von einem der Beispiele 485 bis 488. In Beispiel 489 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Fog-Vorrichtung zu bilden, welche eine Vorrichtung in einer lokalen Domain und eine ferne Vorrichtung in einer fernen Domain umfasst.

[2093] Beispiel 490 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst, die einen Datensatzschlüsselgenerator zur Erzeugung eines Datensatzschlüssels für eine Stufe eines Herstellungsprozesses eines Produkts, einen Privater-Speicher-Kommunikator zur Kommunikation mit einem privaten Speicher zum Speichern des Datensatzschlüssels und zugeordneter für die Stufe von der IoT-Vorrichtung gesammelter Daten und einen Öffentlicher-Speicher-Kom-

munikator zur Kommunikation mit einem öffentlichen Speicher zum Speichern des Datensatzschlüssels im öffentlichen Speicher umfasst.

[2094] Beispiel 491 umfasst den Gegenstand von Beispiel 490. In Beispiel 491 erhält der Datensatzschlüsselgenerator einen Datensatzschlüssel von einem verteilten Ledger.

[2095] Beispiel 492 umfasst den Gegenstand von Beispiel 490 oder 491. In Beispiel 492 wird dem Datensatzschlüsselgenerator ein Datensatzschlüssel von einem zentralen Administrator erteilt.

[2096] Beispiel 493 umfasst den Gegenstand von einem der Beispiele 490 bis 492. In Beispiel 493 umfasst die Vorrichtung einen Datensatzschlüsselspeicher zum Speichern des Datensatzschlüssels für eine Stufe.

[2097] Beispiel 494 umfasst den Gegenstand von einem der Beispiele 490 bis 493. In Beispiel 494 soll der Datensatzschlüsselspeicher einen angehängten Datensatzschlüssel speichern, wobei der angehängte Datensatzschlüssel den an Datensatzschlüssel für vorhergehende Stufen des Herstellungsprozesses angehängten Datensatzschlüssel für die Stufe des Herstellungsprozesses umfasst.

[2098] Beispiel 495 umfasst den Gegenstand von einem der Beispiele 490 bis 494. In Beispiel 495 soll der Öffentlicher-Speicher-Kommunikator den angehängten Datensatzschlüssel für alle Stufen im öffentlichen Speicher speichern.

[2099] Beispiel 496 umfasst den Gegenstand von einem der Beispiele 490 bis 495. In Beispiel 496 umfasst die Vorrichtung eine Datenüberwachungseinrichtung zur Überwachung eines Sensors über eine Schnittstelle und zum Assoziieren der Daten mit dem Datensatzschlüssel.

[2100] Beispiel 497 umfasst den Gegenstand von einem der Beispiele 490 bis 496. In Beispiel 497 umfasst der Sensor einen Temperatursensor, einen Schocksensor oder ein globales Positionsbestimmungssystem oder Kombinationen davon.

[2101] Beispiel 498 umfasst den Gegenstand von einem der Beispiele 490 bis 497. In Beispiel 498 umfasst die Vorrichtung eine Warneinrichtung zum Aktivieren einer Warnangabe, falls eine Versandverlaufsgrenze verletzt wurde.

[2102] Beispiel 499 umfasst den Gegenstand von einem der Beispiele 490 bis 498. In Beispiel 499 umfasst die Versandverlaufsgrenze eine Temperaturgrenze, eine Schockgrenze, eine Lichtgrenze oder Kombinationen davon.

[2103] Beispiel 500 umfasst den Gegenstand von einem der Beispiele 490 bis 499. In Beispiel 500 umfasst die Warnung ein Betätigungselement für eine Lampe, einen Tongenerator oder beides.

[2104] Beispiel 501 umfasst den Gegenstand von einem der Beispiele 490 bis 500. In Beispiel 501 umfasst die Warnung eine über einen Mesh-Sendeempfänger, einen Uplink-Sendeempfänger oder beide gesendete Nachricht.

[2105] Beispiel 502 umfasst ein Verfahren zur Implementation eines Verfolgbarkeitssystems unter Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur Implementation eines Verfolgbarkeitssystems unter Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Empfangen eines Produktversandschlüssels für das Produkt in der IoT-Vorrichtung einer IoT-Vorrichtung mit einem Produkt, das Empfangen eines Verarbeitungsschlüssels für das Produkt in der IoT-Vorrichtung, das Empfangen eines Versandschlüssels für ein verarbeitetes Produkt in der IoT-Vorrichtung, das Empfangen eines Verkaufspunktschlüssels in der IoT-Vorrichtung für das verarbeitete Produkt und das Senden eines angehängten Schlüssels, einschließlich des Produktversandschlüssels, des verarbeiteten Schlüssels, des Versandschlüssels für das verarbeitete Produkt und des Verkaufspunktschlüssels von der IoT-Vorrichtung zu einem öffentlichen Datenspeicher.

[2106] Beispiel 503 umfasst den Gegenstand von Beispiel 502. In Beispiel 503 umfasst das Verfahren das Sammeln von Daten über den Produktionsprozess für das Produkt, das Speichern der Daten für den Produktionsprozess in einem Produktionsspeicher, das Erzeugen eines Produktionsdatenschlüssels in Zusammenhang mit den Daten für den Produktionsprozess und das Speichern des Produktionsdatenschlüssels in der IoT-Vorrichtung.

- [2107]** Beispiel 504 umfasst den Gegenstand von Beispiel 502 oder 503. In Beispiel 504 umfasst das Assoziieren der IoT-Vorrichtung mit dem Produkt das Anhängen der IoT-Vorrichtung an die Produktverpackung.
- [2108]** Beispiel 505 umfasst den Gegenstand von einem der Beispiele 502 bis 504. In Beispiel 505 umfasst das Verfahren das Sammeln von Versanddaten beim Versenden des Produkts, das Speichern der Versanddaten in einem Versandspeicher, das Erzeugen des Produktversandschlüssels und das Assoziieren des Versandschlüssels mit den Versanddaten im Versandspeicher.
- [2109]** Beispiel 506 umfasst den Gegenstand von einem der Beispiele 502 bis 505. In Beispiel 506 umfassen die Daten für das Versenden des Produkts Temperaturmessungen, Beschleunigungskraftmessungen oder Ortskoordinaten oder Kombinationen davon.
- [2110]** Beispiel 507 umfasst den Gegenstand von einem der Beispiele 502 bis 506. In Beispiel 507 umfasst das Verfahren das Sammeln von Verarbeitungsdaten über die Verarbeitung des Produkts zur Bildung des verarbeiteten Produkts, das Speichern der Verarbeitungsdaten für die Verarbeitung des Produkts in einem Verarbeitungsspeicher, das Erzeugen des Verarbeitungsschlüssels und das Assoziieren des Verarbeitungsschlüssels mit den Verarbeitungsdaten im Verarbeitungsspeicher.
- [2111]** Beispiel 508 umfasst den Gegenstand von einem der Beispiele 502 bis 506. In Beispiel 508 umfasst das Verarbeiten des Produkts zur Bildung des verarbeiteten Produkts die Verpackung des Produkts an einem Lagerhaus.
- [2112]** Beispiel 509 umfasst den Gegenstand von einem der Beispiele 502 bis 508. In Beispiel 509 umfasst das Verfahren das Sammeln von Versanddaten für das verarbeitete Produkt beim Versenden des verarbeiteten Produkts, das Speichern der Versanddaten für das verarbeitete Produkt in einem Verarbeitungsspeicher, das Erzeugen des Versandschlüssels für das verarbeitete Produkt und das Assoziieren des Versandschlüssels für das verarbeitete Produkt mit den Versanddaten im Verarbeitungsspeicher.
- [2113]** Beispiel 510 umfasst den Gegenstand von einem der Beispiele 502 bis 509. In Beispiel 510 umfasst das Verfahren das Sammeln von Verkaufspunktdateien über das verarbeitete Produkt am Verkaufspunkt, das Speichern der Verkaufspunktdateien für das verarbeitete Produkt in einem Verkaufsspeicher, das Erzeugen des Verkaufspunktschlüssels und das Assoziieren des Verkaufspunktschlüssels mit den Verkaufspunktdateien im Verkaufsspeicher.
- [2114]** Beispiel 511 umfasst den Gegenstand von einem der Beispiele 502 bis 510. In Beispiel 511 geschieht das Senden von jedem der Schlüssel von der IoT-Vorrichtung zu einem öffentlichen Datenspeicher, wenn das mit der IoT-Vorrichtung assoziierte Produkt verkauft wird.
- [2115]** Beispiel 512 umfasst den Gegenstand von einem der Beispiele 502 bis 511. In Beispiel 512 umfasst das Verfahren das Löschen der Daten aus der IoT-Vorrichtung und das Zurückführen der IoT-Vorrichtung zu einer Produktionsanlage, um sie einem neuen Produkt zuzuordnen.
- [2116]** Beispiel 513 umfasst den Gegenstand von einem der Beispiele 502 bis 512. In Beispiel 513 umfasst das Verfahren das Nachschlagen eines Verfolgbarkeits-Hashes zumindest teilweise auf der Grundlage von Informationen in Zusammenhang mit der Produktverpackung, das Nachschlagen eines Schlüssels in einem öffentlichen Speicher zumindest teilweise auf der Grundlage des Verfolgbarkeits-Hashes und das Zugreifen auf Informationen in einem privaten Speicher zumindest teilweise auf der Grundlage des Schlüssels vom öffentlichen Speicher.
- [2117]** Beispiel 514 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Daten für eine Produktionsstufe in einer IoT-Vorrichtung aufzuzeichnen, die Daten von der IoT-Vorrichtung zu einem privaten Speicher zu übermitteln, einen Datensatzschlüssel für die Stufe in der IoT-Vorrichtung zu erzeugen und den Datensatzschlüssel von der IoT-Vorrichtung zum privaten Speicher zu übermitteln.
- [2118]** Beispiel 515 umfasst den Gegenstand von Beispiel 514. In Beispiel 515 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Datensatzschlüssel an einen Schlüssel erzeugt ist für vorhergehende Stufen anzuhängen, um einen angehängten Schlüssel zu erzeugen.

[2119] Beispiel 516 umfasst den Gegenstand von Beispiel 514 oder 515. In Beispiel 516 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den angehängten Schlüssel zu einem öffentlichen Speicher zu senden.

[2120] Beispiel 517 umfasst den Gegenstand von einem der Beispiele 514 bis 516. In Beispiel 517 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Warnung zu aktivieren, falls Datengrenzen verletzt werden.

[2121] Beispiel 518 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst ein Kommunikationssystem zur Kommunikation mit anderen IoT-Vorrichtungen im IoT-Netz, eine Richtlinien-Entscheidungsmaschine zur Feststellung, welche Richtlinien erzwungen werden, einen Richtlinien-Ablagespeicher zum Speichern der Richtlinien und Zustandsinformationen, die von einer Netzüberwachungseinrichtung übermittelt werden, eine Richtlinien-Erzwingungsmaschine zur Erzwingung der von der Richtlinien-Entscheidungsmaschine festgelegten Richtlinien und eine Peer-Überwachungseinrichtung zum Überwachen von der IoT-Vorrichtung und anderen IoT-Vorrichtungen im IoT-Netz erzwungener Richtlinien.

[2122] Beispiel 519 umfasst den Gegenstand von Beispiel 518. In Beispiel 519 umfasst das Kommunikationssystem einen Mesh-Sendeempfänger, einen Uplink-Sendeempfänger oder eine Netzschnittstellen-Steuereinrichtung oder Kombinationen davon.

[2123] Beispiel 520 umfasst den Gegenstand von Beispiel 518 oder 519. In Beispiel 520 umfasst das IoT-Netz mehrere eine Fog-Vorrichtung bildende Vorrichtungen.

[2124] Beispiel 521 umfasst den Gegenstand von einem der Beispiele 518 bis 520. In Beispiel 521 soll die Richtlinien-Entscheidungsmaschine eine Entscheidung, welche Richtlinien erzwungen werden, auf der Grundlage eines Parameters für die IoT-Vorrichtung fällen.

[2125] Beispiel 522 umfasst den Gegenstand von einem der Beispiele 518 bis 521. In Beispiel 522 umfasst der Parameter die Restkapazität einer Batterie in der IoT-Vorrichtung, andere durch einen Mesh-Sendeempfänger angeschlossene IoT-Vorrichtungen oder einen Status eines Uplink-Sendeempfängers für Vorrichtungen in einer Cloud oder Kombinationen davon.

[2126] Beispiel 523 umfasst den Gegenstand von einem der Beispiele 518 bis 522. In Beispiel 523 ändert die Richtlinien-Entscheidungsmaschine die erzwungenen Richtlinien zumindest teilweise auf der Grundlage einer Änderung des Parameters.

[2127] Beispiel 524 umfasst den Gegenstand von einem der Beispiele 518 bis 523. In Beispiel 524 verteilt die Richtlinien-Entscheidungsmaschine Richtlinien an nicht konfigurierte Knoten.

[2128] Beispiel 525 umfasst den Gegenstand von einem der Beispiele 518 bis 524. In Beispiel 525 sammelt die Peer-Überwachungseinrichtung Informationen und speichert sie in einer Datenbank.

[2129] Beispiel 526 umfasst den Gegenstand von einem der Beispiele 518 bis 525. In Beispiel 526 umfassen die Informationen die aktuelle Vorrichtungskonfiguration, eine Fähigkeit eines Peer-Knotens, einen angebotenen Dienst, eine Knotenanforderung für ein Netz, eine Ressourcenverfügbarkeitsschätzung oder ein ausgelöstes Ereignis oder Kombinationen davon.

[2130] Beispiel 527 umfasst den Gegenstand von einem der Beispiele 518 bis 526. In Beispiel 527 umfasst die Vorrichtung einen Koordinator zum Verteilen von Richtlinien an Peer-Knoten im IoT-Netz.

[2131] Beispiel 528 umfasst den Gegenstand von einem der Beispiele 518 bis 527. In Beispiel 528 umfasst der Koordinator ein Gateway zwischen dem IoT-Netz und einer Cloud-Vorrichtung.

[2132] Beispiel 529 umfasst den Gegenstand von einem der Beispiele 518 bis 528. In Beispiel 529 soll die IoT-Vorrichtung Richtlinien an nächstgelegene Nachbarknoten verteilen.

[2133] Beispiel 530 umfasst den Gegenstand von einem der Beispiele 518 bis 529. In Beispiel 530 kommuniziert die Richtlinien-Entscheidungsmaschine mit Peer-Knoten, um auf Richtlinien zuzugreifen.

[2134] Beispiel 531 umfasst ein Verfahren zur Behandlung der Richtlinienverteilung über IoT-Vorrichtungen in einem IoT-Netz. Das Verfahren zur Behandlung der Richtlinienverteilung über IoT-Vorrichtungen in einem IoT-Netz umfasst das Empfangen einer Entdeckungsnachricht an einem Knoten, wobei die Entdeckungsnachricht neue Richtlinien identifizieren soll, Richtlinien ändern soll oder beides ausführen soll, das Antworten auf die Entdeckungsnachricht mit einer Angebotsnachricht, wobei die Angebotsnachricht Richtlinien identifiziert, das Empfangen einer Akzeptanznachricht, wobei die Akzeptanznachricht die Richtlinien anfordert, und das Antworten mit einer Nachricht, welche die Richtlinien aufweist.

[2135] Beispiel 532 umfasst den Gegenstand von Beispiel 531. In Beispiel 532 umfasst das Verfahren das Installieren von einem Peer-Knoten empfangener Richtlinien und das Aktualisieren eines Status zu einem konfigurierten Knoten.

[2136] Beispiel 533 umfasst den Gegenstand von Beispiel 531 oder 532. In Beispiel 533 umfasst das Verfahren das Empfangen einer aktualisierten Richtlinie in einer Aktualisierungsnachricht.

[2137] Beispiel 534 umfasst den Gegenstand von einem der Beispiele 531 bis 533. In Beispiel 534 umfasst das Verfahren das Ausführen einer Validierung an der in der Aktualisierungsnachricht empfangenen aktualisierten Richtlinie und das Installieren der aktualisierten Richtlinie.

[2138] Beispiel 535 umfasst den Gegenstand von einem der Beispiele 531 bis 534. In Beispiel 535 umfasst die Validierung die Feststellung, ob die neue Richtlinie in Konflikt mit einer aktuellen Richtlinie gelangt, und, falls dies der Fall ist, das Ablehnen der neuen Richtlinie.

[2139] Beispiel 536 umfasst den Gegenstand von einem der Beispiele 531 bis 535. In Beispiel 536 umfasst die Validierung die Feststellung, ob die neue Richtlinie in Konflikt mit einer aktuellen Richtlinie gelangt, und, falls dies der Fall ist, das teilweise Implementieren der neuen Richtlinie.

[2140] Beispiel 537 umfasst den Gegenstand von einem der Beispiele 531 bis 536. In Beispiel 537 umfasst das Verfahren das Senden einer Konfliktwarnnachricht zu einem Peer-Knoten, um den Peer-Knoten vor einem Richtlinienkonflikt zu warnen.

[2141] Beispiel 538 umfasst den Gegenstand von einem der Beispiele 531 bis 537. In Beispiel 538 umfasst das Verfahren das Empfangen einer Entdeckungsnachricht vom Peer-Knoten für die Richtlinienaktualisierung, das Antworten mit einer Angebotsnachricht, das Empfangen einer Akzeptanznachricht vom Peer-Knoten, um anzugeben, dass die Richtlinienaktualisierung gesendet werden kann, und das Senden einer Aktualisierungsnachricht, welche die neue Richtlinie umfasst.

[2142] Beispiel 539 umfasst den Gegenstand von einem der Beispiele 531 bis 538. In Beispiel 539 umfasst das Verfahren das Ausführen einer Validierung an der in der Aktualisierungsnachricht empfangenen aktualisierten Richtlinie und das Installieren der aktualisierten Richtlinie.

[2143] Beispiel 540 umfasst den Gegenstand von einem der Beispiele 531 bis 539. In Beispiel 540 umfasst das Verfahren das Erzeugen einer Datei, die eine Differenz zwischen einer aktuellen Richtlinie und einer neuen Richtlinie umfasst, und das Senden der Datei zu einem Peer-Knoten.

[2144] Beispiel 541 umfasst den Gegenstand von einem der Beispiele 531 bis 540. In Beispiel 541 umfasst das Verfahren die Feststellung, ob ein Peer-Knoten eine Hardwarekapazität für die Richtlinien hat, das Modifizieren der Richtlinien, so dass sie zur Hardwarekapazität des Peer-Knotens passen, und das Senden der modifizierten Richtlinien zum Peer-Knoten.

[2145] Beispiel 542 umfasst den Gegenstand von einem der Beispiele 531 bis 541. In Beispiel 542 umfasst das Verfahren das Bestimmen von Änderungen zwischen neuen Richtlinien und aktuellen Richtlinien und das Senden der Änderungen in Richtlinien zum Peer-Knoten.

[2146] Beispiel 543 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den Prozessor anzuweisen, Richtlinien in anderen Knoten zu entdecken und Richtlinien von durch andere Knoten in einem IoT-Netz gesendeten Nachrichten zu aktualisieren.

[2147] Beispiel 544 umfasst den Gegenstand von Beispiel 543. In Beispiel 544 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Richtlinien von mehreren Knoten zu verketteten.

[2148] Beispiel 545 umfasst den Gegenstand von Beispiel 543 oder 544. In Beispiel 545 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, in Nachrichten von anderen Knoten empfangene Richtlinien zu validieren und Richtlinien abzulehnen, die in Konflikt mit Gruppenzielen gelangen.

[2149] Beispiel 546 umfasst den Gegenstand von einem der Beispiele 543 bis 545. In Beispiel 546 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, implementierte Richtlinien zu ändern, damit sie mit aktuellen Vorrichtungsbedingungen übereinstimmen.

[2150] Beispiel 547 umfasst den Gegenstand von einem der Beispiele 543 bis 546. In Beispiel 547 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Differenz zwischen Richtlinien zu berechnen.

[2151] Beispiel 548 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine Netzsteckervorrichtung umfasst. Die Netzsteckervorrichtung umfasst eine Stromversorgung, um Schaltungsanordnungen innerhalb der Netzsteckervorrichtung Strom bereitzustellen, und eine Stromversorgungs-Steuereinrichtung zur Zufuhr von Strom zu einer externen IoT-Vorrichtung, eine Schnittstelle zum Abfragen der externen IoT-Vorrichtung, um festzustellen, ob die externe IoT-Vorrichtung ausgefallen ist, wobei die Netzsteckervorrichtung den Strom für die externe IoT-Vorrichtung über die Stromversorgungs-Steuereinrichtung zyklisch rückführen soll, falls die externe IoT-Vorrichtung ausgefallen ist.

[2152] Beispiel 549 umfasst den Gegenstand von Beispiel 548. In Beispiel 549 umfasst die Schnittstelle eine Netzschnittstellen-Steuereinrichtung zur Verbindung mit der externen IoT-Vorrichtung, um Daten auszutauschen, Strom bereitzustellen oder beides auszuführen.

[2153] Beispiel 550 umfasst den Gegenstand von Beispiel 548 oder 549. In Beispiel 550 umfasst die Schnittstelle einen universellen seriellen Bus zur Verbindung mit der externen IoT-Vorrichtung, um Daten auszutauschen, Strom bereitzustellen oder beides auszuführen.

[2154] Beispiel 551 umfasst den Gegenstand von einem der Beispiele 548 bis 550. In Beispiel 551 umfasst die Schnittstelle eine JTAG-Schnittstelle zur Verbindung mit der externen IoT-Vorrichtung zur Überwachung des Betriebs, zur Ausführung einer Stromversorgungs-Rücksetzfunktion oder zum fernen Flashen eines nichtflüchtigen Speichers oder zur Ausführung von Kombinationen davon.

[2155] Beispiel 552 umfasst den Gegenstand von einem der Beispiele 548 bis 551. In Beispiel 552 umfasst die Schnittstelle einen Serieller-Peripherieschnittstellen- bzw. Serial-Peripheral-Interface(SPI)-Bus, einen I2C-Bus oder einen I3C-Bus oder eine Kombination davon.

[2156] Beispiel 553 umfasst den Gegenstand von einem der Beispiele 548 bis 552. In Beispiel 553 umfasst die Netzsteckervorrichtung eine HID-Transportschnittstelle zur Verbindung mit einem Sensor.

[2157] Beispiel 554 umfasst den Gegenstand von einem der Beispiele 548 bis 553. In Beispiel 554 umfasst der Sensor einen Umgebungstemperatursensor, einen Feuchtigkeitssensor, einen Näherungssensor oder einen Vorhandenseinssensor oder Kombinationen davon.

[2158] Beispiel 555 umfasst den Gegenstand von einem der Beispiele 548 bis 554. In Beispiel 555 umfasst die Vorrichtung ein vertrauenswürdiges Plattformmodul zur Einrichtung einer vertrauenswürdigen Ausführungsumgebung bzw. Trusted Execute Environment (TEE) für die Netzsteckervorrichtung.

[2159] Beispiel 556 umfasst den Gegenstand von einem der Beispiele 548 bis 555. In Beispiel 556 umfasst die Vorrichtung eine Überwachungseinrichtung zur Überwachung eines Arbeitsvorgangs der externen IoT-Vorrichtung.

[2160] Beispiel 557 umfasst den Gegenstand von einem der Beispiele 548 bis 556. In Beispiel 557 umfasst die Vorrichtung ein Betätigungselement zur zyklischen Rückführung des Stroms der externen IoT-Vorrichtung unter Verwendung der Stromversorgungssteuerung.

[2161] Beispiel 558 umfasst den Gegenstand von einem der Beispiele 548 bis 557. In Beispiel 558 umfasst die Vorrichtung ein Kommunikationssystem, das einen Mesh-Sendeempfänger, einen Uplink-Sendeempfänger oder eine Netzschnittstellen-Steuereinrichtung oder Kombinationen davon umfasst.

[2162] Beispiel 559 umfasst den Gegenstand von einem der Beispiele 558 bis 558. In Beispiel 559 umfasst die Vorrichtung einen Berichtersteller zur Mitteilung einer Bedingung der externen IoT-Vorrichtung durch das Kommunikationssystem.

[2163] Beispiel 560 umfasst den Gegenstand von einem der Beispiele 548 bis 559. In Beispiel 560 umfasst die Vorrichtung mehrere IoT-Vorrichtungen in Kommunikation mit der Netzsteckervorrichtung, wobei die Netzsteckervorrichtung einen Status von jeder der mehreren in Kommunikation mit der Netzsteckervorrichtung stehenden IoT-Vorrichtungen mitteilt.

[2164] Beispiel 561 umfasst ein Verfahren zum Verbessern der Verfügbarkeit einer Internet-der-Dinge(IoT)-Vorrichtung unter Verwendung einer Netzsteckervorrichtung. Das Verfahren zum Verbessern der Verfügbarkeit einer Internet-der-Dinge(IoT)-Vorrichtung unter Verwendung einer Netzsteckervorrichtung umfasst das Entdecken einer Liste behandelter IoT-Vorrichtungen, das Initialisieren der Netzsteckervorrichtung, das Überwachen einer IoT-Vorrichtung, das Mitteilen des Status der IoT-Vorrichtung und das Ausführen einer Aktion zum Wiederherstellen von Operationen, falls die IoT-Vorrichtung ausgefallen ist.

[2165] Beispiel 562 umfasst den Gegenstand von Beispiel 561. In Beispiel 562 umfasst das Entdecken der Liste behandelter IoT-Vorrichtungen das Spezifizieren physikalischer Wege, die mit der Netzsteckervorrichtung verbunden sind, und das Abfragen von jedem der physikalischen Wege, um festzustellen, was physikalisch verbunden ist.

[2166] Beispiel 563 umfasst den Gegenstand von Beispiel 561 oder 562. In Beispiel 563 umfasst das Entdecken der Liste behandelter IoT-Vorrichtungen das zyklische Durchlaufen von Funkeinrichtungen und Kommunikationsstrecken, um nahe gelegene zu behandelnde Vorrichtungen zu entdecken.

[2167] Beispiel 564 umfasst den Gegenstand von einem der Beispiele 561 bis 563. In Beispiel 564 umfasst das Initialisieren der Netzsteckervorrichtung das Laden von Richtlinien von einem fernen Host, einer Orchestrierungsvorrichtung oder von beiden.

[2168] Beispiel 565 umfasst den Gegenstand von einem der Beispiele 561 bis 564. In Beispiel 565 umfassen die Richtlinien einen Satz von Regeln, die Aktionen mit der Liste behandelter IoT-Vorrichtungen assoziieren.

[2169] Beispiel 566 umfasst den Gegenstand von einem der Beispiele 561 bis 565. In Beispiel 566 beruhen die Aktionen zumindest teilweise auf dem Typ der Verbindung mit der IoT-Vorrichtung.

[2170] Beispiel 567 umfasst den Gegenstand von einem der Beispiele 561 bis 566. In Beispiel 567 umfasst das Überwachen einer IoT-Vorrichtung das Überwachen von Kommunikationen von der IoT-Vorrichtung, das Empfangen von Nachrichten von einem Watchdog-Agenten in der IoT-Vorrichtung, das Überwachen auf einen Nichtempfang einer Nachricht vom Watchdog-Agenten in der IoT-Vorrichtung oder Kombinationen davon.

[2171] Beispiel 568 umfasst den Gegenstand von einem der Beispiele 561 bis 567. In Beispiel 568 umfasst das Berichten über den Status der IoT-Vorrichtung das Senden einer Warnung.

[2172] Beispiel 569 umfasst den Gegenstand von einem der Beispiele 561 bis 568. In Beispiel 569 umfasst die Warnung eine E-Mail, ein in ein Protokoll geschriebenes Ereignis oder eine Textnachricht oder Kombinationen davon.

[2173] Beispiel 570 umfasst den Gegenstand von einem der Beispiele 561 bis 569. In Beispiel 570 umfasst das Ausführen der Aktion zur Wiederherstellung von Operationen das ferne Neustarten in der Vorrichtung.

[2174] Beispiel 571 umfasst den Gegenstand von einem der Beispiele 561 bis 570. In Beispiel 571 umfasst das Ausführen der Aktion zur Wiederherstellung von Operationen das Flashen von Firmware in der IoT-Vorrichtung.

[2175] Beispiel 572 umfasst den Gegenstand von einem der Beispiele 561 bis 571. In Beispiel 572 umfasst das Ausführen der Aktion zur Wiederherstellung von Operationen das zyklische Unterbrechen des Stroms zu einer IoT-Vorrichtung, die Strom von der Netzsteckervorrichtung erhält.

[2176] Beispiel 573 umfasst den Gegenstand von einem der Beispiele 561 bis 572. In Beispiel 573 umfasst das Ausführen der Aktion zur Wiederherstellung von Operationen das Schalten von einer Stromversorgung zu einer Batterie.

[2177] Beispiel 574 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, zu behandelnde Internet-der-Dinge(IoT)-Vorrichtungen zu entdecken, eine Liste behandelter IoT-Vorrichtungen zu bilden, eine Netzsteckervorrichtung zu initialisieren und die behandelten IoT-Vorrichtungen zu überwachen.

[2178] Beispiel 575 umfasst den Gegenstand von Beispiel 574. In Beispiel 575 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Netzstecker und angeschlossene IoT-Vorrichtungen bekanntzumachen.

[2179] Beispiel 576 umfasst den Gegenstand von Beispiel 574 oder 575. In Beispiel 576 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Status der behandelten IoT-Vorrichtungen mitzuteilen.

[2180] Beispiel 577 umfasst den Gegenstand von einem der Beispiele 574 bis 576. In Beispiel 577 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, behandelte IoT-Vorrichtungen zu betätigen, um Operationen wiederherzustellen.

[2181] Beispiel 578 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst eine Host-Umgebung, die einen Watchdog-Agenten zum Senden von Watchdog-Nachrichten, welche über die Funktionstüchtigkeit und den Betrieb der Host-Umgebung berichten, umfasst, und eine vertrauenswürdige Zuverlässigkeitsmaschine bzw. Trusted Reliability Engine (TRE), die eine von der Stromversorgung für die Host-Umgebung getrennte Stromversorgung umfasst, eine verteilte TRE-Ledger-Logik zum Schreiben der Watchdog-Nachrichten in eine TRE-Blockchain und eine TRE-Logik zum Anwenden einer Ausfallsicherungsaktion, falls die Host-Umgebung ausfällt.

[2182] Beispiel 579 umfasst den Gegenstand von Beispiel 578. In Beispiel 579 umfasst die Host-Umgebung einen Abbilderzeuger zur Erzeugung eines Abbilds der Host-Umgebung und zum Senden der Abbildkopie zur TRE, um sie als Host-Ersatzabbild bzw. Host Replacement Image (HRI) zu speichern.

[2183] Beispiel 580 umfasst den Gegenstand von Beispiel 578 oder 579. In Beispiel 580 umfasst die Host-Umgebung eine Host-Blockchain-Logik zum Unterhalten einer Host-Blockchain.

[2184] Beispiel 581 umfasst den Gegenstand von einem der Beispiele 578 bis 580. In Beispiel 581 umfasst die Host-Blockchain Watchdog-Nachrichtenblöcke, Peer-Vorrichtungsblöcke oder Identitätsblöcke oder Kombinationen davon.

[2185] Beispiel 582 umfasst den Gegenstand von einem der Beispiele 578 bis 581. In Beispiel 582 umfasst die Host-Umgebung einen Kommunikator zur Kommunikation mit anderen Mesh-Vorrichtungen, Vorrichtungen in einer Cloud oder mit beiden.

[2186] Beispiel 583 umfasst den Gegenstand von einem der Beispiele 578 bis 582. In Beispiel 583 umfasst die TRE ein Kommunikationssystem, um es der TRE zu ermöglichen, mit externen Vorrichtungen zu kommunizieren, falls die Host-Umgebung ausfällt.

[2187] Beispiel 584 umfasst den Gegenstand von einem der Beispiele 578 bis 583. In Beispiel 584 umfasst die TRE ein Host-Ersatzabbild (HRI).

[2188] Beispiel 585 umfasst den Gegenstand von einem der Beispiele 578 bis 584. In Beispiel 585 umfasst das HRI eine Kopie eines Betriebssystems, Treiber und Funktionscode für die IoT-Vorrichtung.

[2189] Beispiel 586 umfasst ein Verfahren zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine (TRE). Das Verfahren zur Implementation eines Ausfallsicherungsmechanismus unter Verwendung einer vertrauenswürdigen Zuverlässigkeitsmaschine (TRE) umfasst das Überwachen einer Host-Umgebung auf einen Ausfall, das Übermitteln einer Watchdog-

Nachricht an eine Blockchain, das Erkennen eines Ausfalls der Host-Umgebung und das Implementieren eines Ausfallprozesses zur Wiederherstellung beim Ausfall der Host-Umgebung.

[2190] Beispiel 587 umfasst den Gegenstand von Beispiel 586. In Beispiel 587 umfasst das Überwachen der Host-Umgebung das Empfangen von Pings von der Host-Umgebung.

[2191] Beispiel 588 umfasst den Gegenstand von Beispiel 586 oder 587. In Beispiel 588 umfasst das Übermitteln der Watchdog-Nachricht das Aufnehmen eines Pings in die Watchdog-Nachricht und das Übergeben der Watchdog-Nachricht an die Blockchain als Transaktion.

[2192] Beispiel 589 umfasst den Gegenstand von einem der Beispiele 586 bis 588. In Beispiel 589 umfasst das Erkennen des Ausfalls der Host-Umgebung die Feststellung, dass während eines ausgewählten Zeitraums keine Pings von der Host-Umgebung empfangen wurden.

[2193] Beispiel 590 umfasst den Gegenstand von einem der Beispiele 586 bis 589. In Beispiel 590 umfasst das Erkennen des Ausfalls der Host-Umgebung die Feststellung, dass über einen Bus der Host-Umgebung keine Kommunikationen stattfinden.

[2194] Beispiel 591 umfasst den Gegenstand von einem der Beispiele 586 bis 590. In Beispiel 591 umfasst das Erkennen des Ausfalls der Host-Umgebung die Feststellung, dass eine CPU angehalten hat.

[2195] Beispiel 592 umfasst den Gegenstand von einem der Beispiele 586 bis 591. In Beispiel 592 umfasst das Erkennen des Ausfalls der Host-Umgebung die Feststellung, dass ein Speicher in der Host-Umgebung ausgefallen ist.

[2196] Beispiel 593 umfasst den Gegenstand von einem der Beispiele 586 bis 592. In Beispiel 593 umfasst der Ausfallprozess die Feststellung, ob die Host-Umgebung lokal wiederherstellbar ist, und, falls dies der Fall ist, das Installieren eines Host-Ersatzabbilds in der Host-Umgebung und das Neustarten der Host-Umgebung.

[2197] Beispiel 594 umfasst den Gegenstand von einem der Beispiele 586 bis 593. In Beispiel 594 umfasst der Ausfallprozess die Feststellung, ob sich eine Ausfallsicherungs Vorrichtung in der Nähe befindet, und, falls dies der Fall ist, das Konfigurieren der Ausfallsicherungs Vorrichtung, um mit der Ausführung einer Funktion der Host-Umgebung zu beginnen.

[2198] Beispiel 595 umfasst den Gegenstand von einem der Beispiele 586 bis 594. In Beispiel 595 umfasst der Ausfallprozess die Feststellung, ob eine Vorrichtung unter Einschluss der Host-Umgebung reparierbar ist, und, falls dies der Fall ist, das Aussenden einer Reparaturdrohne, um die Vorrichtung zu reparieren.

[2199] Beispiel 596 umfasst den Gegenstand von einem der Beispiele 586 bis 595. In Beispiel 596 umfasst der Ausfallprozess die Feststellung, ob eine Vorrichtung unter Einschluss der Host-Umgebung ersetzbar ist, und, falls dies der Fall ist, das Aussenden einer Reparaturdrohne, um die Vorrichtung zu ersetzen.

[2200] Beispiel 597 umfasst den Gegenstand von einem der Beispiele 586 bis 596. In Beispiel 597 umfasst der Ausfallprozess die Feststellung, ob der Ausfall gelöst wurde, und, falls dies der Fall ist, das Dekommissionieren der Host-Umgebung, das Versetzen der TRE in einen Schlafzustand oder beides.

[2201] Beispiel 598 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Host-Umgebung auf eine Herzschnachricht zu überwachen, eine Watchdog(WD)-Nachricht zu erzeugen, die WD-Nachricht zu einer Blockchain zu übermitteln und einen Ausfall in einer Host-Umgebung zu erkennen.

[2202] Beispiel 599 umfasst den Gegenstand von Beispiel 598. In Beispiel 599 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Ausfall in einer lokalen Host-Umgebung zu erkennen und ein Host-Ersatzabbild zu installieren.

[2203] Beispiel 600 umfasst den Gegenstand von Beispiel 598 oder 599. In Beispiel 600 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Ausfall in einer fernen Host-Umgebung zu erkennen und eine Ausfallsicherungs Vorrichtung zu konfigurieren, so dass sie als ferne Host-Umgebung wirkt.

[2204] Beispiel 601 umfasst den Gegenstand von einem der Beispiele 598 bis 600. In Beispiel 601 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Ausfall in einer fernen Host-Umgebung zu erkennen und eine Drohne zur Reparatur oder für den Ersatz einer Vorrichtung, einschließlich der fernen Host-Umgebung, abzusenden.

[2205] Beispiel 602 umfasst den Gegenstand von einem der Beispiele 598 bis 601. In Beispiel 602 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass der Ausfall gelöst wurde, und eine ausgefallene Vorrichtung zu dekommissionieren.

[2206] Beispiel 603 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Kommunikator zum Empfang eines Pakets von einer anderen IoT-Vorrichtung, das einen fraktionellen Schlüssel und einen Offset in einem Nutzinhaltfeld des Frames umfasst, einen Bytevergleicher zum Vergleichen überlappender Bytes des fraktionellen Schlüssels und eines anderen in einem zirkulären Puffer gespeicherten fraktionellen Schlüssels und einen Schlüsselzusammensteller zum Einbringen des fraktionellen Schlüssels in den zirkulären Puffer beim Offset.

[2207] Beispiel 604 umfasst den Gegenstand von Beispiel 603. In Beispiel 604 soll der Bytevergleicher, falls überlappende Bytes des fraktionellen Schlüssels und des anderen im zirkulären Puffer gespeicherten fraktionellen Schlüssels nicht übereinstimmen, verhindern, dass der Schlüsselzusammensteller den fraktionellen Schlüssel in den zirkulären Puffer einbringt.

[2208] Beispiel 605 umfasst den Gegenstand von Beispiel 603 oder 604. In Beispiel 605 soll der Schlüsselzusammensteller einen endgültigen Schlüssel im zirkulären Puffer anhand mehrerer von anderen IoT-Vorrichtungen empfangener fraktioneller Schlüssel bilden.

[2209] Beispiel 606 umfasst den Gegenstand von einem der Beispiele 603 bis 605. In Beispiel 606 umfasst die Vorrichtung einen Schlüsseloperator zur Verwendung eines im zirkulären Puffer zusammengestellten Schlüssels.

[2210] Beispiel 607 umfasst den Gegenstand von einem der Beispiele 603 bis 606. In Beispiel 607 umfasst die Vorrichtung einen Schlüsseloperator zur Verwendung eines im zirkulären Puffer zusammengestellten Schlüssels bei einem Prozess.

[2211] Beispiel 608 umfasst den Gegenstand von einem der Beispiele 603 bis 607. In Beispiel 608 umfasst der Prozess das Bereitstellen des Schlüssels für einen Gateway zur Bestätigung der Identität der IoT-Vorrichtung.

[2212] Beispiel 609 umfasst den Gegenstand von einem der Beispiele 603 bis 608. In Beispiel 609 umfasst der Prozess das Bereitstellen des Schlüssels für einen Gateway zur Bestätigung der Identität einer Fog-Vorrichtung.

[2213] Beispiel 610 umfasst den Gegenstand von einem der Beispiele 603 bis 609. In Beispiel 610 umfasst die Vorrichtung einen Fraktioneller-Schlüssel-Generator zur Erzeugung eines fraktionellen Schlüssels für die IoT-Vorrichtung und den Kommunikator zur Bildung eines Frames, der den fraktionellen Schlüssel in seinem Nutzinhalt aufweist, und zum Senden des Frames zu Peer-Vorrichtungen.

[2214] Beispiel 611 umfasst ein Verfahren zum Zusammenstellen eines vollständigen Schlüssels anhand in einzelnen Knoten in einem IoT-Netz gespeicherter fraktioneller Schlüssel. Das Verfahren zum Zusammenstellen eines vollständigen Schlüssels anhand in einzelnen Knoten in einem IoT-Netz gespeicherter fraktioneller Schlüssel umfasst das Empfangen eines fraktionellen Schlüssels von einer Vorrichtung im IoT-Netz, das Vergleichen überlappender Bytes des fraktionellen Schlüssels mit anderen von anderen Vorrichtungen empfangenen fraktionellen Schlüsseln und das Bilden des vollständigen Schlüssels in einem zirkulären Puffer.

[2215] Beispiel 612 umfasst den Gegenstand von einem der Beispiele 611 bis 612. In Beispiel 612 umfasst das Verfahren das Empfangen eines Frames von der Vorrichtung im IoT-Netz und das Feststellen, ob der Frame in einem Nutzinhaltfeld einen fraktionellen Schlüssel aufweist.

[2216] Beispiel 613 umfasst den Gegenstand von Beispiel 611. In Beispiel 613 umfasst das Verfahren das Speichern des fraktionellen Schlüssels im zirkulären Puffer und, falls der fraktionelle Schlüssel ein Ende des

zirkulären Puffers überlappt, das Schreiben restlicher Bytes an einem entgegengesetzten Ende des zirkulären Puffers.

[2217] Beispiel 614 umfasst den Gegenstand von Beispiel 611 oder 613. In Beispiel 614 umfasst das Verfahren das Empfangen eines zweiten fraktionellen Schlüssels von einer anderen Vorrichtung, die Feststellung, ob Bytes im zweiten fraktionellen Schlüssel bereits in den zirkulären Puffer geschriebene Bytes überlappen, und das Vergleichen der Bytes im zweiten fraktionellen Schlüssel, welche bereits in den zirkulären Puffer geschriebene Bytes überlappen.

[2218] Beispiel 615 umfasst den Gegenstand von einem der Beispiele 611 bis 614. In Beispiel 615 umfasst das Verfahren die Feststellung, dass Bytes im zweiten fraktionellen Schlüssel nicht mit bereits in den zirkulären Puffer geschriebenen überlappenden Bytes übereinstimmen, das Löschen des zweiten fraktionellen Schlüssels und das Senden einer Warnung, dass eine Vorrichtung beeinträchtigt sein kann.

[2219] Beispiel 616 umfasst den Gegenstand von einem der Beispiele 611 bis 615. In Beispiel 616 umfasst das Verfahren die Feststellung, dass alle Bytes im zweiten fraktionellen Schlüssel mit bereits in den zirkulären Puffer geschriebenen überlappenden Bytes übereinstimmen, und das Schreiben des zweiten fraktionellen Schlüssels in den zirkulären Puffer.

[2220] Beispiel 617 umfasst den Gegenstand von einem der Beispiele 611 bis 616. In Beispiel 617 umfasst das Verfahren die Feststellung, dass alle Schlüssel empfangen wurden und dass der zirkuläre Puffer den vollständigen Schlüssel umfasst, und das Bereitstellen des vollständigen Schlüssels für eine andere Vorrichtung.

[2221] Beispiel 618 umfasst den Gegenstand von einem der Beispiele 611 bis 617. In Beispiel 618 haben die fraktionellen Schlüssel unterschiedliche Längen.

[2222] Beispiel 619 umfasst den Gegenstand von einem der Beispiele 611 bis 618. In Beispiel 619 wird der vollständige Schlüssel zusammengestellt, bevor alle fraktionellen Schlüssel empfangen wurden.

[2223] Beispiel 620 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen fraktionellen Schlüssel von einer Vorrichtung zu empfangen, Bytes im fraktionellen Schlüssel mit bereits in einen zirkulären Puffer geschriebenen Bytes zu vergleichen und den fraktionellen Schlüssel in den zirkulären Puffer zu schreiben.

[2224] Beispiel 621 umfasst den Gegenstand von einem der Beispiele 620 bis 621. In Beispiel 621 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Frame von einer Vorrichtung zu empfangen, und festzustellen, dass der Frame einen fraktionellen Schlüssel in einem Nutzinhaltsfeld umfasst.

[2225] Beispiel 622 umfasst den Gegenstand von Beispiel 620. In Beispiel 622 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen fraktionellen Schlüssel zu einer anderen Vorrichtung zu übertragen.

[2226] Beispiel 623 umfasst den Gegenstand von Beispiel 620 oder 622. In Beispiel 623 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen im zirkulären Puffer zusammengestellten vollständigen Schlüssel zu verwenden.

[2227] Beispiel 624 umfasst den Gegenstand von einem der Beispiele 620 bis 623. In Beispiel 624 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass überlappende Bytes zwischen einem fraktionellen Schlüssel und in den zirkulären Puffer geschriebenen Bytes nicht übereinstimmen, und die Verwendung des fraktionellen Schlüssels zu verhindern.

[2228] Beispiel 625 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Kommunikator zum Austauschen von Paketen mit anderen Vorrichtungen durch einen Sendeempfänger, eine Netzschnittstellen-Steuereinrichtung oder beide, einen Schlüsselgenerator zur Erzeugung eines Schlüssels durch Ausführen einer logischen Operation an einem vollständigen fraktionellen Schlüssel und einem lokalen Schlüssel und einen Transaktor zur Verwendung des Schlüssels, um eine Transaktion durch den Kommunikator zu einem Netz zu übertragen.

[2229] Beispiel 626 umfasst den Gegenstand von Beispiel 625. In Beispiel 626 umfasst die IoT-Vorrichtung einen Schlüssellebensdauer-Zeitgeber zur Steuerung eines Zeitraums, während dessen der Schlüssel verwendet werden kann, bevor ein neuer Schlüssel erzeugt wird.

[2230] Beispiel 627 umfasst den Gegenstand von Beispiel 625 oder 626. In Beispiel 627 umfasst die IoT-Vorrichtung einen zirkulären Puffer zum Speichern eines Ergebnisses der logischen Operation zwischen dem vollständigen fraktionellen Schlüssel und dem lokalen Schlüssel.

[2231] Beispiel 628 umfasst den Gegenstand von einem der Beispiele 625 bis 627. In Beispiel 628 umfasst die IoT-Vorrichtung den vollständigen fraktionellen Schlüssel, wobei der vollständige fraktionelle Schlüssel anhand von anderen IoT-Vorrichtungen übertragener fraktioneller Schlüssel zusammengestellt wurde.

[2232] Beispiel 629 umfasst den Gegenstand von einem der Beispiele 625 bis 628. In Beispiel 629 umfasst die logische Operation eine zwischen dem vollständigen fraktionellen Schlüssel und dem lokalen Schlüssel ausgeführte Exklusiv-ODER-Operation.

[2233] Beispiel 630 umfasst den Gegenstand von einem der Beispiele 625 bis 629. In Beispiel 630 umfasst die Transaktion ein Kauf, eine Miete, eine Zahlung für einen Zugriff oder Kombinationen davon.

[2234] Beispiel 631 umfasst den Gegenstand von einem der Beispiele 625 bis 630. In Beispiel 631 soll der Schlüsselgenerator einen lokalen Schlüssel von einer Blockchain erhalten.

[2235] Beispiel 632 umfasst ein Verfahren zur Erzeugung eines Schlüssels nach Bedarf in einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur Erzeugung eines Schlüssels nach Bedarf in einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Erzeugen eines neuen Schlüssels durch Ausführen einer logischen Operation zwischen einem vollständigen fraktionellen Schlüssel und einem lokalen Schlüssel, das Verifizieren des neuen Schlüssels und das Verwenden des neuen Schlüssels zur Übermittlung einer Transaktion.

[2236] Beispiel 633 umfasst den Gegenstand von Beispiel 632. In Beispiel 633 umfasst das Verfahren das Empfangen mehrerer fraktioneller Schlüssel von anderen IoT-Vorrichtungen und das Zusammenstellen der mehreren fraktionellen Schlüssel zur Bildung des vollständigen fraktionellen Schlüssels.

[2237] Beispiel 634 umfasst den Gegenstand von Beispiel 632 oder 633. In Beispiel 634 umfasst das Verfahren die Feststellung, ob ein Schlüssel-Offset-Wert empfangen wurde, und, falls dies nicht der Fall ist, das Erzeugen des Schlüssel-Offset-Werts.

[2238] Beispiel 635 umfasst den Gegenstand von einem der Beispiele 632 bis 634. In Beispiel 635 umfasst das Verfahren das Verwenden des Schlüssel-Offset-Werts zur Bestimmung eines Anfangspunkts in einem zirkulären Puffer für die logische Operation zwischen dem vollständigen fraktionellen Schlüssel und dem lokalen Schlüssel.

[2239] Beispiel 636 umfasst den Gegenstand von einem der Beispiele 632 bis 635. In Beispiel 636 umfasst die logische Operation eine zwischen jedem Bit des vollständigen fraktionellen Schlüssels und jedem Bit des lokalen Schlüssels ausgeführte Exklusiv-ODER-Operation.

[2240] Beispiel 637 umfasst den Gegenstand von einem der Beispiele 632 bis 636. In Beispiel 637 umfasst das Verfahren die Feststellung, ob der neue Schlüssel abgelaufen ist, und, falls dies der Fall ist, das Erzeugen eines anderen neuen Schlüssels.

[2241] Beispiel 638 umfasst den Gegenstand von einem der Beispiele 632 bis 637. In Beispiel 638 umfasst das Verfahren das Erzeugen eines anderen neuen Schlüssels, nachdem der neue Schlüssel in einer Transaktion verwendet wurde.

[2242] Beispiel 639 umfasst den Gegenstand von einem der Beispiele 632 bis 638. In Beispiel 639 umfasst das Verfahren das Löschen des neuen Schlüssels, nachdem er für die Übergabe einer Transaktion verwendet wurde.

[2243] Beispiel 640 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen fraktionel-

len Schlüssel zu empfangen, einen Offset-Wert zu empfangen und einen neuen Schlüssel zumindest teilweise auf der Grundlage des fraktionellen Schlüssels und des Offset-Werts zu erzeugen.

[2244] Beispiel 641 umfasst den Gegenstand von Beispiel 640. In Beispiel 641 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, mehrere fraktionelle Schlüssel von Inter-net-der-Dinge(IoT)-Vorrichtungen in einem Netz zu empfangen und die mehreren fraktionellen Schlüssel zusammenzustellen, um einen vollständigen fraktionellen Schlüssel zu bilden.

[2245] Beispiel 642 umfasst den Gegenstand von Beispiel 640 oder 641. In Beispiel 642 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Exklusiv-ODER-Operation zwischen jedem Bit des vollständigen fraktionellen Schlüssels und jedem Bit eines lokalen Schlüssels, beginnend beim Offset-Wert, auszuführen, um den neuen Schlüssel zu erzeugen.

[2246] Beispiel 643 umfasst den Gegenstand von einem der Beispiele 640 bis 642. In Beispiel 643 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Daten unter Verwendung des neuen Schlüssels zu verschlüsseln.

[2247] Beispiel 644 umfasst den Gegenstand von einem der Beispiele 640 bis 643. In Beispiel 644 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den neuen Schlüssel nach einer ausgewählten Zeit ablaufen zu lassen und einen anderen neuen Schlüssel zu erzeugen.

[2248] Beispiel 645 umfasst den Gegenstand von einem der Beispiele 640 bis 644. In Beispiel 645 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine durch den neuen Schlüssel validierte Transaktion auszuführen, den neuen Schlüssel ablaufen zu lassen, nachdem die Transaktion abgeschlossen wurde, und einen anderen neuen Schlüssel zu erzeugen.

[2249] Beispiel 646 umfasst den Gegenstand von einem der Beispiele 640 bis 645. In Beispiel 646 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Schlüssel von einer Blockchain zu erhalten.

[2250] Beispiel 647 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Inter-net-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Zusammenhangsidentifizierer zum Bestimmen eines Zusammenhangs für die Erzeugung von Seeds, einen Seed-Baum-Generator zur Erzeugung eines Seed-Baums anhand des Zusammenhangs und einen Seed-Generator zur Erzeugung eines Seeds für jeden Knoten im Seed-Baum.

[2251] Beispiel 648 umfasst den Gegenstand von Beispiel 647. In Beispiel 648 kann der Zusammenhang zumindest teilweise auf einer Zeit, einem Ort oder einer IP-Adresse oder Kombinationen davon beruhen.

[2252] Beispiel 649 umfasst den Gegenstand von Beispiel 647 oder 648. In Beispiel 649 zerlegt der Seed-Baum-Generator den Zusammenhang zur Bildung des Seed-Baums.

[2253] Beispiel 650 umfasst den Gegenstand von einem der Beispiele 647 bis 649. In Beispiel 650 extrahiert der Seed-Baum-Generator einen Teil des Zusammenhangs an jedem Knoten, wobei der Teil einen Zahlenwert umfasst.

[2254] Beispiel 651 umfasst den Gegenstand von einem der Beispiele 647 bis 650. In Beispiel 651 verwendet der Seed-Generator den Zahlenwert als Seed für die Erzeugung einer Zufallszahl.

[2255] Beispiel 652 umfasst den Gegenstand von einem der Beispiele 647 bis 651, einschließlich eines Kommunikators für den Austausch von Paketen mit anderen IoT-Vorrichtungen zu 652. In Beispiel 652 umfassen die Pakete den Zusammenhang, eine hierarchische Ebene oder einen Root-Seed oder Kombinationen davon.

[2256] Beispiel 653 umfasst den Gegenstand von einem der Beispiele 647 bis 652. In Beispiel 653 erzeugt der Seed-Generator einen Root-Seed für einen oberen Knoten in einem Seed-Baum.

[2257] Beispiel 654 umfasst den Gegenstand von einem der Beispiele 647 bis 653. In Beispiel 654 umfasst die IoT-Vorrichtung einen Fraktioneller-SchlüsselZusammensteller zum Zusammenstellen eines vollständigen

fraktionellen Schlüssels anhand von anderen IoT-Vorrichtungen über einen Kommunikator empfangener Teile fraktioneller Schlüssel.

[2258] Beispiel 655 umfasst den Gegenstand von einem der Beispiele 647 bis 654. In Beispiel 655 umfasst die IoT-Vorrichtung eine Verschlüsselungs-/Entschlüsselungseinrichtung zur Verschlüsselung oder Entschlüsselung von Kommunikationen unter Verwendung eines anhand des Seeds erzeugten Schlüssels.

[2259] Beispiel 656 umfasst ein Verfahren zur Erzeugung eines geteilten Geheimnisses für sichere Kommunikationen zwischen IoT-Vorrichtungen. Das Verfahren zur Erzeugung eines geteilten Geheimnisses für sichere Kommunikationen zwischen IoT-Vorrichtungen umfasst das Identifizieren eines gemeinsamen Attributs zwischen den IoT-Vorrichtungen, das Zerlegen des Attributs zur Bildung eines Seed-Baums, das Erzeugen eines Seeds für eine Root des Seed-Baums und das Bereitstellen des Seeds und des Attributs für jede teilnehmende Vorrichtung.

[2260] Beispiel 657 umfasst den Gegenstand von Beispiel 656. In Beispiel 657 umfasst das Zerlegen des Attributs das Zuweisen einer Teilmenge des Attributs zu jedem Knoten in einer Hierarchie.

[2261] Beispiel 658 umfasst den Gegenstand von Beispiel 656 oder 657. In Beispiel 658 umfasst das Verfahren das Verwenden des Seeds für die Root des Seed-Baums zur Erzeugung eines neuen Seeds für einen Knoten im Seed-Baum und das Verwenden des neuen Seeds zur Erzeugung eines kryptographischen Schlüssels.

[2262] Beispiel 659 umfasst den Gegenstand von einem der Beispiele 656 bis 658. In Beispiel 659 umfasst das Verfahren das Verwenden eines kryptographischen Geheimnisses zur Unterteilung des Seeds in Anteile und das Bereitstellen der Anteile über Vorrichtungen.

[2263] Beispiel 660 umfasst den Gegenstand von einem der Beispiele 656 bis 659. In Beispiel 660 umfasst das Verfahren die Verwendung eines Netzes, um die jeweiligen Anteile zu erhalten, und das Neuzusammenstellen des Seeds anhand der Anteile.

[2264] Beispiel 661 umfasst den Gegenstand von einem der Beispiele 656 bis 660. In Beispiel 661 umfasst das Verfahren das Empfangen mehrerer fraktioneller Schlüssel von anderen IoT-Vorrichtungen in einem Netz und das Zusammenstellen eines vollständigen fraktionellen Schlüssels anhand der mehreren fraktionellen Schlüssel.

[2265] Beispiel 662 umfasst den Gegenstand von einem der Beispiele 656 bis 661. In Beispiel 662 umfasst das Verfahren das Verschlüsseln von Daten unter Verwendung eines für einen Knoten im Seed-Baum erzeugten kryptographischen Schlüssels, das Senden der Daten zu einer anderen IoT-Vorrichtung und das Entschlüsseln der Daten unter Verwendung eines für den Knoten im Seed-Baum, der in der anderen IoT-Vorrichtung gespeichert ist, erzeugten kryptographischen Schlüssels.

[2266] Beispiel 663 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Seed-Baum für einen Zusammenhang zu erzeugen, einen Root-Seed für den Zusammenhang zu erzeugen, den Zusammenhang anderen Vorrichtungen bereitzustellen und den Root-Seed anderen Vorrichtungen bereitzustellen.

[2267] Beispiel 664 umfasst den Gegenstand von Beispiel 663. In Beispiel 664 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Seed für jeden Knoten im Seed-Baum zu erzeugen, den Seed für die Erzeugung eines kryptographischen Schlüssels zu verwenden und den kryptographischen Schlüssel zum Verschlüsseln zu anderen Vorrichtungen gesendeter Daten zu verwenden.

[2268] Beispiel 665 umfasst den Gegenstand von Beispiel 663 oder 664. In Beispiel 665 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den Zusammenhang von einer anderen Vorrichtung zu empfangen, den Root-Seed von einer anderen Vorrichtung zu empfangen, einen lokalen Seed-Baum für den Zusammenhang zu erzeugen und den Root-Seed zur Erzeugung eines lokalen kryptographischen Schlüssels für jeden Knoten im lokalen Seed-Baum zu verwenden.

[2269] Beispiel 666 umfasst das nichtflüchtige maschinenlesbare Medium aus den Beispielen 663 bis 665. In Beispiel 666 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen fraktionellen Schlüssel zu empfangen, einen Offset-Wert zu empfangen und

einen neuen Schlüssel zumindest teilweise auf der Grundlage des fraktionellen Schlüssels und des Offset-Werts zu erzeugen.

[2270] Beispiel 667 umfasst den Gegenstand von einem der Beispiele 663 bis 666. In Beispiel 667 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, mehrere fraktionelle Schlüssel von Internet-der-Dinge(IoT)-Vorrichtungen in einem Netz zu empfangen und die mehreren fraktionellen Schlüssel zusammenzustellen, um einen vollständigen fraktionellen Schlüssel zu bilden.

[2271] Beispiel 668 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz einen IoT-Server umfasst. Der IoT-Server umfasst eine sichere Boot-/Messeinrichtung zur Verwendung eines vertrauenswürdigen Plattformmoduls (TPM), um eine vertrauenswürdige Ausführungsumgebung (TEE) zu erzeugen, einen Vertrauensanker zur Bestätigung der Identität eines Diensteanbieters, einen Authentifizierer zum Authentifizieren von Kommunikationen mit einem IoT-Client unter Verwendung eines symmetrischen Schlüssels (SK), einen Schlüsselmanager zur Feststellung, ob ein Schlüssel abgelaufen ist, und einen Schlüsselgenerator zur Erzeugung des Schlüssels.

[2272] Beispiel 669 umfasst den Gegenstand von Beispiel 668. In Beispiel 669 umfasst der Vertrauensanker einen Hash eines öffentlichen Schlüssels oder einen zertifizierten Weg oder eine Kette zu einer vertrauenswürdigen Autoritäts-Root.

[2273] Beispiel 670 umfasst den Gegenstand von Beispiel 668 oder 679. In Beispiel 670 ist der SK ein durch den Schlüsselgenerator erzeugter temporärer symmetrischer Schlüssel bzw. Temporal Symmetric Key (TSK).

[2274] Beispiel 671 umfasst den Gegenstand von einem der Beispiele 668 bis 670. In Beispiel 671 umfasst der IoT-Server einen öffentlichen Schlüssel (PK) zum Entschlüsseln von Nachrichten von einem Diensteanbieter.

[2275] Beispiel 672 umfasst den Gegenstand von einem der Beispiele 668 bis 671. In Beispiel 672 weist der IoT-Server eine Ablaufzeit für den öffentlichen Schlüssel auf.

[2276] Beispiel 673 umfasst den Gegenstand von einem der Beispiele 668 bis 672. In Beispiel 673 umfasst der IoT-Server einen vom Diensteanbieter empfangenen SK.

[2277] Beispiel 674 umfasst den Gegenstand von einem der Beispiele 668 bis 673. In Beispiel 674 weist der IoT-Server eine Ablaufzeit für den SK auf.

[2278] Beispiel 675 umfasst den Gegenstand von einem der Beispiele 668 bis 674. In Beispiel 675 umfasst der IoT-Server einen Diensteanbieter-Berechtigungsnachweis zur Validierung des IoT-Servers beim Diensteanbieter.

[2279] Beispiel 676 umfasst den Gegenstand von einem der Beispiele 668 bis 675. In Beispiel 676 umfasst die Vorrichtung den einen SK für die Kommunikation umfassenden IoT-Client.

[2280] Beispiel 677 umfasst den Gegenstand von einem der Beispiele 668 bis 676. In Beispiel 677 umfasst die Vorrichtung den einen Status für einen öffentlichen Schlüssel aufweisenden IoT-Server.

[2281] Beispiel 678 umfasst den Gegenstand von einem der Beispiele 668 bis 677. In Beispiel 678 umfasst die Vorrichtung eine Entität zur Erkennung, dass ein öffentlicher Schlüssel beschädigt wurde, und zum Senden einer Widerrufsnachricht zum IoT-Server.

[2282] Beispiel 679 umfasst ein Verfahren zur einheitlichen Schlüsselbehandlung in einer IoT-Netzumgebung. Das Verfahren zur einheitlichen Schlüsselbehandlung in einer IoT-Netzumgebung umfasst das Senden einer Anforderung von einem IoT-Client zu einem Diensteanbieter für einen Kommunikationsschlüssel, das Empfangen des Kommunikationsschlüssels am IoT-Client vom Diensteanbieter, das Senden des Kommunikationsschlüssels zu einem IoT-Server vom IoT-Client und das Kommunizieren mit dem IoT-Server unter Verwendung eines symmetrischen Schlüssels zum Entschlüsseln vom IoT-Server empfangener Nachrichten.

[2283] Beispiel 680 umfasst den Gegenstand von Beispiel 679. In Beispiel 680 umfasst der Kommunikationsschlüssel den symmetrischen Schlüssel.

[2284] Beispiel 681 umfasst den Gegenstand von Beispiel 679 oder 680. In Beispiel 681 umfasst der Kommunikationsschlüssel ein vom IoT-Server bereitgestelltes Zertifikat.

[2285] Beispiel 682 umfasst den Gegenstand von einem der Beispiele 679 bis 681, einschließlich des Empfangs eines temporären symmetrischen Schlüssels am IoT-Client vom IoT-Server für 682. In Beispiel 682 umfasst der temporäre symmetrische Schlüssel den symmetrischen Schlüssel.

[2286] Beispiel 683 umfasst den Gegenstand von einem der Beispiele 679 bis 682. In Beispiel 683 umfasst das Verfahren das Anfordern von Berechtigungsnachweisen für den IoT-Server von einem Dienstanbieter für sichere Kommunikationen und das Empfangen eines Vertrauensankers am IoT-Server vom Dienstanbieter.

[2287] Beispiel 684 umfasst den Gegenstand von einem der Beispiele 679 bis 683. In Beispiel 684 umfasst das Verfahren das Erzeugen eines temporären symmetrischen Schlüssels im IoT-Server.

[2288] Beispiel 685 umfasst den Gegenstand von einem der Beispiele 679 bis 684. In Beispiel 685 umfasst das Verfahren das Empfangen einer Widerrufsnachricht am IoT-Server zum Widerrufen des Kommunikationsschlüssels.

[2289] Beispiel 686 umfasst den Gegenstand von einem der Beispiele 679 bis 685. In Beispiel 686 umfasst das Verfahren das Ablaufenlassen des Kommunikationsschlüssels und das Anfordern eines neuen Kommunikationsschlüssels, der vom Dienstanbieter bereitzustellen ist.

[2290] Beispiel 687 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, sich bei einem Dienstanbieter zu authentifizieren, einen Schlüssel vom Dienstanbieter zu erhalten, einer Vorrichtung einen Kommunikationsschlüssel bereitzustellen und unter Verwendung des Schlüssels mit der Vorrichtung zu kommunizieren, um Daten zu verschlüsseln und zu entschlüsseln.

[2291] Beispiel 688 umfasst den Gegenstand von Beispiel 687. In Beispiel 688 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Schlüssel von der Vorrichtung zu empfangen.

[2292] Beispiel 689 umfasst den Gegenstand von Beispiel 687 oder 688. In Beispiel 689 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Kommunikationsschlüssel ansprechend auf den vom Dienstanbieter empfangenen Schlüssel zu erzeugen.

[2293] Beispiel 690 umfasst den Gegenstand von einem der Beispiele 687 bis 689. In Beispiel 690 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, ob der Schlüssel eine vorgegebene Lebensdauer überschritten hat.

[2294] Beispiel 691 umfasst den Gegenstand von einem der Beispiele 687 bis 690. In Beispiel 691 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Schlüssel zu widerrufen und die Authentifizierung am Dienstanbieter zu wiederholen.

[2295] Beispiel 692 umfasst den Gegenstand von einem der Beispiele 687 bis 691. In Beispiel 692 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen widerrufenen oder abgelaufenen Schlüssel wiederaufzufrischen.

[2296] Beispiel 693 umfasst den Gegenstand von einem der Beispiele 687 bis 692. In Beispiel 693 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Widerrufsnachricht zu empfangen und den Schlüssel zu widerrufen.

[2297] Beispiel 694 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Dienstspezifizierer zum Spezifizieren von Diensten, die der IoT-Vorrichtung zur Verfügung stehen, von Diensten, die von der IoT-Vorrichtung bereitgestellt werden können, oder von beiden, einen Vertragsspezifizierer zum Entdecken eines Vertrags für die IoT-Vorrichtung und eine Vertragsbeitrittsfunktion zum Hinzufügen der IoT-Vorrichtung zum Vertrag.

[2298] Beispiel 695 umfasst den Gegenstand von Beispiel 694. In Beispiel 695 umfasst die IoT-Vorrichtung eine Blockchain-Logik zum Teilen und Unterhalten einer Blockchain über ein Netz von IoT-Vorrichtungen, wobei die Blockchain Dienste, Verträge, Identitäten, Attribute oder Kombinationen davon umfasst.

[2299] Beispiel 696 umfasst den Gegenstand von Beispiel 694 oder 695. In Beispiel 696 umfasst die Blockchain eine Liste erzeugter Vorrichtungen, wobei die Liste erzeugter Vorrichtungen die zum Vertrag hinzugefügten Vorrichtungen umfasst.

[2300] Beispiel 697 umfasst den Gegenstand von einem der Beispiele 694 bis 696. In Beispiel 697 umfasst die Blockchain eine Vorrichtungsattributliste für alle Vorrichtungen in der Liste erzeugter Vorrichtungen, einschließlich Zusammenhangseigenschaften, bekanntgemachter Dienste oder beider für die Vorrichtung.

[2301] Beispiel 698 umfasst den Gegenstand von einem der Beispiele 694 bis 697. In Beispiel 698 umfasst die IoT-Vorrichtung eine Vertragsverlassensfunktion zum Beenden der Teilnahme der IoT-Vorrichtung an einem Vertrag.

[2302] Beispiel 699 umfasst den Gegenstand von einem der Beispiele 694 bis 698. In Beispiel 699 umfasst die IoT-Vorrichtung eine Tokenausgabefunktion zur Ausgabe von Tokens an Vorrichtungen.

[2303] Beispiel 700 umfasst den Gegenstand von einem der Beispiele 694 bis 699. In Beispiel 700 umfasst die IoT-Vorrichtung eine Token-Widerrufsfunktion zum Ungültigmachen an eine Vorrichtung ausgegebener Tokens, wenn die Vorrichtung den Vertrag verlässt.

[2304] Beispiel 701 umfasst den Gegenstand von einem der Beispiele 694 bis 700. In Beispiel 701 umfasst die IoT-Vorrichtung ein vertrauenswürdiges Plattformmodul zur Ausführung von Messungen für eine vertrauenswürdige Ausführungsumgebung während eines Boot-Prozesses.

[2305] Beispiel 702 umfasst ein Verfahren zur Behandlung eines Lebenszyklus von Vorrichtungen. Das Verfahren zur Behandlung eines Lebenszyklus von Vorrichtungen umfasst das Booten einer IoT-Vorrichtung in eine sichere Enklave, das Ausführen eines Identitäts-Clients in der sicheren Enklave, das Erhalten einer Identität für die IoT-Vorrichtung, das Erzeugen einer Kommissionierungstransaktion für die IoT-Vorrichtung, das Spezifizieren der IoT-Vorrichtung zur Verfügung stehender Verträge und das Hinzufügen der IoT-Vorrichtung zu einem Vertrag.

[2306] Beispiel 703 umfasst den Gegenstand von Beispiel 702. In Beispiel 703 umfasst das Erhalten einer Identität für die IoT-Vorrichtung die Spezifikation von Diensten, von denen die Identität erhalten werden kann, das Auswählen eines Dienstes zum Erhalten der Identität und das Anfordern der Identität vom Dienst.

[2307] Beispiel 704 umfasst den Gegenstand von Beispiel 702 oder 703. In Beispiel 704 umfasst die Identität einen DNS-Namen, einen NetBIOS-Namen, eine IP-Adresse oder eine UUID oder Kombinationen davon.

[2308] Beispiel 705 umfasst den Gegenstand von einem der Beispiele 702 bis 704. In Beispiel 705 wird die Identität zumindest teilweise auf der Grundlage des Vertrags ausgewählt.

[2309] Beispiel 706 umfasst den Gegenstand von einem der Beispiele 702 bis 705. In Beispiel 706 umfasst das Verfahren das Senden einer Warnnachricht, falls das Erhalten der Identität fehlschlägt.

[2310] Beispiel 707 umfasst den Gegenstand von einem der Beispiele 702 bis 706. In Beispiel 707 umfasst das Verfahren das Zuweisen eines anfänglichen Guthabenbetrags, wenn die Identität erhalten wird.

[2311] Beispiel 708 umfasst den Gegenstand von einem der Beispiele 702 bis 707. In Beispiel 708 umfasst das Hinzufügen der IoT-Vorrichtung zum Vertrag das Senden einer Gebühr zu einer Wallet-Adresse für einen Eigentümer des Vertrags.

[2312] Beispiel 709 umfasst den Gegenstand von einem der Beispiele 702 bis 708. In Beispiel 709 umfasst das Verfahren das Abschließen des Erfüllens von Anforderungen für den Beitritt zum Vertrag, bevor dem Vertrag beigetreten wird.

[2313] Beispiel 710 umfasst den Gegenstand von einem der Beispiele 702 bis 709. In Beispiel 710 umfassen die Anforderungen das Verschlüsseln eines Speichers vor dem Beitritt zum Vertrag.

[2314] Beispiel 711 umfasst den Gegenstand von einem der Beispiele 702 bis 710. In Beispiel 711 umfasst das Verfahren das Hinzufügen der IoT-Vorrichtung zu einer Liste erzeugter Vorrichtungen in Zusammenhang mit dem Vertrag.

[2315] Beispiel 712 umfasst den Gegenstand von einem der Beispiele 702 bis 711. In Beispiel 712 umfasst das Verfahren das Bekanntmachen von Vorrichtungsattributen für die IoT-Vorrichtung.

[2316] Beispiel 713 umfasst den Gegenstand von einem der Beispiele 702 bis 712. In Beispiel 713 umfasst das Verfahren das Identifizieren eines Mechanismus zum Attestieren von jedem der Vorrichtungsattribute.

[2317] Beispiel 714 umfasst den Gegenstand von einem der Beispiele 702 bis 713. In Beispiel 714 umfasst das Verfahren das Anfordern von Tokens für das Funktionieren unter dem Vertrag.

[2318] Beispiel 715 umfasst den Gegenstand von einem der Beispiele 702 bis 714. In Beispiel 715 umfasst das Verfahren das Präsentieren eines Tokens für einen Eigentümer eines Dienstes, um Zugang zum Dienst zu ermöglichen.

[2319] Beispiel 716 umfasst den Gegenstand von einem der Beispiele 702 bis 715. In Beispiel 716 umfasst das Verfahren das Kommissionieren der IoT-Vorrichtung für den Betrieb unter dem Vertrag und das Ausführen von Operationen unter dem Vertrag.

[2320] Beispiel 717 umfasst den Gegenstand von einem der Beispiele 702 bis 716. In Beispiel 717 umfasst das Verfahren das Dekommissionieren der IoT-Vorrichtung und das Erfüllen von Bedingungen, die erforderlich sind, um den Vertrag zu verlassen.

[2321] Beispiel 718 umfasst den Gegenstand von einem der Beispiele 702 bis 717. In Beispiel 718 umfasst das Verfahren das Ausführen eines Rücksetzens in einer Fabrik nach dem Verlassen des Vertrags.

[2322] Beispiel 719 umfasst den Gegenstand von einem der Beispiele 702 bis 718. In Beispiel 719 umfasst das Verfahren das Senden einer Lebenssendenachricht zu einem Wartungsdienstanbieter nach dem Verlassen des Vertrags.

[2323] Beispiel 720 umfasst den Gegenstand von einem der Beispiele 702 bis 719. In Beispiel 720 umfasst das Verfahren das Rückerstatten verbleibender Guthabenbeträge für die IoT-Vorrichtung, wenn die IoT-Vorrichtung den Vertrag verlässt.

[2324] Beispiel 721 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, in eine sichere Enklave zu booten, eine Identität zu erhalten, verfügbare Verträge zu spezifizieren und einem Vertrag beizutreten.

[2325] Beispiel 722 umfasst den Gegenstand von Beispiel 721. In Beispiel 722 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Schlüssel für die Verwendung als Blockchain-Client zu erzeugen.

[2326] Beispiel 723 umfasst den Gegenstand von Beispiel 721 oder 722. In Beispiel 723 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Attribute für eine IoT-Vorrichtung zu veröffentlichen.

[2327] Beispiel 724 umfasst den Gegenstand von einem der Beispiele 721 bis 723. In Beispiel 724 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Tokens für den Betrieb unter einem Vertrag anzufordern.

[2328] Beispiel 725 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst ein Bloomfilter zum Speichern von Informationen über Elemente in einem K-Bucket, eine Blockchain-Logik zum Erzeugen von Einträgen in einer Blockchain, einschließlich Werte vom Bloomfilter, einen Inhaltserzeuger zum Erzeugen von Hash-Codes für das Bloomfilter und einen Suchmanager zum Suchen eines Bloomfilters, um die Wahrscheinlichkeit zu bestimmen, dass ein Suchziel vorhanden ist.

- [2329]** Beispiel 726 umfasst den Gegenstand von Beispiel 725. In Beispiel 726 umfasst das Bloomfilter eine Speicherstruktur, die eine Bitsequenz umfasst.
- [2330]** Beispiel 727 umfasst den Gegenstand von Beispiel 725 oder 726. In Beispiel 727 umfasst die Bitsequenz für jedes der Elemente berechnete überschriebene Hash-Werte.
- [2331]** Beispiel 728 umfasst den Gegenstand von einem der Beispiele 725 bis 727. In Beispiel 728 umfasst das K-Bucket eine Gruppe von Knoten in Zusammenhang mit der IoT-Vorrichtung.
- [2332]** Beispiel 729 umfasst den Gegenstand von einem der Beispiele 727 bis 728. In Beispiel 729 umfassen die Elemente Ressourcen, Dienste, Verträge oder IoT-Vorrichtungsidentitäten oder Kombinationen davon.
- [2333]** Beispiel 730 umfasst den Gegenstand von einem der Beispiele 725 bis 729, einschließlich einer Verteilte-Hashtags- bzw. Distributed-Hash-Tag(DHT)-Datenbank zu 730. In Beispiel 730 umfasst die DHT-Datenbank einen individuellen Eintrag für jedes Element im K-Bucket.
- [2334]** Beispiel 731 umfasst den Gegenstand von einem der Beispiele 725 bis 730. In Beispiel 731 umfasst die Vorrichtung einen Inhaltslokalisierer zum Durchsuchen der DHT-Datenbank, um festzustellen, ob ein Element vorhanden ist.
- [2335]** Beispiel 732 umfasst den Gegenstand von einem der Beispiele 725 bis 731. In Beispiel 732 soll der Inhaltserzeuger Inhalt erzeugen, der eine universelle Ressourcenkennung bzw. Universal Resource Identifier (URI), Metadaten oder Hash-Codes oder Kombinationen davon umfasst.
- [2336]** Beispiel 733 umfasst den Gegenstand von einem der Beispiele 725 bis 732. In Beispiel 733 soll der Suchmanager Zölle für weitere Hops in einer Suche bezahlen.
- [2337]** Beispiel 734 umfasst ein Verfahren zur Ressourcenentdeckung. Das Verfahren zur Ressourcenentdeckung umfasst das Berechnen eines Hash-Codes für ein Suchziel, das Vergleichen von Bits des Hash-Codes mit in einem Bloomfilter festgelegten Bits und das Ausführen einer Suche für eine verteilte Hash-Tabelle bzw. Distributed Hash Table (DHT) für den Hash-Code des Suchziels, falls die Bits des Hash-Codes mit im Bloomfilter festgelegten Bits übereinstimmen.
- [2338]** Beispiel 735 umfasst den Gegenstand von Beispiel 734. In Beispiel 735 umfasst das Verfahren das Ausstrahlen des Hash-Codes des Suchziels an Knoten innerhalb eines lokalen K-Buckets und das Ausführen einer Durchsuchung der DHT an jeglichen der Knoten innerhalb des lokalen K-Buckets, wofür die Bits des Hash-Codes mit den im Bloomfilter festgelegten Bits übereinstimmen.
- [2339]** Beispiel 736 umfasst den Gegenstand von Beispiel 734 oder 735. In Beispiel 736 umfasst das Verfahren das Feststellen, dass die Suche in einem lokalen K-Bucket nicht erfolgreich war, das Feststellen von Zollgebühren für das Senden des Hash-Codes für das Suchziel zu einem fernen K-Bucket, das Zahlen der Zollgebühren und das Senden des Hash-Codes für das Suchziel zum fernen K-Bucket, um die Suche fortzusetzen.
- [2340]** Beispiel 737 umfasst den Gegenstand von einem der Beispiele 734 bis 736. In Beispiel 737 umfasst das Verfahren das Feststellen, dass die Suche in einem lokalen K-Bucket nicht erfolgreich war, das Feststellen einer Zollgebühr für das Senden des Hash-Codes für das Suchziel zu einem fernen K-Bucket und das Beenden der Suche, falls die Zollgebühren eine vorgegebene Grenze überschreiten.
- [2341]** Beispiel 738 umfasst den Gegenstand von einem der Beispiele 734 bis 737. In Beispiel 738 umfasst das Verfahren das Initialisieren der DHT, das Erzeugen einer Blockchain-Datenbank, das Erzeugen eines Geneseblocks in der Blockchain-Datenbank und das Kopieren der Blockchain-Datenbank und der DHT zu jedem der mehreren Teilnehmer.
- [2342]** Beispiel 739 umfasst den Gegenstand von einem der Beispiele 734 bis 738. In Beispiel 739 umfasst das Verfahren das Speichern des Bloomfilters in der Blockchain-Datenbank als Transaktion.
- [2343]** Beispiel 740 umfasst den Gegenstand von einem der Beispiele 734 bis 739. In Beispiel 740 umfasst das Verfahren das Speichern eines Zeigers in der Blockchain-Datenbank als Transaktion, wobei der Zeiger den Ort der DHT umfasst.

[2344] Beispiel 741 umfasst den Gegenstand von einem der Beispiele 734 bis 740. In Beispiel 741 umfasst das Verfahren das Erzeugen von Inhalt für die Blockchain-Datenbank, die DHT oder beide, einschließlich des Erzeugens eines Element-Hash-Codes, des Speicherns des Element-Hash-Codes in der DHT, des Erzeugens einer einheitlichen Ressourcenkennung (URI) für in der DHT gespeicherte Daten und des Speicherns der URI und des Element-Hash-Codes in der Blockchain-Datenbank.

[2345] Beispiel 742 umfasst den Gegenstand von einem der Beispiele 734 bis 741. In Beispiel 742 umfasst das Verfahren das Speichern von Metadaten für die URI und des Element-Hash-Codes in der Blockchain-Datenbank, wobei die Metadaten Aktionen für Inhaltserzeuger steuern.

[2346] Beispiel 743 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Inhalt für das Lokalisieren von Ressourcen durch Berechnen eines Hash-Codes für ein Suchziel nachzuschlagen und Bits des Hash-Codes mit in einem Bloomfilter festgelegten Bits zu vergleichen.

[2347] Beispiel 744 umfasst den Gegenstand von Beispiel 743. In Beispiel 744 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass die Bits des Hash-Codes mit im Bloomfilter festgelegten Bits übereinstimmen, und eine DHT zu durchsuchen, um festzustellen, ob sich der Hash-Code in der DHT befindet.

[2348] Beispiel 745 umfasst den Gegenstand von Beispiel 743 oder 744. In Beispiel 745 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Blockchain-Datenbank zu erzeugen.

[2349] Beispiel 746 umfasst den Gegenstand von einem der Beispiele 743 bis 745. In Beispiel 746 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Inhalt für die Blockchain-Datenbank zu erzeugen, einschließlich des Berechnens eines Element-Hash-Codes für jedes von mehreren Elementen.

[2350] Beispiel 747 umfasst den Gegenstand von einem der Beispiele 743 bis 746. In Beispiel 747 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, jeden Element-Hash-Code in der Blockchain-Datenbank zu speichern.

[2351] Beispiel 748 umfasst den Gegenstand von einem der Beispiele 743 bis 747. In Beispiel 748 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Gebühr für das Senden einer Suche zu anderen Knoten zu zahlen, falls die Suche in lokalen Knoten nicht erfolgreich war.

[2352] Beispiel 749 umfasst eine Vorrichtung zur Verwendung in einem Internetder-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Erlaubnisanleitungsverfasser zum Verfassen einer Erlaubnisanleitung für mehrere entdeckte Peers, wobei die mehreren entdeckten Peers jeweils einen Parameter haben und wobei eine Bedingung der Erlaubnisanleitung ansprechend darauf erzeugt wird, dass die Bedingung von wenigstens zwei der mehreren entdeckten Peers zugelassen wird. Der Parameter jedes entdeckbaren Peers von den mehreren entdeckten Peers umfasst einen Bereich zulässiger Bedingungen für einen zugeordneten Peer und einen Aktionsausführer zur Ausführung einer Aktion der Erlaubnisanleitung ansprechend auf eine Erkennung, dass eine Kondition der Bedingung erfüllt ist.

[2353] Beispiel 750 umfasst den Gegenstand von Beispiel 749. In Beispiel 750 umfasst der Erlaubnisanleitungsverfasser eine Funktion zum Auflisten der Bedingungen und Konditionen der mehreren entdeckten Peers.

[2354] Beispiel 751 umfasst den Gegenstand von Beispiel 749 oder 750. In Beispiel 751 umfasst der Erlaubnisanleitungsverfasser eine Auflistung der Dienstqualitätsbedingungen und -konditionen für die mehreren entdeckten Peers.

[2355] Beispiel 752 umfasst den Gegenstand von einem der Beispiele 749 bis 751. In Beispiel 752 umfasst der Erlaubnisanleitungsverfasser eine Auflistung der Datenebenenbedingungen und -konditionen für die mehreren entdeckten Peers.

[2356] Beispiel 753 umfasst den Gegenstand von einem der Beispiele 752 bis 752. In Beispiel 753 soll die Datenebene einen Prozess dafür angeben, wie die Daten den Peers zuzuführen sind und von diesen zu verbrauchen sind.

[2357] Beispiel 754 umfasst den Gegenstand von einem der Beispiele 749 bis 753. In Beispiel 754 umfasst die Erlaubnisanleitung eine Lebenszeit.

[2358] Beispiel 755 umfasst den Gegenstand von einem der Beispiele 749 bis 754. In Beispiel 755 umfasst die Erlaubnisanleitung einen Protokollumwandlungsvermittler zur Behandlung des Beitretens zur Erlaubnisanleitung und ihres Verlassens durch einen Peer.

[2359] Beispiel 756 umfasst den Gegenstand von einem der Beispiele 749 bis 755. In Beispiel 756 umfasst die Ausführung einer Aktion der Erlaubnisanleitung das Autokommissionieren eines Dienstes für einen Peer, wodurch der Peer angewiesen wird, Daten zu verarbeiten.

[2360] Beispiel 757 umfasst den Gegenstand von einem der Beispiele 749 bis 756. In Beispiel 757 umfasst die Erlaubnisanleitung eine Präambel zur Behandlung des Austausches einer Konfiguration zwischen den mehreren entdeckten Peers.

[2361] Beispiel 758 umfasst den Gegenstand von einem der Beispiele 749 bis 757. In Beispiel 758 bezieht sich die Bedingung auf eine zwischen den mehreren entdeckten Peers zu zahlende Zahlungsrate und wird eine abschließende Zahlung zwischen Peers vorgenommen, nachdem erkannt wurde, dass ein Peer von den mehreren entdeckten Peers die Teilnahme an der Erlaubnisanleitung beendet.

[2362] Beispiel 759 umfasst ein Verfahren zur Aufgabendefinition und Kommissionierung in einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur Aufgabendefinition und Kommissionierung in einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Entwerfen einer Erlaubnisanleitung für mehrere entdeckte Peers, wobei die mehreren entdeckten Peers jeweils einen Parameter haben und wobei eine Bedingung der Erlaubnisanleitung ansprechend darauf erzeugt wird, dass die Bedingung von wenigstens zwei der mehreren entdeckten Peers zugelassen wird, und das Ausführen einer Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt ist.

[2363] Beispiel 760 umfasst den Gegenstand von Beispiel 759. In Beispiel 760 umfasst das Verfassen der Erlaubnisanleitung eine Funktion zum Auflisten der Bedingungen und Konditionen der mehreren entdeckten Peers.

[2364] Beispiel 761 umfasst den Gegenstand von einem der Beispiele 759 bis 760. In Beispiel 761 umfasst das Verfassen der Erlaubnisanleitung das Auflisten der Dienstqualitätsbedingungen und -konditionen für die mehreren entdeckten Peers.

[2365] Beispiel 762 umfasst den Gegenstand von einem der Beispiele 759 bis 761. In Beispiel 762 umfasst das Verfassen der Erlaubnisanleitung das Auflisten der Datenebenenbedingungen und -konditionen für die mehreren entdeckten Peers.

[2366] Beispiel 763 umfasst den Gegenstand von einem der Beispiele 759 bis 76. In Beispiel 763 soll die Datenebene einen Prozess dafür angeben, wie die Daten den Peers zuzuführen sind und von diesen zu verbrauchen sind.

[2367] Beispiel 764 umfasst den Gegenstand von einem der Beispiele 759 bis 763. In Beispiel 764 umfasst die Erlaubnisanleitung eine Lebenszeit.

[2368] Beispiel 765 umfasst den Gegenstand von einem der Beispiele 759 bis 764. In Beispiel 765 umfasst die Erlaubnisanleitung einen Protokollumwandlungsvermittler zur Behandlung des Beitretens zur Erlaubnisanleitung und ihres Verlassens durch einen Peer.

[2369] Beispiel 766 umfasst den Gegenstand von einem der Beispiele 759 bis 765. In Beispiel 766 umfasst die Ausführung einer Aktion der Erlaubnisanleitung das Autokommissionieren eines Dienstes für einen Peer, wodurch der Peer angewiesen wird, Daten zu verarbeiten.

[2370] Beispiel 767 umfasst den Gegenstand von einem der Beispiele 759 bis 766. In Beispiel 767 umfasst die Erlaubnisanleitung eine Präambel zur Behandlung des Austausches einer Konfiguration zwischen den mehreren entdeckten Peers.

[2371] Beispiel 768 umfasst den Gegenstand von einem der Beispiele 759 bis 767. In Beispiel 768 bezieht sich die Bedingung auf eine zwischen den mehreren entdeckten Peers zu zahlende Zahlungsrate und wird eine abschließende Zahlung zwischen Peers vorgenommen, nachdem erkannt wurde, dass ein Peer von den mehreren entdeckten Peers die Teilnahme an der Erlaubnisanleitung beendet.

[2372] Beispiel 769 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Erlaubnisanleitung für mehrere entdeckte Peers zu verfassen, wobei die mehreren entdeckten Peers jeweils einen Parameter haben, und wobei eine Bedingung der Erlaubnisanleitung ansprechend darauf erzeugt wird, dass die Bedingung von wenigstens zwei der mehreren entdeckten Peers erlaubt wird, und eine Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt ist, auszuführen.

[2373] Beispiel 770 umfasst den Gegenstand von Beispiel 769. In Beispiel 770 umfasst das Verfassen der Erlaubnisanleitung eine Funktion zum Auflisten der Bedingungen und Konditionen der mehreren entdeckten Peers.

[2374] Beispiel 771 umfasst den Gegenstand von Beispiel 769 oder 770. In Beispiel 771 umfasst das Verfassen der Erlaubnisanleitung das Auflisten der Dienstqualitätsbedingungen und -konditionen für die mehreren entdeckten Peers.

[2375] Beispiel 772 umfasst den Gegenstand von einem der Beispiele 769 bis 771. In Beispiel 772 umfasst das Verfassen der Erlaubnisanleitung das Auflisten der Datenebenenbedingungen und -konditionen für die mehreren entdeckten Peers.

[2376] Beispiel 773 umfasst den Gegenstand von einem der Beispiele 769 bis 772. In Beispiel 773 soll die Datenebene einen Prozess dafür angeben, wie die Daten den Peers zuzuführen sind und von diesen zu verbrauchen sind.

[2377] Beispiel 774 umfasst den Gegenstand von einem der Beispiele 769 bis 773. In Beispiel 774 umfasst die Erlaubnisanleitung eine Lebenszeit.

[2378] Beispiel 775 umfasst den Gegenstand von einem der Beispiele 769 bis 774. In Beispiel 775 umfasst die Erlaubnisanleitung einen Protokollumwandlungsvermittler zur Behandlung des Beitretens zur Erlaubnisanleitung und ihres Verlassens durch einen Peer.

[2379] Beispiel 776 umfasst den Gegenstand von einem der Beispiele 769 bis 775. In Beispiel 776 umfasst die Ausführung einer Aktion der Erlaubnisanleitung das Autokommissionieren eines Dienstes für einen Peer, wodurch der Peer angewiesen wird, Daten zu verarbeiten.

[2380] Beispiel 777 umfasst den Gegenstand von einem der Beispiele 769 bis 776. In Beispiel 777 umfasst die Erlaubnisanleitung eine Präambel zur Behandlung des Austausches einer Konfiguration zwischen den mehreren entdeckten Peers.

[2381] Beispiel 778 umfasst den Gegenstand von einem der Beispiele 769 bis 777. In Beispiel 778 bezieht sich die Bedingung auf eine zwischen den mehreren entdeckten Peers zu zahlende Zahlungsrate und wird eine abschließende Zahlung zwischen Peers vorgenommen, nachdem erkannt wurde, dass ein Peer von den mehreren entdeckten Peers die Teilnahme an der Erlaubnisanleitung beendet.

[2382] Beispiel 779 umfasst eine Vorrichtung zur Verwendung in einem Internetder-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Verfasser einer Schwebender-Dienst-Erlaubnisanleitung zum Verfassen einer Schwebender-Dienst-Erlaubnisanleitung für mehrere entdeckte Hosts, wobei die mehreren entdeckten Hosts jeweils darauf beurteilt werden, ob sie einen Parameter erfüllen. Die Vorrichtung umfasst auch einen Host-Hardwarewähler zum Wählen einer Host-Hardware für den schwebenden Dienst auf der Grundlage einer Datenstruktur des schwebenden Dienstes, einen Ausführer der Schwebender-Dienst-Erlaubnisanleitung zum Ausführen der Schwebender-Dienst-Erlaubnisanleitung unter Verwendung der Host-Hardware und einen Wertübertrager zur Übertragung von Wert auf ein Dienst-Wallet in Zusammen-

menhang mit dem schwebenden Dienst ansprechend darauf, dass erkannt wurde, dass eine Bedingung der schwebenden Erlaubnisanleitung erreicht wurde.

[2383] Beispiel 780 umfasst den Gegenstand von Beispiel 779. In Beispiel 780 leitet der schwebende Dienst eine Werttransaktion zwischen dem Dienst-Wallet und einem Host-Wallet ein.

[2384] Beispiel 781 umfasst den Gegenstand von Beispiel 779 oder 780. In Beispiel 781 hält das Dienst-Wallet einen Blockchain-codierten Wert.

[2385] Beispiel 782 umfasst den Gegenstand von einem der Beispiele 779 bis 781. In Beispiel 782 ist die Datenstruktur eine Entscheidungsmatrix.

[2386] Beispiel 783 umfasst den Gegenstand von einem der Beispiele 779 bis 782. In Beispiel 783 listet die Entscheidungsmatrix ein vom schwebenden Dienst gesuchtes Merkmal, eine Anzahl verfügbarer Hosts und einen Bewertungswert von jedem der Hosts in Bezug auf das in der Entscheidungsmatrix aufgelistete Merkmal auf.

[2387] Beispiel 784 umfasst den Gegenstand von einem der Beispiele 779 bis 783. In Beispiel 784 wählt der schwebende Dienst einen Host auf der Grundlage eines besten Werts aus, der anhand der Kosten pro Stunde, dividiert durch die Anzahl der Merkmale, berechnet wird, wobei die Qualitätsmetrik eine zufrieden stellende Verwendung des schwebenden Dienstes angibt und die Kosten pro Stunde projizierte Kosten pro Stunde des Betriebs des schwebenden Dienstes unter Verwendung eines bewerteten Hosts sind.

[2388] Beispiel 785 umfasst den Gegenstand von einem der Beispiele 779 bis 784. In Beispiel 785 gewichten die Merkmale des schwebenden Dienstes die Merkmale bei einer Wertberechnung unter Verwendung der Entscheidungsmatrix auf verschiedene Weise.

[2389] Beispiel 786 umfasst den Gegenstand von einem der Beispiele 779 bis 785. In Beispiel 786 gibt die Schwebender-Dienst-Erlaubnisanleitung Strafen an, die ansprechend auf eine erkannte Verletzung der Dienst-erlaubnisanleitung gegen einen Host zu erwägen sind, wobei die Strafen von einem Host-Wallet einzuziehen sind.

[2390] Beispiel 787 umfasst den Gegenstand von einem der Beispiele 779 bis 786. In Beispiel 787 hört der schwebende Dienst auf zu funktionieren, wenn das Dienst-Wallet einen Wert von Null hat.

[2391] Beispiel 788 umfasst den Gegenstand von einem der Beispiele 779 bis 787. In Beispiel 788 gibt die Erlaubnisanleitung ansprechend auf eine Erkennung, dass das Dienst-Wallet einen Auslöseschwellenwert erreicht hat, an, dass ein Dienst-Wallet einen Wert übertragen soll.

[2392] Beispiel 789 umfasst ein Verfahren zur Behandlung eines schwebenden Dienstes in einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur Behandlung eines schwebenden Dienstes in einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Verfassen einer Schwebender-Dienst-Erlaubnisanleitung für mehrere entdeckte Hosts, wobei die mehreren entdeckten Hosts jeweils auf eine Erfüllung eines Parameters bewertet werden, das Auswählen einer Host-Hardware für den schwebenden Dienst auf der Grundlage einer Datenstruktur des schwebenden Dienstes, das Ausführen der Schwebender-Dienst-Erlaubnisanleitung unter Verwendung der Host-Hardware und das Übertragen von Wert auf ein dem schwebenden Dienst zugeordnetes Dienst-Wallet ansprechend auf eine Erkennung, dass eine Bedingung der Schwebende-Erlaubnisanleitung erreicht wurde.

[2393] Beispiel 790 umfasst den Gegenstand von Beispiel 789. In Beispiel 790 leitet der schwebende Dienst eine Werttransaktion zwischen dem Dienst-Wallet und einem Host-Wallet ein.

[2394] Beispiel 791 umfasst den Gegenstand von Beispiel 789 oder 790. In Beispiel 791 hält das Dienst-Wallet einen Blockchain-codierten Wert.

[2395] Beispiel 792 umfasst den Gegenstand von einem der Beispiele 789 bis 791. In Beispiel 792 ist die Datenstruktur eine Entscheidungsmatrix.

[2396] Beispiel 793 umfasst den Gegenstand von einem der Beispiele 789 bis 792. In Beispiel 793 listet die Entscheidungsmatrix ein vom schwebenden Dienst gesuchtes Merkmal, eine Anzahl verfügbarer Hosts und

einen Bewertungswert von jedem der Hosts in Bezug auf das in der Entscheidungsmatrix aufgelistete Merkmal auf.

[2397] Beispiel 794 umfasst den Gegenstand von einem der Beispiele 789 bis 793. In Beispiel 794 wählt der schwebende Dienst einen Host auf der Grundlage eines besten Werts aus, der anhand der Kosten pro Stunde, dividiert durch die Anzahl der Merkmale, berechnet wird, wobei die Qualitätsmetrik eine zufrieden stellende Verwendung des schwebenden Dienstes angibt und die Kosten pro Stunde projizierte Kosten pro Stunde des Betriebs des schwebenden Dienstes unter Verwendung eines bewerteten Hosts sind.

[2398] Beispiel 795 umfasst den Gegenstand von einem der Beispiele 789 bis 794. In Beispiel 795 gewichten die Merkmale des schwebenden Dienstes die Merkmale bei einer Wertberechnung unter Verwendung der Entscheidungsmatrix auf verschiedene Weise.

[2399] Beispiel 796 umfasst den Gegenstand von einem der Beispiele 789 bis 795. In Beispiel 796 gibt die Schwebender-Dienst-Erlaubnisanleitung Strafen an, die ansprechend auf eine erkannte Verletzung der Dienst-erlaubnisanleitung gegen einen Host zu erwägen sind, wobei die Strafen von einem Host-Wallet einzuziehen sind.

[2400] Beispiel 797 umfasst den Gegenstand von einem der Beispiele 789 bis 796. In Beispiel 797 hört der schwebende Dienst auf zu funktionieren, wenn das Dienst-Wallet einen Wert von Null hat.

[2401] Beispiel 798 umfasst den Gegenstand von einem der Beispiele 789 bis 797. In Beispiel 798 gibt die Erlaubnisanleitung ansprechend auf eine Erkennung, dass das Dienst-Wallet einen Auslöseschwellenwert erreicht hat, an, dass ein Dienst-Wallet einen Wert übertragen soll.

[2402] Beispiel 799 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Schwebender-Dienst-Erlaubnisanleitung für mehrere entdeckte Hosts zu verfassen, wobei die mehreren entdeckten Hosts jeweils auf eine Erfüllung eines Parameters bewertet werden, eine Host-Hardware für den schwebenden Dienst auf der Grundlage einer Datenstruktur des schwebenden Dienstes auszuwählen, die Schwebender-Dienst-Erlaubnisanleitung unter Verwendung der Host-Hardware auszuführen und ansprechend auf eine Erkennung, dass eine Bedingung der Schwebende-Erlaubnisanleitung erreicht wurde, Wert auf ein dem schwebenden Dienst zugeordnetes Dienst-Wallet zu übertragen.

[2403] Beispiel 800 umfasst den Gegenstand von Beispiel 799. In Beispiel 800 leitet der schwebende Dienst eine Werttransaktion zwischen dem Dienst-Wallet und einem Host-Wallet ein.

[2404] Beispiel 801 umfasst den Gegenstand von Beispiel 799 oder 800. In Beispiel 801 hält das Dienst-Wallet einen Blockchain-codierten Wert.

[2405] Beispiel 802 umfasst den Gegenstand von einem der Beispiele 799 bis 801. In Beispiel 802 ist die Datenstruktur eine Entscheidungsmatrix.

[2406] Beispiel 803 umfasst den Gegenstand von einem der Beispiele 779 bis 802. In Beispiel 803 listet die Entscheidungsmatrix ein vom schwebenden Dienst gesuchtes Merkmal, eine Anzahl verfügbarer Hosts und einen Bewertungswert von jedem der Hosts in Bezug auf das in der Entscheidungsmatrix aufgelistete Merkmal auf.

[2407] Beispiel 804 umfasst den Gegenstand von einem der Beispiele 779 bis 803. In Beispiel 804 wählt der schwebende Dienst einen Host auf der Grundlage eines besten Werts aus, der anhand der Kosten pro Stunde, dividiert durch die Anzahl der Merkmale, berechnet wird, wobei die Qualitätsmetrik eine zufrieden stellende Verwendung des schwebenden Dienstes angibt und die Kosten pro Stunde projizierte Kosten pro Stunde des Betriebs des schwebenden Dienstes unter Verwendung eines bewerteten Hosts sind.

[2408] Beispiel 805 umfasst den Gegenstand von einem der Beispiele 779 bis 804. In Beispiel 805 gewichten die Merkmale des schwebenden Dienstes die Merkmale bei einer Wertberechnung unter Verwendung der Entscheidungsmatrix auf verschiedene Weise.

[2409] Beispiel 806 umfasst den Gegenstand von einem der Beispiele 799 bis 805. In Beispiel 806 gibt die Schwebender-Dienst-Erlaubnisanleitung Strafen an, die ansprechend auf eine erkannte Verletzung der Dienst-

terlaubnisanleitung gegen einen Host zu erwägen sind, wobei die Strafen von einem Host-Wallet einzuziehen sind.

[2410] Beispiel 807 umfasst den Gegenstand von einem der Beispiele 799 bis 806. In Beispiel 807 hört der schwebende Dienst auf zu funktionieren, wenn das Dienst-Wallet einen Wert von Null hat.

[2411] Beispiel 808 umfasst den Gegenstand von einem der Beispiele 799 bis 807. In Beispiel 808 gibt die Erlaubnisanleitung ansprechend auf eine Erkennung, dass das Dienst-Wallet einen Auslöseschwellenwert erreicht hat, an, dass ein Dienst-Wallet einen Wert übertragen soll.

[2412] Beispiel 809 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Erlaubnisanleitungsverfasser zum Verfassen einer Erlaubnisanleitung für einen ersten entdeckten Peer, einschließlich eines ersten Parameters und eines ersten Parameterwerts. Die Vorrichtung umfasst auch einen zweiten entdeckten Peer, einschließlich eines zweiten Parameters und eines zweiten Parameterwerts, einen Parametergewichtsberechner zur Berechnung eines ersten Parametergewichts und eines zweiten Parametergewichts durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts, einen Bedingungsgenerator zur Erzeugung einer Bedingung der Erlaubnisanleitung ansprechend darauf, dass eine vorgeschlagene Bedingung innerhalb vom ersten Parameter und vom zweiten Parameter vorgeschlagener Bereiche liegt, wobei der erste Parameter durch das erste Parametergewicht angepasst wird und der zweite Parameter durch das zweite Parametergewicht angepasst wird, und einen Aktionsausführer zur Ausführung einer Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt wurde.

[2413] Beispiel 810 umfasst den Gegenstand von Beispiel 809. In Beispiel 810 umfasst die Vorrichtung einen Prozessor zur Verarbeitung einer Anforderung von einem Kandidaten-Peer an die Erlaubnisanleitung zur Aufnahme eines Beitrittsparameters und eines Beitrittsparameterwerts.

[2414] Beispiel 811 umfasst den Gegenstand von Beispiel 809 oder 810. In Beispiel 811 berechnet der Prozessor ein Beitrittsparametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts mit dem Beitrittsparameterwert.

[2415] Beispiel 812 umfasst den Gegenstand von einem der Beispiele 809 bis 811. In Beispiel 812 beziehen sich der erste Parameter und der zweite Parameter auf akzeptable Datenwertbereiche für einen ersten bzw. einen zweiten Knoten.

[2416] Beispiel 813 umfasst den Gegenstand von einem der Beispiele 809 bis 812. In Beispiel 813 werden die akzeptablen Datenwertbereiche mit einer Kostenfunktion berechnet.

[2417] Beispiel 814 umfasst den Gegenstand von einem der Beispiele 809 bis 813. In Beispiel 814 soll die Kostenfunktion Betriebskosten eines die Erlaubnisanleitung implementierenden Knotens berechnen und kombinieren.

[2418] Beispiel 815 umfasst den Gegenstand von einem der Beispiele 809 bis 814. In Beispiel 815 umfassen die Betriebskosten Energiekosten, laufende Kosten und/oder Wartungskosten für den Betrieb einer Vorrichtung, den Datentransport und Speichervorrichtungen.

[2419] Beispiel 816 umfasst den Gegenstand von einem der Beispiele 809 bis 815. In Beispiel 816 beziehen sich die Datenwertbereiche auf eine Berechnung des Werts der Daten als Funktion der Anzahl der Datenquellen.

[2420] Beispiel 817 umfasst den Gegenstand von einem der Beispiele 809 bis 816. In Beispiel 817 sind die Daten von mehreren Sensoren synthetisierte abgeleitete Daten.

[2421] Beispiel 818 umfasst den Gegenstand von einem der Beispiele 809 bis 817. In Beispiel 818 nimmt der Wert von Daten zu, wenn die Rate der Datennachfrage zunimmt.

[2422] Beispiel 819 umfasst ein Verfahren zur Verhandlung mit einem Wert versehener Dateneinheiten in einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur Verhandlung mit einem Wert versehener Dateneinheiten in einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Verfassen einer Erlaubnisanleitung für einen ersten entdeckten Peer, einschließlich eines ersten Parameters und eines ersten Parameterwerts, und

einen zweiten entdeckten Peer, einschließlich eines zweiten Parameters und eines zweiten Parameterwerts, das Berechnen eines ersten Parametergewichts und eines zweiten Parametergewichts durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts, das Erzeugen einer Bedingung der Erlaubnisanleitung ansprechend darauf, dass eine vorgeschlagene Bedingung innerhalb vom ersten Parameter und vom zweiten Parameter vorgeschlagener Bereiche liegt, wobei der erste Parameter durch das erste Parametergewicht angepasst wird und der zweite Parameter durch das zweite Parametergewicht angepasst wird, und das Ausführen einer Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt wurde.

[2423] Beispiel 820 umfasst den Gegenstand von Beispiel 819. In Beispiel 820 umfasst das Verfahren das Empfangen einer Anforderung von einem Kandidaten-Peer an die Erlaubnisanleitung zur Aufnahme eines Beitrittsparameters und eines Beitrittsparameterwerts.

[2424] Beispiel 821 umfasst den Gegenstand von Beispiel 819 oder 820. In Beispiel 821 umfasst das Verfahren das Berechnen eines Beitrittsparametergewichts durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts mit dem Beitrittsparameterwert.

[2425] Beispiel 822 umfasst den Gegenstand von einem der Beispiele 819 bis 821. In Beispiel 822 beziehen sich der erste Parameter und der zweite Parameter auf akzeptable Datenwertbereiche für einen ersten bzw. einen zweiten Knoten.

[2426] Beispiel 823 umfasst den Gegenstand von einem der Beispiele 819 bis 822. In Beispiel 823 werden die akzeptablen Datenwertbereiche mit einer Kostenfunktion berechnet.

[2427] Beispiel 824 umfasst den Gegenstand von einem der Beispiele 819 bis 823. In Beispiel 824 soll die Kostenfunktion Betriebskosten eines die Erlaubnisanleitung implementierenden Knotens berechnen und kombinieren.

[2428] Beispiel 825 umfasst den Gegenstand von einem der Beispiele 819 bis 824. In Beispiel 825 umfassen die Betriebskosten Energiekosten, laufende Kosten und/oder Wartungskosten für den Betrieb der Vorrichtung, den Datentransport und Speichervorrichtungen.

[2429] Beispiel 826 umfasst den Gegenstand von einem der Beispiele 819 bis 825. In Beispiel 826 beziehen sich die Datenwertbereiche auf eine Berechnung des Werts der Daten als Funktion der Anzahl der Datenquellen.

[2430] Beispiel 827 umfasst den Gegenstand von einem der Beispiele 819 bis 826. In Beispiel 827 sind die Daten von mehreren Sensoren synthetisierte abgeleitete Daten.

[2431] Beispiel 828 umfasst den Gegenstand von einem der Beispiele 819 bis 827. In Beispiel 828 nimmt der Wert von Daten zu, wenn die Rate der Datennachfrage zunimmt.

[2432] Beispiel 829 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Erlaubnisanleitung für einen ersten entdeckten Peer, einschließlich eines ersten Parameters und eines ersten Parameterwerts, und einen zweiten entdeckten Peer, einschließlich eines zweiten Parameters und eines zweiten Parameterwerts, zu verfassen, ein erstes Parametergewicht und ein zweites Parametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts zu berechnen, eine Bedingung der Erlaubnisanleitung ansprechend darauf zu erzeugen, dass eine vorgeschlagene Bedingung innerhalb vom ersten Parameter und vom zweiten Parameter vorgeschlagener Bereiche liegt, wobei der erste Parameter durch das erste Parametergewicht angepasst wird und der zweite Parameter durch das zweite Parametergewicht angepasst wird, und eine Aktion der Erlaubnisanleitung ansprechend auf die Erkennung, dass eine Kondition der Bedingung erfüllt wurde, auszuführen.

[2433] Beispiel 830 umfasst den Gegenstand von Beispiel 829. In Beispiel 830 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine von einem Kandidaten-Peer empfangene Anforderung zu verarbeiten, wobei die Anforderung einen Beitrittsparameter und einen Beitrittsparameterwert umfasst.

[2434] Beispiel 831 umfasst den Gegenstand von Beispiel 829 oder 830. In Beispiel 831 umfasst das nicht-flüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Beitrittsparametergewicht durch Vergleichen des ersten Parameterwerts und des zweiten Parameterwerts mit dem Beitrittsparameterwert zu berechnen.

[2435] Beispiel 832 umfasst den Gegenstand von einem der Beispiele 829 bis 831. In Beispiel 832 beziehen sich der erste Parameter und der zweite Parameter auf akzeptable Datenwertbereiche für einen ersten bzw. einen zweiten Knoten.

[2436] Beispiel 833 umfasst den Gegenstand von einem der Beispiele 829 bis 832. In Beispiel 833 werden die akzeptablen Datenwertbereiche mit einer Kostenfunktion berechnet.

[2437] Beispiel 834 umfasst den Gegenstand von einem der Beispiele 829 bis 833. In Beispiel 834 soll die Kostenfunktion Betriebskosten eines die Erlaubnisanleitung implementierenden Knotens berechnen und kombinieren.

[2438] Beispiel 835 umfasst den Gegenstand von einem der Beispiele 829 bis 834. In Beispiel 835 umfassen die Betriebskosten Energiekosten, laufende Kosten und/oder Wartungskosten für den Betrieb einer Vorrichtung, den Datentransport und Speichervorrichtungen.

[2439] Beispiel 836 umfasst den Gegenstand von einem der Beispiele 829 bis 835. In Beispiel 836 beziehen sich die Datenwertbereiche auf eine Berechnung des Werts der Daten als Funktion der Anzahl der Datenquellen.

[2440] Beispiel 837 umfasst den Gegenstand von einem der Beispiele 829 bis 836. In Beispiel 837 sind die Daten von mehreren Sensoren synthetisierte abgeleitete Daten.

[2441] Beispiel 838 umfasst den Gegenstand von einem der Beispiele 829 bis 837. In Beispiel 838 nimmt der Wert von Daten zu, wenn die Rate der Datennachfrage zunimmt.

[2442] Beispiel 839 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Vorrichtungsidentitätsgenerator zur Erzeugung einer Vorrichtungsidentität für eine Vorrichtung als Blockchain-Client, einen Nachrichtenveröffentlicher zur Veröffentlichung einer Entdeckungsausstrahlungsnachricht von der Vorrichtung, einen Netzbewerber, damit sich die Vorrichtung bewerben kann, sich ansprechend darauf, dass die Vorrichtung auf der Grundlage der veröffentlichten Entdeckungsausstrahlungsnachricht eine Antwort vom DNAP empfängt, einem Dezentralisiertes-Netzzugangs-Proxy- bzw. Decentralized-Network-Access-Proxy(DNAP)-Netz anzuschließen, einen Vorrichtungsbeschreiber zur Beschreibung der Identität und von Attributen der Vorrichtung für das DNAP und einen Paketabsender zum Senden eines Pakets von der Vorrichtung durch das Netz ansprechend darauf, dass der Vorrichtung durch das Netz auf der Grundlage der Identität und der Attribute der Vorrichtung Zugang gewährt wird.

[2443] Beispiel 840 umfasst den Gegenstand von Beispiel 839. In Beispiel 840 fordert die Vorrichtung Tokens vom DNAP an.

[2444] Beispiel 841 umfasst den Gegenstand von Beispiel 839 oder 840. In Beispiel 841 gewähren die Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Netzdaten anders als über Peer-to-Peer.

[2445] Beispiel 842 umfasst den Gegenstand von einem der Beispiele 839 bis 841. In Beispiel 842 gewähren Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Daten auf einer Schicht einer Open-System-Interconnection-Schicht eines Netzes.

[2446] Beispiel 843 umfasst den Gegenstand von einem der Beispiele 839 bis 842. In Beispiel 843 hängt das Paket ein Token an und soll die Kombination aus dem Paket und dem Token zur Überprüfung zum DNAP gesendet werden, wobei das DNAP sowohl das Paket als auch das Token ablehnt, wenn festgestellt wird, dass das Token nicht vom DNAP akzeptiert wird.

[2447] Beispiel 844 umfasst den Gegenstand von einem der Beispiele 839 bis 843. In Beispiel 844 ist das Token für die Verwendung mit einer Schwellenanzahl von Paketen, einem Schwellenvolumen von Daten und einem Schwellenzeitraum gültig.

[2448] Beispiel 845 umfasst den Gegenstand von einem der Beispiele 839 bis 844. In Beispiel 845 speichert die Vorrichtung einen Transaktionsdatensatz von der Vorrichtung empfangener und gesendeter Transaktionen, wobei der Transaktionsdatensatz mit dem DNAP zu teilen ist.

[2449] Beispiel 846 umfasst den Gegenstand von einem der Beispiele 839 bis 845. In Beispiel 846 erzeugt die Vorrichtung Schlüssel zur Angabe des Ursprungs eines von der Vorrichtung gesendeten Pakets.

[2450] Beispiel 847 umfasst den Gegenstand von einem der Beispiele 839 bis 846. In Beispiel 847 ist die Vorrichtung eine Blockchain-fähige Vorrichtung und speichert die Vorrichtung alle von der Vorrichtung gesendeten und empfangenen Transaktionen in der Blockchain.

[2451] Beispiel 848 umfasst den Gegenstand von einem der Beispiele 839 bis 847. In Beispiel 848 werden Beschreibungen der Vorrichtungsattribute außerhalb einer Blockchain gespeichert.

[2452] Beispiel 849 umfasst ein Verfahren zur sicheren Kommunikation mit einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur sicheren Kommunikation mit einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Erzeugen einer Vorrichtungsidentität für eine Vorrichtung als Blockchain-Client, das Veröffentlichen einer Entdeckungsausstrahlungsnachricht von der Vorrichtung, das Bewerben der Vorrichtung, sich ansprechend darauf, dass die Vorrichtung auf der Grundlage der veröffentlichten Entdeckungsausstrahlungsnachricht eine Antwort vom DNAP empfängt, einem Dezentralisiertes-Netzzugangs-Proxy(DNAP)-Netz anzuschließen, das Beschreiben der Identität und der Attribute der Vorrichtung für das DNAP und das Senden eines Pakets von der Vorrichtung durch das Netz ansprechend darauf, dass der Vorrichtung durch das Netz auf der Grundlage der Identität und der Attribute der Vorrichtung Zugang gewährt wird.

[2453] Beispiel 850 umfasst den Gegenstand von Beispiel 849. In Beispiel 850 fordert die Vorrichtung Tokens vom DNAP an.

[2454] Beispiel 851 umfasst den Gegenstand von Beispiel 849 oder 850. In Beispiel 851 gewähren die Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Netzdaten anders als über Peer-to-Peer.

[2455] Beispiel 852 umfasst den Gegenstand von einem der Beispiele 849 bis 851. In Beispiel 852 gewähren Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Daten auf einer Schicht einer Open-System-Interconnection-Schicht eines Netzes.

[2456] Beispiel 853 umfasst den Gegenstand von einem der Beispiele 849 bis 852. In Beispiel 853 hängt das Paket ein Token an und soll die Kombination aus dem Paket und dem Token zur Überprüfung zum DNAP gesendet werden, wobei das DNAP sowohl das Paket als auch das Token ablehnt, wenn festgestellt wird, dass das Token nicht vom DNAP akzeptiert wird.

[2457] Beispiel 854 umfasst den Gegenstand von einem der Beispiele 849 bis 853. In Beispiel 854 ist das Token für die Verwendung mit einer Schwellenanzahl von Paketen, einem Schwellenvolumen von Daten und einem Schwellenzeitraum gültig.

[2458] Beispiel 855 umfasst den Gegenstand von einem der Beispiele 849 bis 854. In Beispiel 855 speichert die Vorrichtung einen Transaktionsdatensatz von der Vorrichtung empfangener und gesendeter Transaktionen, wobei der Transaktionsdatensatz mit dem DNAP zu teilen ist.

[2459] Beispiel 856 umfasst den Gegenstand von einem der Beispiele 849 bis 855. In Beispiel 856 erzeugt die Vorrichtung Schlüssel zur Angabe des Ursprungs eines von der Vorrichtung gesendeten Pakets.

[2460] Beispiel 857 umfasst den Gegenstand von einem der Beispiele 849 bis 856. In Beispiel 857 ist die Vorrichtung eine Blockchain-fähige Vorrichtung und speichert die Vorrichtung alle von der Vorrichtung gesendeten und empfangenen Transaktionen in der Blockchain.

[2461] Beispiel 858 umfasst den Gegenstand von einem der Beispiele 849 bis 857. In Beispiel 858 werden Beschreibungen der Vorrichtungsattribute außerhalb einer Blockchain gespeichert.

[2462] Beispiel 859 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Vorrichtungsidentität für eine Vorrichtung als Blockchain-Client zu erzeugen, eine Entdeckungsausstrahlungsnach-

richt von der Vorrichtung zu veröffentlichen, von der Vorrichtung eine Bewerbung vorzunehmen, sich ansprechend darauf, dass die Vorrichtung auf der Grundlage der veröffentlichten Entdeckungsausstrahlungsnachricht eine Antwort vom DNAP empfängt, einem Dezentralisiertes-Netzzugangs-Proxy(DNAP)-Netz anzuschließen, die Identität und Attribute der Vorrichtung für das DNAP zu beschreiben und ein Paket von der Vorrichtung ansprechend darauf durch das Netz zu senden, dass der Vorrichtung durch das Netz auf der Grundlage der Identität und der Attribute der Vorrichtung Zugang gewährt wird.

[2463] Beispiel 860 umfasst den Gegenstand von Beispiel 859. In Beispiel 860 fordert die Vorrichtung Tokens vom DNAP an.

[2464] Beispiel 861 umfasst den Gegenstand von Beispiel 859 oder 860. In Beispiel 861 gewähren die Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Netzdaten anders als über Peer-to-Peer.

[2465] Beispiel 862 umfasst den Gegenstand von einem der Beispiele 859 bis 861. In Beispiel 862 gewähren Tokens der Vorrichtung die Fähigkeit zum Senden und Empfangen von Daten auf einer Schicht einer Open-System-Interconnection-Schicht eines Netzes.

[2466] Beispiel 863 umfasst den Gegenstand von einem der Beispiele 859 bis 862. In Beispiel 863 hängt das Paket ein Token an und soll die Kombination aus dem Paket und dem Token zur Überprüfung zum DNAP gesendet werden, wobei das DNAP sowohl das Paket als auch das Token ablehnt, wenn festgestellt wird, dass das Token nicht vom DNAP akzeptiert wird.

[2467] Beispiel 864 umfasst den Gegenstand von einem der Beispiele 859 bis 863. In Beispiel 864 ist das Token für die Verwendung mit einer Schwellenanzahl von Paketen, einem Schwellenvolumen von Daten und einem Schwellenzeitraum gültig.

[2468] Beispiel 865 umfasst den Gegenstand von einem der Beispiele 859 bis 864. In Beispiel 865 speichert die Vorrichtung einen Transaktionsdatensatz von der Vorrichtung empfangener und gesendeter Transaktionen, wobei der Transaktionsdatensatz mit dem DNAP zu teilen ist.

[2469] Beispiel 866 umfasst den Gegenstand von einem der Beispiele 859 bis 865. In Beispiel 866 erzeugt die Vorrichtung Schlüssel zur Angabe des Ursprungs eines von der Vorrichtung gesendeten Pakets.

[2470] Beispiel 867 umfasst den Gegenstand von einem der Beispiele 859 bis 866. In Beispiel 867 ist die Vorrichtung eine Blockchain-fähige Vorrichtung und speichert die Vorrichtung alle von der Vorrichtung gesendeten und empfangenen Transaktionen in der Blockchain.

[2471] Beispiel 868 umfasst den Gegenstand von einem der Beispiele 859 bis 867. In Beispiel 868 werden Beschreibungen der Vorrichtungsattribute außerhalb einer Blockchain gespeichert.

[2472] Beispiel 869 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Vorrichtungsregistrator zum Registrieren einer Vorrichtung an einem ersten Netz durch ein Portal zu einem zweiten Netz, wobei das zweite Netz autorisiert ist, auf das erste Netz zuzugreifen, einen Vorrichtungsaufnehmer zum Aufnehmen einer Vorrichtung in eine Erlaubnisanleitung durch Zustimmung zu Verpflichtungen der Erlaubnisanleitung, einen Token-Antragsteller zur Anforderung eines Tokens unter Verwendung einer Funktion der Erlaubnisanleitung, wobei das Token die Vorrichtung als für den Zugang zum zweiten Netz authentifiziert identifiziert, und einen Anforderungsabsender zum Senden einer Authentifizierungsanforderung von der Vorrichtung zum ersten Netz, wobei das erste Netz die Authentifizierung ansprechend auf die Erkennung des Tokens bestätigt.

[2473] Beispiel 870 umfasst den Gegenstand von Beispiel 869. In Beispiel 870 umfasst die Vorrichtung einen Zahlungsaustausch mit einem Wallet im zweiten Netz.

[2474] Beispiel 871 umfasst den Gegenstand von Beispiel 869 oder 870. In Beispiel 871 führt die Anforderung für das Token dazu, dass eine Reservierung von Münzen in einer Abrechnungs-Blockchain einem auf einer Sidechain erzeugten Token entspricht.

[2475] Beispiel 872 umfasst den Gegenstand von einem der Beispiele 869 bis 871. In Beispiel 872 wird das Token ansprechend auf eine Authentifizierung der Vorrichtung durch Präsentieren des Tokens beim ersten Netz durch die Vorrichtung auf einer Sidechain verbraucht.

[2476] Beispiel 873 umfasst den Gegenstand von einem der Beispiele 869 bis 872. In Beispiel 873 wird eine Münze der Blockchain ansprechend auf die Erkennung, dass ein Token von einer Sidechain widerrufen und verbraucht wurde, freigegeben.

[2477] Beispiel 874 umfasst den Gegenstand von einem der Beispiele 869 bis 873. In Beispiel 874 umfasst das Beitreten zur Erlaubnisanleitung das Bereitstellen von Attributen der Vorrichtung zur Erlaubnisanleitung für ein Attributfilter von der Vorrichtung, um zu validieren, dass die Attribute der Vorrichtung im ersten Netz erlaubt sind.

[2478] Beispiel 875 umfasst den Gegenstand von einem der Beispiele 869 bis 874. In Beispiel 875 umfassen die Attribute ein Attribut eines Benutzerprofils, das aktiv ist, während die Vorrichtung der Erlaubnisanleitung beitrifft.

[2479] Beispiel 876 umfasst den Gegenstand von einem der Beispiele 869 bis 875. In Beispiel 876 zerstört sich das Token selbst als Form der Autorisierung der Vorrichtung, nachdem es verwendet wurde.

[2480] Beispiel 877 umfasst den Gegenstand von einem der Beispiele 869 bis 876. In Beispiel 877 wird die Vorrichtung auf der Grundlage der Authentifizierung beim ersten Netz, dass die Vorrichtung Berechtigungsnachweise hat, um auf das zweite Netz zuzugreifen, autorisiert, auf das erste Netz zuzugreifen.

[2481] Beispiel 878 umfasst den Gegenstand von einem der Beispiele 869 bis 877. In Beispiel 878 verstreicht die Autorisierung der Vorrichtung zur Verwendung des ersten Netzes auf der Grundlage der Anzahl der Zugriffe, des Datenvolumens, worauf durch das erste Netz zugegriffen wird, und der Zeit des gewährten Zugriffs.

[2482] Beispiel 879 umfasst ein Verfahren zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Registrieren einer Vorrichtung bei einem ersten Netz durch ein Portal zu einem zweiten Netz, wobei das zweite Netz berechtigt ist, auf das erste Netz zuzugreifen, das Hinzufügen einer Vorrichtung zu einer Erlaubnisanleitung durch Zustimmung zu Verpflichtungen der Erlaubnisanleitung, das Anfordern eines Tokens unter Verwendung einer Funktion der Erlaubnisanleitung, wobei das Token die Vorrichtung als für den Zugriff auf das zweite Netz authentifiziert identifiziert, und das Senden einer Authentifizierungsanforderung von der Vorrichtung zum ersten Netz, wobei das erste Netz die Authentifizierung ansprechend auf die Erkennung des Tokens bestätigt.

[2483] Beispiel 880 umfasst das Verfahren von Beispiel 879. In Beispiel 879 umfasst die Vorrichtung einen Zahlungsaustausch mit einem Wallet im zweiten Netz.

[2484] Beispiel 881 umfasst den Gegenstand von Beispiel 879 oder 880. In Beispiel 881 führt die Anforderung für das Token dazu, dass eine Reservierung von Münzen in einer Abrechnungs-Blockchain einem auf einer Sidechain erzeugten Token entspricht.

[2485] Beispiel 882 umfasst den Gegenstand von einem der Beispiele 879 bis 881. In Beispiel 882 wird das Token ansprechend auf eine Authentifizierung der Vorrichtung durch Präsentieren des Tokens beim ersten Netz durch die Vorrichtung auf einer Sidechain verbraucht.

[2486] Beispiel 883 umfasst den Gegenstand von einem der Beispiele 879 bis 882. In Beispiel 883 wird eine Münze der Blockchain ansprechend auf die Erkennung, dass ein Token von einer Sidechain widerrufen und verbraucht wurde, freigegeben.

[2487] Beispiel 884 umfasst den Gegenstand von einem der Beispiele 879 bis 883. In Beispiel 884 umfasst das Beitreten zur Erlaubnisanleitung das Bereitstellen von Attributen der Vorrichtung zur Erlaubnisanleitung für ein Attributfilter von der Vorrichtung, um zu validieren, dass die Attribute der Vorrichtung im ersten Netz erlaubt sind.

[2488] Beispiel 885 umfasst den Gegenstand von einem der Beispiele 879 bis 884. In Beispiel 885 umfassen die Attribute ein Attribut eines Benutzerprofils, das aktiv ist, während die Vorrichtung der Erlaubnisanleitung beitrifft.

[2489] Beispiel 886 umfasst den Gegenstand von einem der Beispiele 879 bis 885. In Beispiel 886 zerstört sich das Token selbst als Form der Autorisierung der Vorrichtung, nachdem es verwendet wurde.

[2490] Beispiel 887 umfasst den Gegenstand von einem der Beispiele 879 bis 886. In Beispiel 887 wird die Vorrichtung auf der Grundlage der Authentifizierung beim ersten Netz, dass die Vorrichtung Berechtigungsnachweise hat, um auf das zweite Netz zuzugreifen, autorisiert, auf das erste Netz zuzugreifen.

[2491] Beispiel 888 umfasst den Gegenstand von einem der Beispiele 879 bis 887. In Beispiel 888 verstreich die Autorisierung der Vorrichtung zur Verwendung des ersten Netzes auf der Grundlage der Anzahl der Zugriffe, des Datenvolumens, worauf durch das erste Netz zugegriffen wird, und der Zeit des gewährten Zugriffs.

[2492] Beispiel 889 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Vorrichtung durch ein Portal zu einem zweiten Netz beim ersten Netz zu registrieren, wobei das zweite Netz berechtigt ist, auf das erste Netz zuzugreifen, eine Vorrichtung durch Zustimmung zu Verpflichtungen der Erlaubnisanleitung zu einer Erlaubnisanleitung hinzuzufügen, ein Token unter Verwendung einer Funktion der Erlaubnisanleitung anzufordern, wobei das Token die Vorrichtung als für den Zugriff auf das zweite Netz authentifiziert identifiziert, und eine Authentifizierungsanforderung von der Vorrichtung zum ersten Netz zu senden, wobei das erste Netz die Authentifizierung ansprechend auf die Erkennung des Tokens bestätigt.

[2493] Beispiel 890 umfasst den Gegenstand von Beispiel 889. In Beispiel 890 umfasst die Vorrichtung einen Zahlungsaustausch mit einem Wallet im zweiten Netz.

[2494] Beispiel 891 umfasst den Gegenstand von Beispiel 889 oder 890. In Beispiel 891 führt die Anforderung für das Token dazu, dass eine Reservierung von Münzen in einer Abrechnungs-Blockchain einem auf einer Sidechain erzeugten Token entspricht.

[2495] Beispiel 892 umfasst den Gegenstand von einem der Beispiele 889 bis 891. In Beispiel 892 wird das Token ansprechend auf eine Authentifizierung der Vorrichtung durch Präsentieren des Tokens beim ersten Netz durch die Vorrichtung auf einer Sidechain verbraucht.

[2496] Beispiel 893 umfasst den Gegenstand von einem der Beispiele 889 bis 892. In Beispiel 893 wird eine Münze der Blockchain ansprechend auf die Erkennung, dass ein Token von einer Sidechain widerrufen und verbraucht wurde, freigegeben.

[2497] Beispiel 894 umfasst den Gegenstand von einem der Beispiele 889 bis 893. In Beispiel 894 umfasst das Beitreten zur Erlaubnisanleitung das Bereitstellen von Attributen der Vorrichtung zur Erlaubnisanleitung für ein Attributfilter von der Vorrichtung, um zu validieren, dass die Attribute der Vorrichtung im ersten Netz erlaubt sind.

[2498] Beispiel 895 umfasst den Gegenstand von einem der Beispiele 889 bis 894. In Beispiel 895 umfassen die Attribute ein Attribut eines Benutzerprofils, das aktiv ist, während die Vorrichtung der Erlaubnisanleitung beitrifft.

[2499] Beispiel 896 umfasst den Gegenstand von einem der Beispiele 889 bis 895. In Beispiel 896 zerstört sich das Token selbst als Form der Autorisierung der Vorrichtung, nachdem es verwendet wurde.

[2500] Beispiel 897 umfasst den Gegenstand von einem der Beispiele 889 bis 896. In Beispiel 897 wird die Vorrichtung auf der Grundlage der Authentifizierung beim ersten Netz durch das zweite Netz, dass die Vorrichtung Berechtigungsnachweise hat, um auf das zweite Netz zuzugreifen, autorisiert, auf das erste Netz zuzugreifen.

[2501] Beispiel 898 umfasst den Gegenstand von einem der Beispiele 889 bis 897. In Beispiel 898 verstreich die Autorisierung der Vorrichtung zur Verwendung des ersten Netzes auf der Grundlage der Anzahl der Zugriffe, des Datenvolumens, worauf durch das erste Netz zugegriffen wird, und der Zeit des gewährten Zugriffs.

[2502] Beispiel 899 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Identitätsverifizierer zum Verifizieren der Identität einer Authentifizierungsanforderung mit einer dezentralisierten Anwendungsprogramm-schnittstelle (API). Die Vorrichtung umfasst auch die von einem Fernauthentifizierungs-Einwähl-Benutzerdienst- bzw. Remote-Authentication-Dial-In-User-Service(RADIUS)-Client empfangene Authentifizierungsanforderung, wobei die dezentralisierte API die Identität durch Senden einer Anforderung an einen verteilten Ledger und Zurückgeben einer Antwort ansprechend auf den Empfang einer Identitätsverifizierungsantwort

vom verteilten Ledger verifizieren soll, und einen Antwortzurückgeber zum Zurückgeben einer Antwort auf die Authentifizierungsanforderung an den RADIUS-Client ansprechend auf den Empfang der Antwort von der dezentralisierten API, wobei der RADIUS-Client ansprechend auf eine Antwort zur authentifizierten Identität eine Transaktion vornimmt.

[2503] Beispiel 900 umfasst den Gegenstand von Beispiel 899. In Beispiel 900 umfasst die Transaktion einen Benutzernamen, ein Passwort und/oder Metadaten.

[2504] Beispiel 901 umfasst den Gegenstand von Beispiel 899 oder 900. In Beispiel 901 umfasst die Transaktion eine Werttransaktion.

[2505] Beispiel 902 umfasst den Gegenstand von einem der Beispiele 899 bis 901. In Beispiel 902 ist die Transaktion eine Kryptowährungstransaktion.

[2506] Beispiel 903 umfasst den Gegenstand von einem der Beispiele 899 bis 902. In Beispiel 903 umfasst die Authentifizierungsanforderung eine Anforderung einer Netzadresse.

[2507] Beispiel 904 umfasst den Gegenstand von einem der Beispiele 899 bis 903. In Beispiel 904 umfasst die Netzadresse einen voll qualifizierten Domainnamen für den RADIUS-Server und eine Internetprotokolladresse für den RADIUS-Server.

[2508] Beispiel 905 umfasst den Gegenstand von einem der Beispiele 899 bis 904. In Beispiel 905 verifiziert der RADIUS-Server einen öffentlichen Schlüssel durch Anfordern des Orts des öffentlichen Schlüssels von einer Blockchain.

[2509] Beispiel 906 umfasst den Gegenstand von einem der Beispiele 899 bis 905. In Beispiel 906 geschieht die Anforderung an einen RADIUS-Server außerhalb der Chain ansprechend darauf, dass ein RADIUS-Client eine Bestätigung einer authentifizierten Identität empfängt.

[2510] Beispiel 907 umfasst den Gegenstand von einem der Beispiele 899 bis 906. In Beispiel 907 führt der RADIUS-Server eine Protokollierung und Abrechnung von Anforderungen, die der RADIUS-Server empfängt, aus.

[2511] Beispiel 908 umfasst den Gegenstand von einem der Beispiele 899 bis 907. In Beispiel 908 wird die Antwort auf die Authentifizierungsanforderung als unveränderlicher Datensatz in einer Blockchain gespeichert.

[2512] Beispiel 909 umfasst ein Verfahren zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer Internet-der-Dinge(IoT)-Vorrichtung. Das Verfahren zur dezentralisierten Autorisierung, Authentifizierung und Abrechnung mit einer Internet-der-Dinge(IoT)-Vorrichtung umfasst das Verifizieren der Identität einer Authentifizierungsanforderung mit einem verteilten Ledger, wobei die Authentifizierungsanforderung von einem Fernauthentifizierungs-Einwähl-Benutzer-dienst(RADIUS)-Client empfangen wird, das Senden einer Anforderung zu einem RADIUS-Server ansprechend auf den Empfang einer positiven Identitätsverifizierungsantwort vom verteilten Ledger und das Zurückgeben einer Antwort auf die Authentifizierungsanforderung an den RADIUS-Client ansprechend auf den Empfang einer Antwort vom RADIUS-Server, wobei der RADIUS-Client ansprechend auf eine Antwort der authentifizierten Identität eine Transaktion mit dem RADIUS-Server vornimmt.

[2513] Beispiel 910 umfasst den Gegenstand von Beispiel 909. In Beispiel 910 umfasst die Transaktion einen Benutzernamen, ein Passwort und Metadaten.

[2514] Beispiel 911 umfasst den Gegenstand von Beispiel 909 oder 910. In Beispiel 911 umfasst die Transaktion eine Werttransaktion.

[2515] Beispiel 912 umfasst den Gegenstand von einem der Beispiele 909 bis 911. In Beispiel 912 ist die Transaktion eine Kryptowährungstransaktion.

[2516] Beispiel 913 umfasst den Gegenstand von einem der Beispiele 909 bis 912. In Beispiel 913 umfasst die Authentifizierungsanforderung eine Anforderung einer Netzadresse.

[2517] Beispiel 914 umfasst den Gegenstand von einem der Beispiele 909 bis 913. In Beispiel 914 umfasst die Netzadresse einen voll qualifizierten Domainnamen für den RADIUS-Server und eine Internetprotokolladresse für den RADIUS-Server.

[2518] Beispiel 915 umfasst den Gegenstand von einem der Beispiele 909 bis 914. In Beispiel 915 verifiziert der RADIUS-Server einen öffentlichen Schlüssel durch Anfordern des Orts des öffentlichen Schlüssels von einer Blockchain.

[2519] Beispiel 916 umfasst den Gegenstand von einem der Beispiele 909 bis 915. In Beispiel 916 geschieht die Anforderung an einen RADIUS-Server außerhalb der Chain ansprechend darauf, dass ein RADIUS-Client eine Bestätigung einer authentifizierten Identität empfängt.

[2520] Beispiel 917 umfasst den Gegenstand von einem der Beispiele 909 bis 916. In Beispiel 917 führt der RADIUS-Server eine Protokollierung und Abrechnung von Anforderungen, die der RADIUS-Server empfängt, aus.

[2521] Beispiel 918 umfasst den Gegenstand von einem der Beispiele 917 bis 917. In Beispiel 918 wird die Antwort auf die Authentifizierungsanforderung als unveränderlicher Datensatz in einer Blockchain gespeichert.

[2522] Beispiel 919 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Identität einer Authentifizierungsanforderung mit einem verteilten Ledger zu verifizieren, wobei die Authentifizierungsanforderung von einem Fernauthentifizierungs-Einwähl-Benutzerdienst(RADIUS)-Client empfangen wird, ansprechend auf den Empfang einer positiven Identitätsverifizierungsantwort vom verteilten Ledger eine Anforderung zu einem RADIUS-Server zu senden und ansprechend auf den Empfang einer Antwort vom RADIUS-Server eine Antwort auf die Authentifizierungsanforderung zum RADIUS-Client zurückzugeben, wobei der RADIUS-Client ansprechend auf eine Antwort der authentifizierten Identität eine Transaktion mit dem RADIUS-Server vornimmt.

[2523] Beispiel 920 umfasst den Gegenstand von Beispiel 919. In Beispiel 920 umfasst die Transaktion einen Benutzernamen, ein Passwort und Metadaten.

[2524] Beispiel 921 umfasst den Gegenstand von Beispiel 919 oder 920. In Beispiel 921 umfasst die Transaktion eine Werttransaktion.

[2525] Beispiel 922 umfasst den Gegenstand von einem der Beispiele 919 bis 921. In Beispiel 922 ist die Transaktion eine Kryptowährungstransaktion.

[2526] Beispiel 923 umfasst den Gegenstand von einem der Beispiele 919 bis 922. In Beispiel 923 umfasst die Authentifizierungsanforderung eine Anforderung einer Netzadresse.

[2527] Beispiel 924 umfasst den Gegenstand von einem der Beispiele 919 bis 923. In Beispiel 924 umfasst die Netzadresse einen voll qualifizierten Domainnamen für den RADIUS-Server und eine Internetprotokolladresse für den RADIUS-Server.

[2528] Beispiel 925 umfasst den Gegenstand von einem der Beispiele 919 bis 924. In Beispiel 925 verifiziert der RADIUS-Server einen öffentlichen Schlüssel durch Anfordern des Orts des öffentlichen Schlüssels von einer Blockchain.

[2529] Beispiel 926 umfasst den Gegenstand von einem der Beispiele 919 bis 925. In Beispiel 926 geschieht die Anforderung an einen RADIUS-Server außerhalb der Chain ansprechend darauf, dass ein RADIUS-Client eine Bestätigung einer authentifizierten Identität empfängt.

[2530] Beispiel 927 umfasst den Gegenstand von einem der Beispiele 919 bis 926. In Beispiel 927 führt der RADIUS-Server eine Protokollierung und Abrechnung von Anforderungen, die der RADIUS-Server empfängt, aus.

[2531] Beispiel 928 umfasst den Gegenstand von einem der Beispiele 919 bis 927. In Beispiel 928 wird die Antwort auf die Authentifizierungsanforderung als unveränderlicher Datensatz in einer Blockchain gespeichert.

[2532] Beispiel 929 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Vorrichtungsverbinder zum Verbinden einer Vorrichtung mit einem Netz einer dezentralisierten Datenbank, einen Namespaceentdecker zum Entdecken eines Namespace eines Knotens der dezentralisierten Datenbank, einen Partitionserzeuger zur Erzeugung einer geteilten Datenbankpartition ansprechend auf die Akzeptanz durch den Knoten, einen Dienstbekanntmacher zum Bekanntmachen eines Dienstes für die dezentralisierte Datenbank und einen Datenweiterleiter zum Weiterleiten während der Ausführung eines Dienstes empfangener und erzeugter Daten zwischen einer privaten Datenbankpartition und einer geteilten Datenbankpartition.

[2533] Beispiel 930 umfasst den Gegenstand von Beispiel 929. In Beispiel 930 ist die geteilte Datenbankpartition permissioniert und verschlüsselt.

[2534] Beispiel 931 umfasst den Gegenstand von Beispiel 929 oder 930. In Beispiel 931 werden Kopien in einer geteilten Datenbankpartition gespeicherter Daten ansprechend darauf zu einem zweiten Knoten repliziert, dass der zweite Knoten Vorrechte präsentiert, welche die Berechtigung des zweiten Knotens, die Daten zu kopieren, angeben.

[2535] Beispiel 932 umfasst den Gegenstand von einem der Beispiele 929 bis 931. In Beispiel 932 installiert die Vorrichtung dezentralisierte Datenbanksoftware ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2536] Beispiel 933 umfasst den Gegenstand von einem der Beispiele 929 bis 932. In Beispiel 933 erzeugt die Vorrichtung eine geteilte Datenbankpartition ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2537] Beispiel 934 umfasst den Gegenstand von einem der Beispiele 929 bis 933. In Beispiel 934 fordert die Vorrichtung den Beitritt zur dezentralisierten Datenbank ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an.

[2538] Beispiel 935 umfasst den Gegenstand von einem der Beispiele 929 bis 934. In Beispiel 935 repliziert die Vorrichtung eine geteilte Knotenpartition zur Speicherung in der geteilten Datenbankpartition ansprechend auf die Erzeugung der geteilten Datenbankpartition.

[2539] Beispiel 936 umfasst den Gegenstand von einem der Beispiele 929 bis 935. In Beispiel 936 werden die Daten in der geteilten Partition zur Speicherung in einer geteilten Knotenpartition ansprechend darauf repliziert, dass die Daten zur geteilten Datenbankpartition weitergeleitet werden.

[2540] Beispiel 937 umfasst den Gegenstand von einem der Beispiele 929 bis 936. In Beispiel 937 empfängt die Vorrichtung die Akzeptanz an der dezentralisierten Datenbank ansprechend darauf, dass der Knoten für die Akzeptanz der Vorrichtung stimmt.

[2541] Beispiel 938 umfasst den Gegenstand von einem der Beispiele 929 bis 937. In Beispiel 938 wird die Vorrichtung ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an der dezentralisierten Datenbank akzeptiert.

[2542] Beispiel 939 umfasst ein Verfahren zum Beitritt zu einer dezentralisierten Datenbank. Das Verfahren zum Beitritt zu einer dezentralisierten Datenbank umfasst das Verbinden einer Vorrichtung mit einem Netz einer dezentralisierten Datenbank, das Entdecken eines Namespace eines Knotens der dezentralisierten Datenbank, das Erzeugen einer geteilten Datenbankpartition ansprechend auf die Akzeptanz durch den Knoten, das Bekanntmachen eines Dienstes für die dezentralisierte Datenbank und das Übertragen während der Ausführung eines Dienstes empfangener und erzeugter Daten zwischen einer privaten Datenbankpartition und einer geteilten Datenbankpartition.

[2543] Beispiel 940 umfasst den Gegenstand von Beispiel 939. In Beispiel 940 ist die geteilte Datenbankpartition permissioniert und verschlüsselt.

[2544] Beispiel 941 umfasst den Gegenstand von Beispiel 939 oder 940. In Beispiel 941 werden Kopien in einer geteilten Datenbankpartition gespeicherter Daten ansprechend darauf zu einem zweiten Knoten repliziert, dass der zweite Knoten Vorrechte präsentiert, welche die Berechtigung des zweiten Knotens, die Daten zu kopieren, angeben.

[2545] Beispiel 942 umfasst den Gegenstand von einem der Beispiele 939 bis 941. In Beispiel 942 installiert die Vorrichtung dezentralisierte Datenbanksoftware ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2546] Beispiel 943 umfasst den Gegenstand von einem der Beispiele 939 bis 942. In Beispiel 943 erzeugt die Vorrichtung eine geteilte Datenbankpartition ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2547] Beispiel 944 umfasst den Gegenstand von einem der Beispiele 939 bis 943. In Beispiel 944 fordert die Vorrichtung den Beitritt zur dezentralisierten Datenbank ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an.

[2548] Beispiel 945 umfasst den Gegenstand von einem der Beispiele 939 bis 944. In Beispiel 945 repliziert die Vorrichtung eine geteilte Knotenpartition zur Speicherung in der geteilten Datenbankpartition ansprechend auf die Erzeugung der geteilten Datenbankpartition.

[2549] Beispiel 946 umfasst den Gegenstand von einem der Beispiele 939 bis 945. In Beispiel 946 werden die Daten in der geteilten Partition zur Speicherung in einer geteilten Knotenpartition ansprechend darauf repliziert, dass die Daten zur geteilten Datenbankpartition weitergeleitet werden.

[2550] Beispiel 947 umfasst den Gegenstand von einem der Beispiele 939 bis 946. In Beispiel 947 empfängt die Vorrichtung die Akzeptanz an der dezentralisierten Datenbank ansprechend darauf, dass der Knoten für die Akzeptanz der Vorrichtung stimmt.

[2551] Beispiel 948 umfasst den Gegenstand von einem der Beispiele 939 bis 947. In Beispiel 948 wird die Vorrichtung ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an der dezentralisierten Datenbank akzeptiert.

[2552] Beispiel 949 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Vorrichtung mit einem Netz einer dezentralisierten Datenbank zu verbinden, einen Namespace eines Knotens der dezentralisierten Datenbank zu entdecken, ansprechend auf die Akzeptanz durch den Knoten eine geteilte Datenbankpartition zu erzeugen, der dezentralisierten Datenbank einen Dienst bekanntzumachen und ansprechend auf die Ausführung eines Dienstes empfangene und erzeugte Daten zwischen einer privaten Datenbankpartition und einer geteilten Datenbankpartition zu übertragen.

[2553] Beispiel 950 umfasst den Gegenstand von Beispiel 949. In Beispiel 950 ist die geteilte Datenbankpartition permissioniert und verschlüsselt.

[2554] Beispiel 951 umfasst den Gegenstand von Beispiel 949 oder 950. In Beispiel 951 werden Kopien in einer geteilten Datenbankpartition gespeicherter Daten ansprechend darauf zu einem zweiten Knoten repliziert, dass der zweite Knoten Vorrechte präsentiert, welche die Berechtigung des zweiten Knotens, die Daten zu kopieren, angeben.

[2555] Beispiel 952 umfasst den Gegenstand von einem der Beispiele 949 bis 951. In Beispiel 952 installiert die Vorrichtung dezentralisierte Datenbanksoftware ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2556] Beispiel 953 umfasst den Gegenstand von einem der Beispiele 949 bis 952. In Beispiel 953 erzeugt die Vorrichtung eine geteilte Datenbankpartition ansprechend auf die Verbindung mit dem Netz einer dezentralisierten Datenbank.

[2557] Beispiel 954 umfasst den Gegenstand von einem der Beispiele 949 bis 954. In Beispiel 954 fordert die Vorrichtung den Beitritt zur dezentralisierten Datenbank ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an.

[2558] Beispiel 955 umfasst den Gegenstand von einem der Beispiele 949 bis 954. In Beispiel 955 repliziert die Vorrichtung eine geteilte Knotenpartition zur Speicherung in der geteilten Datenbankpartition ansprechend auf die Erzeugung der geteilten Datenbankpartition.

[2559] Beispiel 956 umfasst den Gegenstand von einem der Beispiele 949 bis 955. In Beispiel 956 werden die Daten in der geteilten Partition zur Speicherung in einer geteilten Knotenpartition ansprechend darauf repliziert, dass die Daten zur geteilten Datenbankpartition weitergeleitet werden.

[2560] Beispiel 957 umfasst den Gegenstand von einem der Beispiele 949 bis 956. In Beispiel 957 empfängt die Vorrichtung die Akzeptanz an der dezentralisierten Datenbank ansprechend darauf, dass der Knoten für die Akzeptanz der Vorrichtung stimmt.

[2561] Beispiel 958 umfasst den Gegenstand von einem der Beispiele 949 bis 957. In Beispiel 958 wird die Vorrichtung ansprechend auf die Entdeckung des Namespace des Knotens der dezentralisierten Datenbank an der dezentralisierten Datenbank akzeptiert.

[2562] Beispiel 959 umfasst eine Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz. Die Vorrichtung zur Verwendung in einem Internet-der-Dinge(IoT)-Netz umfasst einen Berechtigungsnachweiserteiler zur Erteilung eines Berechtigungsnachweises an eine Aufrufer-Entität, wobei der Berechtigungsnachweis mehrere Schichten einer Autorisierungsstruktur umfasst, und einen Objekt-Entitäts-Bereitsteller zur Bereitstellung einer Zugriffssteuerungsliste, die eine Referenz zu einem Zielobjekt und eine Erlaubnis spezifiziert, für eine Objekt-Entität. Die Vorrichtung umfasst auch einen Berechtigungsnachweispräsentierer zum Präsentieren eines Autorisierungsberechtigungsnachweises für die Objekt-Entität und einen Zugriffssteuerungslisten-Richtlinien-anwender zum Anwenden einer Zugriffssteuerungslisten-Richtlinie, um auf der Grundlage eines Vergleichs, ob der Berechtigungsnachweis die Aufrufer-Entität überlappt, ob das Zielobjekt eine Anforderung überlappt, ob eine Anzahl von Vorrichtungsschichtidentifikationen mit einer Anzahl von Berechtigungsnachweisidentifikationen übereinstimmt und ob eine Anzahl von Zielschichtidentifikationen mit einer Anzahl von Anforderungsschichtidentifikationen übereinstimmt, festzustellen, ob ein Zugriff für eine IoT-Vorrichtung erlaubt ist.

[2563] Beispiel 960 umfasst den Gegenstand von Beispiel 959. In Beispiel 960 ist der Berechtigungsnachweis eine sechsschichtige Erlaubnis.

[2564] Beispiel 961 umfasst den Gegenstand von Beispiel 959 oder 960. In Beispiel 961 umfasst die sechsschichtige Erlaubnis eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht.

[2565] Beispiel 962 umfasst den Gegenstand von einem der Beispiele 959 bis 961. In Beispiel 962 umfassen die mehreren Schichten eine Plattformschicht, um eine physikalische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2566] Beispiel 963 umfasst den Gegenstand von einem der Beispiele 959 bis 962. In Beispiel 963 umfassen die mehreren Schichten eine Vorrichtungsschicht, um eine logische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2567] Beispiel 964 umfasst den Gegenstand von einem der Beispiele 959 bis 963. In Beispiel 964 umfassen die mehreren Schichten eine Sammelschicht für eine logische Struktur einer Ressource, wobei die Ressource eine logische Struktur für einen Datensatz umfasst, wobei der Datensatz eine logische Struktur einer Eigenschaft umfasst und wobei die Eigenschaft eine atomare Datenstruktur und eine komplexe Datenstruktur umfasst.

[2568] Beispiel 965 umfasst den Gegenstand von einem der Beispiele 959 bis 964. In Beispiel 965 ist die Eigenschaft eine komplexe Datenstruktur und ist die komplexe Datenstruktur für eine Struktur vorgesehen, die unter Verwendung einer Datenmodelliersprache definierbar ist.

[2569] Beispiel 966 umfasst den Gegenstand von einem der Beispiele 959 bis 965. In Beispiel 966 umfasst die Eigenschaft eine atomare Datenstruktur und ist die atomare Datenstruktur eine Zeichenkette, eine Zahl und ein Datum.

[2570] Beispiel 967 umfasst den Gegenstand von einem der Beispiele 959 bis 966. In Beispiel 967 wird der Autorisierungsberechtigungsnachweis für die Objekt-Entität durch Beschränkungen beschränkt, die dem Objekt durch die Physikalität der Objektdaten auf der Grundlage einer Erzeugen-, Lesen-, Aktualisieren-, Löschen- und Benachrichtigen- bzw. Create, Read, Update, Delete, and Notify(CRUDN)-Lebenszyklusbenachrichtigung auferlegt werden.

[2571] Beispiel 968 umfasst den Gegenstand von einem der Beispiele 959 bis 967. In Beispiel 968 gibt der Berechtigungsnachweis die Installation durch einen Hersteller an.

[2572] Beispiel 969 umfasst ein Verfahren zur Zugriffssteuerung bei einem IoT-Objekt. Das Verfahren zur Zugriffssteuerung bei einem IoT-Objekt umfasst das Ausgeben eines Berechtigungsnachweises an eine Aufrufer-Entität, wobei der Berechtigungsnachweis mehrere Schichten der Autorisierungsstruktur umfasst, das Versehen einer Objekt-Entität mit einer Zugriffssteuerungsliste, die eine Referenz für ein Zielobjekt und eine Erlaubnis spezifiziert, das Präsentieren eines Autorisierungsberechtigungsnachweises für die Objekt-Entität und das Anwenden einer Zugriffssteuerungslisten-Richtlinie, um auf der Grundlage eines Vergleichs, ob der Berechtigungsnachweis die Aufrufer-Entität überlappt, ob das Zielobjekt eine Anforderung überlappt, ob eine Anzahl von Vorrichtungsschichtidentifikationen mit einer Anzahl von Berechtigungsnachweisidentifikationen übereinstimmt und ob eine Anzahl von Zielschichtidentifikationen mit einer Anzahl von Anforderungsschichtidentifikationen übereinstimmt, festzustellen, ob ein Zugriff für eine IoT-Vorrichtung erlaubt ist.

[2573] Beispiel 970 umfasst den Gegenstand von Beispiel 969. In Beispiel 970 ist der Berechtigungsnachweis eine sechsschichtige Erlaubnis.

[2574] Beispiel 971 umfasst den Gegenstand von Beispiel 969 oder 970. In Beispiel 971 umfasst die sechsschichtige Erlaubnis eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht.

[2575] Beispiel 972 umfasst den Gegenstand von einem der Beispiele 969 bis 971. In Beispiel 972 umfassen die mehreren Schichten eine Plattformschicht, um eine physikalische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2576] Beispiel 973 umfasst den Gegenstand von einem der Beispiele 969 bis 972. In Beispiel 973 umfassen die mehreren Schichten eine Vorrichtungsschicht, um eine logische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2577] Beispiel 974 umfasst den Gegenstand von einem der Beispiele 969 bis 973. In Beispiel 974 umfassen die mehreren Schichten eine Sammelschicht für eine logische Struktur einer Ressource, wobei die Ressource eine logische Struktur für einen Datensatz umfasst, wobei der Datensatz eine logische Struktur einer Eigenschaft umfasst und wobei die Eigenschaft eine atomare Datenstruktur und eine komplexe Datenstruktur umfasst.

[2578] Beispiel 975 umfasst den Gegenstand von einem der Beispiele 969 bis 974. In Beispiel 975 ist die Eigenschaft eine komplexe Datenstruktur und ist die komplexe Datenstruktur für eine Struktur vorgesehen, die unter Verwendung einer Datenmodelliersprache definierbar ist.

[2579] Beispiel 976 umfasst den Gegenstand von einem der Beispiele 969 bis 975. In Beispiel 976 umfasst die Eigenschaft eine atomare Datenstruktur und ist die atomare Datenstruktur eine Zeichenkette, eine Zahl und ein Datum.

[2580] Beispiel 977 umfasst den Gegenstand von einem der Beispiele 969 bis 976. In Beispiel 977 wird der Autorisierungsberechtigungsnachweis für die Objekt-Entität durch Beschränkungen beschränkt, die dem Objekt durch die Physikalität der Objektdaten auf der Grundlage einer Erzeugen-, Lesen-, Aktualisieren-, Löschen- und Benachrichtigen- bzw. Create, Read, Update, Delete, and Notify(CRUDN)-Lebenszyklusbenachrichtigung auferlegt werden.

[2581] Beispiel 978 umfasst den Gegenstand von einem der Beispiele 969 bis 977. In Beispiel 978 gibt der Berechtigungsnachweis die Installation durch einen Hersteller an.

[2582] Beispiel 979 umfasst ein nichtflüchtiges maschinenlesbares Medium. Das nichtflüchtige maschinenlesbare Medium umfasst Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einer Aufrufer-Entität einen Berechtigungsnachweis zu erteilen, wobei der Berechtigungsnachweis eine Anzahl der Schichten der Autorisierungsstruktur aufweist, eine Objekt-Entität mit einer Zugriffssteuerungsliste zu versehen, die eine Referenz zu einem Zielobjekt und eine Erlaubnis spezifiziert, der Objekt-Entität einen Autorisierungsberechtigungsnachweis zu präsentieren und eine Zugriffssteuerungslisten-Richtlinie anzuwenden, um auf der Grundlage eines Vergleichs, ob der Berechtigungsnachweis die Aufrufer-Entität überlappt, ob das Zielobjekt eine Anforderung überlappt, ob eine Anzahl von Vorrichtungsschichtidentifikationen mit einer Anzahl von Be-

rechtigungsnachweisidentifikationen übereinstimmt und ob eine Anzahl von Zielschichtidentifikationen mit einer Anzahl von Anforderungsschichtidentifikationen übereinstimmt, festzustellen, ob ein Zugriff für eine IoT-Vorrichtung erlaubt ist.

[2583] Beispiel 980 umfasst den Gegenstand von Beispiel 979. In Beispiel 980 ist der Berechtigungsnachweis eine sechsschichtige Erlaubnis.

[2584] Beispiel 981 umfasst den Gegenstand von Beispiel 979 oder 980. In Beispiel 981 umfasst die sechsschichtige Erlaubnis eine Plattformschicht, eine Vorrichtungsschicht, eine Sammelschicht, eine Ressourcenschicht, eine Aufzeichnungsschicht und eine Eigenschaftsschicht.

[2585] Beispiel 982 umfasst den Gegenstand von einem der Beispiele 979 bis 981. In Beispiel 982 umfassen die mehreren Schichten eine Plattformschicht, um eine physikalische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2586] Beispiel 983 umfasst den Gegenstand von einem der Beispiele 979 bis 982. In Beispiel 983 umfassen die mehreren Schichten eine Vorrichtungsschicht, um eine logische Instanz eines Computers widerzuspiegeln, und sie weisen Rechen-, Netzwerk-, Speicher-, Mess- und Betätigungsfähigkeiten auf.

[2587] Beispiel 984 umfasst den Gegenstand von einem der Beispiele 979 bis 983. In Beispiel 984 umfassen die mehreren Schichten eine Sammelschicht für eine logische Struktur einer Ressource, wobei die Ressource eine logische Struktur für einen Datensatz umfasst, wobei der Datensatz eine logische Struktur einer Eigenschaft umfasst und wobei die Eigenschaft eine atomare Datenstruktur und eine komplexe Datenstruktur umfasst.

[2588] Beispiel 985 umfasst den Gegenstand von einem der Beispiele 979 bis 984. In Beispiel 985 ist die Eigenschaft eine komplexe Datenstruktur und ist die komplexe Datenstruktur für eine Struktur vorgesehen, die unter Verwendung einer Datenmodelliersprache definierbar ist.

[2589] Beispiel 986 umfasst den Gegenstand von einem der Beispiele 979 bis 985. In Beispiel 986 umfasst die Eigenschaft eine atomare Datenstruktur und ist die atomare Datenstruktur eine Zeichenkette, eine Zahl und ein Datum.

[2590] Beispiel 987 umfasst den Gegenstand von einem der Beispiele 979 bis 986. In Beispiel 987 wird der Autorisierungsberechtigungsnachweis für die Objekt-Entität durch Beschränkungen beschränkt, die dem Objekt durch die Physikalität der Objektdaten auf der Grundlage einer Erzeugen-, Lesen-, Aktualisieren-, Löschen- und Benachrichtigen- bzw. Create, Read, Update, Delete, and Notify(CRUDN)-Lebenszyklusbenachrichtigung auferlegt werden.

[2591] Beispiel 988 umfasst den Gegenstand von einem der Beispiele 979 bis 987. In Beispiel 988 gibt der Berechtigungsnachweis die Installation durch einen Hersteller an.

[2592] Beispiel 989 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst auch einen Ressourcenhardwarekomponentenidentifizierer zum Identifizieren einer von der IoT-Vorrichtung gesteuerten Ressourcenhardwarekomponente, wobei die Ressourcenhardwarekomponente eine Fähigkeitsschwelle aufweist, einen Prozessor zum Verarbeiten einer von einem externen Modul empfangenen Angabe einer Anforderung externer Modulhardware, einen Externes-Modul-Vergleicher zum Vergleichen der Anforderung externer Modulhardware mit der Fähigkeitsschwelle der Ressourcenhardwarekomponente der IoT-Vorrichtung und einen Sender zum Senden eines Deaktivierungssignals zum externen Modul ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt.

[2593] Beispiel 990 umfasst den Gegenstand von Beispiel 989. In Beispiel 990 sendet die IoT-Vorrichtung ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, eine Anforderung zu einer Master-Vorrichtung, wobei die Anforderung an die Master-Vorrichtung fordert, dass eine zweite Ressourcenhardwarekomponente für die Steuerung durch die IoT-Vorrichtung zugewiesen wird.

[2594] Beispiel 991 umfasst den Gegenstand von Beispiel 989 oder 990. In Beispiel 991 umfasst die IoT-Vorrichtung eine zweite Ressourcenhardwarekomponente, die von der IoT gesteuert wird, wobei die erste

Ressourcenhardwarekomponente und die zweite Ressourcenhardwarekomponente gepoolt werden können, so dass die Fähigkeitsschwelle die Summe der Fähigkeitsschwellen der ersten Ressourcenhardware und der zweiten Ressourcenhardware ist.

[2595] Beispiel 992 umfasst den Gegenstand von einem der Beispiele 989 bis 991. In Beispiel 992 umfasst das externe Modul eine Modulressource, die mit einer ersten Ressourcenhardwarekomponente zur Verwendung durch die IoT-Vorrichtung zu poolen ist.

[2596] Beispiel 993 umfasst den Gegenstand von einem der Beispiele 989 bis 992. In Beispiel 993 umfasst die Ressourcenhardwarekomponente eine Stromquelle, eine Verarbeitungsressource, eine integrierte Kommunikationskomponente, einen Kontextsensor und ein Kontextbetätigungselement, eine Signalaufbereitungsschaltung, eine flüchtige Speicherressource oder eine permanente Speicherressource.

[2597] Beispiel 994 umfasst den Gegenstand von einem der Beispiele 989 bis 993. In Beispiel 994 umfasst die Fähigkeitsschwelle eine minimale Funktionskompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine minimale Fähigkeit zur Zusammenarbeit angegeben wird, und eine vollständige Kompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine Fähigkeit angegeben wird, bei den höchsten Fähigkeiten des externen Moduls zu funktionieren.

[2598] Beispiel 995 umfasst den Gegenstand von einem der Beispiele 989 bis 994. In Beispiel 995 soll die IoT-Vorrichtung eine nicht erfüllte Kapazitätsschwelle durch Aktivieren eines sichtbaren Indikators angeben.

[2599] Beispiel 996 umfasst den Gegenstand von einem der Beispiele 989 bis 995. In Beispiel 996 soll die IoT-Vorrichtung das externe Modul ansprechend auf die Erfüllung der Fähigkeitsschwelle der Steuerung durch die IoT-Vorrichtung unterwerfen.

[2600] Beispiel 997 umfasst den Gegenstand von einem der Beispiele 989 bis 996. In Beispiel 997 soll die IoT-Vorrichtung ansprechend darauf, dass die Lebensdauer des externen Moduls kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung eines aktualisierten externen Moduls senden.

[2601] Beispiel 998 umfasst den Gegenstand von einem der Beispiele 989 bis 997. In Beispiel 998 soll die IoT-Vorrichtung ansprechend darauf, dass die Lebensdauer der Ressourcenhardwarekomponente kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung einer aktualisierten Ressourcenhardwarekomponente senden.

[2602] Beispiel 999 umfasst ein Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur Zuordnung von Ressourcen und Anforderungen selbstbeschreibender Hardware. Das Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur Zuordnung von Ressourcen und Anforderungen selbstbeschreibender Hardware umfasst das Identifizieren einer von der IoT-Vorrichtung gesteuerten Ressourcenhardwarekomponente, wobei die Ressourcenhardwarekomponente eine Fähigkeitsschwelle aufweist, das Verarbeiten einer von einem externen Modul empfangenen Angabe einer Anforderung externer Modulhardware, das Vergleichen der Anforderung externer Modulhardware mit der Fähigkeitsschwelle der Ressourcenhardwarekomponente der IoT-Vorrichtung und das Senden eines Deaktivierungssignals zum externen Modul ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt.

[2603] Beispiel 1000 umfasst den Gegenstand von Beispiel 999. In Beispiel 1000 umfasst das Verfahren das Senden einer Anforderung zu einer Master-Vorrichtung ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, wobei die Anforderung an die Master-Vorrichtung fordert, dass eine zweite Ressourcenhardwarekomponente für die Steuerung durch die IoT-Vorrichtung zugewiesen wird.

[2604] Beispiel 1001 umfasst den Gegenstand von Beispiel 999 oder 1000. In Beispiel 1001 umfasst das Verfahren eine zweite Ressourcenhardwarekomponente, die von der IoT-Vorrichtung gesteuert wird, wobei die erste Ressourcenhardwarekomponente und die zweite Ressourcenhardwarekomponente gepoolt werden können, so dass die Fähigkeitsschwelle die Summe der Fähigkeitsschwellen der ersten Ressourcenhardware und der zweiten Ressourcenhardware ist.

[2605] Beispiel 1002 umfasst den Gegenstand von einem der Beispiele 999 bis 1001. In Beispiel 1002 umfasst das externe Modul eine Modulressource, die mit einer ersten Ressourcenhardwarekomponente zur Verwendung durch die IoT-Vorrichtung zu poolen ist.

[2606] Beispiel 1003 umfasst den Gegenstand von einem der Beispiele 999 bis 1002. In Beispiel 1003 umfasst die Ressourcenhardwarekomponente eine Stromquelle, eine Verarbeitungsressource, eine integrierte Kommunikationskomponente, einen Kontextsensor und ein Kontextbetätigungselement, eine Signalaufbereitungsschaltung, eine flüchtige Speicherressource oder eine permanente Speicherressource.

[2607] Beispiel 1004 umfasst den Gegenstand von einem der Beispiele 999 bis 1003. In Beispiel 1004 umfasst die Fähigkeitsschwelle eine minimale Funktionskompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine minimale Fähigkeit zur Zusammenarbeit angegeben wird, und eine vollständige Kompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine Fähigkeit angegeben wird, bei den höchsten Fähigkeiten des externen Moduls zu funktionieren.

[2608] Beispiel 1005 umfasst den Gegenstand von einem der Beispiele 999 bis 1004. In Beispiel 1005 umfasst das Verfahren die Angabe einer nicht erfüllten Kapazitätsschwelle durch Aktivieren eines sichtbaren Indikators.

[2609] Beispiel 1006 umfasst den Gegenstand von einem der Beispiele 999 bis 1005. In Beispiel 1006 umfasst das Verfahren das Bringen des externen Moduls unter die Kontrolle der IoT-Vorrichtung ansprechend darauf, dass die Fähigkeitsschwelle erfüllt wird.

[2610] Beispiel 1007 umfasst den Gegenstand von einem der Beispiele 999 bis 1006. In Beispiel 1007 wird ansprechend darauf, dass die Lebenszeit des externen Moduls kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung eines aktualisierten externen Moduls gesendet.

[2611] Beispiel 1008 umfasst den Gegenstand von einem der Beispiele 999 bis 1007. In Beispiel 1008 wird ansprechend darauf, dass die Lebenszeit einer Ressourcenhardwarekomponente kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung einer aktualisierten Ressourcenhardwarekomponente gesendet.

[2612] Beispiel 1009 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine von einer IoT-Vorrichtung gesteuerte Ressourcenhardwarekomponente zu identifizieren, wobei die Ressourcenhardwarekomponente eine Fähigkeitsschwelle aufweist, eine empfangene Angabe einer Anforderung externer Modulhardware von einem externen Modul zu verarbeiten, die Anforderung externer Modulhardware mit der Fähigkeitsschwelle der Ressourcenhardwarekomponente der IoT-Vorrichtung zu vergleichen und ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, ein Deaktivierungssignal zum externen Modul zu senden.

[2613] Beispiel 1010 umfasst den Gegenstand von Beispiel 1009. In Beispiel 1010 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ansprechend darauf, dass die Anforderung externer Modulhardware die Fähigkeitsschwelle der Ressourcenhardwarekomponente nicht erfüllt, eine Anforderung zu einer Master-Vorrichtung zu senden, wobei die Anforderung an die Master-Vorrichtung fordert, dass eine zweite Ressourcenhardwarekomponente für die Steuerung durch die IoT-Vorrichtung zugewiesen wird.

[2614] Beispiel 1011 umfasst den Gegenstand von Beispiel 1009 oder 1010. In Beispiel 1011 umfasst das nichtflüchtige maschinenlesbare Medium eine zweite von der IoT-Vorrichtung gesteuerte Ressourcenhardwarekomponente, wobei die erste Ressourcenhardwarekomponente und die zweite Ressourcenhardwarekomponente gepoolt werden können, so dass die Fähigkeitsschwelle die Summe der Fähigkeitsschwellen der ersten Ressourcenhardware und der zweiten Ressourcenhardware ist.

[2615] Beispiel 1012 umfasst den Gegenstand von einem der Beispiele 1009 bis 1011. In Beispiel 1012 umfasst das externe Modul eine Modulressource, die mit einer ersten Ressourcenhardwarekomponente zur Verwendung durch die IoT-Vorrichtung zu poolen ist.

[2616] Beispiel 1013 umfasst den Gegenstand von einem der Beispiele 1009 bis 1012. In Beispiel 1013 umfasst die Ressourcenhardwarekomponente eine Stromquelle, eine Verarbeitungsressource, eine integrierte Kommunikationskomponente, einen Kontextsensor und ein Kontextbetätigungselement, eine Signalaufbereitungsschaltung, eine flüchtige Speicherressource oder eine permanente Speicherressource.

[2617] Beispiel 1014 umfasst den Gegenstand von einem der Beispiele 1009 bis 1013. In Beispiel 1014 umfasst die Fähigkeitsschwelle eine minimale Funktionskompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine minimale Fähigkeit zur Zusammenarbeit angegeben wird, und eine vollständige Kompatibilität zwischen der Ressourcenhardwarekomponente und dem externen Modul, wodurch eine Fähigkeit angegeben wird, bei den höchsten Fähigkeiten des externen Moduls zu funktionieren.

[2618] Beispiel 1015 umfasst den Gegenstand von einem der Beispiele 1009 bis 1014. In Beispiel 1015 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine nicht erfüllte Fähigkeitsschwelle durch Aktivieren eines sichtbaren Indikators anzugeben.

[2619] Beispiel 1016 umfasst den Gegenstand von einem der Beispiele 1009 bis 1015. In Beispiel 1016 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, das externe Modul ansprechend darauf, dass die Fähigkeitsschwelle erfüllt wird, unter die Steuerung der IoT-Vorrichtung zu stellen.

[2620] Beispiel 1017 umfasst den Gegenstand von einem der Beispiele 1009 bis 1016. In Beispiel 1017 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ansprechend darauf, dass die Lebenszeit des externen Moduls kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung eines aktualisierten externen Moduls zu senden.

[2621] Beispiel 1018 umfasst den Gegenstand von einem der Beispiele 1009 bis 1017. In Beispiel 1018 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ansprechend darauf, dass die Lebenszeit einer Ressourcenhardwarekomponente kürzer als die Betriebsdauer der IoT-Vorrichtung ist, eine Anforderung einer aktualisierten Ressourcenhardwarekomponente zu übertragen.

[2622] Beispiel 1019 umfasst eine Vorrichtung zur Anwendungsenergieberechnung. Die Vorrichtung zur Anwendungsenergieberechnung umfasst einen Energieressourcenspezifizierer zum Spezifizieren von Energieressourcen für eine Vorrichtung, die dafür ausgelegt ist, eine Anzahl von Modulen mit Strom zu versorgen und zu steuern, und einen Energiebedarfsspezifizierer zum Spezifizieren des Energiebedarfs für die mehreren Module. Die Vorrichtung umfasst auch einen Anwendungszerleger zum Zerlegen einer Anwendung in Aufgaben, die, wenn sie aktiviert werden, vollständig auf einem von den mehreren Modulen funktionieren, einen Stromverbrauchsidentifizierer zum Identifizieren des Stromverbrauchs des einen von den mehreren Modulen für jeden aktiven Zeitraum, einen Aufgabenaktivierungszähler zum Zählen der Anzahl der Male, mit denen die Aufgabe das eine von den mehreren Modulen in einer Zeiteinheit aktiviert, und einen Gesamtenergieberechner zum Berechnen der in der Zeiteinheit vom einen von den mehreren Modulen verbrauchten Gesamtenergie auf der Grundlage der Zeit, während derer die Aufgabe aktiv ist, der Zeitdauer des Aufgabenabschlusses auf dem einen Modul und der vom Modul während des aktiven Zeitraums benötigten Energie.

[2623] Beispiel 1020 umfasst den Gegenstand von Beispiel 1019. In Beispiel 1020 umfasst die Vorrichtung einen Vorrichtungslebenszeitgenerator zur Erzeugung der Vorrichtungslebenszeit auf der Grundlage der in der Zeiteinheit von den mehreren Modulen verbrauchten Gesamtenergie und der Energieressourcen für die Vorrichtung.

[2624] Beispiel 1021 umfasst den Gegenstand von Beispiel 1019 oder 1020. In Beispiel 1021 umfasst die Vorrichtung einen Vorrichtungsanweiser, um die Vorrichtung anzuweisen, die Anwendung ansprechend darauf zu starten, dass die berechnete Vorrichtungslebenszeit größer oder gleich einer vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegten minimalen Zeitschwelle ist.

[2625] Beispiel 1022 umfasst den Gegenstand von einem der Beispiele 1019 bis 1021. In Beispiel 1022 umfasst die Vorrichtung einen Vorrichtungsanweiser, um anzuweisen, dass eine Anforderung einer neuen Konfiguration ansprechend darauf zu einer Master-Vorrichtung gesendet wird, dass die berechnete Vorrichtungslebenszeit kleiner als eine vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegte minimale Zeitschwelle ist.

[2626] Beispiel 1023 umfasst den Gegenstand von einem der Beispiele 1019 bis 1022. In Beispiel 1023 umfasst die Anforderung einer neuen Konfiguration eine detaillierte Auflistung zumindest eines der Folgenden: des Energieverbrauchs für die Aufgabe auf dem Modul, der vom Modul über die Zeiteinheit verbrauchten Energie und des Energieverbrauchs für einen Modultyp in der Anzahl von Modulen.

[2627] Beispiel 1024 umfasst den Gegenstand von einem der Beispiele 1019 bis 1023. In Beispiel 1024 geschieht ein Aktivierungsschritt vor dem Spezifizieren von Energieressourcen, wobei der Aktivierungsschritt das Bereitstellen von Strom für die Vorrichtung, das Aktualisieren einer Hardwarekonfiguration der Vorrichtung und eine Aktualisierung des Anwendungscodes umfasst.

[2628] Beispiel 1025 umfasst den Gegenstand von einem der Beispiele 1019 bis 1024. In Beispiel 1025 umfasst die Vorrichtung einen Vorrichtungsanweiser, um die Vorrichtung anzuweisen, die für eine erste Konfiguration berechnete Gesamtenergie pro Zeiteinheit zu speichern, wobei der Vorrichtungsanweiser die Vorrichtung anweisen soll, von einer Master-Vorrichtung eine zweite Konfiguration anzufordern, einen zweiten Gesamtenergieberechner zur Berechnung einer zweiten Gesamtenergie pro Einheit für die zweite Konfiguration der Vorrichtung, einen Gesamtenergievergleicher zum Vergleichen der Gesamtenergie pro Einheit der ersten Konfiguration mit der zweiten Gesamtenergie pro Einheit für die zweite Konfiguration und den Vorrichtungsanweiser, um die Vorrichtung anzuweisen, entweder die erste Konfiguration oder die zweite Konfiguration auf der Grundlage des Vergleichs der Gesamtenergie pro Einheit und der zweiten Gesamtenergie pro Einheit anzufordern.

[2629] Beispiel 1026 umfasst den Gegenstand von einem der Beispiele 1019 bis 1025. In Beispiel 1026 umfasst die Vorrichtung einen Vorrichtungsanweiser, um die Vorrichtung anzuweisen, die Anwendung zu starten, falls festgestellt wird, dass die Vorrichtung durch eine festverdrahtete Ressource mit Strom versorgt wird.

[2630] Beispiel 1027 umfasst den Gegenstand von einem der Beispiele 1019 bis 1026. In Beispiel 1027 werden beim Identifizieren des Stromverbrauchs des einen der mehreren Module für jeden aktiven Zeitraum vom Modul historische Energieverbrauchswerte in Zusammenhang mit dem Modul für die Aufgabe abgerufen.

[2631] Beispiel 1028 umfasst den Gegenstand von einem der Beispiele 1019 bis 1027. In Beispiel 1028 umfasst die Vorrichtung einen Anwendungsschlafmodusidentifizierer zur Identifikation, ob die Anwendung eine Schlafmodusfunktion hat, einen Zeitmesser zum Messen der Zeit, während derer sich die Anwendung in einer Zeiteinheit in einem Schlafmodus befindet, und einen Gesamtenergieberechner zur Berechnung der verbrauchten Gesamtenergie auf der Grundlage des Gesamtstroms, der auf der Grundlage der Zeit gespart wird, während derer sich ein Prozessor für eine Anwendung im Schlafmodus befindet.

[2632] Beispiel 1029 umfasst ein Verfahren zur Bestimmung des Anwendungsenergiebedarfs. Das Verfahren zur Bestimmung des Anwendungsenergiebedarfs umfasst das Spezifizieren von Energieressourcen für eine Vorrichtung, die dafür ausgelegt ist, mehrere Module mit Strom zu versorgen und zu steuern, das Spezifizieren des Energiebedarfs für die mehreren Module, das Zerlegen einer Anwendung in Aufgaben, die, wenn sie aktiviert werden, vollständig auf einem der mehreren Module funktionieren, das Identifizieren des Stromverbrauchs des einen der mehreren Module für jeden aktiven Zeitraum, das Zählen der Anzahl der Male, mit denen die Aufgabe das eine der mehreren Module in einer Zeiteinheit aktiviert, und das Berechnen der in der Zeiteinheit von dem einen der mehreren Module verwendeten Gesamtenergie auf der Grundlage der Zeit, während derer die Aufgabe aktiv ist, der Zeitdauer des Aufgabenabschlusses auf dem einen Modul und der vom Modul während des aktiven Zeitraums benötigten Energie.

[2633] Beispiel 1030 umfasst den Gegenstand von Beispiel 1029. In Beispiel 1030 umfasst das Verfahren das Erzeugen einer Vorrichtungslebenszeit auf der Grundlage der in der Zeiteinheit verwendeten Gesamtenergie für die mehreren Module und auf den Energieressourcen für die Vorrichtung.

[2634] Beispiel 1031 umfasst den Gegenstand von Beispiel 1029 oder 1030. In Beispiel 1031 wird beim Verfahren die Vorrichtung angewiesen, die Anwendung ansprechend darauf zu starten, dass die berechnete Vorrichtungslebenszeit größer oder gleich einer vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegten minimalen Zeitschwelle ist.

[2635] Beispiel 1032 umfasst den Gegenstand von einem der Beispiele 1029 bis 1031. In Beispiel 1032 wird beim Verfahren angewiesen, dass eine Anforderung einer neuen Konfiguration ansprechend darauf zu einer Master-Vorrichtung gesendet wird, dass die berechnete Vorrichtungslebenszeit kürzer als eine vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegte minimale Zeitschwelle ist.

[2636] Beispiel 1033 umfasst den Gegenstand von einem der Beispiele 1029 bis 1032. In Beispiel 1033 umfasst die Anforderung einer neuen Konfiguration eine detaillierte Auflistung zumindest eines der Folgenden: des Energieverbrauchs für die Aufgabe auf dem Modul, der vom Modul über die Zeiteinheit verbrauchten Energie und des Energieverbrauchs für einen Modultyp in der Anzahl von Modulen.

[2637] Beispiel 1034 umfasst den Gegenstand von einem der Beispiele 1029 bis 1033. In Beispiel 1034 geschieht ein Aktivierungsschritt vor dem Spezifizieren von Energieressourcen, wobei der Aktivierungsschritt das Bereitstellen von Strom für die Vorrichtung, das Aktualisieren einer Hardwarekonfiguration der Vorrichtung oder eine Aktualisierung des Anwendungscodes umfasst.

[2638] Beispiel 1035 umfasst den Gegenstand von einem der Beispiele 1029 bis 1034. In Beispiel 1035 umfasst das Verfahren das Anweisen der Vorrichtung, die für eine erste Konfiguration berechnete Gesamtenergie pro Zeiteinheit zu speichern, das Anweisen der Vorrichtung, von einer Master-Vorrichtung eine zweite Konfiguration anzufordern, das Berechnen einer zweiten Gesamtenergie pro Einheit für die zweite Konfiguration der Vorrichtung, das Vergleichen der Gesamtenergie pro Einheit der ersten Konfiguration mit der zweiten Gesamtenergie pro Einheit für die zweite Konfiguration und das Anweisen der Vorrichtung, entweder die erste Konfiguration oder die zweite Konfiguration anzufordern, auf der Grundlage des Vergleichs der Gesamtenergie pro Einheit und der zweiten Gesamtenergie pro Einheit.

[2639] Beispiel 1036 umfasst den Gegenstand von einem der Beispiele 1029 bis 1035. In Beispiel 1036 wird beim Verfahren die Vorrichtung angewiesen, die Anwendung zu starten, falls festgestellt wird, dass die Vorrichtung durch eine festverdrahtete Ressource mit Strom versorgt wird.

[2640] Beispiel 1037 umfasst den Gegenstand von einem der Beispiele 1029 bis 1036. In Beispiel 1037 werden beim Identifizieren des Stromverbrauchs des einen der mehreren Module für jeden aktiven Zeitraum vom Modul historische Energieverbrauchswerte in Zusammenhang mit dem Modul für die Aufgabe abgerufen.

[2641] Beispiel 1038 umfasst den Gegenstand von einem der Beispiele 1029 bis 1037. In Beispiel 1038 umfasst das Verfahren das Feststellen, ob die Anwendung eine Schlafmodusfunktion hat, das Messen der Zeit, während derer sich die Anwendung in einer Zeiteinheit in einem Schlafmodus befindet, und das Berechnen der verbrauchten Gesamtenergie auf der Grundlage des Gesamtstroms, der auf der Grundlage der Zeit gespart wird, während derer sich ein Prozessor für eine Anwendung im Schlafmodus befindet.

[2642] Beispiel 1039 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Energieressourcen für eine Vorrichtung zu spezifizieren, die dafür ausgelegt ist, mehrere Module mit Strom zu versorgen und zu steuern, den Energiebedarf für die mehreren Module zu spezifizieren, eine Anwendung in Aufgaben zu zerlegen, die, wenn sie aktiviert werden, vollständig auf einem der mehreren Module funktionieren, den Stromverbrauch des einen der mehreren Module für jeden aktiven Zeitraum zu identifizieren, die Anzahl der Male zu zählen, mit denen die Aufgabe das eine der mehreren Module in einer Zeiteinheit aktiviert, und die in der Zeiteinheit von dem einen der mehreren Module verwendete Gesamtenergie auf der Grundlage der Zeit, während derer die Aufgabe aktiv ist, der Zeitdauer des Aufgabenabschlusses auf dem einen Modul und der vom Modul während des aktiven Zeitraums benötigten Energie zu berechnen.

[2643] Beispiel 1040 umfasst den Gegenstand von Beispiel 1039. In Beispiel 1040 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Vorrichtungslebenszeit auf der Grundlage der in der Zeiteinheit verbrauchten Gesamtenergie für die mehreren Module und auf der Grundlage der Energieressourcen für die Vorrichtung zu erzeugen.

[2644] Beispiel 1041 umfasst den Gegenstand von Beispiel 1039 oder 1040. In Beispiel 1041 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Anwendung ansprechend darauf zu starten, dass die berechnete Vorrichtungslebenszeit größer oder gleich einer vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegten minimalen Zeitschwelle ist.

[2645] Beispiel 1042 umfasst den Gegenstand von einem der Beispiele 1039 bis 1041. In Beispiel 1042 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, das Senden einer neuen Konfiguration zu einer Master-Vorrichtung ansprechend darauf zu fordern, dass die berechnete Vorrichtungslebenszeit kürzer als eine vor Beginn der Anwendungsanalyse für die Vorrichtung festgelegte minimale Zeitschwelle ist.

[2646] Beispiel 1043 umfasst den Gegenstand von einem der Beispiele 1039 bis 1042. In Beispiel 1043 umfasst die Anforderung einer neuen Konfiguration eine detaillierte Auflistung zumindest eines der Folgenden: des Energieverbrauchs für die Aufgabe auf dem Modul, der vom Modul über die Zeiteinheit verbrauchten Energie und des Energieverbrauchs für einen Modultyp in der Anzahl von Modulen.

[2647] Beispiel 1044 umfasst den Gegenstand von einem der Beispiele 1039 bis 1043. In Beispiel 1044 geschieht ein Aktivierungsschritt vor dem Spezifizieren von Energieressourcen, wobei der Aktivierungsschritt das Bereitstellen von Strom für die Vorrichtung, das Aktualisieren einer Hardwarekonfiguration der Vorrichtung und eine Aktualisierung des Anwendungscodes umfasst.

[2648] Beispiel 1045 umfasst den Gegenstand von einem der Beispiele 1039 bis 1044. In Beispiel 1045 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die für eine erste Konfiguration berechnete Gesamtenergie pro Zeiteinheit zu speichern, die Vorrichtung anzuweisen, von einer Master-Vorrichtung eine zweite Konfiguration anzufordern, eine zweite Gesamtenergie pro Einheit für die zweite Konfiguration der Vorrichtung zu berechnen, die Gesamtenergie pro Einheit der ersten Konfiguration mit der zweiten Gesamtenergie pro Einheit für die zweite Konfiguration zu vergleichen und die Vorrichtung anzuweisen, auf der Grundlage des Vergleichs der Gesamtenergie pro Einheit und der zweiten Gesamtenergie pro Einheit entweder die erste Konfiguration oder die zweite Konfiguration anzufordern.

[2649] Beispiel 1046 umfasst den Gegenstand von einem der Beispiele 1039 bis 1045. In Beispiel 1046 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Anwendung zu starten, falls festgestellt wird, dass die Vorrichtung durch eine festverdrahtete Ressource mit Strom versorgt wird.

[2650] Beispiel 1047 umfasst den Gegenstand von einem der Beispiele 1039 bis 1046. In Beispiel 1047 werden beim Identifizieren des Stromverbrauchs des einen der mehreren Module für jeden aktiven Zeitraum vom Modul historische Energieverbrauchswerte in Zusammenhang mit dem Modul für die Aufgabe abgerufen.

[2651] Beispiel 1048 umfasst den Gegenstand von einem der Beispiele 1039 bis 1047. In Beispiel 1048 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, ob die Anwendung eine Schlafmodusfunktion hat, die Zeit zu messen, während derer sich die Anwendung in einer Zeiteinheit in einem Schlafmodus befindet, und die verbrauchte Gesamtenergie auf der Grundlage des Gesamtstroms zu berechnen, der auf der Grundlage der Zeit gespart wird, während derer sich der Prozessor für eine Anwendung im Schlafmodus befindet.

[2652] Beispiel 1049 umfasst eine Vorrichtung zur Erzeugung eigens eingerichteter Hardware zur Signalaufbereitung. Die Vorrichtung zur Erzeugung eigens eingerichteter Hardware zur Signalaufbereitung umfasst eine selbstbeschreibende Vorrichtung. Die Vorrichtung umfasst auch ein feldprogrammierbares Gate-Array zur Erkennung einer Verbindung mit einem selbstbeschreibenden Sensormodul und einen Prozessor zur Verarbeitung vom selbstbeschreibenden Sensormodul empfangener Daten, wobei die empfangenen Daten Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul angeben, zur Erzeugung von Code, der auf dem feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist, und zur Modifikation vom selbstbeschreibenden Sensormodul empfangener Rohdaten zur Erzeugung von signalaufbereiteten Daten unter Verwendung des feldprogrammierbaren Gate-Arrays auf der Grundlage der Signalaufbereitungsinformationen.

[2653] Beispiel 1050 umfasst den Gegenstand von Beispiel 1049. In Beispiel 1050 soll der Prozessor erkennen, dass das feldprogrammierbare Gate-Array den Code des feldprogrammierbaren Gate-Arrays nicht unterstützen kann.

[2654] Beispiel 1051 umfasst den Gegenstand von Beispiel 1049 oder 1050. In Beispiel 1051 soll der Prozessor existierende Blöcke des feldprogrammierbaren Gate-Arrays ansprechend auf die Erkennung, dass die Blöcke des feldprogrammierbaren Gate-Arrays für die Signalaufbereitung der vom selbstbeschreibenden Sensormodul empfangenen Rohdaten ausreichen, mit den existierenden Blöcken des feldprogrammierbaren Gate-Arrays verbinden.

[2655] Beispiel 1052 umfasst den Gegenstand von einem der Beispiele 1049 bis 1051. In Beispiel 1052 soll der Prozessor identifizierte Eigenschaften des feldprogrammierbaren Gate-Arrays und Eigenschaften selbstbeschreibender Vorrichtungen übertragen.

[2656] Beispiel 1053 umfasst den Gegenstand von einem der Beispiele 1049 bis 1052. In Beispiel 1053 soll der Prozessor eine Anforderung zu einer Master-Vorrichtung senden, wobei die Anforderung Eigenschaften des feldprogrammierbaren Gate-Arrays und Eigenschaften einer selbstbeschreibenden Vorrichtung aufweist, wobei die Anforderung ansprechend auf die Erkennung zu senden ist, dass die selbstbeschreibende Vorrich-

tung beschränkt ist und das feldprogrammierbare Gate-Array der selbstbeschreibenden Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke aufweist.

[2657] Beispiel 1054 umfasst den Gegenstand von einem der Beispiele 1049 bis 1053. In Beispiel 1054 soll die Anforderung einen von der Master-Vorrichtung erzeugten Code des feldprogrammierbaren Gate-Arrays erzeugen und zurückgeben.

[2658] Beispiel 1055 umfasst den Gegenstand von einem der Beispiele 1049 bis 1054. In Beispiel 1055 umfasst die selbstbeschreibende Vorrichtung eine Batteriestromquelle oder eine Internetverbindung auf der Grundlage einer drahtlosen Technologie, oder ein Prozessor mit einem Modus mit geringer Leistungsaufnahme wird als beschränkte selbstbeschreibende Vorrichtung angesehen.

[2659] Beispiel 1056 umfasst den Gegenstand von einem der Beispiele 1049 bis 1055. In Beispiel 1056 soll der Prozessor ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen zu einer unbeschränkten Vorrichtung senden.

[2660] Beispiel 1057 umfasst den Gegenstand von einem der Beispiele 1049 bis 1056. In Beispiel 1057 soll der Prozessor zur unbeschränkten Vorrichtung eine Anforderung senden, einen Rückkehr-Code eines feldprogrammierbaren Gate-Arrays zu erzeugen, der am feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist.

[2661] Beispiel 1058 umfasst den Gegenstand von einem der Beispiele 1049 bis 1057. In Beispiel 1058 soll der Prozessor die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen zu einer zweiten beschränkten Vorrichtung senden, nachdem festgestellt wurde, dass die selbstbeschreibende Vorrichtung beschränkt ist, wobei die zweite beschränkte Vorrichtung auf eine nicht beschränkte Vorrichtung zugreifen kann.

[2662] Beispiel 1059 umfasst ein Verfahren zur Erzeugung von eigens eingerichteter Signalaufbereitungshardware. Das Verfahren zur Erzeugung von eigens eingerichteter Signalaufbereitungshardware umfasst das Erkennen eines selbstbeschreibenden Sensormoduls ansprechend auf eine Verbindung des selbstbeschreibenden Sensormoduls mit einer selbstbeschreibenden Vorrichtung, wobei die selbstbeschreibende Vorrichtung ein feldprogrammierbares Gate-Array umfasst, das Verarbeiten vom selbstbeschreibenden Sensormodul empfangener Daten, wobei die empfangenen Daten Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul angeben, das Erzeugen von Code für ein feldprogrammierbares Gate-Array, der auf dem feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist, und das Modifizieren vom selbstbeschreibenden Sensormodul empfangener Rohdaten zur Erzeugung von signalaufbereiteten Daten unter Verwendung des feldprogrammierbaren Gate-Arrays auf der Grundlage der Signalaufbereitungsinformationen.

[2663] Beispiel 1060 umfasst den Gegenstand von Beispiel 1059. In Beispiel 1060 umfasst das Verfahren die Erkennung, dass das feldprogrammierbare Gate-Array den Code des feldprogrammierbaren Gate-Arrays nicht unterstützen kann.

[2664] Beispiel 1061 umfasst den Gegenstand von Beispiel 1059 oder 1060. In Beispiel 1061 umfasst das Verfahren das Verbinden existierender Blöcke des feldprogrammierbaren Gate-Arrays ansprechend auf die Erkennung, dass die Blöcke des feldprogrammierbaren Gate-Arrays für die Signalaufbereitung der vom selbstbeschreibenden Sensormodul empfangenen Rohdaten ausreichen.

[2665] Beispiel 1062 umfasst den Gegenstand von einem der Beispiele 1059 bis 1061. In Beispiel 1062 umfasst das Verfahren das Senden identifizierter Eigenschaften des feldprogrammierbaren Gate-Arrays und der selbstbeschreibenden Vorrichtung.

[2666] Beispiel 1063 umfasst den Gegenstand von einem der Beispiele 1059 bis 1062. In Beispiel 1063 umfasst das Verfahren das Senden einer Anforderung zu einer Master-Vorrichtung, wobei die Anforderung Eigenschaften des feldprogrammierbaren Gate-Arrays und der selbstbeschreibenden Vorrichtung aufnehmen soll, die Anforderung ansprechend auf die Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, und die Erkennung, dass das feldprogrammierbare Gate-Array der selbstbeschreibenden Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke hat, zu senden ist.

[2667] Beispiel 1064 umfasst den Gegenstand von einem der Beispiele 1059 bis 1063. In Beispiel 1064 soll die Anforderung einen von der Master-Vorrichtung erzeugten Code des feldprogrammierbaren Gate-Arrays erzeugen und zurückgeben.

[2668] Beispiel 1065 umfasst den Gegenstand von einem der Beispiele 1059 bis 1064. In Beispiel 1065 umfasst die selbstbeschreibende Vorrichtung eine Batteriestromquelle oder eine Internetverbindung auf der Grundlage einer drahtlosen Technologie, oder ein Prozessor mit einem Modus mit geringer Leistungsaufnahme wird als beschränkte selbstbeschreibende Vorrichtung angesehen.

[2669] Beispiel 1066 umfasst den Gegenstand von einem der Beispiele 1059 bis 1065. In Beispiel 1066 umfasst das Verfahren das Senden der Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und von Vorrichtungsinformationen zu einer unbeschränkten Vorrichtung ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist.

[2670] Beispiel 1067 umfasst den Gegenstand von einem der Beispiele 1059 bis 1066. In Beispiel 1067 umfasst das Verfahren das Senden einer Anforderung zur Erzeugung eines Rückkehr-Codes eines feldprogrammierbaren Gate-Arrays, der auf dem feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist, zur unbeschränkten Vorrichtung.

[2671] Beispiel 1068 umfasst den Gegenstand von einem der Beispiele 1059 bis 1067. In Beispiel 1068 umfasst das Verfahren das Senden der Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und von Vorrichtungsinformationen zu einer zweiten beschränkten Vorrichtung ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, wobei die zweite beschränkte Vorrichtung auf eine unbeschränkte Vorrichtung zugreifen kann.

[2672] Beispiel 1069 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, ein selbstbeschreibendes Sensormodul ansprechend auf eine Verbindung des selbstbeschreibenden Sensormoduls mit einer selbstbeschreibenden Vorrichtung zu erkennen, wobei die selbstbeschreibende Vorrichtung ein feldprogrammierbares Gate-Array umfasst, vom selbstbeschreibenden Sensormodul empfangene Daten zu verarbeiten, wobei die empfangenen Daten Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul angeben, Code des feldprogrammierbaren Gate-Arrays, der auf dem feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist, zu erzeugen und vom selbstbeschreibenden Sensormodul empfangene Rohdaten zu modifizieren, um signalaufbereitete Daten unter Verwendung des feldprogrammierbaren Gate-Arrays auf der Grundlage der Signalaufbereitungsinformationen zu erzeugen.

[2673] Beispiel 1070 umfasst den Gegenstand von Beispiel 1069. In Beispiel 1070 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, festzustellen, dass das feldprogrammierbare Gate-Array den Code für das feldprogrammierbare Gate-Array nicht unterstützen kann.

[2674] Beispiel 1071 umfasst den Gegenstand von Beispiel 1069 oder 1070. In Beispiel 1071 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, existierende Blöcke des feldprogrammierbaren Gate-Arrays ansprechend auf die Erkennung zu verbinden, dass die Blöcke des feldprogrammierbaren Gate-Arrays für die Signalaufbereitung der vom selbstbeschreibenden Sensormodul empfangenen Rohdaten ausreichen.

[2675] Beispiel 1072 umfasst den Gegenstand von einem der Beispiele 1069 bis 1071. In Beispiel 1072 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, identifizierte Eigenschaften des feldprogrammierbaren Gate-Arrays und der selbstbeschreibenden Vorrichtung zu übertragen.

[2676] Beispiel 1073 umfasst den Gegenstand von einem der Beispiele 1069 bis 1072. In Beispiel 1073 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Anforderung zu einer Master-Vorrichtung zu senden, wobei die Anforderung Eigenschaften des feldprogrammierbaren Gate-Arrays und der selbstbeschreibenden Vorrichtung aufweisen soll, wobei die Anforderung ansprechend auf die Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, und die Erkennung, dass das feldprogrammierbare Gate-Array der selbstbeschreibenden Vorrichtung nicht genügend konfigurierbare Signalaufbereitungsblöcke hat, zu senden ist.

[2677] Beispiel 1074 umfasst den Gegenstand von einem der Beispiele 1069 bis 1073. In Beispiel 1074 soll die Anforderung einen von der Master-Vorrichtung erzeugten Code des feldprogrammierbaren Gate-Arrays erzeugen und zurückgeben.

[2678] Beispiel 1075 umfasst den Gegenstand von einem der Beispiele 1069 bis 1074. In Beispiel 1075 umfasst die selbstbeschreibende Vorrichtung eine Batteriestromquelle oder eine Internetverbindung auf der Grundlage einer drahtlosen Technologie, oder ein Prozessor mit einem Modus mit geringer Leistungsaufnahme wird als beschränkte selbstbeschreibende Vorrichtung angesehen.

[2679] Beispiel 1076 umfasst den Gegenstand von einem der Beispiele 1069 bis 1075. In Beispiel 1076 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen ansprechend auf eine Erkennung, dass die selbstbeschreibende Vorrichtung beschränkt ist, zu einer unbeschränkten Vorrichtung zu senden.

[2680] Beispiel 1077 umfasst den Gegenstand von einem der Beispiele 1069 bis 1076. In Beispiel 1077 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, zur unbeschränkten Vorrichtung eine Anforderung zu senden, einen Rückkehr-Code eines feldprogrammierbaren Gate-Arrays, der auf dem feldprogrammierbaren Gate-Array der selbstbeschreibenden Vorrichtung zu implementieren ist, zu erzeugen.

[2681] Beispiel 1078 umfasst den Gegenstand von einem der Beispiele 1069 bis 1077. In Beispiel 1078 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Signalaufbereitungsinformationen für das selbstbeschreibende Sensormodul und Vorrichtungsinformationen zu einer zweiten beschränkten Vorrichtung zu senden, nachdem erkannt wurde, dass die selbstbeschreibende Vorrichtung beschränkt ist, wobei die zweite beschränkte Vorrichtung auf eine unbeschränkte Vorrichtung zugreifen kann.

[2682] Beispiel 1079 umfasst eine Vorrichtung. Die Vorrichtung umfasst eine Internet-der-Dinge(IoT)-Vorrichtung. Die IoT-Vorrichtung umfasst einen Subnetz-Funktionstüchtigkeitsdatenanforderer zum Anfordern von Subnetz-Funktionstüchtigkeitsdaten von einem Subnetz ansprechend auf eine empfangene Netzfunktionstüchtigkeitsanforderung für ein Netz, einen Netz-Shadow-Filter-Modifizierer zur Modifikation eines Netz-Shadow-Filters auf der Grundlage anhand Blockchain-Daten erzeugter empfangener Subnetz-Funktionstüchtigkeitsdaten und einen Berichtsbereitsteller zum Bereitstellen eines Berichts zur Netzfunktionstüchtigkeit auf der Grundlage des Netz-Shadow-Filters.

[2683] Beispiel 1080 umfasst den Gegenstand von Beispiel 1079. In Beispiel 1080 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von einer Vorrichtung ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2684] Beispiel 1081 umfasst den Gegenstand von Beispiel 1079 oder 1080. In Beispiel 1081 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren eines Subnetz-Shadow-Filters unter Verwendung anhand empfangener Vorrichtungsfunktionstüchtigkeitsdaten erzeugter Blockchain-Daten zu erzeugen.

[2685] Beispiel 1082 umfasst den Gegenstand von einem der Beispiele 1079 bis 1081. In Beispiel 1082 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von mehreren Vorrichtungen ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2686] Beispiel 1083 umfasst den Gegenstand von einem der Beispiele 1079 bis 1082. In Beispiel 1083 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren des Subnetz-Shadow-Filters unter Verwendung mehrerer empfangener Vorrichtungsfunktionstüchtigkeitsdaten durch die Verwendung eines logischen Operators zum Vergleich der mehreren empfangenen Funktionstüchtigkeitsdaten zu erzeugen.

[2687] Beispiel 1084 umfasst den Gegenstand von einem der Beispiele 1079 bis 1083. In Beispiel 1084 gibt die Vorrichtung Vorrichtungsfunktionstüchtigkeitsdaten auf der Grundlage eines Vorrichtungs-Shadow-Filters zurück, wobei das Vorrichtungs-Shadow-Filter auf der Grundlage eines Vorrichtungs-Bloomfilters zu erzeugen ist, das ein Nichtvorhandensein geplanter Funktionstüchtigkeitsnachrichten an der Vorrichtung verfolgt.

[2688] Beispiel 1085 umfasst den Gegenstand von einem der Beispiele 1079 bis 1084. In Beispiel 1085 umfasst das Vorrichtung-Shadow-Filter mehrere Shadow-Filter, die jeweils einem Zeitintervall mehrerer Bloomfilter entsprechen.

[2689] Beispiel 1086 umfasst den Gegenstand von einem der Beispiele 1079 bis 1085. In Beispiel 1086 arbeitet das Netz-Shadow-Filter durch Anwendung eines auf einen Satz, der wenigstens das Subnetz und ein anderes Subnetz umfasst, angewendeten logischen Operators.

[2690] Beispiel 1087 umfasst den Gegenstand von einem der Beispiele 1079 bis 1086. In Beispiel 1087 umfasst die Vorrichtung einen Subnetz-Funktionstüchtigkeitsdatenanforderer zum Anfordern von Subnetz-Funktionstüchtigkeitsdaten von mehreren logisch im Netz organisierten Subnetzen und einen Netz-Shadow-Filter-Modifizierer zum Modifizieren eines Netz-Shadow-Filters auf der Grundlage mehrerer empfangener Subnetz-Funktionstüchtigkeitsdaten.

[2691] Beispiel 1088 umfasst den Gegenstand von einem der Beispiele 1079 bis 1087. In Beispiel 1088 beruhen die Blockchain-Daten auf Bloomfiltern, die in eine Blockchain-Transaktion aufgenommen werden können, um einen historischen Datensatz der Netzfunktionstüchtigkeit an einem Endpunkt einer Netztopographie zu erzeugen.

[2692] Beispiel 1089 umfasst ein Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zum Berichten über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen. Das Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zum Berichten über die Funktionstüchtigkeit eines Netzes und von Netzvorrichtungen umfasst das Anfordern von Subnetz-Funktionstüchtigkeitsdaten von einem Subnetz ansprechend auf eine empfangene Netzfunktionstüchtigkeitsanforderung für ein Netz, das Modifizieren eines Netz-Shadow-Filters auf der Grundlage anhand Blockchain-Daten erzeugter empfangener Subnetz-Funktionstüchtigkeitsdaten und das Bereitstellen eines Berichts der Netzfunktionstüchtigkeit auf der Grundlage des Netz-Shadow-Filters.

[2693] Beispiel 1090 umfasst den Gegenstand von Beispiel 1089. In Beispiel 1090 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von einer Vorrichtung ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2694] Beispiel 1091 umfasst den Gegenstand von Beispiel 1089 oder 1090. In Beispiel 1091 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren eines Subnetz-Shadow-Filters unter Verwendung anhand empfangener Vorrichtungsfunktionstüchtigkeitsdaten erzeugter Blockchain-Daten zu erzeugen.

[2695] Beispiel 1092 umfasst den Gegenstand von einem der Beispiele 1089 bis 1091. In Beispiel 1092 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von mehreren Vorrichtungen ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2696] Beispiel 1093 umfasst den Gegenstand von einem der Beispiele 1089 bis 1092. In Beispiel 1093 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren des Subnetz-Shadow-Filters unter Verwendung mehrerer empfangener Vorrichtungsfunktionstüchtigkeitsdaten durch die Verwendung eines logischen Operators zum Vergleich der mehreren empfangenen Funktionstüchtigkeitsdaten zu erzeugen.

[2697] Beispiel 1094 umfasst den Gegenstand von einem der Beispiele 1089 bis 1093. In Beispiel 1094 gibt die Vorrichtung Vorrichtungsfunktionstüchtigkeitsdaten auf der Grundlage eines Vorrichtung-Shadow-Filters zurück, wobei das Vorrichtung-Shadow-Filter auf der Grundlage eines Vorrichtung-Bloomfilters zu erzeugen ist, das ein Nichtvorhandensein geplanter Funktionstüchtigkeitsnachrichten an der Vorrichtung verfolgt.

[2698] Beispiel 1095 umfasst den Gegenstand von einem der Beispiele 1089 bis 1094. In Beispiel 1095 umfasst das Vorrichtung-Shadow-Filter mehrere Shadow-Filter, die jeweils einem Zeitintervall mehrerer Bloomfilter entsprechen.

[2699] Beispiel 1096 umfasst den Gegenstand von einem der Beispiele 1089 bis 1095. In Beispiel 1096 arbeitet das Netz-Shadow-Filter durch Anwendung eines auf einen Satz, der wenigstens das Subnetz und ein anderes Subnetz umfasst, angewendeten logischen Operators.

[2700] Beispiel 1097 umfasst den Gegenstand von einem der Beispiele 1089 bis 1096. In Beispiel 1097 umfasst das Verfahren das Anfordern von Subnetz-Funktionstüchtigkeitsdaten von mehreren logisch im Netz organisierten Subnetzen und das Modifizieren eines Netz-Shadow-Filters auf der Grundlage mehrerer empfangener Subnetz-Funktionstüchtigkeitsdaten.

[2701] Beispiel 1098 umfasst den Gegenstand von einem der Beispiele 1089 bis 1097. In Beispiel 1098 beruhen die Blockchain-Daten auf Bloomfiltern, die in eine Blockchain-Transaktion aufgenommen werden können, um einen historischen Datensatz der Netzfunktionstüchtigkeit an einem Endpunkt einer Netztopographie zu erzeugen.

[2702] Beispiel 1099 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Subnetz-Funktionstüchtigkeitsdaten von einem Subnetz ansprechend auf eine empfangene Netzfunktionstüchtigkeitsanforderung für ein Netz anzufordern, ein Netz-Shadow-Filter auf der Grundlage anhand Blockchain-Daten erzeugter empfangener Subnetz-Funktionstüchtigkeitsdaten zu modifizieren und einen Bericht über die Netzfunktionstüchtigkeit auf der Grundlage des Netz-Shadow-Filters bereitzustellen.

[2703] Beispiel 1100 umfasst den Gegenstand von Beispiel 1099. In Beispiel 1100 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von einer Vorrichtung ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2704] Beispiel 1101 umfasst den Gegenstand von Beispiel 1099 oder 1100. In Beispiel 1101 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren eines Subnetz-Shadow-Filters unter Verwendung anhand empfangener Vorrichtungsfunktionstüchtigkeitsdaten erzeugter Blockchain-Daten zu erzeugen.

[2705] Beispiel 1102 umfasst den Gegenstand von einem der Beispiele 1099 bis 1101. In Beispiel 1102 soll das Subnetz Vorrichtungsfunktionstüchtigkeitsdaten von mehreren Vorrichtungen ansprechend auf den Empfang der Subnetz-Funktionstüchtigkeitsdatenanforderung anfordern.

[2706] Beispiel 1103 umfasst den Gegenstand von einem der Beispiele 1099 bis 1102. In Beispiel 1103 wird das Subnetz angewiesen, Subnetz-Funktionstüchtigkeitsdaten durch Modifizieren des Subnetz-Shadow-Filters unter Verwendung mehrerer empfangener Vorrichtungsfunktionstüchtigkeitsdaten durch die Verwendung eines logischen Operators zum Vergleich der mehreren empfangenen Funktionstüchtigkeitsdaten zu erzeugen.

[2707] Beispiel 1104 umfasst den Gegenstand von einem der Beispiele 1099 bis 1103. In Beispiel 1104 gibt die Vorrichtung Vorrichtungsfunktionstüchtigkeitsdaten auf der Grundlage eines Vorrichtungs-Shadow-Filters zurück, wobei das Vorrichtungs-Shadow-Filter auf der Grundlage eines Vorrichtungs-Bloomfilters zu erzeugen ist, das ein Nichtvorhandensein geplanter Funktionstüchtigkeitsnachrichten an der Vorrichtung verfolgt.

[2708] Beispiel 1105 umfasst den Gegenstand von einem der Beispiele 1099 bis 1104. In Beispiel 1105 umfasst das Vorrichtungs-Shadow-Filter mehrere Shadow-Filter, die jeweils einem Zeitintervall mehrerer Bloomfilter entsprechen.

[2709] Beispiel 1106 umfasst den Gegenstand von einem der Beispiele 1099 bis 1105. In Beispiel 1106 arbeitet das Netz-Shadow-Filter durch Anwendung eines auf einen Satz, der wenigstens das Subnetz und ein anderes Subnetz umfasst, angewendeten logischen Operators.

[2710] Beispiel 1107 umfasst den Gegenstand von einem der Beispiele 1099 bis 1106. In Beispiel 1107 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Subnetz-Funktionstüchtigkeitsdaten von mehreren logisch im Netz organisierten Subnetzen anzufordern und ein Netz-Shadow-Filter auf der Grundlage mehrerer empfangener Subnetz-Funktionstüchtigkeitsdaten zu modifizieren.

[2711] Beispiel 1108 umfasst den Gegenstand von einem der Beispiele 1099 bis 1107. In Beispiel 1108 beruhen die Blockchain-Daten auf Bloomfiltern, die in eine Blockchain-Transaktion aufgenommen werden können, um einen historischen Datensatz der Netzfunktionstüchtigkeit an einem Endpunkt einer Netztopographie zu erzeugen.

[2712] Beispiel 1109 umfasst eine Vorrichtung, die ein Internet-der-Dinge(IoT)-Vorrichtung umfasst. Die Internet-der-Dinge(IoT)-Vorrichtung umfasst einen Steuerkanal-Dateneinfüger zum Einfügen von Steuerkanaldaten in einen Weitbereichsnetz-Frame, einen Frame-Sender zum Senden des Weitbereichsnetz-Frames zu einer ersten Vorrichtung mit einem ersten Übertragungsprotokoll und einer zweiten Vorrichtung mit einem zweiten Protokoll, wobei die Steuerkanaldaten Zugriff sowohl auf die erste Vorrichtung und das erste Vorrichtungsprotokoll als auch auf die zweite Vorrichtung und das zweite Vorrichtungsprotokoll erhalten sollen, und einen Datenmodifizierer zum Modifizieren von Daten eines sendenden Knotens ansprechend darauf, dass die erste Vorrichtung eine erkannte Übertragung von einem sendenden Knoten zurückgibt.

[2713] Beispiel 1110 umfasst den Gegenstand von Beispiel 1109. In Beispiel 1110 umfasst der Weitbereichsnetz-Frame ein Anwendungsnutzzinhalts-Frame-Feld zur Einkapselung der Steuerkanaldaten vor und während der Übertragung.

[2714] Beispiel 1111 umfasst den Gegenstand von Beispiel 1109 oder 1110. In Beispiel 1111 umfassen die Steuerkanaldaten Befehle für die erste Vorrichtung, einen Nutzzinhalt von einem sendenden Knoten zu sammeln, einen Zeitstempel aus dem Nutzzinhalt zu extrahieren und Nutzzinhalte, die einen Zeitstempel und eine Zonenkennung umfassen, zurückzugeben.

[2715] Beispiel 1112 umfasst den Gegenstand von einem der Beispiele 1109 bis 1111. In Beispiel 1112 sendet der Frame-Sender den Weitbereichsnetz-Frame durch Befehle an die erste Vorrichtung, als ein Zwischensender zu wirken und zur zweiten Vorrichtung zurückzusenden, zur zweiten Vorrichtung.

[2716] Beispiel 1113 umfasst den Gegenstand von einem der Beispiele 1109 bis 1112. In Beispiel 1113 umfassen die Steuerkanaldaten einen Nachrichten-Header, ein Zonenkennungsfeld, ein Steuernachrichtsdatenfeld und ein Sicherheitsfeld.

[2717] Beispiel 1114 umfasst den Gegenstand von einem der Beispiele 1109 bis 1113. In Beispiel 1114 beruht die Modifikation der Daten des sendenden Knotens auf dem Extrahieren einer Zonenkennung und eines Zeitstempels, wodurch eine erkannte Übertragung von einem sendenden Knoten zurückgegeben wird.

[2718] Beispiel 1115 umfasst den Gegenstand von einem der Beispiele 1109 bis 1114. In Beispiel 1115 umfassen die Steuerkanaldaten Befehle an die erste Vorrichtung, ihre Zonenkennung und mehrere Zeitstempel von einem sendenden Knoten zurückzugeben.

[2719] Beispiel 1116 umfasst den Gegenstand von einem der Beispiele 1109 bis 1115. In Beispiel 1116 umfassen die Steuerkanaldaten Befehle an die zweite Vorrichtung, an die erste Vorrichtung ihre Zonenkennung und einen Zeitstempel von einem sendenden Knoten zurückzugeben, wobei die zweite Vorrichtung keinen Kommunikationsweg zu einer Servervorrichtung hat, außer indem durch die erste Vorrichtung geleitet wird.

[2720] Beispiel 1117 umfasst den Gegenstand von einem der Beispiele 1109 bis 1116. In Beispiel 1117 umfasst die Vorrichtung einen Berechner zum Berechnen des Orts des sendenden Knotens unter Verwendung einer hyperbolischen Funktion und einer Anzahl von Zeitstempeln von Paketen, die von der ersten Vorrichtung und der zweiten Vorrichtung empfangen werden, sowie der Zonenkennung der ersten Vorrichtung und der zweiten Vorrichtung.

[2721] Beispiel 1118 umfasst den Gegenstand von einem der Beispiele 1109 bis 1117. In Beispiel 1118 umfasst die Vorrichtung einen Berechner zum Berechnen des Orts des sendenden Knotens auf der Grundlage der Sendedurchlaufzeit zwischen einer signalerfassenden Vorrichtung und einer Signalempfangsvorrichtung.

[2722] Beispiel 1119 umfasst ein Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur Entdeckung von Ressourcen und für die Sektoridentifikation geographischer Orte. Das Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur Entdeckung von Ressourcen und zur Sektoridentifikation geographischer Orte umfasst das Einfügen von Steuerkanaldaten in einen Weitbereichsnetz-Frame, das Senden des Weitbereichsnetz-Frames zu einer ersten Vorrichtung mit einem ersten Übertragungsprotokoll und einer zweiten Vorrichtung mit einem zweiten Protokoll, wobei die Steuerkanaldaten dazu dienen, Zugang sowohl zur ersten Vorrichtung und zum ersten Vorrichtungsprotokoll als auch zur zweiten Vorrichtung und zum zweiten Vorrichtungsprotokoll zu erhalten, und das Modifizieren der Daten des sendenden Knotens ansprechend darauf, dass die erste Vorrichtung eine erkannte Übertragung von einem sendenden Knoten zurückgibt.

[2723] Beispiel 1120 umfasst den Gegenstand von Beispiel 1119. In Beispiel 1120 umfasst der Weitbereichsnetz-Frame ein Anwendungsnutzzinhalts-Frame-Feld zur Einkapselung der Steuerkanaldaten vor und während der Übertragung.

[2724] Beispiel 1121 umfasst den Gegenstand von Beispiel 1119 oder 1120. In Beispiel 1121 umfassen die Steuerkanaldaten Befehle für die erste Vorrichtung, einen Nutzzinhalt von einem sendenden Knoten zu sammeln, einen Zeitstempel aus dem Nutzzinhalt zu extrahieren und Nutzzinhalte, die einen Zeitstempel und eine Zonenkennung umfassen, zurückzugeben.

[2725] Beispiel 1122 umfasst den Gegenstand von einem der Beispiele 1119 bis 1121. In Beispiel 1122 umfasst das Verfahren das Senden des Weitbereichsnetz-Frames zur zweiten Vorrichtung durch Befehle an die erste Vorrichtung, als Zwischensender zu wirken und zur zweiten Vorrichtung zurückzusenden.

[2726] Beispiel 1123 umfasst den Gegenstand von einem der Beispiele 1119 bis 1122. In Beispiel 1123 umfassen die Steuerkanaldaten einen Nachrichten-Header, ein Zonenkennungsfeld, ein Steuernachrichtsdatenfeld und ein Sicherheitsfeld.

[2727] Beispiel 1124 umfasst den Gegenstand von einem der Beispiele 1119 bis 1123. In Beispiel 1124 beruht die Modifikation der Daten des sendenden Knotens auf dem Extrahieren einer Zonenkennung und eines Zeitstempels, wodurch eine erkannte Übertragung von einem sendenden Knoten zurückgegeben wird.

[2728] Beispiel 1125 umfasst den Gegenstand von einem der Beispiele 1119 bis 1124. In Beispiel 1125 umfassen die Steuerkanaldaten Befehle an die erste Vorrichtung, ihre Zonenkennung und mehrere Zeitstempel von einem sendenden Knoten zurückzugeben.

[2729] Beispiel 1126 umfasst den Gegenstand von einem der Beispiele 1119 bis 1125. In Beispiel 1126 umfassen die Steuerkanaldaten Befehle an die zweite Vorrichtung, an die erste Vorrichtung ihre Zonenkennung und einen Zeitstempel von einem sendenden Knoten zurückzugeben, wobei die zweite Vorrichtung keinen Kommunikationsweg zu einer Servervorrichtung hat, außer indem durch die erste Vorrichtung geleitet wird.

[2730] Beispiel 1127 umfasst den Gegenstand von einem der Beispiele 1119 bis 1126. In Beispiel 1127 umfasst das Verfahren das Berechnen eines Orts des sendenden Knotens unter Verwendung einer hyperbolischen Funktion und mehrerer Zeitstempel von Paketen, die von der ersten Vorrichtung und der zweiten Vorrichtung empfangen werden, sowie der Zonenkennung der ersten Vorrichtung und der zweiten Vorrichtung.

[2731] Beispiel 1128 umfasst den Gegenstand von einem der Beispiele 1119 bis 1127. In Beispiel 1128 umfasst das Verfahren das Berechnen des Orts des sendenden Knotens auf der Grundlage der Sendedurchlaufzeit zwischen einer signalerfassenden Vorrichtung und einer Signalempfangsvorrichtung.

[2732] Beispiel 1129 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Steuerkanaldaten in einen Weitbereichsnetz-Frame einzufügen, den Weitbereichsnetz-Frame mit einem ersten Übertragungsprotokoll zu einer ersten Vorrichtung und mit einem zweiten Protokoll zu einer zweiten Vorrichtung zu übertragen, wobei die Steuerkanaldaten dazu dienen, Zugang sowohl zur ersten Vorrichtung und zum ersten Vorrichtungsprotokoll als auch zur zweiten Vorrichtung und zum zweiten Vorrichtungsprotokoll zu erhalten, und die Daten des sendenden Knotens ansprechend darauf zu modifizieren, dass die erste Vorrichtung eine erkannte Übertragung von einem sendenden Knoten zurückgibt.

[2733] Beispiel 1130 umfasst den Gegenstand von Beispiel 1129. In Beispiel 1130 umfasst der Weitbereichsnetz-Frame ein Anwendungsnutzzinhalts-Frame-Feld zur Einkapselung der Steuerkanaldaten vor und während der Übertragung.

[2734] Beispiel 1131 umfasst den Gegenstand von Beispiel 1129 oder 1130. In Beispiel 1131 umfassen die Steuerkanaldaten Befehle für die erste Vorrichtung, einen Nutzzinhalt von einem sendenden Knoten zu sammeln, einen Zeitstempel aus dem Nutzzinhalt zu extrahieren und Nutzzinhalte, die einen Zeitstempel und eine Zonenkennung umfassen, zurückzugeben.

[2735] Beispiel 1132 umfasst den Gegenstand von einem der Beispiele 1129 bis 1131. In Beispiel 1132 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen,

den Weitbereichsnetz-Frame durch Befehle an die erste Vorrichtung, als Zwischensender zu wirken und zur zweiten Vorrichtung zurückzusenden, zur zweiten Vorrichtung zu senden.

[2736] Beispiel 1133 umfasst den Gegenstand von einem der Beispiele 1129 bis 1132. In Beispiel 1133 umfassen die Steuerkanaldaten einen Nachrichten-Header, ein Zonenkennungsfeld, ein Steuernachrichtsdatenfeld und ein Sicherheitsfeld.

[2737] Beispiel 1134 umfasst den Gegenstand von einem der Beispiele 1129 bis 1133. In Beispiel 1134 beruht die Modifikation der Daten des sendenden Knotens auf dem Extrahieren einer Zonenkennung und eines Zeitstempels, wodurch eine erkannte Übertragung von einem sendenden Knoten zurückgegeben wird.

[2738] Beispiel 1135 umfasst den Gegenstand von einem der Beispiele 1129 bis 1134. In Beispiel 1135 umfassen die Steuerkanaldaten Befehle an die erste Vorrichtung, ihre Zonenkennung und mehrere Zeitstempel von einem sendenden Knoten zurückzugeben.

[2739] Beispiel 1136 umfasst den Gegenstand von einem der Beispiele 1129 bis 1135. In Beispiel 1136 umfassen die Steuerkanaldaten Befehle an die zweite Vorrichtung, an die erste Vorrichtung ihre Zonenkennung und einen Zeitstempel von einem sendenden Knoten zurückzugeben, wobei die zweite Vorrichtung keinen Kommunikationsweg zu einer Servervorrichtung hat, außer indem durch die erste Vorrichtung geleitet wird.

[2740] Beispiel 1137 umfasst den Gegenstand von einem der Beispiele 1129 bis 1136. In Beispiel 1137 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den Ort des sendenden Knotens unter Verwendung einer hyperbolischen Funktion und einer Anzahl von Zeitstempeln von Paketen, die von der ersten Vorrichtung und der zweiten Vorrichtung empfangen werden, sowie der Zonenkennung der ersten Vorrichtung und der zweiten Vorrichtung zu berechnen.

[2741] Beispiel 1138 umfasst den Gegenstand von einem der Beispiele 1129 bis 1137. In Beispiel 1138 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den Ort des sendenden Knotens auf der Grundlage der Sendedurchlaufzeit zwischen einer signalerfassenden Vorrichtung und einer Signalempfangsvorrichtung zu berechnen.

[2742] Beispiel 1139 umfasst eine Vorrichtung, die eine Internet-der-Dinge(IoT)-Vorrichtung umfasst. Die Internet-der-Dinge(IoT)-Vorrichtung umfasst einen Datenlader zum Laden von Daten in eine ausgewählte interaktive Analytikumgebung auf der Grundlage eines repräsentativen Datensatzes aus einer Umgebung, wobei die interaktive Analytikumgebung wenigstens eine von einer Daten-Mining-Umgebung und einer Datenfusionsumgebung ist, einen Übereinstimmungsfeststeller zur Feststellung, ob ein vorgeschlagenes Modell mit Daten übereinstimmt, wobei das vorgeschlagene Modell eine Arbeitslast zerlegbarer Abschnitte zur Verarbeitung auf eine Anzahl von Vorrichtungen aufweist, und eine Arbeitslast-Zuordnungseinrichtung zum Zuordnen der Arbeitslast zu einer Vorrichtung für die Ausführung durch die Vorrichtung.

[2743] Beispiel 1140 umfasst den Gegenstand von Beispiel 1139. In Beispiel 1140 nimmt die Daten-Mining-Umgebung Daten von einer Anwendungsdomain und von einer gespeicherten Wissensdatenbank entsprechend dem repräsentativen Datensatz aus einer Umgebung auf.

[2744] Beispiel 1141 umfasst den Gegenstand von Beispiel 1139 oder 1140. In Beispiel 1141 werden die Daten vor dem Laden in eine ausgewählte interaktive Analytikumgebung gereinigt und organisiert.

[2745] Beispiel 1142 umfasst den Gegenstand von einem der Beispiele 1139 bis 1141. In Beispiel 1142 umfasst die Vorrichtung einen Zerleger zum Zerlegen der Arbeitslast vor der Zuordnung der Arbeitslast zur Vorrichtung für die Ausführung durch die Vorrichtung.

[2746] Beispiel 1143 umfasst den Gegenstand von einem der Beispiele 1139 bis 1142. In Beispiel 1143 wird die Arbeitslast durch Parallelität und eine modulare seriell abhängige Aufgabe zerlegt.

[2747] Beispiel 1144 umfasst den Gegenstand von einem der Beispiele 1139 bis 1143. In Beispiel 1144 wird die Arbeitslast für eine Klassifizierungsaufgabe zerlegt, wobei eine Vorverarbeitungs-, eine Merkmalsextraktions- und eine Klassifizierungsaufgabe getrennt werden können.

[2748] Beispiel 1145 umfasst den Gegenstand von einem der Beispiele 1139 bis 1144. In Beispiel 1145 umfasst die Vorrichtung einen Prozessor zum Berechnen der Funktionsweise einer Plattform in Bezug auf die nach der Zuordnung verwendeten Ressourcen und die Ausführung der Arbeitslast an der Vorrichtung.

[2749] Beispiel 1146 umfasst den Gegenstand von einem der Beispiele 1139 bis 1145. In Beispiel 1146 gibt die interaktive Analytik eine zeitliche Beziehung zwischen einer ersten Datenprobe und einer zweiten Datenprobe an, wobei die interaktive Analytik ferner eine Beziehung zwischen einer ersten Probe und einem ersten Sensor und der zweiten Datenprobe und einem zweiten Sensor angibt.

[2750] Beispiel 1147 umfasst den Gegenstand von einem der Beispiele 1139 bis 1146. In Beispiel 1147 umfasst die Arbeitslast ein Profil von Rechenanforderungen, Speichieranforderungen und Netzanforderungen der Arbeitslast.

[2751] Beispiel 1148 umfasst den Gegenstand von einem der Beispiele 1139 bis 1147. In Beispiel 1148 umfasst die Arbeitslast mehrere Teilarbeitslasten an der Vorrichtung, wobei die Arbeitslast eine durch Sortieren der Arbeitslasten in Bezug auf eine benötigte Ressource und eine verfügbare Ressource und Paaren einer Arbeitslast mit einer Vorrichtung auf der Grundlage der benötigten Ressource und der verfügbaren Ressource erhaltene Übereinstimmung umfasst.

[2752] Beispiel 1149 umfasst ein Verfahren, das eine Datenanalytik von IoT-Systemen bereitstellt. Das Verfahren zum Bereitstellen einer Datenanalytik von IoT-Systemen umfasst das Laden von Daten in eine ausgewählte interaktive Analytikumgebung auf der Grundlage eines repräsentativen Datensatzes aus einer Umgebung, wobei die interaktive Analytikumgebung eine Daten-Mining-Umgebung und eine Datenfusionsumgebung ist, das Feststellen, ob ein vorgeschlagenes Modell mit Daten übereinstimmt, wobei das vorgeschlagene Modell eine Arbeitslast zerlegbarer Abschnitte zur Verarbeitung auf mehreren Vorrichtungen umfasst, und das Zuordnen der Arbeitslast zu einer Vorrichtung zur Ausführung durch die Vorrichtung.

[2753] Beispiel 1150 umfasst den Gegenstand von Beispiel 1149. In Beispiel 1150 nimmt die Daten-Mining-Umgebung Daten von einer Anwendungsdomain und von einer gespeicherten Wissensdatenbank entsprechend dem repräsentativen Datensatz aus einer Umgebung auf.

[2754] Beispiel 1151 umfasst den Gegenstand von Beispiel 1149 oder 1150. In Beispiel 1151 werden die Daten vor dem Laden in eine ausgewählte interaktive Analytikumgebung gereinigt und organisiert.

[2755] Beispiel 1152 umfasst den Gegenstand von einem der Beispiele 1149 bis 1151. In Beispiel 1152 umfasst das Verfahren das Zerlegen der Arbeitslast vor dem Zuordnen der Arbeitslast zur Vorrichtung zur Ausführung durch die Vorrichtung.

[2756] Beispiel 1153 umfasst den Gegenstand von einem der Beispiele 1149 bis 1152. In Beispiel 1153 wird die Arbeitslast durch Parallelität und eine modulare seriell abhängige Aufgabe zerlegt.

[2757] Beispiel 1154 umfasst den Gegenstand von einem der Beispiele 1149 bis 1153. In Beispiel 1154 wird die Arbeitslast für eine Klassifizierungsaufgabe zerlegt, wobei eine Vorverarbeitungs-, eine Merkmalsextraktions- und eine Klassifizierungsaufgabe getrennt werden können.

[2758] Beispiel 1155 umfasst den Gegenstand von einem der Beispiele 1149 bis 1154. In Beispiel 1155 umfasst das Verfahren das Berechnen der Funktionsweise einer Plattform in Bezug auf nach der Zuordnung verwendete Ressourcen und die Ausführung der Arbeitslast auf der Vorrichtung.

[2759] Beispiel 1156 umfasst den Gegenstand von einem der Beispiele 1149 bis 1155. In Beispiel 1156 gibt die interaktive Analytik eine zeitliche Beziehung zwischen einer ersten Datenprobe und einer zweiten Datenprobe an, wobei die interaktive Analytik ferner eine Beziehung zwischen einer ersten Probe und einem ersten Sensor und der zweiten Datenprobe und einem zweiten Sensor angibt.

[2760] Beispiel 1157 umfasst den Gegenstand von einem der Beispiele 1149 bis 1156. In Beispiel 1157 umfasst die Arbeitslast ein Profil von Rechenanforderungen, Speichieranforderungen und Netzanforderungen der Arbeitslast.

[2761] Beispiel 1158 umfasst den Gegenstand von einem der Beispiele 1149 bis 1157. In Beispiel 1158 umfasst die Arbeitslast mehrere Teilarbeitslasten an der Vorrichtung, wobei die Arbeitslast eine durch Sortieren

der Arbeitslasten in Bezug auf eine benötigte Ressource und eine verfügbare Ressource und Paaren einer Arbeitslast mit einer Vorrichtung auf der Grundlage der benötigten Ressource und der verfügbaren Ressource erhaltene Übereinstimmung umfasst.

[2762] Beispiel 1159 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, Daten auf der Grundlage eines repräsentativen Datensatzes aus einer Umgebung in eine ausgewählte interaktive Analytikumgebung zu laden, wobei die interaktive Analytikumgebung wenigstens eine von einer Daten-Mining-Umgebung und einer Datenfusionsumgebung ist, festzustellen, ob ein vorgeschlagenes Modell mit Daten übereinstimmt, wobei das vorgeschlagene Modell eine Arbeitslast zerlegbarer Abschnitte zur Verarbeitung auf mehreren Vorrichtungen aufweisen soll, und die Arbeitslast einer Vorrichtung zur Ausführung durch die Vorrichtung zuzuordnen.

[2763] Beispiel 1160 umfasst den Gegenstand von Beispiel 1159. In Beispiel 1160 nimmt die Daten-Mining-Umgebung Daten von einer Anwendungsdomain und von einer gespeicherten Wissensdatenbank entsprechend dem repräsentativen Datensatz aus einer Umgebung auf.

[2764] Beispiel 1161 umfasst den Gegenstand von Beispiel 1159 oder 1160. In Beispiel 1161 werden die Daten vor dem Laden in eine ausgewählte interaktive Analytikumgebung gereinigt und organisiert.

[2765] Beispiel 1162 umfasst den Gegenstand von einem der Beispiele 1159 bis 1161. In Beispiel 1162 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Arbeitslast vor der Zuordnung zur Vorrichtung für die Ausführung durch die Vorrichtung zu zerlegen.

[2766] Beispiel 1163 umfasst den Gegenstand von einem der Beispiele 1159 bis 1162. In Beispiel 1163 wird die Arbeitslast durch Parallelität und eine modulare seriell abhängige Aufgabe zerlegt.

[2767] Beispiel 1164 umfasst den Gegenstand von einem der Beispiele 1159 bis 1163. In Beispiel 1164 wird die Arbeitslast für eine Klassifizierungsaufgabe zerlegt, wobei eine Vorverarbeitungs-, eine Merkmalsextraktions- und eine Klassifizierungsaufgabe getrennt werden können.

[2768] Beispiel 1165 umfasst den Gegenstand von einem der Beispiele 1159 bis 1164. In Beispiel 1165 umfasst das maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, die Funktionsweise einer Plattform in Bezug auf die nach der Zuordnung verwendeten Ressourcen und die Ausführung der Arbeitslast auf der Vorrichtung zu berechnen.

[2769] Beispiel 1166 umfasst den Gegenstand von einem der Beispiele 1159 bis 1165. In Beispiel 1166 gibt die interaktive Analytik eine zeitliche Beziehung zwischen einer ersten Datenprobe und einer zweiten Datenprobe an, wobei die interaktive Analytik ferner eine Beziehung zwischen einer ersten Probe und einem ersten Sensor und der zweiten Datenprobe und einem zweiten Sensor angibt.

[2770] Beispiel 1167 umfasst den Gegenstand von einem der Beispiele 1159 bis 1166. In Beispiel 1167 umfasst die Arbeitslast ein Profil von Rechenanforderungen, Speichieranforderungen und Netzanforderungen der Arbeitslast.

[2771] Beispiel 1168 umfasst den Gegenstand von einem der Beispiele 1159 bis 1167. In Beispiel 1168 umfasst die Arbeitslast mehrere Teilarbeitslasten an der Vorrichtung, wobei die Arbeitslast eine durch Sortieren der Arbeitslasten in Bezug auf eine benötigte Ressource und eine verfügbare Ressource und Paaren einer Arbeitslast mit einer Vorrichtung auf der Grundlage der benötigten Ressource und der verfügbaren Ressource erhaltene Übereinstimmung umfasst.

[2772] Beispiel 1169 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen IoT-Netztopologieidentifizierer zur Identifikation einer IoT-Netztopologie, die Verbindungen zwischen einer Anzahl von IoT-Knoten in einem IoT-Netz zeigt, einen IoT-Knotenressourcenidentifizierer zur Identifikation von IoT-Knotenressourcen jedes in der IoT-Netztopologie identifizierten IoT-Knotens, einen Identifizierer der neuronalen Netztopologie zur Identifikation der neuronalen Netztopologie von Knotenabständen und Knotenorten, einen Zuordnungsoptimierer zur Optimierung einer Zuordnung auf der Grundlage der IoT-Knotenressourcen, der Knotenabstände und der Knotenorte und einen Prozessor für zerlegbare Aufgaben zur Verarbeitung einer zerlegbaren Aufgabe in den mehreren IoT-Knoten auf der Grundlage einer Überlagerung der neuronalen Netztopologie auf dem IoT-Netz.

[2773] Beispiel 1170 umfasst den Gegenstand von Beispiel 1169. In Beispiel 1170 ist das neuronale Netz ein künstliches neuronales Netz.

[2774] Beispiel 1171 umfasst den Gegenstand von Beispiel 1169 oder 1170. In Beispiel 1171 bewahrt die optimierte Zuordnung den Ort der Verarbeitung der zerlegbaren Aufgabe über das IoT-Netz durch die Verwendung der Knotenorte der mehreren Knoten zur Identifikation eines Knotens oder der mehreren Knoten, die sich am selben physikalischen Ort befinden.

[2775] Beispiel 1172 umfasst den Gegenstand von einem der Beispiele 1169 bis 1171. In Beispiel 1172 zeigt die IoT-Netztopologie Knoteneigenschaften, einschließlich Abständen zwischen Knoten, einer Clusterbildung, Dispersionsinformationen, empfangener Signalstärken und/oder Signal-Rausch-Verhältnisse.

[2776] Beispiel 1173 umfasst den Gegenstand von einem der Beispiele 1169 bis 1172. In Beispiel 1173 umfasst die Identifikation der IoT-Netztopologie das Bestimmen der Nähe der mehreren Knoten zueinander, der aktuellen Energieniveaus der mehreren Knoten und eine Signalmessung für die mehreren Knoten.

[2777] Beispiel 1174 umfasst den Gegenstand von einem der Beispiele 1169 bis 1173. In Beispiel 1174 kann das Abrufen der Signalmessung der mehreren Knoten durch Abrufen des Indikators der empfangenen Signalstärke oder der Ausstrahlungsleistung geschehen.

[2778] Beispiel 1175 umfasst den Gegenstand von einem der Beispiele 1169 bis 1174. In Beispiel 1175 umfassen die IoT-Knotenressourcen das Energieniveau, den verfügbaren Speicherplatz, die aktuelle Verarbeitungslast, die Netzkapazitäten, die verfügbare Betriebszeit oder Knotenzuverlässigkeitsinformationen.

[2779] Beispiel 1176 umfasst den Gegenstand von einem der Beispiele 1169 bis 1175. In Beispiel 1176 umfasst die Verarbeitung einer zerlegbaren Aufgabe in den mehreren IoT-Knoten das Senden von Teilen der zerlegbaren Aufgabe zu den mehreren IoT-Knoten auf der Grundlage davon, ob die IoT-Knoten als sich an einem physikalischen Ort auf demselben Netz befindend identifiziert wurden.

[2780] Beispiel 1177 umfasst den Gegenstand von einem der Beispiele 1169 bis 1176. In Beispiel 1177 umfasst die Überlagerung der Topologie des neuronalen Netzes drei Schichten, einschließlich einer Eingangsschicht, einer verborgenen Schicht und einer Ausgangsschicht.

[2781] Beispiel 1178 umfasst den Gegenstand von einem der Beispiele 1169 bis 1177. In Beispiel 1178 umfasst das Optimieren einer Zuordnung eine Übertragungszeit für die Übertragung von Informationen von Eingangsknoten zu Ausgangsknoten.

[2782] Beispiel 1179 umfasst ein Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur verteilten Zuordnung eines neuronalen Netzes und zur Ressourcenverwaltung. Das Verfahren zur Verwendung einer Internet-der-Dinge(IoT)-Vorrichtung zur verteilten Zuordnung eines neuronalen Netzes und zur Ressourcenverwaltung umfasst das Identifizieren einer IoT-Netztopologie, die Verbindungen zwischen mehreren IoT-Knoten in einem IoT-Netz zeigt, das Identifizieren von IoT-Knotenressourcen jedes in der IoT-Netztopologie identifizierten IoT-Knotens, das Identifizieren einer Topologie des neuronalen Netzes von Knotenabständen und Knotenorten, das Optimieren einer Zuordnung auf der Grundlage der IoT-Knotenressourcen, der Knotenabstände und der Knotenorte und die Verarbeitung einer zerlegbaren Aufgabe in den mehreren IoT-Knoten auf der Grundlage einer Überlagerung der Topologie des neuronalen Netzes über dem IoT-Netz.

[2783] Beispiel 1180 umfasst den Gegenstand von Beispiel 1179. In Beispiel 1180 ist das neuronale Netz ein künstliches neuronales Netz.

[2784] Beispiel 1181 umfasst den Gegenstand von Beispiel 1179 oder 1180. In Beispiel 1181 bewahrt die optimierte Zuordnung den Ort der Verarbeitung der zerlegbaren Aufgabe über das IoT-Netz durch die Verwendung der Knotenorte der mehreren Knoten zur Identifikation eines Knotens oder der mehreren Knoten, die sich am selben physikalischen Ort befinden.

[2785] Beispiel 1182 umfasst den Gegenstand von einem der Beispiele 1179 bis 1181. In Beispiel 1182 zeigt die IoT-Netztopologie Knoteneigenschaften, einschließlich Abständen zwischen Knoten, einer Clusterbildung, Dispersionsinformationen, empfangener Signalstärken und Signal-Rausch-Verhältnisse.

[2786] Beispiel 1183 umfasst den Gegenstand von einem der Beispiele 1179 bis 1182. In Beispiel 1183 umfasst die Identifikation der IoT-Netztopologie das Bestimmen der Nähe der mehreren Knoten zueinander, der aktuellen Energieniveaus der mehreren Knoten und eine Signalmessung für die mehreren Knoten.

[2787] Beispiel 1184 umfasst den Gegenstand von einem der Beispiele 1179 bis 1183. In Beispiel 1184 kann das Abrufen der Signalmessung der mehreren Knoten durch Abrufen des Indikators der empfangenen Signalstärke oder der Ausstrahlungsleistung geschehen.

[2788] Beispiel 1185 umfasst den Gegenstand von einem der Beispiele 1179 bis 1184. In Beispiel 1185 umfassen die IoT-Knotenressourcen das Energieniveau, den verfügbaren Speicherplatz, die aktuelle Verarbeitungslast, die Netzkapazitäten, die verfügbare Betriebszeit oder Knotenzuverlässigkeitsinformationen.

[2789] Beispiel 1186 umfasst den Gegenstand von einem der Beispiele 1179 bis 1185. In Beispiel 1186 umfasst die Verarbeitung einer zerlegbaren Aufgabe in den mehreren IoT-Knoten das Senden von Teilen der zerlegbaren Aufgabe zu den mehreren IoT-Knoten auf der Grundlage davon, ob die IoT-Knoten als sich an einem physikalischen Ort auf demselben Netz befindend identifiziert wurden.

[2790] Beispiel 1187 umfasst den Gegenstand von einem der Beispiele 1179 bis 1186. In Beispiel 1187 umfasst die Überlagerung der Topologie des neuronalen Netzes drei Schichten, einschließlich einer Eingangsschicht, einer verborgenen Schicht und einer Ausgangsschicht.

[2791] Beispiel 1188 umfasst den Gegenstand von einem der Beispiele 1179 bis 1187. In Beispiel 1188 umfasst das Optimieren einer Zuordnung eine Übertragungszeit für die Übertragung von Informationen von Eingangsknoten zu Ausgangsknoten.

[2792] Beispiel 1189 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine IoT-Netztopologie zu identifizieren, die Verbindungen zwischen mehreren IoT-Knoten in einem IoT-Netz zeigt, IoT-Knotenressourcen jedes in der IoT-Netztopologie identifizierten IoT-Knotens zu identifizieren, eine Topologie eines neuronalen Netzes von Knotenabständen und Knotenorten zu identifizieren, eine Zuordnung auf der Grundlage der IoT-Knotenressourcen, der Knotenabstände und der Knotenorte zu optimieren und eine zerlegbare Aufgabe in den mehreren IoT-Knoten auf der Grundlage einer Überlagerung der Topologie des neuronalen Netzes auf dem IoT-Netz zu verarbeiten.

[2793] Beispiel 1190 umfasst den Gegenstand von Beispiel 1189. In Beispiel 1190 ist das neuronale Netz ein künstliches neuronales Netz.

[2794] Beispiel 1191 umfasst den Gegenstand von Beispiel 1189 oder 1190. In Beispiel 1191 bewahrt die optimierte Zuordnung den Ort der Verarbeitung der zerlegbaren Aufgabe über das IoT-Netz durch die Verwendung der Knotenorte der mehreren Knoten zur Identifikation eines Knotens oder der mehreren Knoten, die sich am selben physikalischen Ort befinden.

[2795] Beispiel 1192 umfasst den Gegenstand von einem der Beispiele 1189 bis 1191. In Beispiel 1192 zeigt die IoT-Netztopologie Knoteneigenschaften, einschließlich Abständen zwischen Knoten, einer Clusterbildung, Dispersionsinformationen, empfangener Signalstärken und Signal-Rausch-Verhältnisse.

[2796] Beispiel 1193 umfasst den Gegenstand von einem der Beispiele 1189 bis 1192. In Beispiel 1193 umfasst die Identifikation der IoT-Netztopologie das Bestimmen der Nähe der mehreren Knoten zueinander, der aktuellen Energieniveaus der mehreren Knoten und eine Signalmessung für die mehreren Knoten.

[2797] Beispiel 1194 umfasst den Gegenstand von einem der Beispiele 1189 bis 1193. In Beispiel 1194 kann das Abrufen der Signalmessung der mehreren Knoten durch Abrufen des Indikators der empfangenen Signalstärke oder der Ausstrahlungsleistung geschehen.

[2798] Beispiel 1195 umfasst den Gegenstand von einem der Beispiele 1189 bis 1194. In Beispiel 1195 umfassen die IoT-Knotenressourcen das Energieniveau, den verfügbaren Speicherplatz, die aktuelle Verarbeitungslast, die Netzkapazitäten, die verfügbare Betriebszeit oder Knotenzuverlässigkeitsinformationen.

[2799] Beispiel 1196 umfasst den Gegenstand von einem der Beispiele 1189 bis 1195. In Beispiel 1196 umfasst die Verarbeitung einer zerlegbaren Aufgabe in den mehreren IoT-Knoten das Senden von Teilen der

zerlegbaren Aufgabe zu den mehreren IoT-Knoten auf der Grundlage davon, ob die IoT-Knoten als sich an einem physikalischen Ort auf demselben Netz befindend identifiziert wurden.

[2800] Beispiel 1197 umfasst den Gegenstand von einem der Beispiele 1189 bis 1196. In Beispiel 1197 umfasst die Überlagerung der Topologie des neuronalen Netzes drei Schichten, einschließlich einer Eingangsschicht, einer verborgenen Schicht und einer Ausgangsschicht.

[2801] Beispiel 1198 umfasst den Gegenstand von einem der Beispiele 1189 bis 1197. In Beispiel 1198 umfasst das Optimieren einer Zuordnung eine Übertragungszeit für die Übertragung von Informationen von Eingangsknoten zu Ausgangsknoten.

[2802] Beispiel 1199 umfasst eine Vorrichtung, die eine Internet-der-Dinge(IoT)-Vorrichtung umfasst. Die Vorrichtung umfasst eine Internet-der-Dinge(IoT)-Vorrichtung. Die IoT-Vorrichtung umfasst eine Blockchain-Logik zum Unterhalten einer Blockchain in der IoT-Vorrichtung und zum Übertragen der Blockchain über andere IoT-Vorrichtungen, einen Merkle-Baum, der Hash-Codeeinträge in Zusammenhang mit jedem Block in der Blockchain umfasst, wobei ein Eintrag im Merkle-Baum eine Referenz zu einem Merkle-Baum niedrigerer Ebene in Zusammenhang mit einer Blockchain niedrigerer Ebene für ein Netz niedrigerer Ebene umfasst, und einen Lokalisierer zum Durchsuchen des Merkle-Baums nach einem Ziel-Hash-Code zum Lokalisieren eines Zielblocks in der Blockchain, wobei, falls der Lokalisierer die Referenz zum Merkle-Baum niedrigerer Ebene findet, der Lokalisierer den Merkle-Baum niedrigerer Ebene nach dem Zielblock durchsuchen soll.

[2803] Beispiel 1200 umfasst den Gegenstand von Beispiel 1199. In Beispiel 1200 umfasst die Referenz eine Netzreferenz mit einer Verbindung zum Merkle-Baum niedrigerer Ebene im Netz niedrigerer Ebene.

[2804] Beispiel 1201 umfasst den Gegenstand von Beispiel 1199 oder 1200. In Beispiel 1201 umfasst die Referenz eine Verbindung zu einer Kopie des Merkle-Baums niedrigerer Ebene in der IoT-Vorrichtung.

[2805] Beispiel 1202 umfasst den Gegenstand von einem der Beispiele 1199 bis 1201. In Beispiel 1202 umfasst die Vorrichtung einen Indexierer zur Bildung des Merkle-Baums für die Blockchain.

[2806] Beispiel 1203 umfasst den Gegenstand von einem der Beispiele 1199 bis 1202. In Beispiel 1203 umfasst die Vorrichtung einen Cache-Erzeuger zur Bildung eines kohärenten Caches in der IoT-Vorrichtung, wobei der kohärente Cache eine Kopie des Merkle-Baums niedrigerer Ebene umfasst.

[2807] Beispiel 1204 umfasst den Gegenstand von einem der Beispiele 1199 bis 1203. In Beispiel 1204 umfasst die Vorrichtung einen Cache-Agenten zur Unterhaltung eines kohärenten Caches in der IoT-Vorrichtung, wobei der Cache-Agent Benachrichtigungen von einer Vorrichtung im Netz niedrigerer Ebene subskribiert, um den Cache-Agenten über Cache-Kohärenzereignisse zu informieren.

[2808] Beispiel 1205 umfasst den Gegenstand von einem der Beispiele 1199 bis 1204. In Beispiel 1205 soll der Cache-Agent Cache-Kohärenzereignisse einem Cache-Agenten höherer Ebene in einem Netz höherer Ebene bekannt machen.

[2809] Beispiel 1206 umfasst ein Verfahren zum Lokalisieren eines Blocks in einer hierarchischen Blockchain. Das Verfahren zum Lokalisieren eines Blocks in einer hierarchischen Blockchain umfasst das Erhalten eines Hash-Werts für den Block, das Setzen einer aktuellen Blockchain auf eine Root einer hierarchischen Blockchain auf einer aktuellen Netzebene, das Vergleichen des Block-Hash-Werts mit Werten in einem Merkle-Baum für die aktuelle Blockchain, das Feststellen, ob der Hash-Wert einem Sync-Block für eine nachgeordnete Blockchain zugeordnet ist, und, falls dies der Fall ist, das Setzen der aktuellen Blockchain auf die nachgeordnete Blockchain, das Lokalisieren des Blocks in der aktuellen Blockchain und das Abrufen des Blocks.

[2810] Beispiel 1207 umfasst den Gegenstand von Beispiel 1206. In Beispiel 1207 umfasst das Erhalten des Hash-Werts das Nachschlagen des Hash-Werts in einer Tabelle.

[2811] Beispiel 1208 umfasst den Gegenstand von Beispiel 1206 oder 1207. In Beispiel 1208 umfasst das Erhalten des Hash-Werts das Berechnen des Hash-Werts.

[2812] Beispiel 1209 umfasst den Gegenstand von einem der Beispiele 1206 bis 1208. In Beispiel 1209 umfasst das Setzen der aktuellen Blockchain auf die nachgeordnete Blockchain das Setzen einer Netzreferenz auf eine Blockchain niedrigerer Ebene und das Fortsetzen einer Suche in der Blockchain niedrigerer Ebene.

[2813] Beispiel 1210 umfasst den Gegenstand von einem der Beispiele 1206 bis 1209. In Beispiel 1210 umfasst das Setzen der aktuellen Blockchain auf die nachgeordnete Blockchain das Setzen eines Werts für einen Zeiger auf eine Kopie der Blockchain niedrigerer Ebene in einem kohärenten Cache.

[2814] Beispiel 1211 umfasst den Gegenstand von einem der Beispiele 1206 bis 1210. In Beispiel 1211 umfasst das Lokalisieren des Blocks in der aktuellen Blockchain die Feststellung, dass der Block-Hash-Wert mit einem Wert in einem Merkle-Baum für die aktuelle Blockchain übereinstimmt, und das Erhalten eines Zeigers auf den Block vom Merkle-Baum.

[2815] Beispiel 1212 umfasst den Gegenstand von einem der Beispiele 1206 bis 1211. In Beispiel 1212 umfasst das Verfahren das Bilden der hierarchischen Blockchain, einschließlich des Schreibens von Transaktionsdaten in eine aktuelle Blockchain für ein lokales IoT-Netz, das Feststellen, ob ein Block ein Sync-Block ist, und, falls dies der Fall ist, das Bilden einer den Hash des Sync-Blocks enthaltenden Übertragung, das Senden des Hashes des Sync-Blocks zu einem IoT-Netz nächsthöherer Ebene und das Übermitteln der Übertragung zu einem aktuellen Block in dem Netz nächsthöherer Ebene.

[2816] Beispiel 1213 umfasst den Gegenstand von einem der Beispiele 1206 bis 1212. In Beispiel 1213 umfasst das Verfahren das Bilden eines kohärenten Caches der hierarchischen Blockchain, einschließlich des Subskribierens der aktuellen Blockchain für einen veröffentlichenden Cache-Agenten in einer nachgeordneten Blockchain, das Akzeptieren einer Registrierung der aktuellen Blockchain in der nachgeordneten Blockchain und das Warten auf Veröffentlichungsereignisse.

[2817] Beispiel 1214 umfasst den Gegenstand von einem der Beispiele 1206 bis 1213. In Beispiel 1214 umfasst das Verfahren das Aufrechterhalten eines kohärenten Caches der hierarchischen Blockchain, einschließlich des Empfangens einer Benachrichtigung eines Cache-Kohärenzereignisses in einer nachgeordneten Blockchain, das Kopieren eines Merkle-Baumwegs von der nachgeordneten Blockchain und das Veröffentlichen von ihm bei einem Cache-Agenten an einer Subskribenten-Blockchain und das Ersetzen eines der nachgeordneten Blockchain entsprechenden aktuellen gecachten Merkle-Baumwegs.

[2818] Beispiel 1215 umfasst den Gegenstand von einem der Beispiele 1206 bis 1214. In Beispiel 1215 umfasst das Verfahren das Feststellen, ob der Weg einen neuen Zweig des Merkle-Baumwegs bildet, und, falls dies der Fall ist, das Bilden einer neuen lokalen Root in der aktuellen Blockchain, das Machen der aktuellen Blockchain gleich der lokalen Root und das Wiederholen des Bildens der neuen lokalen Root, bis die lokale Root gleich einer globalen Root ist.

[2819] Beispiel 1216 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine hierarchische Blockchain und einen hierarchischen Index der hierarchischen Blockchain zu bilden.

[2820] Beispiel 1217 umfasst den Gegenstand von Beispiel 1216. In Beispiel 1217 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Referenz zu einer Blockchain niedrigerer Ebene in den hierarchischen Index einzufügen.

[2821] Beispiel 1218 umfasst den Gegenstand von Beispiel 1216 oder 1217. In Beispiel 1218 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen kohärenten Cache von Merkle-Bäumen auf niedrigeren Ebenen in einem Internet-der-Dinge(IoT)-Netz zu bilden.

[2822] Beispiel 1219 umfasst den Gegenstand von einem der Beispiele 1216 bis 1218. In Beispiel 1219 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Merkle-Bäume von niedrigeren Ebenen in einem Internet der Dinge (IoT) zu einer aktuellen Ebene im IoT-Netz zu kopieren.

[2823] Beispiel 1220 umfasst den Gegenstand von einem der Beispiele 1216 bis 1219. In Beispiel 1220 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Kohärenz eines Caches von Merkle-Bäumen von niedrigeren Ebenen in einem IoT-Netz aufrechtzuerhalten.

[2824] Beispiel 1221 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst eine Bloomfilter-Themenliste mit

einem Hash-Code eines Themas, einen Subskriptions-Manager zum Vergleichen der Bloomfilter-Themenliste mit einer Liste verfügbaren Inhalts und einen Inhaltslokalisierer zum Bereitstellen von Inhalt in Zusammenhang mit einem Hash-Code.

[2825] Beispiel 1222 umfasst den Gegenstand von Beispiel 1221. In Beispiel 1222 umfasst die IoT-Vorrichtung eine Whitelist-Maske unter Einschluss von Hash-Codes erlaubten Inhalts.

[2826] Beispiel 1223 umfasst den Gegenstand von Beispiel 1221 oder 1222. In Beispiel 1223 umfasst die IoT-Vorrichtung eine Blacklist-Maske unter Einschluss von Hash-Codes verbotenen Inhalts.

[2827] Beispiel 1224 umfasst den Gegenstand von einem der Beispiele 1221 bis 1223. In Beispiel 1224 umfasst die IoT-Vorrichtung die Aufnahme eines Hash-Codeberechners zum Erhalten eines Hash-Codes für ein Thema.

[2828] Beispiel 1225 umfasst den Gegenstand von einem der Beispiele 1221 bis 1224. In Beispiel 1225 umfasst die IoT-Vorrichtung die Aufnahme eines vertrauenswürdigen Plattformmoduls zum Einrichten einer vertrauenswürdigen Ausführungsumgebung.

[2829] Beispiel 1226 umfasst den Gegenstand von einem der Beispiele 1221 bis 1225. In Beispiel 1226 umfasst die IoT-Vorrichtung einen Subskribenten, dem Inhalt von einer anderen Vorrichtung bereitgestellt wird.

[2830] Beispiel 1227 umfasst den Gegenstand von einem der Beispiele 1221 bis 1226. In Beispiel 1227 umfasst die IoT-Vorrichtung einen Veröffentlichler, der einer anderen Vorrichtung Inhalt bereitstellt.

[2831] Beispiel 1228 umfasst den Gegenstand von einem der Beispiele 1221 bis 1227. In Beispiel 1228 umfasst die IoT-Vorrichtung einen Router, der mehreren Vorrichtungen Inhalt bereitstellt.

[2832] Beispiel 1229 umfasst den Gegenstand von einem der Beispiele 1221 bis 1228. In Beispiel 1229 umfassen die mehreren Vorrichtungen Subskribenten, Veröffentlichler oder Router oder Kombinationen davon.

[2833] Beispiel 1230 umfasst ein Verfahren zum Implementieren eines Veröffentlichler-Subskribieren-Inhaltsverteilungsmodells unter Verwendung von Bloomfiltern. Das Verfahren zum Implementieren eines Veröffentlichler-Subskribieren-Inhaltsverteilungsmodells unter Verwendung von Bloomfiltern umfasst das Empfangen einer Registrierung eines Subskribenten-Bloomfilters, einschließlich eines Hash-Codes eines Themas, das Berechnen eines Inhaltsabgriffs des Subskribenten-Bloomfilters mit einem Veröffentlichler-Bloomfilter, einschließlich mehrerer Hash-Codes von Themen, und das Bereitstellen von Inhalt für einen Subskribenten für den Hash-Code, falls der Inhaltsabgriff eine Bitübereinstimmung zwischen im Subskribenten-Bloomfilter gesetzten Bits und im Veröffentlichler-Bloomfilter gesetzten Bits angibt.

[2834] Beispiel 1231 umfasst den Gegenstand von Beispiel 1230. In Beispiel 1231 umfasst das Verfahren das Berechnen eines Whitelist-Abgriffs des Subskribenten-Bloomfilters mit einer Whitelist-Maske, einschließlich mehrerer Hash-Codes erlaubter Themen.

[2835] Beispiel 1232 umfasst den Gegenstand von Beispiel 1230 oder 1231. In Beispiel 1232 umfasst das Verfahren das Berechnen des Whitelist-Abgriffs durch Ausführen einer UND-Funktion zwischen dem Subskribenten-Bloomfilter und der Whitelist-Maske.

[2836] Beispiel 1233 umfasst den Gegenstand von einem der Beispiele 1230 bis 1232. In Beispiel 1233 umfasst das Verfahren das Berechnen eines Blacklist-Abgriffs des Subskribenten-Bloomfilters mit einer Blacklist-Maske, die mehrere Hash-Codes verbotener Themen umfasst.

[2837] Beispiel 1234 umfasst den Gegenstand von einem der Beispiele 1230 bis 1233. In Beispiel 1234 umfasst das Verfahren das Berechnen des Blacklist-Abgriffs durch Berechnen eines Zwischen-Bloomfilters als NAND-Funktion zwischen dem Subskribenten-Bloomfilter und der Blacklist-Maske und das Berechnen des Blacklist-Abgriffs als UND-Funktion zwischen dem Zwischen-Bloomfilter und dem Subskribenten-Bloomfilter.

[2838] Beispiel 1235 umfasst den Gegenstand von einem der Beispiele 1230 bis 1234. In Beispiel 1235 umfasst das Verfahren das Bereitstellen des Inhalts für den Subskribenten, falls ein Whitelist-Abgriff eine Bitübereinstimmung zwischen gesetzten Bits im Subskribenten-Bloomfilter und gesetzten Bits in einer Whitelist-Mas-

ke angibt, und gibt ein Blacklist-Abgriff keine Bitübereinstimmung zwischen gesetzten Bits im Subskribenten-Bloomfilter und gesetzten Bits in einer Blacklist-Maske an.

[2839] Beispiel 1236 umfasst den Gegenstand von einem der Beispiele 1230 bis 1235. In Beispiel 1236 umfasst das Verfahren das Erzeugen einer Bloomfilter-Themenliste durch Berechnen eines Hash-Codes jedes von mehreren Themen und das Überschreiben eines Bloomfilters mit dem Hash-Code für jedes der mehreren Themen.

[2840] Beispiel 1237 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, ein Subskriptions-Bloomfilter zu registrieren, einen Schnitt des Subskriptions-Bloomfilters mit einem Inhaltsfilter zu berechnen und, falls der Schnitt nicht null ist, Inhalt in Zusammenhang mit dem Subskriptions-Bloomfilter bereitzustellen.

[2841] Beispiel 1238 umfasst den Gegenstand von Beispiel 1237. In Beispiel 1238 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Whitelist-Maske zu registrieren, die erlaubten Inhalt angibt, und eine Blacklist-Maske zu registrieren, die verbotenen Inhalt angibt.

[2842] Beispiel 1239 umfasst den Gegenstand von Beispiel 1237 oder 1238. In Beispiel 1239 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass ein Hash-Code des Subskriptions-Bloomfilters einen von Null verschiedenen Abgriff mit der Whitelist-Maske hat, und, falls dies der Fall ist, den Inhalt bereitzustellen.

[2843] Beispiel 1240 umfasst den Gegenstand von einem der Beispiele 1237 bis 1239. In Beispiel 1240 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, festzustellen, dass ein Hash-Code des Subskriptions-Bloomfilters einen von Null verschiedenen Abgriff mit der Blacklist-Maske hat, und, falls dies der Fall ist, zu verhindern, dass der Inhalt bereitgestellt wird.

[2844] Beispiel 1241 umfasst den Gegenstand von einem der Beispiele 1237 bis 1240. In Beispiel 1241 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Hash-Code für ein Thema zu bestimmen und den Hash-Code in das Subskriptions-Bloomfilter zu schreiben.

[2845] Beispiel 1242 umfasst den Gegenstand von einem der Beispiele 1237 bis 1241. In Beispiel 1242 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Hash-Code für ein Thema zu bestimmen, eine Exklusiv-ODER-Operation am Hash-Code auszuführen und das Subskriptions-Bloomfilter zu veranlassen, ein neues Bloomfilter zu erzeugen, und das Subskriptions-Bloomfilter durch das neue Bloomfilter zu ersetzen.

[2846] Beispiel 1243 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Die IoT-Vorrichtung umfasst einen Themenklassifizierer zur Feststellung, dass ein Thema verschlüsselten Inhalt umfasst, einen Benachrichtiger zum Senden einer Benachrichtigung zu anderen Vorrichtungen, dass das Thema verschlüsselten Inhalt umfasst, und einen Schlüssel-Subskribenten zum Subskribieren des Themas einschließlich eines Themenschlüssels für den verschlüsselten Inhalt.

[2847] Beispiel 1244 umfasst den Gegenstand von Beispiel 1243. In Beispiel 1244 umfasst das IoT-Netz einen Veröffentlichender zum Bereitstellen des verschlüsselten Inhalt umfassenden Themas.

[2848] Beispiel 1245 umfasst den Gegenstand von Beispiel 1243 oder 1244. In Beispiel 1245 umfasst das IoT-Netz einen Router in Kommunikation mit einem Veröffentlichender, um den verschlüsselten Inhalt für das Thema zu erhalten.

[2849] Beispiel 1246 umfasst den Gegenstand von einem der Beispiele 1243 bis 1245. In Beispiel 1246 soll der Router den Themenschlüssel vom Veröffentlichender für den verschlüsselten Inhalt erhalten.

[2850] Beispiel 1247 umfasst den Gegenstand von einem der Beispiele 1243 bis 1246. In Beispiel 1247 umfasst das IoT-Netz einen Router, der durch eine Kette von Routern in Kommunikation mit einem Veröffentlichender steht.

[2851] Beispiel 1248 umfasst den Gegenstand von einem der Beispiele 1243 bis 1247. In Beispiel 1248 soll der Schlüssel-Subskribent des Routers den Themenschlüssel von einem Peer-Knoten in der Kette von Routern erhalten.

[2852] Beispiel 1249 umfasst den Gegenstand von einem der Beispiele 1243 bis 1248. In Beispiel 1249 umfasst der Router einen Cache für den Themenschlüssel, wobei der Schlüssel-Subskribent präemptiv den Themenschlüssel von einem Peer-Knoten in der Kette von Routern erhält, um den Cache vorzuwärmen.

[2853] Beispiel 1250 umfasst den Gegenstand von einem der Beispiele 1243 bis 1249. In Beispiel 1250 umfasst das IoT-Netz einen Subskribenten in Kommunikation mit einem Router, wobei der Subskribent einen Verbraucher des verschlüsselten Inhalts umfasst.

[2854] Beispiel 1251 umfasst den Gegenstand von einem der Beispiele 1243 bis 1250. In Beispiel 1251 soll der Schlüssel-Subskribent des Subskribenten den Themenschlüssel vom Router erhalten.

[2855] Beispiel 1252 umfasst den Gegenstand von einem der Beispiele 1243 bis 1251. In Beispiel 1252 umfasst der Subskribent einen Entschlüsseler zum Entschlüsseln des verschlüsselten Inhalts unter Verwendung des Themenschlüssels.

[2856] Beispiel 1253 umfasst ein Verfahren zum Behandeln von Verschlüsselungsschlüsseln in einer Veröffentlichung-Subskription-Inhaltsverteilungsumgebung. Das Verfahren zur Behandlung von Verschlüsselungsschlüsseln in einer Veröffentlichung-Subskription-Inhaltsverteilungsumgebung umfasst das Empfangen einer Themenbenachrichtigung, dass ein verschlüsselter Inhalt umfassendes Thema verfügbar ist, das Subskribieren eines Schlüsselbehandlungsthemas für einen Themenschlüssel, das Empfangen einer Benachrichtigung, dass der Themenschlüssel verfügbar ist, und das Anfordern des Themenschlüssels, um einen Cache präemptiv vorzuwärmen.

[2857] Beispiel 1254 umfasst den Gegenstand von Beispiel 1253. In Beispiel 1254 umfasst das Verfahren das Empfangen einer Anforderung von einem Subskribenten für den Themenschlüssel und das Bereitstellen des Themenschlüssels vom Cache.

[2858] Beispiel 1255 umfasst den Gegenstand von Beispiel 1253 oder 1254. In Beispiel 1255 umfasst das Verfahren das Bilden des Schlüsselbehandlungsthemas zur Benachrichtigung von Vorrichtungen in Bezug auf die Verfügbarkeit des Themenschlüssels.

[2859] Beispiel 1256 umfasst den Gegenstand von einem der Beispiele 1253 bis 1255. In Beispiel 1256 umfasst das Verfahren die Feststellung, ob sich der Themenschlüssel im Cache befindet, und, falls dies nicht der Fall ist, das Anfordern des Themenschlüssels von einem Peer-Knoten.

[2860] Beispiel 1257 umfasst den Gegenstand von einem der Beispiele 1253 bis 1256. In Beispiel 1257 umfasst das Verfahren das Bilden eines Bloomfilters, einschließlich Hash-Codes von verfügbaren veröffentlichten Themen.

[2861] Beispiel 1258 umfasst den Gegenstand von einem der Beispiele 1253 bis 1257. In Beispiel 1258 umfasst das Anfordern des Themenschlüssels das Ausgeben eines GET-Befehls, um den Themenschlüssel zu erhalten.

[2862] Beispiel 1259 umfasst den Gegenstand von einem der Beispiele 1253 bis 1258. In Beispiel 1259 umfasst das Verfahren das Verschlüsseln des Themenschlüssels unter Verwendung eines seitenspezifischen Schlüssels zum Schützen des Themenschlüssels vor einem Abfangen während Übertragungen zwischen Weiterleitungsknoten und Subskribentenknoten.

[2863] Beispiel 1260 umfasst den Gegenstand von einem der Beispiele 1253 bis 1259. In Beispiel 1260 umfasst das Verfahren das Herunterladen des verschlüsselten Inhalts von einem öffentlichen Ablagespeicher.

[2864] Beispiel 1261 umfasst den Gegenstand von einem der Beispiele 1253 bis 1260. In Beispiel 1261 umfasst das Verfahren das Entschlüsseln des verschlüsselten Inhalts unter Verwendung des Themenschlüssels.

[2865] Beispiel 1262 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Themenbenachrichtigung über verschlüsselten Inhalt

zu empfangen, ein Schlüsselbehandlungsthema nach Empfang der Themenbenachrichtigung zu subskribieren und den Themenschlüssel nach dem Empfang einer Schlüsselbenachrichtigung, dass der Themenschlüssel verfügbar ist, von einem Peer-Knoten zu erhalten.

[2866] Beispiel 1263 umfasst den Gegenstand von Beispiel 1262. In Beispiel 1263 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, ein Bloomfilter zu erzeugen, das Hash-Codes verfügbaren Inhalts umfasst.

[2867] Beispiel 1264 umfasst den Gegenstand von Beispiel 1262 oder 1263. In Beispiel 1264 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, die Themenbenachrichtigung zu einem Peer-Router zu senden.

[2868] Beispiel 1265 umfasst den Gegenstand von einem der Beispiele 1262 bis 1264. In Beispiel 1265 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Subskribenten über das verschlüsselten Inhalt umfassende Thema zu benachrichtigen.

[2869] Beispiel 1266 umfasst den Gegenstand von einem der Beispiele 1262 bis 1265. In Beispiel 1266 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Themenschlüssel zu einem Subskribenten zu senden, nachdem eine Anforderung des Themenschlüssels vom Subskribenten empfangen wurde.

[2870] Beispiel 1267 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine IoT-Vorrichtung umfasst. Diese umfasst auch einen Attestierer zum Bereitstellen eines Gruppenmitgliedschafts-Berechtigungsnachweises für die IoT-Vorrichtung zu einem Schlüsselverteilungszentrum und einen Subskribenten zum Zuführen eines Hash-Code eines interessierenden Themas umfassenden Bloomfilters und zum Empfangen eines Schlüssels vom Schlüsselverteilungszentrum, um interessierenden Inhalt zu entschlüsseln.

[2871] Beispiel 1268 umfasst den Gegenstand von Beispiel 1267. In Beispiel 1268 umfasst das interessierende Thema ein öffentliches Thema, ein privates Thema, ein Schlüsselbehandlungsthema oder ein Sicherheitsniveauthema oder Kombinationen davon.

[2872] Beispiel 1269 umfasst den Gegenstand von Beispiel 1267 oder 1268. In Beispiel 1269 umfasst die Vorrichtung einen Themenbenennungsserver zum Anmelden einer Vorrichtung an einem Thema und zur Ausgabe des Gruppenmitgliedschafts-Berechtigungsnachweises an die IoT-Vorrichtung.

[2873] Beispiel 1270 umfasst den Gegenstand von einem der Beispiele 1267 bis 1269. In Beispiel 1270 umfasst die Vorrichtung das Schlüsselverteilungszentrum zur Überprüfung der Identität der IoT-Vorrichtung und zum Bereitstellen eines Themengruppenschlüssels für die IoT-Vorrichtung.

[2874] Beispiel 1271 umfasst den Gegenstand von einem der Beispiele 1267 bis 1270. In Beispiel 1271 umfasst die Vorrichtung einen Integritätserzwinger zum Blockieren von Leseoperationen zu einem niedrigeren Niveau, von Schreiboperationen zu einem höheren Niveau oder von beiden.

[2875] Beispiel 1272 umfasst den Gegenstand von einem der Beispiele 1267 bis 1271. In Beispiel 1272 kann ein Niveau ein Sicherheitsniveau, eine Netzebene oder beide sein.

[2876] Beispiel 1273 umfasst den Gegenstand von einem der Beispiele 1267 bis 1272. In Beispiel 1273 umfasst die Vorrichtung einen Vertraulichkeitserzwinger zum Blockieren von Leseoperationen zu einem höheren Niveau, von Schreiboperationen zu einem niedrigeren Niveau oder von beiden.

[2877] Beispiel 1274 umfasst den Gegenstand von einem der Beispiele 1267 bis 1273. In Beispiel 1274 kann ein Niveau ein Sicherheitsniveau, eine Netzebene oder beide sein.

[2878] Beispiel 1275 umfasst ein Verfahren zur Behandlung der Themenverschlüsselung in einer Veröffentlichung-Subskription-Umgebung. Das Verfahren zur Behandlung der Themenverschlüsselung in einer Veröffentlichung-Subskription-Umgebung umfasst das Subskribieren eines verschlüsselten Themas an einem Router, das Empfangen einer Benachrichtigung des verschlüsselten Themas, das verschlüsselten Inhalt umfasst, das Senden einer Subskribentenattestierungsnachricht zu einem Schlüsselverteilungszentrum (KDC), das

Empfangen eines Themenverschlüsselungsschlüssels vom KDC und das Entschlüsseln des verschlüsselten Inhalts unter Verwendung des Themenverschlüsselungsschlüssels.

[2879] Beispiel 1276 umfasst den Gegenstand von Beispiel 1275. In Beispiel 1276 umfasst das Verfahren das Empfangen einer Benachrichtigung für das verschlüsselte Thema, einschließlich weiteren verschlüsselten Inhalts, und das Entschlüsseln des weiteren verschlüsselten Inhalts unter Verwendung des Themenverschlüsselungsschlüssels.

[2880] Beispiel 1277 umfasst den Gegenstand von Beispiel 1275 oder 1276. In Beispiel 1277 umfasst das Verfahren das Erzeugen des Themenverschlüsselungsschlüssels, das Übermitteln des Themenverschlüsselungsschlüssels zum KDC in einer Veröffentlicherrattestierungsnachricht und das Veröffentlichen verschlüsselten Inhalts für einen Router.

[2881] Beispiel 1278 umfasst den Gegenstand von einem der Beispiele 1275 bis 1277. In Beispiel 1278 umfasst das Verfahren das Empfangen einer Subskription für das verschlüsselte Thema von einem Subskribenten, das Empfangen einer Veröffentlichung des verschlüsselten Themas von einem Veröffentlicherr und das Bereitstellen einer Benachrichtigung der Veröffentlichung mit dem verschlüsselten Inhalt für den Subskribenten.

[2882] Beispiel 1279 umfasst den Gegenstand von einem der Beispiele 1275 bis 1278. In Beispiel 1279 umfasst das Verfahren das Erzwingen einer Integritätsrichtlinie durch Verhindern eines Herunterlesens zu einem niedrigeren Niveau, eines Heraufschreibens zu einem höheren Niveau oder von beiden.

[2883] Beispiel 1280 umfasst den Gegenstand von einem der Beispiele 1275 bis 1279. In Beispiel 1280 ist ein Niveau ein Sicherheitsniveau, eine Netzebene oder beides.

[2884] Beispiel 1281 umfasst den Gegenstand von einem der Beispiele 1275 bis 1280. In Beispiel 1281 umfasst das Verfahren das Erzwingen einer Vertraulichkeitsrichtlinie durch Verhindern eines Hochlesens zu einem höheren Niveau, eines Herunterschreibens zu einem niedrigeren Niveau oder von beiden.

[2885] Beispiel 1282 umfasst den Gegenstand von einem der Beispiele 1275 bis 1281. In Beispiel 1282 ist ein Niveau ein Sicherheitsniveau, eine Netzebene oder beides.

[2886] Beispiel 1283 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, einen Themenverschlüsselungsschlüssel zu erzeugen, den Themenverschlüsselungsschlüssel zu einem Schlüsselverteilungszentrum (KDC) zu übermitteln und unter Verwendung des Themenverschlüsselungsschlüssels verschlüsselten Inhalt für einen Router zu veröffentlichen.

[2887] Beispiel 1284 umfasst den Gegenstand von Beispiel 1283. In Beispiel 1284 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, verschlüsselten Inhalt zu einem Subskribenten zu senden.

[2888] Beispiel 1285 umfasst den Gegenstand von Beispiel 1283 oder 1284. In Beispiel 1285 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, den Themenverschlüsselungsschlüssel vom Schlüsselverteilungszentrum zu erhalten.

[2889] Beispiel 1286 umfasst den Gegenstand von einem der Beispiele 1283 bis 1285. In Beispiel 1286 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Inhalt unter Verwendung des Themenverschlüsselungsschlüssels zu entschlüsseln.

[2890] Beispiel 1287 umfasst den Gegenstand von einem der Beispiele 1283 bis 1286. In Beispiel 1287 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Identität unter Verwendung eines durch einen Themenbenennungsserver bereitgestellten Zertifikats zu attestieren.

[2891] Beispiel 1288 umfasst den Gegenstand von einem der Beispiele 1283 bis 1287. In Beispiel 1288 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Integritätsrichtlinie zu erzwingen.

[2892] Beispiel 1289 umfasst den Gegenstand von einem der Beispiele 1283 bis 1288. In Beispiel 1289 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, eine Vertraulichkeitsrichtlinie zu erzwingen.

[2893] Beispiel 1290 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz einen scheuen Roboter umfasst. Der scheue Roboter umfasst Sensoren zum Erkennen eines Vorhandenseins, ein Antriebssystem zum Bewegen des scheuen Roboters, eine Kontextmaschine zur Verwendung von Daten von den Sensoren zur Steuerung des Antriebssystems zumindest teilweise auf der Grundlage des erkannten Vorhandenseins und ein Netzsystem zum Warnen anderer Vorrichtungen vor dem Vorhandensein.

[2894] Beispiel 1291 umfasst den Gegenstand von Beispiel 1290. In Beispiel 1291 umfassen die Sensoren eine Kamera zum Erhalten von Bilddaten.

[2895] Beispiel 1292 umfasst den Gegenstand von Beispiel 1290 oder 1291. In Beispiel 1292 umfassen die Sensoren einen thermischen Sensor zum Erkennen einer Wärmesignatur eines Lebewesens.

[2896] Beispiel 1293 umfasst den Gegenstand von einem der Beispiele 1290 bis 1292. In Beispiel 1293 umfassen die Sensoren ein Mikrofonfeld zum Bestimmen des Orts eines Tons, des Abstands von einer Tonquelle oder von beiden.

[2897] Beispiel 1294 umfasst den Gegenstand von einem der Beispiele 1290 bis 1293. In Beispiel 1294 umfassen die Sensoren einen Radarsensor zur Bestimmung der Entfernung von Objekten.

[2898] Beispiel 1295 umfasst den Gegenstand von einem der Beispiele 1290 bis 1294. In Beispiel 1295 umfasst das Vorhandensein ein sich bewegendes Objekt.

[2899] Beispiel 1296 umfasst den Gegenstand von einem der Beispiele 1290 bis 1295. In Beispiel 1296 umfasst der scheue Roboter ein Geolokationssystem zur Verwendung eines globalen Positionsbestimmungssystems zur Bestimmung einer Position oder einer Orientierung oder von beiden.

[2900] Beispiel 1297 umfasst den Gegenstand von einem der Beispiele 1290 bis 1296. In Beispiel 1297 umfasst der scheue Roboter ein Robotersystem zum Steuern von Vorrichtungen zum Aufsammeln von Objekten, zur Verbindung mit einer Ladeeinheit, zum Bereitstellen eines externen Ladeanschlusses oder von Kombinationen davon.

[2901] Beispiel 1298 umfasst den Gegenstand von einem der Beispiele 1290 bis 1297. In Beispiel 1298 umfasst der scheue Roboter ein Ladesystem zum automatischen Koppeln des scheuen Roboters mit einer externen Stromversorgung zum Laden einer Batterie.

[2902] Beispiel 1299 umfasst den Gegenstand von einem der Beispiele 1290 bis 1298. In Beispiel 1299 umfasst der scheue Roboter eine vertrauenswürdige Ausführungsumgebung (TEE).

[2903] Beispiel 1300 umfasst den Gegenstand von einem der Beispiele 1290 bis 1299. In Beispiel 1300 umfasst der scheue Roboter ein vertrauenswürdiges Plattformmodul (TPM) zur Einrichtung einer vertrauenswürdigen Ausführungsumgebung (TEE) während des Bootens.

[2904] Beispiel 1301 umfasst den Gegenstand von einem der Beispiele 1290 bis 1300. In Beispiel 1301 umfasst der scheue Roboter ein Stromausgabesystem, um externen Vorrichtungen Strom bereitzustellen.

[2905] Beispiel 1302 umfasst den Gegenstand von einem der Beispiele 1290 bis 1301. In Beispiel 1302 umfasst die Vorrichtung einen Schwarm von Robotern in einem ähnlichen geographischen Gebiet zum Austausch von Informationen über ein Vorhandensein.

[2906] Beispiel 1303 umfasst ein Verfahren zum Steuern eines scheuen Roboters in einem Roboterschwarm. Das Verfahren zum Steuern eines scheuen Roboters in einem Roboterschwarm umfasst das Erkennen eines Vorhandenseins in der Nähe des scheuen Roboters, das Anweisen eines Antriebssystems, den scheuen Roboter anzuhalten, falls er sich bewegt, und das Warnen anderer Roboter in einem Schwarm über das Vorhandensein.

[2907] Beispiel 1304 umfasst den Gegenstand von Beispiel 1303. In Beispiel 1304 umfasst das Erkennen des Vorhandenseins das Erkennen einer Bewegung eines Objekts in der Nähe des scheuen Roboters und das Klassifizieren, dass das Objekt ein Lebewesen hat.

[2908] Beispiel 1305 umfasst den Gegenstand von Beispiel 1303 oder 1304. In Beispiel 1305 umfasst das Erkennen des Vorhandenseins das Erhalten eines Bilds eines Objekts in der Nähe des scheuen Roboters und das Verarbeiten des Bilds zur Identifikation einer Person oder eines Tiers im Bild.

[2909] Beispiel 1306 umfasst den Gegenstand von einem der Beispiele 1303 bis 1305. In Beispiel 1306 umfasst das Erkennen des Vorhandenseins das Empfangen einer Warnnachricht, einschließlich eines von einer anderen Vorrichtung erkannten Vorhandenseins.

[2910] Beispiel 1307 umfasst den Gegenstand von einem der Beispiele 1303 bis 1306. In Beispiel 1307 umfasst das Anweisen des Antriebssystems, den scheuen Roboter anzuhalten, das Bewegen des scheuen Roboters zu einem Ort außerhalb des Wegs des ankommenden Verkehrs und das Deaktivieren des Antriebssystems.

[2911] Beispiel 1308 umfasst den Gegenstand von einem der Beispiele 1303 bis 1307. In Beispiel 1308 umfasst das Verfahren das Erkennen, dass eine Batteriereserve eine vorgegebene Grenze unterschreitet, das Bewegen des scheuen Roboters zu einer Ladestation und das automatische Koppeln des scheuen Roboters mit einer Ladevorrichtung.

[2912] Beispiel 1309 umfasst den Gegenstand von einem der Beispiele 1303 bis 1308. In Beispiel 1309 umfasst das Verfahren das Feststellen, dass sich der scheue Roboter zu einem neuen Ort bewegen soll, das Erhalten des aktuellen Orts, das Zuordnen eines Wegs zum neuen Ort und das Bewegen des scheuen Roboters vom aktuellen Ort zum Zielort.

[2913] Beispiel 1310 umfasst den Gegenstand von einem der Beispiele 1303 bis 1309. In Beispiel 1310 umfasst das Bewegen des scheuen Roboters das Verfolgen einer Richtlinie.

[2914] Beispiel 1311 umfasst den Gegenstand von einem der Beispiele 1303 bis 1310. In Beispiel 1311 umfasst das Warnen der anderen Roboter das Erkennen eines sich durch eine Zone um den scheuen Roboter bewegendes Objekts und das Senden einer Warnung zu den am nächsten benachbarten Robotern, um sie über die Erkennung zu informieren.

[2915] Beispiel 1312 umfasst den Gegenstand von einem der Beispiele 1303 bis 1311. In Beispiel 1312 umfasst die Warnung den Ort des Objekts, die Bewegung des Objekts oder die Gewissheit der Erkennung des Objekts oder Kombinationen davon.

[2916] Beispiel 1313 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, das Vorhandensein eines Objekts zu erkennen, festzustellen, dass sich das Objekt in einer Liste von Objekten in einer Richtlinie befindet, und eine Warnung über das Vorhandensein an andere Roboter in einem Schwarm zu übermitteln.

[2917] Beispiel 1314 umfasst den Gegenstand von Beispiel 1313. In Beispiel 1314 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, festzustellen, dass das Objekt ein Lebewesen ist, und ein Antriebssystem anzuweisen, den scheuen Roboter anzuhalten.

[2918] Beispiel 1315 umfasst den Gegenstand von Beispiel 1313 oder 1314. In Beispiel 1315 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den scheuen Roboter zu einem neuen Ort zu bewegen und eine Funktion am neuen Ort auszuführen.

[2919] Beispiel 1316 umfasst den Gegenstand von einem der Beispiele 1313 bis 1315. In Beispiel 1316 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, den scheuen Roboter in eine vertrauenswürdige Ausführungsumgebung (TEE) zu booten und den scheuen Roboter durch Laden von Richtlinien und Starten einer Kontextmaschine zu initialisieren.

[2920] Beispiel 1317 umfasst den Gegenstand von einem der Beispiele 1313 bis 1316. In Beispiel 1317 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, einen Prozessor anweisen, mit dem scheuen Roboter gekoppelte Sensoren zu entdecken und Treiber für die Sensoren zu laden.

[2921] Beispiel 1318 umfasst eine Vorrichtung. Die Vorrichtung umfasst ein Internet-der-Dinge(IoT)-Netz, wobei das IoT-Netz eine Pre-first-Responder-Vorrichtung (PFRD) umfasst, einschließlich eines Geolokationssystems zur Bestimmung des Orts der PFRD, eines Antriebssystems der Bewegung der PFRD zu einer Ereignisszene und einer Szenenbewertungs-Steuereinrichtung zum Bereitstellen von Funktionalität für die PFRD.

[2922] Beispiel 1319 umfasst den Gegenstand von Beispiel 1318. In Beispiel 1319 umfasst die Funktionalität das Steuern einer Bewegung, das Erfassen und Übertragen von Bildern oder das Bereitstellen von Netzkommunikationen für Benutzervorrichtungen oder Kombinationen davon.

[2923] Beispiel 1320 umfasst den Gegenstand von Beispiel 1318 oder 1319. In Beispiel 1320 umfasst die Vorrichtung ein Netzkommunikationssystem zum Bereitstellen von Kommunikationen für die PFRD, von Kommunikationen zwischen der PFRD und einem Notfall-Responder (ER) oder von Kommunikationen zwischen der PFRD und einer Basisstation oder Kombinationen davon.

[2924] Beispiel 1321 umfasst den Gegenstand von einem der Beispiele 1318 bis 1320. In Beispiel 1321 umfasst die Vorrichtung eine vertrauenswürdige Ausführungsumgebung (TEE) zum Speichern eines Entropiemultiplexier(EM)-Baums.

[2925] Beispiel 1322 umfasst den Gegenstand von einem der Beispiele 1318 bis 1321. In Beispiel 1322 soll der EM-Baum einem Wegpunkt zuordnen, zu dem mehrere PFRD kommissioniert werden können, um zu antworten.

[2926] Beispiel 1323 umfasst den Gegenstand von einem der Beispiele 1318 bis 1322. In Beispiel 1323 umfasst der EM-Baum eine Ortskoordinate und eine vorgesehene Ankunftszeit für jeden Wegpunkt.

[2927] Beispiel 1324 umfasst den Gegenstand von einem der Beispiele 1318 bis 1323. In Beispiel 1324 soll der EM-Baum Seed-Werte für die Erzeugung von Identifikationsschlüsseln an einem Wegpunkt bereitstellen.

[2928] Beispiel 1325 umfasst den Gegenstand von einem der Beispiele 1318 bis 1324. In Beispiel 1325 umfasst die Vorrichtung einen Fahrtwegplaner zum Bereitstellen von Weginformationen zu einem Bestimmungs-ort.

[2929] Beispiel 1326 umfasst den Gegenstand von einem der Beispiele 1318 bis 1325. In Beispiel 1326 umfassen die Weginformationen Szenen, Wegpunkte, Orte, Bestimmungsorte oder Schnittpunkte oder Kombinationen davon.

[2930] Beispiel 1327 umfasst den Gegenstand von einem der Beispiele 1318 bis 1326. In Beispiel 1327 umfassen die Weginformationen alternative Wege.

[2931] Beispiel 1328 umfasst den Gegenstand von einem der Beispiele 1318 bis 1327. In Beispiel 1328 umfasst die PFRD ein unbemanntes Luftfahrzeug.

[2932] Beispiel 1329 umfasst den Gegenstand von einem der Beispiele 1318 bis 1328. In Beispiel 1329 umfasst die PFRD einen Schlüssel für Kommunikationen innerhalb eines einzigen Zuständigkeitsbereichs.

[2933] Beispiel 1330 umfasst den Gegenstand von einem der Beispiele 1318 bis 1329. In Beispiel 1330 umfasst die PFRD mehrere Schlüssel für Kommunikationen innerhalb mehrerer unabhängiger Zuständigkeitsbereiche.

[2934] Beispiel 1331 umfasst den Gegenstand von einem der Beispiele 1318 bis 1330. In Beispiel 1331 umfasst die PFRD einen Schlüssel für mehrere Zuständigkeitsbereiche für Kommunikationen innerhalb überlappender Zuständigkeitsbereiche.

[2935] Beispiel 1332 umfasst ein Verfahren zur Behandlung einer Pre-first-Responder-Vorrichtung (PFRD). Das Verfahren zur Behandlung einer Pre-first-Responder-Vorrichtung (PFRD) umfasst das Empfangen eines Befehls zur Teilnahme an einem Ereignis durch die PFRD, das Erhalten eines Schlüssels für einen Zuständigkeitsbereich, das Registrieren beim Zuständigkeitsbereich und das Bereitstellen von Unterstützung für Notfall-Responder beim Ereignis.

[2936] Beispiel 1333 umfasst den Gegenstand von Beispiel 1332. In Beispiel 1333 umfasst das Verfahren die Feststellung, ob die PFRD in einem einzigen Zuständigkeitsbereich arbeitet, das Verhandeln eines Schlüssels von dem einzelnen Zuständigkeitsbereich, die Registrierung der PFRD beim einzelnen Zuständigkeitsbereich unter Verwendung des Schlüssels und das Einleiten einer Kommunikation mit Notfall-Respondern, falls die Registrierung erfolgreich war.

[2937] Beispiel 1334 umfasst den Gegenstand von Beispiel 1332 oder 1333. In Beispiel 1334 umfasst das Verfahren die Feststellung, ob die PFRD in überlappenden Zuständigkeitsbereichen arbeitet, das Verhandeln eines geteilten Schlüssels von den überlappenden Zuständigkeitsbereichen, das Registrieren der PFRD bei jedem der überlappenden Zuständigkeitsbereiche unter Verwendung des geteilten Schlüssels und das Einleiten einer Kommunikation mit Notfall-Respondern, falls die Registrierung bei allen überlappenden Zuständigkeitsbereichen erfolgreich war.

[2938] Beispiel 1335 umfasst den Gegenstand von einem der Beispiele 1332 bis 1334. In Beispiel 1335 umfasst das Verfahren die Feststellung, ob die PFRD in mehreren nicht überlappenden Zuständigkeitsbereichen arbeitet, das Verhandeln eines individuellen Schlüssels von jedem der nicht überlappenden Zuständigkeitsbereiche, das Registrieren der PFRD bei jedem der nicht überlappenden Zuständigkeitsbereiche unter Verwendung des geeigneten individuellen Schlüssels und das Einleiten einer Kommunikation mit Notfall-Respondern, falls die Registrierung bei allen nicht überlappenden Zuständigkeitsbereichen erfolgreich war.

[2939] Beispiel 1336 umfasst den Gegenstand von einem der Beispiele 1332 bis 1335. In Beispiel 1336 umfasst das Verfahren die Feststellung, dass die Registrierung bei einem Zuständigkeitsbereich nicht erfolgreich war, und das Warnen eines Zuständigkeitsbereichs-Steuernetzes.

[2940] Beispiel 1337 umfasst den Gegenstand von einem der Beispiele 1332 bis 1336. In Beispiel 1337 umfasst das Verfahren das Abrufen von Richtlinien aus einem sicheren Speicher in einer vertrauenswürdigen Ausführungsumgebung und das Aktivieren einer anfänglichen Flugplanrichtlinie.

[2941] Beispiel 1338 umfasst den Gegenstand von einem der Beispiele 1332 bis 1337. In Beispiel 1338 umfasst die anfängliche Flugplanrichtlinie autonome Operationen, einschließlich des Lösen von einem Ankerpunkt, des Schwebens, des Erfassens von Bildern oder der Ausführung einer Bilderkennung oder Kombinationen davon.

[2942] Beispiel 1339 umfasst den Gegenstand von einem der Beispiele 1332 bis 1338. In Beispiel 1339 umfasst das Verfahren die Feststellung, ob die Netzverbindbarkeit schlecht ist, und das Implementieren von Verbindbarkeitsrichtlinien zur Verbesserung der Kommunikation.

[2943] Beispiel 1340 umfasst den Gegenstand von einem der Beispiele 1332 bis 1339. In Beispiel 1340 umfasst das Verfahren das Abrufen von Fernbehandlungsrichtlinien aus einem sicheren Speicher in einer vertrauenswürdigen Ausführungsumgebung und das Zulassen einer Fernflugbehandlung durch Notfall-Responder.

[2944] Beispiel 1341 umfasst ein nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, die, wenn sie ausgeführt werden, einen Prozessor anweisen, eine Pre-first-Responder-Vorrichtung (PFRD) zur Reaktion auf ein Ereignis auszulösen, Richtlinien für Operationen der PFRD abzurufen und auf das Ereignis zu reagieren.

[2945] Beispiel 1342 umfasst den Gegenstand von Beispiel 1341. In Beispiel 1342 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Fernbehandlungsrichtlinien in der PFRD zu implementieren und eine Fernoperationsbehandlung durch einen Notfall-Responder zu ermöglichen.

[2946] Beispiel 1343 umfasst den Gegenstand von Beispiel 1341 oder 1342. In Beispiel 1343 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Richtlinien zur Verbesserung von Kommunikationen zu implementieren.

[2947] Beispiel 1344 umfasst den Gegenstand von einem der Beispiele 1341 bis 1343. In Beispiel 1344 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, einen Fernnetzzugriff für Notfall-Responder bereitzustellen.

[2948] Beispiel 1345 umfasst den Gegenstand von einem der Beispiele 1341 bis 1344. In Beispiel 1345 umfasst das nichtflüchtige maschinenlesbare Medium Befehle, die, wenn sie ausgeführt werden, den Prozessor anweisen, Fahrpläne von einem Fahrtwegplaner zu akzeptieren.

[2949] Beispiel 1346 umfasst eine Vorrichtung, die Mittel zur Ausführung eines Verfahrens nach einem anderen Beispiel umfasst.

[2950] Beispiel 1347 umfasst einen maschinenlesbaren Speicher, der maschinenlesbare Befehle umfasst, um, wenn sie ausgeführt werden, ein Verfahren nach einem anderen Beispiel zu implementieren oder eine Vorrichtung nach einem anderen Beispiel zu verwirklichen.

[2951] Einige Ausführungsformen können in Hardware, Firmware und Software oder einer Kombination davon implementiert werden. Einige Ausführungsformen können auch als auf einem maschinenlesbaren Medium gespeicherte Befehle, die von einer Rechenplattform gelesen und ausgeführt werden können, um die hier beschriebenen Operationen auszuführen, implementiert werden. Ein maschinenlesbares Medium kann einen Mechanismus zum Speichern oder Übertragen von Informationen in einer von einer Maschine (beispielsweise einem Computer) lesbaren Form umfassen. Beispielsweise kann ein maschinenlesbares Medium unter anderem einen Nurlesespeicher (ROM), einen Direktzugriffsspeicher (RAM) Magnetplatten-Speichermedien, optische Speichermedien, Flash-Speichervorrichtungen oder elektrische, optische, akustische Signale oder eine andere Form sich ausbreitender Signale, beispielsweise Trägerwellen, Infrarotsignale, Digitalsignale oder die Schnittstellen, welche Signale aussenden und/oder empfangen, umfassen.

[2952] Eine Ausführungsform ist eine Implementation oder ein Beispiel. Ein in dieser Beschreibung auftretender Bezug auf „eine Ausführungsform“, „eine einzige Ausführungsform“, „einige Ausführungsformen“, „verschiedene Ausführungsformen“ oder „andere Ausführungsformen“ bedeutet, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder eine bestimmte Eigenschaft, die in Zusammenhang mit den Ausführungsformen beschrieben wird, in zumindest einigen Ausführungsformen, jedoch nicht notwendigerweise in allen Ausführungsformen, der Techniken enthalten ist. Die verschiedenen Instanzen von „eine Ausführungsform“, „eine einzige Ausführungsform“ oder „einige Ausführungsformen“ beziehen sich nicht notwendigerweise alle auf die gleichen Ausführungsformen. Elemente oder Aspekte von einer Ausführungsform können mit Elementen oder Aspekten einer anderen Ausführungsform kombiniert werden.

[2953] Nicht alle Komponenten, Merkmale, Strukturen, Eigenschaften usw., die hier beschrieben und erläutert wurden, müssen in einer bestimmten Ausführungsform oder in bestimmten Ausführungsformen enthalten sein. Falls die Beschreibung aussagt, dass eine Komponente, ein Merkmal, eine Struktur oder eine Eigenschaft enthalten sein „kann“ oder „könnte“, braucht diese bestimmte Komponente, dieses bestimmte Merkmal, diese bestimmte Struktur oder diese bestimmte Eigenschaft beispielsweise nicht unbedingt enthalten zu sein. Falls sich die Beschreibung oder ein Anspruch auf „ein“ Element bezieht, bedeutet dies nicht, dass es nur ein einziges von diesem Element gibt. Falls sich die Beschreibung oder Ansprüche auf „ein zusätzliches“ Element bezieht, schließt dies nicht aus, dass es mehr als eines von dem zusätzlichen Element gibt.

[2954] Es sei bemerkt, dass, wenngleich einige Ausführungsformen mit Bezug auf bestimmte Implementationen beschrieben wurden, andere Implementationen gemäß einigen Ausführungsformen möglich sind. Zusätzlich brauchen die Anordnung und/oder die Reihenfolge von Schaltungselementen oder anderen Merkmalen, die in den Zeichnungen dargestellt sind und/oder hier beschrieben sind, nicht in der bestimmten dargestellten und beschriebenen Weise angeordnet sein. Viele andere Anordnungen sind gemäß einigen Ausführungsformen möglich.

[2955] Bei jedem in einer Figur dargestellten System können die Elemente in einigen Fällen jeweils die gleiche Bezugszahl oder eine andere Bezugszahl haben, um nahezulegen, dass die dargestellten Elemente verschiedenen und/oder ähnlich sein könnten. Ein Element kann jedoch flexibel genug sein, um verschiedene Implementationen zu haben und mit einigen oder allen der hier dargestellten oder beschriebenen Systeme zu wirken. Die verschiedenen in den Figuren dargestellten Elemente können gleich oder verschieden sein. Was als erstes und was als zweites Element bezeichnet wird, ist willkürlich.

[2956] Die Techniken sind nicht auf die bestimmten hier aufgelisteten Einzelheiten beschränkt. Tatsächlich werden Fachleute auf dem Gebiet, die den Vorteil dieser Offenbarung haben, verstehen, dass innerhalb des Geltungsbereichs der vorliegenden Techniken viele andere Abänderungen der vorstehenden Beschreibung

und Zeichnungen vorgenommen werden können. Dementsprechend wird der Geltungsbereich der Techniken durch die folgenden Ansprüche einschließlich jeglicher Änderungen daran definiert.

Patentansprüche

1. Vorrichtung, welche ein zusammengesetztes Objekt in einem IoT-Netz umfasst, umfassend:
einen Vorrichtungseigentümer, der Folgendes umfasst:
einen Nameserver zum Bereitstellen von Namen für Sub-Objekte, die das zusammengesetzte Objekt bilden, und
eine Sub-Objekt-Liste der das zusammengesetzte Objekt bildenden Sub-Objekte und
mehrere Sub-Objekte, die das zusammengesetzte Objekt bilden, und
eine Blockchain, welche die das zusammengesetzte Objekt bildenden Sub-Objekte aufzeichnet.
2. Vorrichtung nach Anspruch 1, wobei ein Sub-Objekt ein aus Sub-Objekten niedrigerer Ebene gebildetes zusammengesetztes Objekt umfasst.
3. Vorrichtung nach Anspruch 1, wobei ein Sub-Objekt ein atomares Objekt umfasst.
4. Vorrichtung nach Anspruch 1, wobei der Name des zusammengesetzten Objekts einen anhand der Namen der mehreren Sub-Objekte berechneten Hash umfasst.
5. Vorrichtung nach Anspruch 1, wobei jedes Sub-Objekt einen Gruppenschlüssel umfasst, der es dem Sub-Objekt erlaubt, im Auftrag der Gruppe zu wirken.
6. Vorrichtung nach einem der Ansprüche 1-5, wobei der Vorrichtungseigentümer einen EPID-Server umfasst.
7. Vorrichtung nach einem der Ansprüche 1-5, wobei der Vorrichtungseigentümer einen Proxy-Broker umfasst.
8. Vorrichtung nach einem der Ansprüche 1-5, wobei der Vorrichtungseigentümer eine Blockchain umfasst.
9. Vorrichtung nach einem der Ansprüche 1-5, wobei die Blockchain einen Datensatz des zusammengesetzten Objekts umfasst.
10. Verfahren zur Bildung eines zusammengesetzten Objekts in einem IoT-Netz, welches Folgendes umfasst:
Bilden einer Liste von Sub-Objekten in einem Vorrichtungseigentümer, Erzeugen einer Sammlungsgruppenkennung,
Übergeben der Sammlungsgruppenkennung an eine Blockchain in einer Blockchain-Transaktion und
Erhalten eines Gruppennamens von der Blockchain in einem Nameserver.
11. Verfahren nach Anspruch 10, welches Folgendes umfasst:
Feststellen anhand der Blockchain, ob die Sammlungsgruppenkennung bereits in Gebrauch ist, und, falls dies der Fall ist,
Erzeugen einer neuen Sammlungsgruppenkennung.
12. Verfahren nach Anspruch 10, welches Folgendes umfasst:
Akzeptieren einer Beitrittsanforderung von einem Sub-Objekt, Bestätigen, dass das Sub-Objekt ein Gruppenmitglied ist,
Nachschlagen des Namens des Sub-Objekts in der Blockchain und Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom Nameserver.
13. Verfahren nach Anspruch 10, welches Folgendes umfasst:
Feststellen, ob die Gruppenmitgliedschaft privat ist, und, falls dies der Fall ist, Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom Vorrichtungseigentümer, der als Proxy für den Nameserver wirkt.
14. Verfahren nach Anspruch 10, umfassend das Erzeugen der Sammlungsgruppenkennung durch:
Kombinieren der Namen des Sub-Objekts zur Bildung einer Kombination und Berechnen eines Hash-Codes der Kombination.

15. Verfahren nach Anspruch 10, umfassend das Erzeugen eines Namens für ein Sub-Objekt durch:
Kombinieren der Namen aller das Sub-Objekt bildenden Sub-Sub-Objekte zur Bildung einer Kombination und
Berechnen eines Hash-Codes der Kombination.

16. Verfahren nach einem der Ansprüche 10-15, welches Folgendes umfasst:
Bestätigen, dass eine Blockchain-Transaktion in einer Gruppe von Vorrichtungen in einem Mesh-Netz gültig ist, und
Umkehren der Blockchain-Transaktion, falls sie nicht gültig ist.

17. Vorrichtung zur Bildung eines zusammengesetzten Objekts in einem IoT-Netz, welche Folgendes umfasst:
Mittel zum Bilden einer Liste von Sub-Objekten in einem Vorrichtungseigentümer,
Mittel zum Erzeugen einer Sammlungsgruppenkennung,
Mittel zum Übergeben der Sammlungsgruppenkennung an eine Blockchain in einer Blockchain-Transaktion und
Mittel zum Erhalten eines Gruppennamens von der Blockchain in einem Nameserver.

18. Vorrichtung nach Anspruch 17, welche Folgendes umfasst:
Mittel zum Feststellen anhand der Blockchain, ob die Sammlungsgruppenkennung bereits in Gebrauch ist, und
Mittel zum Erzeugen einer neuen Sammlungsgruppenkennung, falls die Sammlungsgruppenkennung bereits in Gebrauch ist.

19. Vorrichtung nach Anspruch 17, welche Folgendes umfasst:
Mittel zum Akzeptieren einer Beitrittsanforderung von einem Sub-Objekt,
Mittel zum Bestätigen, dass das Sub-Objekt ein Gruppenmitglied ist,
Mittel zum Nachschlagen des Namens des Sub-Objekts in der Blockchain und
Mittel zum Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom Nameserver.

20. Vorrichtung nach Anspruch 17, welche Folgendes umfasst:
Mittel zum Feststellen, ob die Gruppenmitgliedschaft privat ist, und
Mittel zum Bereitstellen eines Gruppenschlüssels für das Sub-Objekt vom als Proxy für den Nameserver wirkenden Vorrichtungseigentümer, falls die Gruppenmitgliedschaft privat ist.

21. Vorrichtung nach Anspruch 17, wobei die Mittel zum Erzeugen der Sammlungsgruppenkennung Folgendes umfassen:
Mittel zum Kombinieren der Namen des Sub-Objekts zur Bildung einer Kombination und
Mittel zum Berechnen eines Hash-Codes der Kombination.

22. Vorrichtung nach Anspruch 17, welche Mittel zum Erzeugen eines Namens für ein Sub-Objekt umfasst, wobei die Mittel zum Erzeugen eines Namens für ein Sub-Objekt Folgendes umfassen:
Mittel zum Kombinieren der Namen aller das Sub-Objekt bildenden Sub-Sub-Objekte zur Bildung einer Kombination und
Mittel zum Berechnen eines Hash-Codes der Kombination.

23. Vorrichtung nach einem der Ansprüche 17-22, welche Folgendes umfasst:
Mittel zum Bestätigen, dass die Blockchain-Transaktion in einer Gruppe von Vorrichtungen in einem Mesh-Netz gültig ist, und
Mittel zum Umkehren der Blockchain-Transaktion, falls sie nicht gültig ist.

24. Nichtflüchtiges maschinenlesbares Medium, das Befehle umfasst, um einen Prozessor anzuweisen, Folgendes auszuführen:
Speichern einer Liste von Sub-Objekten für eine Gruppe,
Berechnen einer Sammlungsgruppenidentität für die Gruppe und
Bereitstellen von Gruppenidentitäts-Berechtigungsnachweisen für Sub-Objekte in der Gruppe.

25. Nichtflüchtiges maschinenlesbares Medium nach Anspruch 24, welches Befehle umfasst, um den Prozessor anzuweisen, als ein Proxy-Server für Sub-Objekte zu wirken.

26. Nichtflüchtiges maschinenlesbares Medium nach Anspruch 24, welches Befehle umfasst, um den Prozessor anzuweisen, Folgendes auszuführen:

Übergeben einer eine Sammlungsgruppenidentität umfassenden Transaktion an eine Blockchain und Migrieren der Blockchain zu anderen Vorrichtungen in einem Mesh.

27. Nichtflüchtiges maschinenlesbares Medium nach einem der Ansprüche 24-26, welches eine Transaktionsblöcke umfassende Blockchain umfasst, wobei ein Transaktionsblock eine Sammlungsgruppenidentität umfasst.

Es folgen 275 Seiten Zeichnungen

Anhängende Zeichnungen

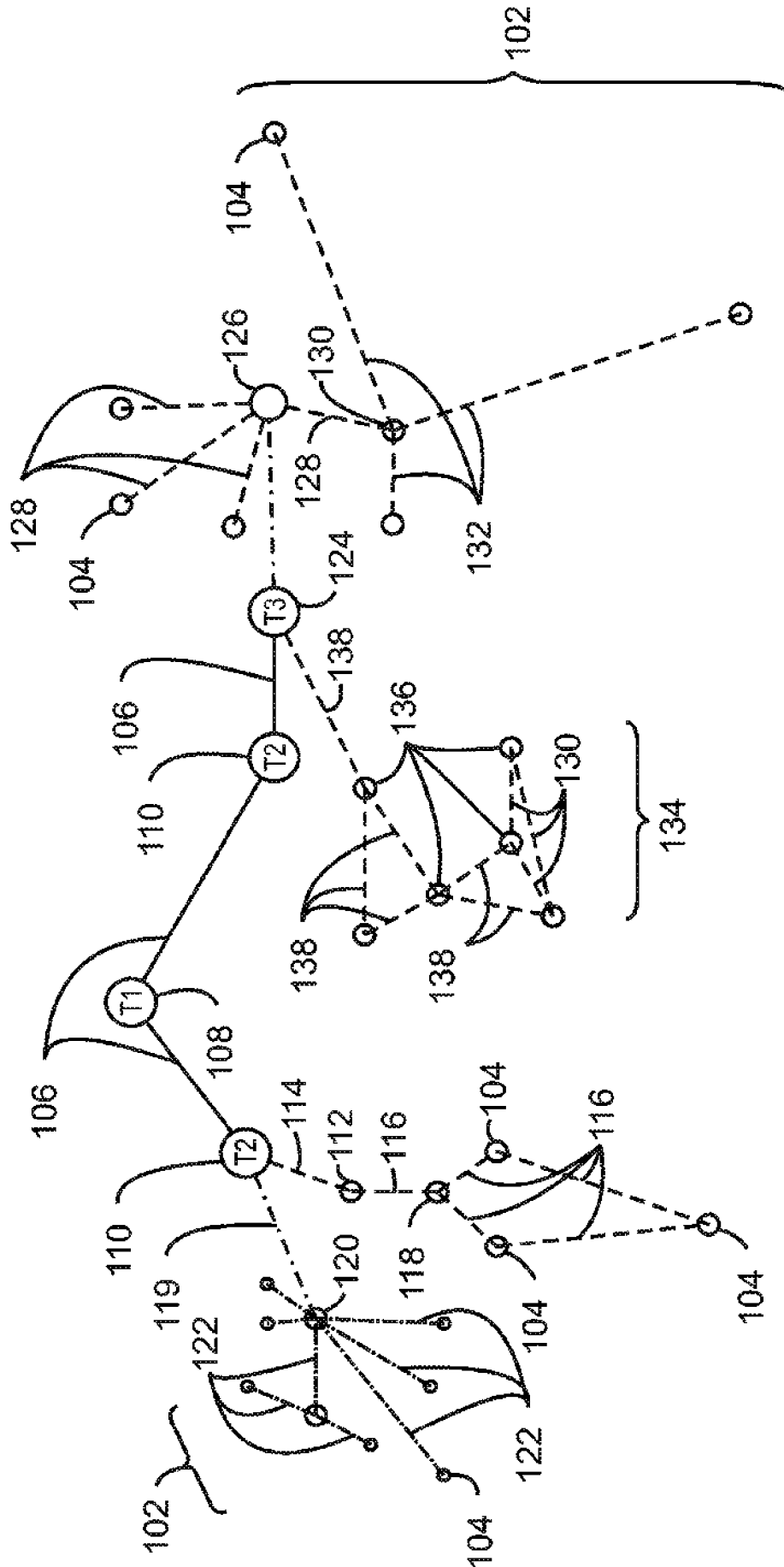
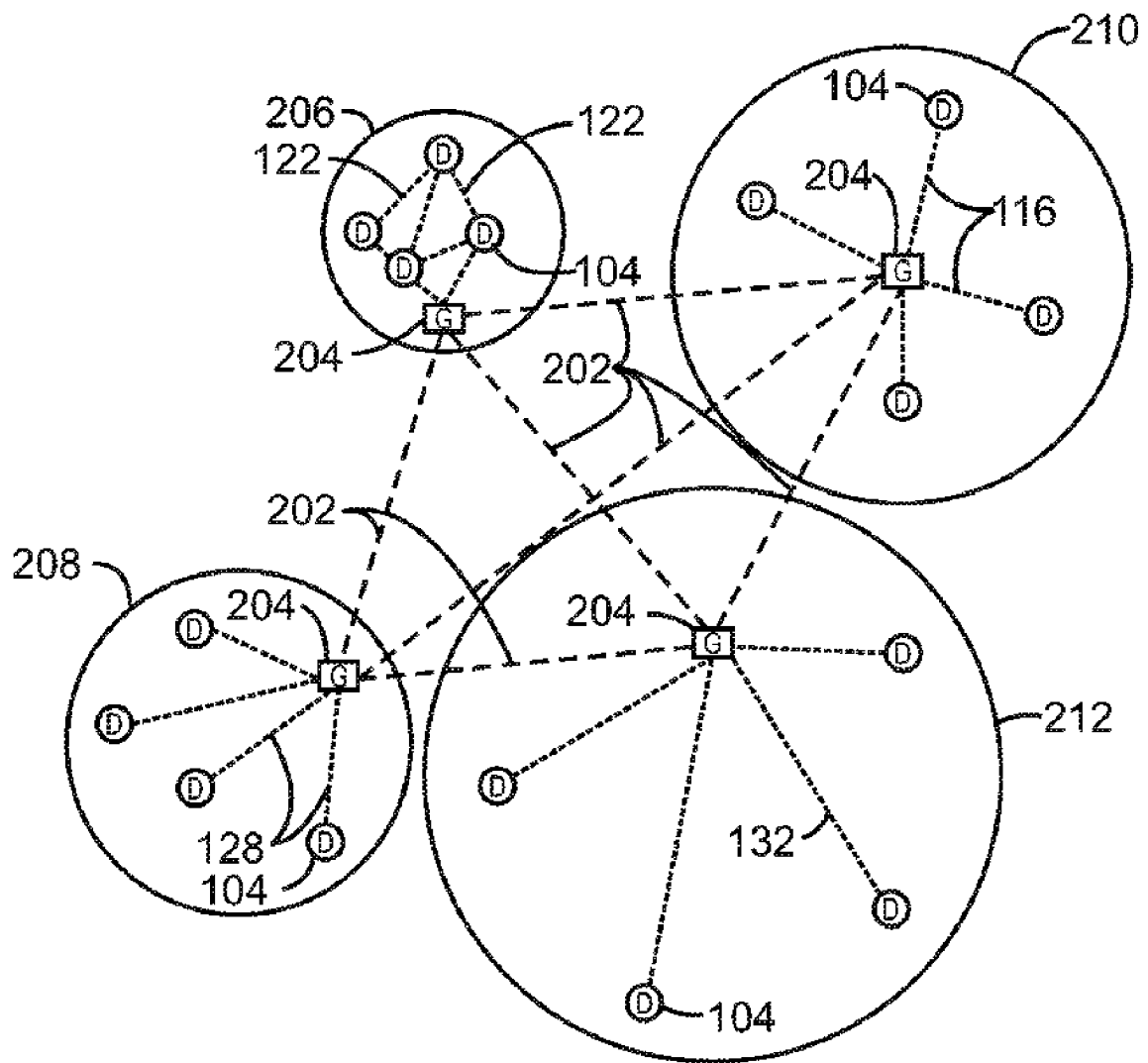
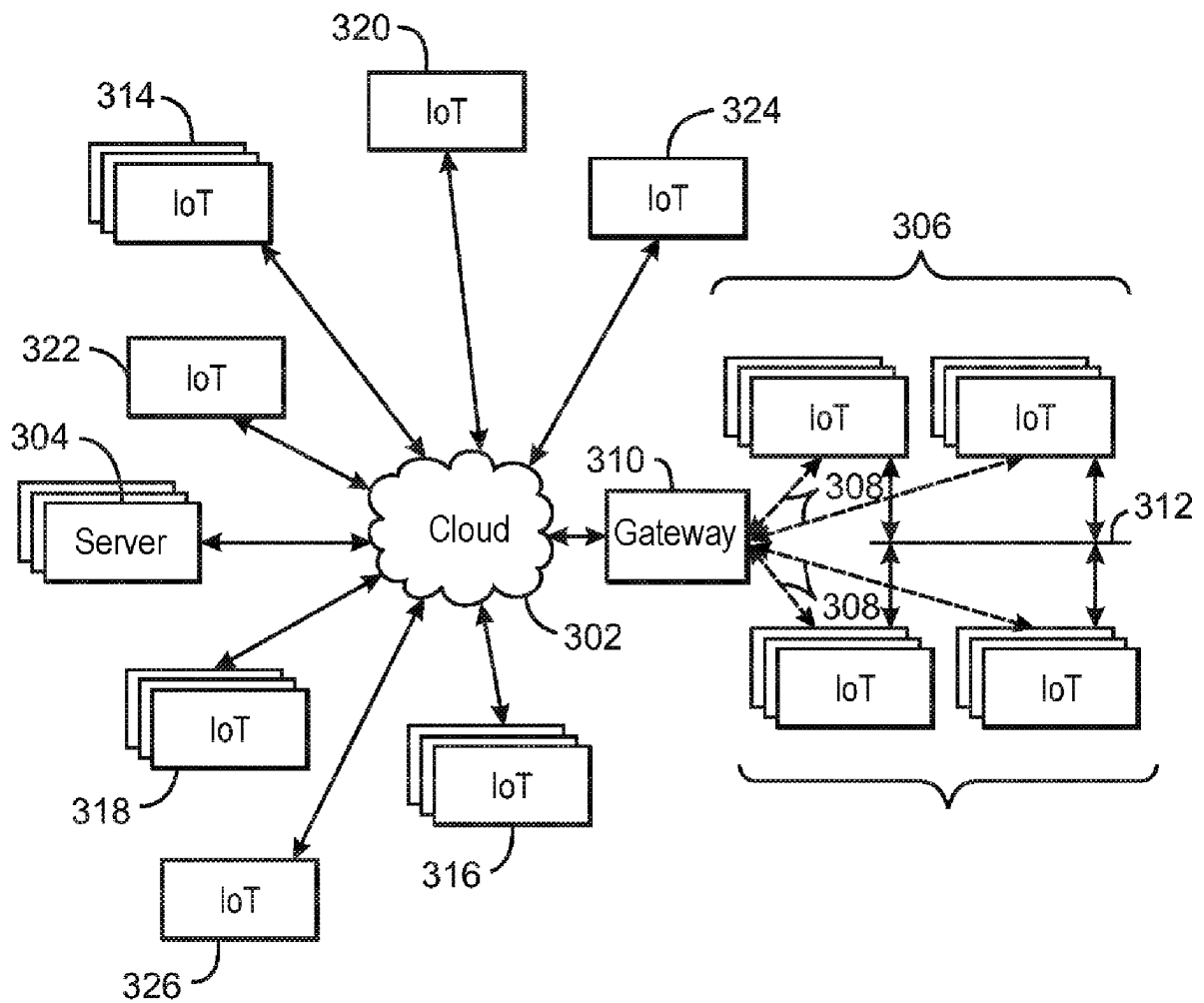


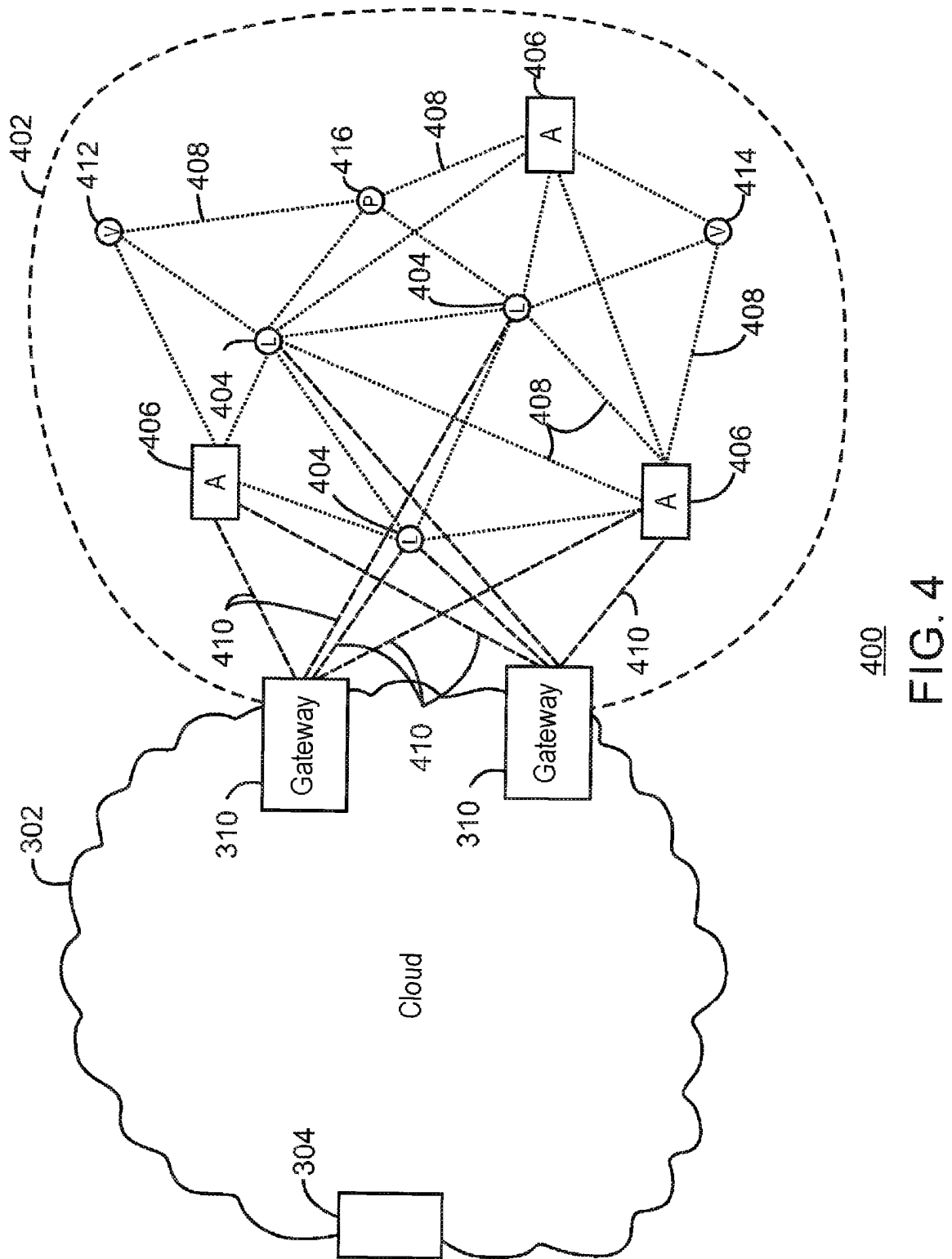
FIG. 1

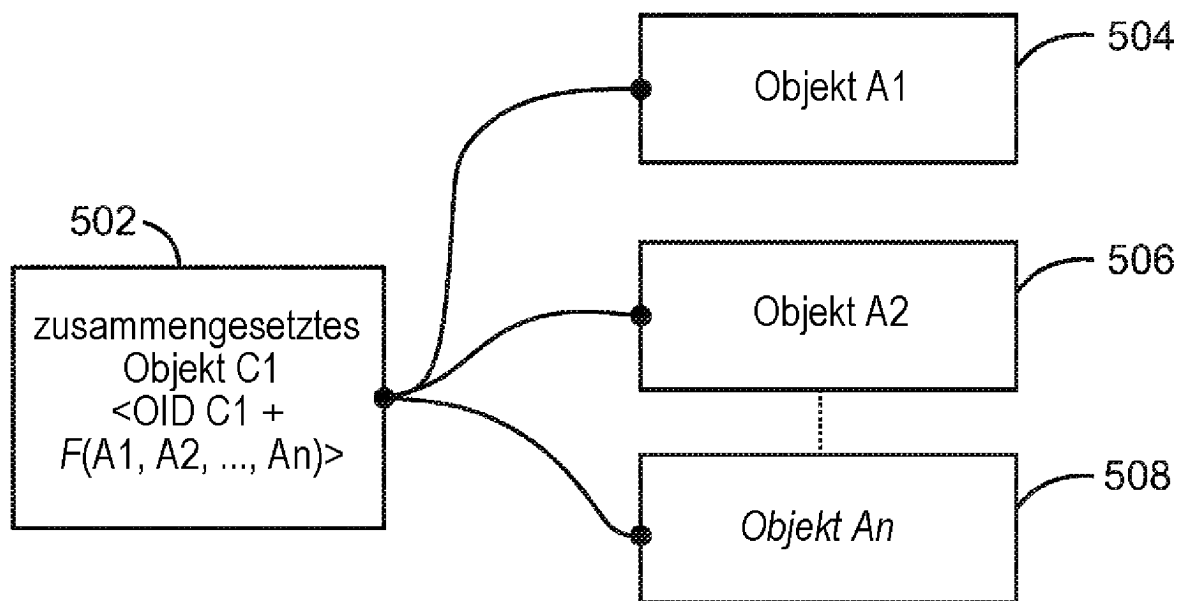


200
FIG. 2

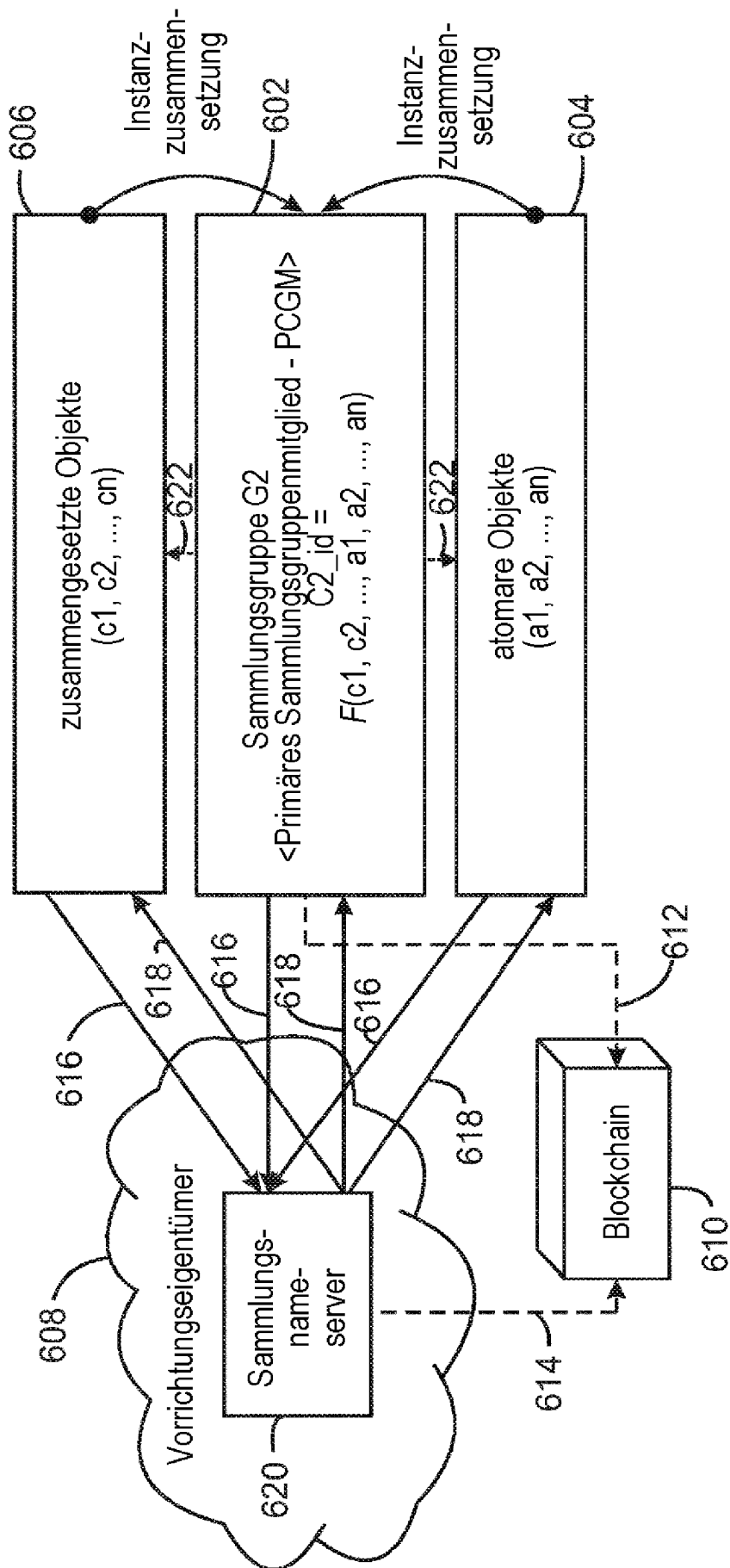


300
FIG. 3



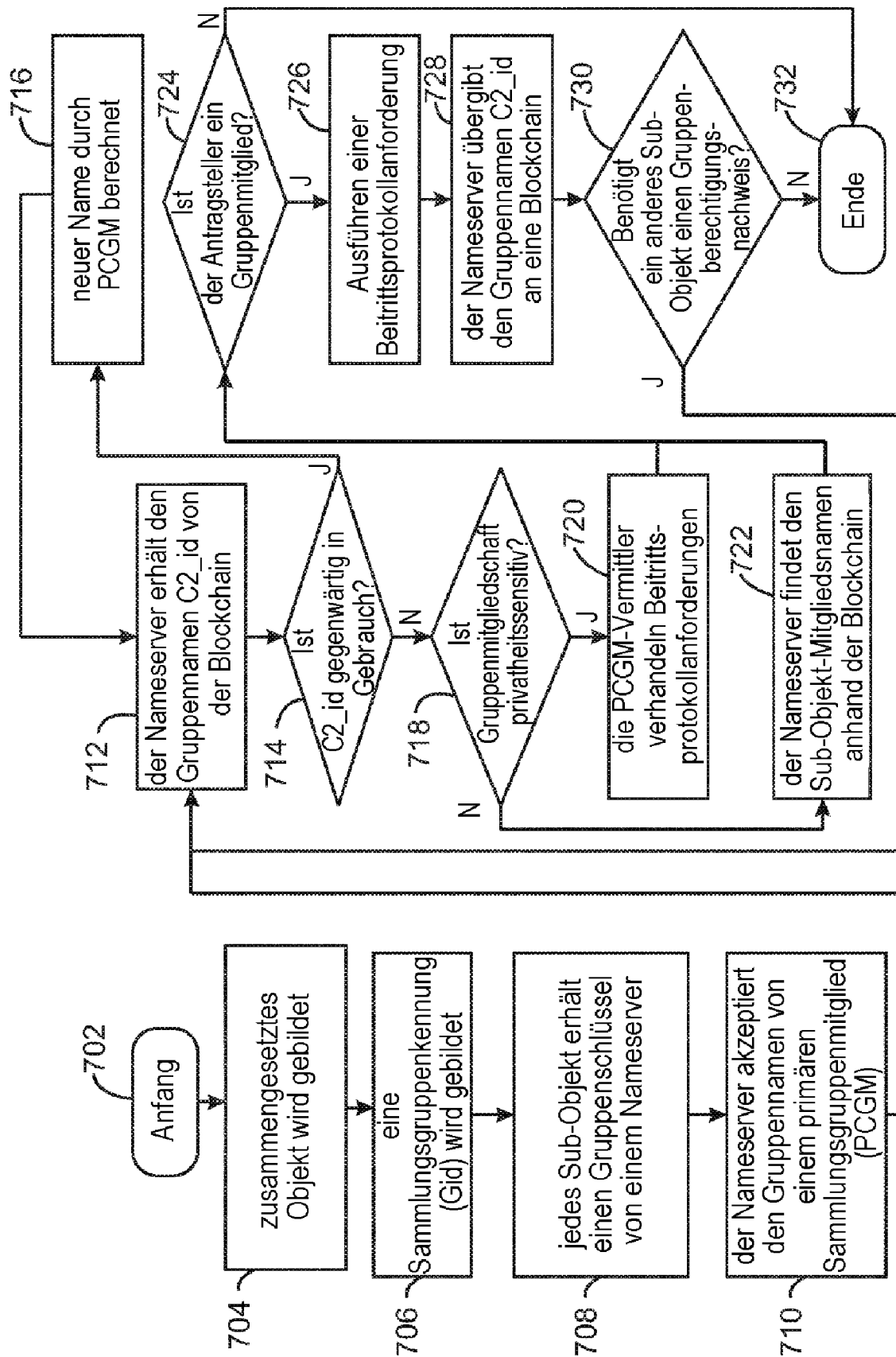


500
FIG. 5



600

FIG. 6



700

FIG. 7

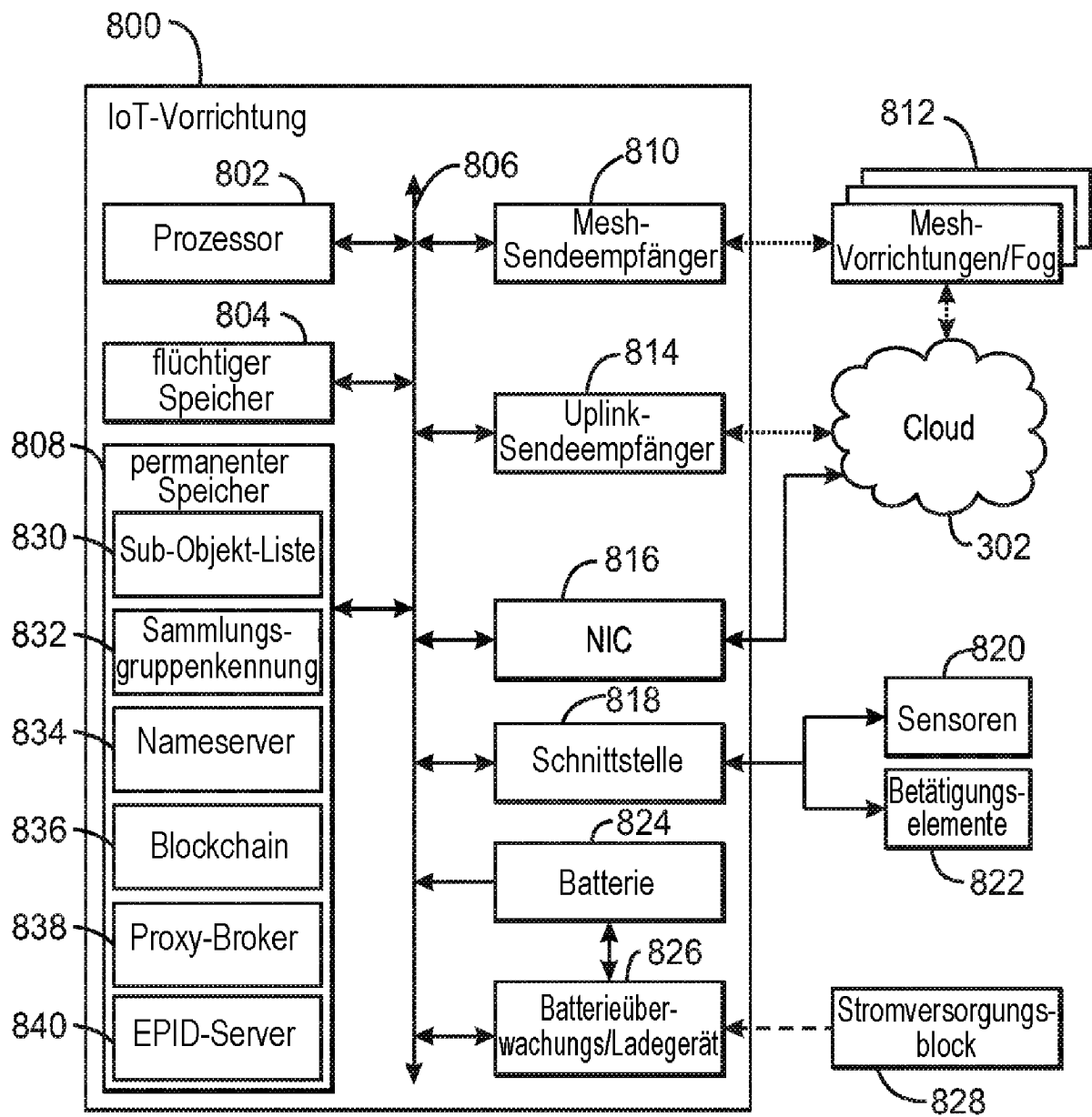


FIG. 8

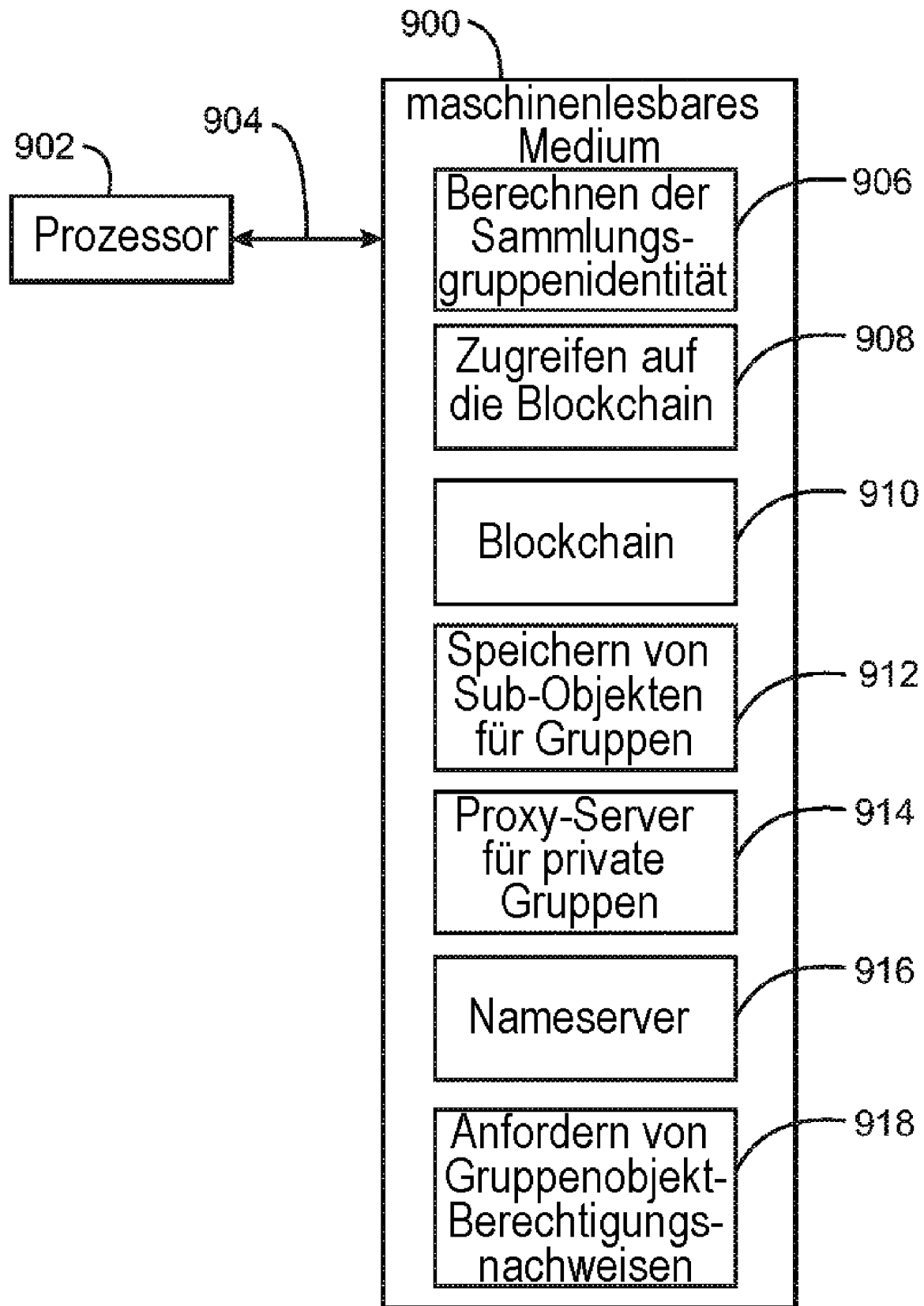
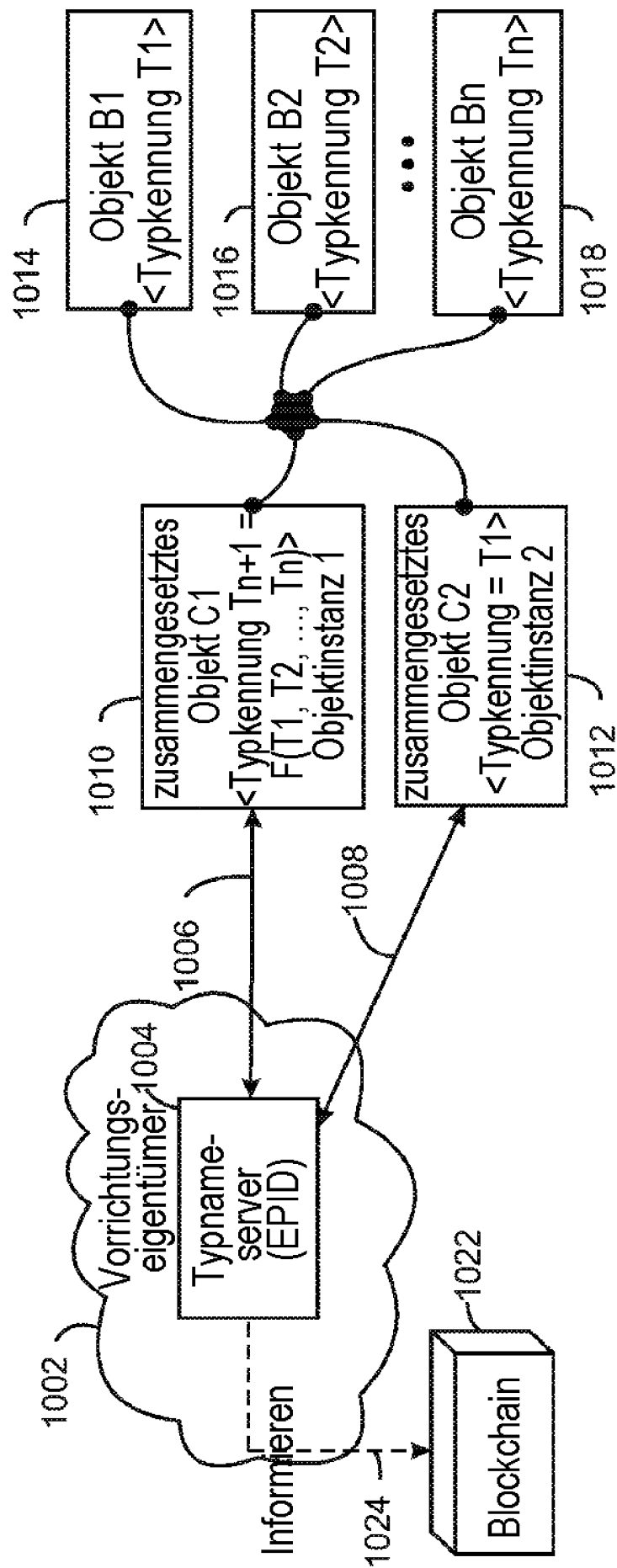
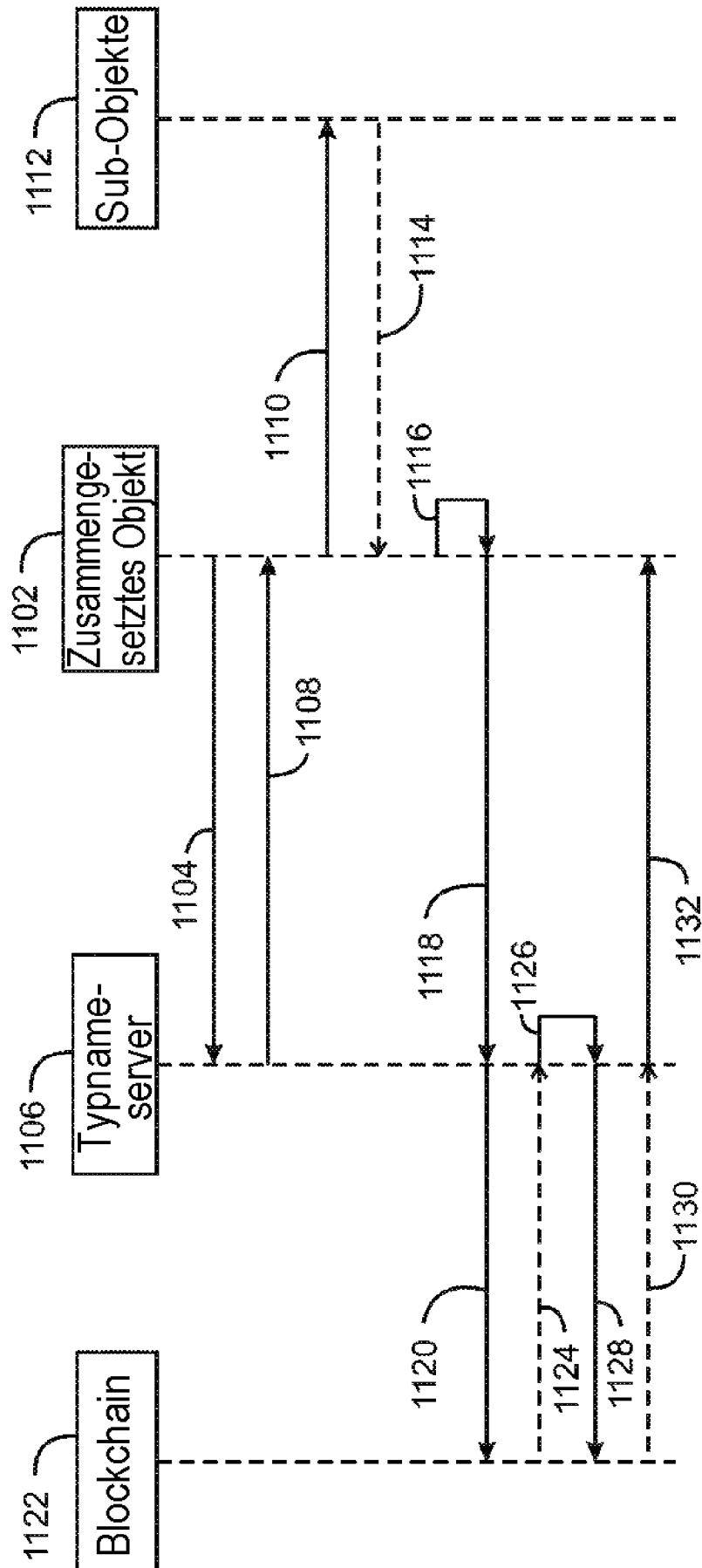


FIG. 9

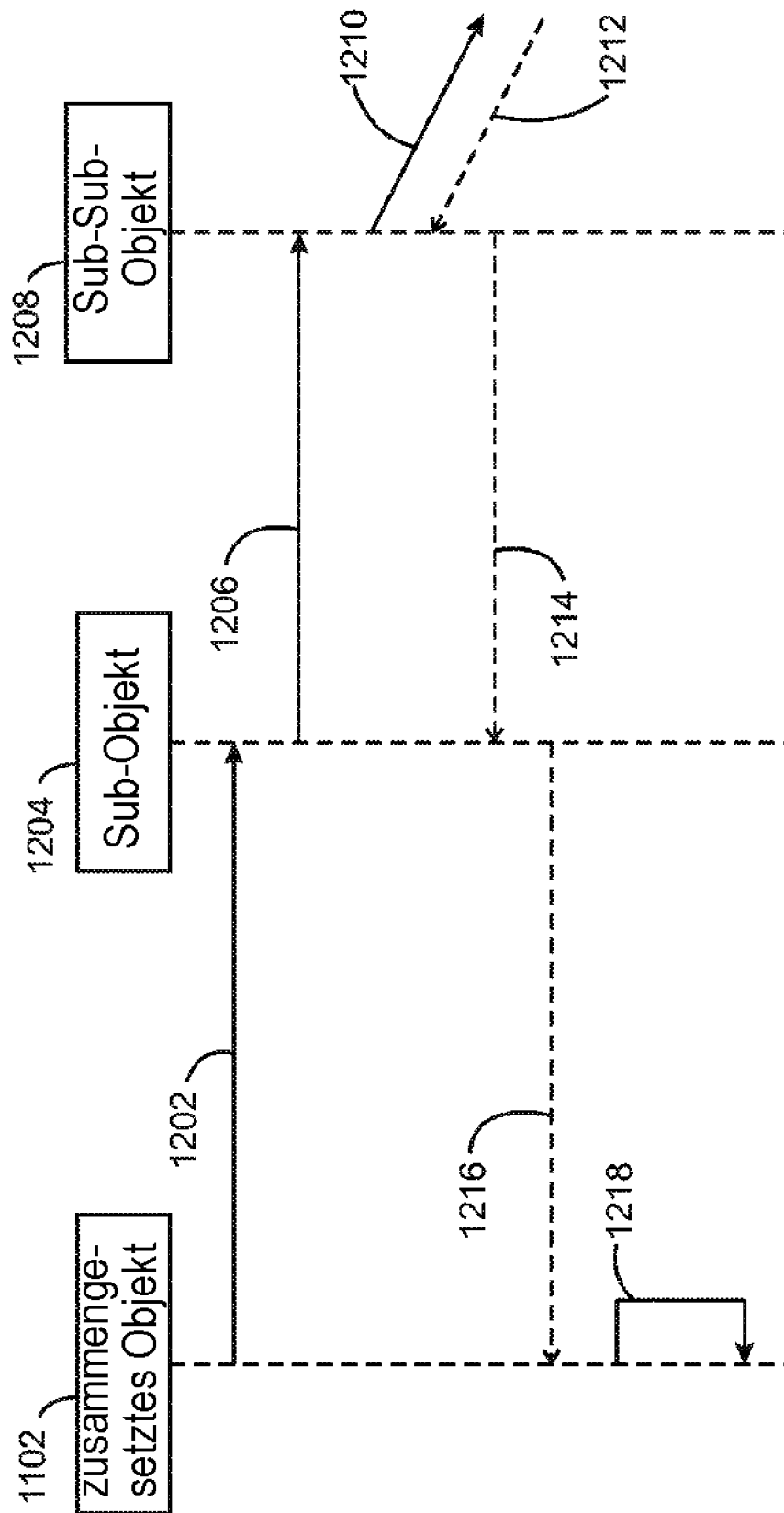


1000

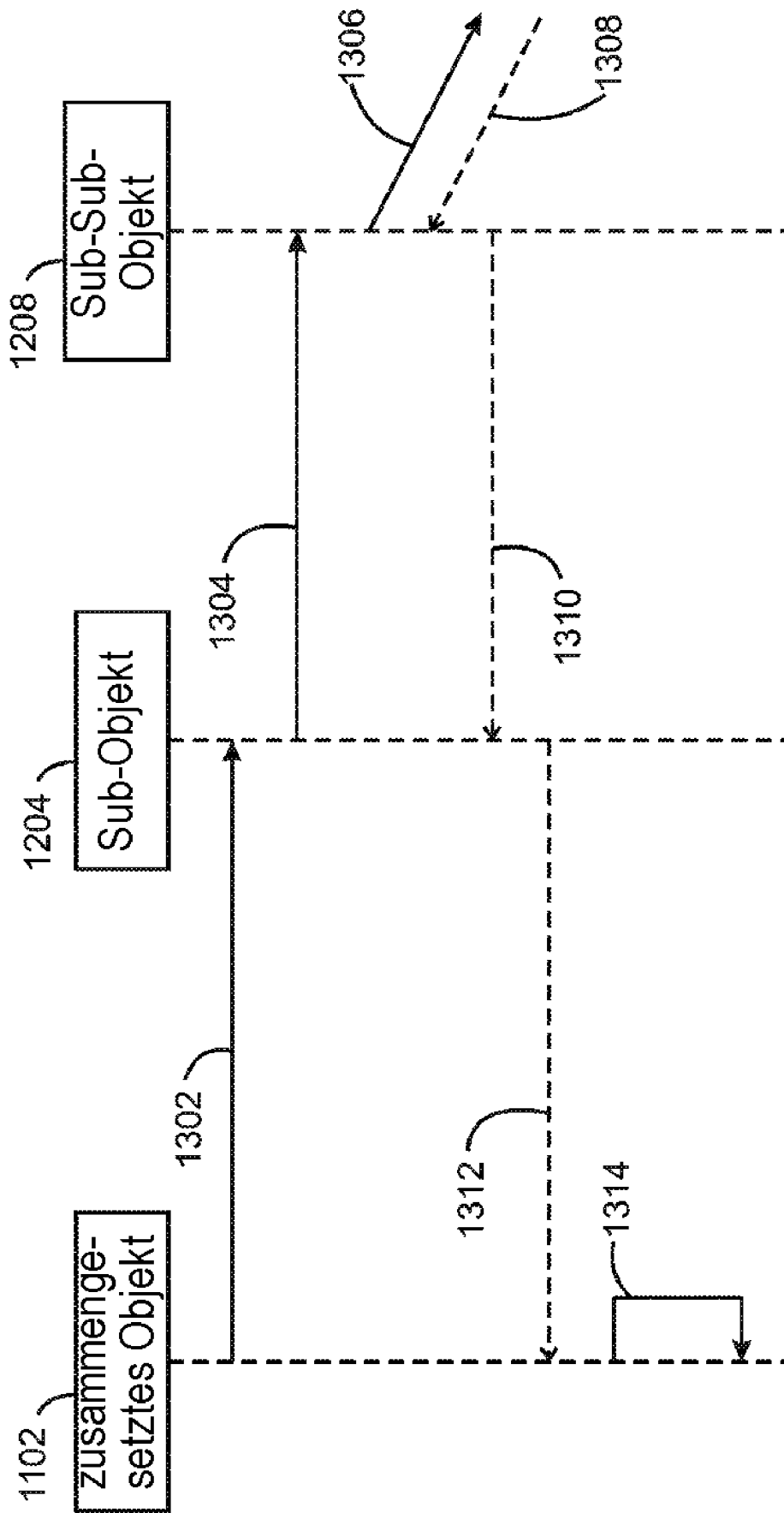
FIG. 10



1100
FIG. 11



1200
FIG. 12



1300

FIG. 13

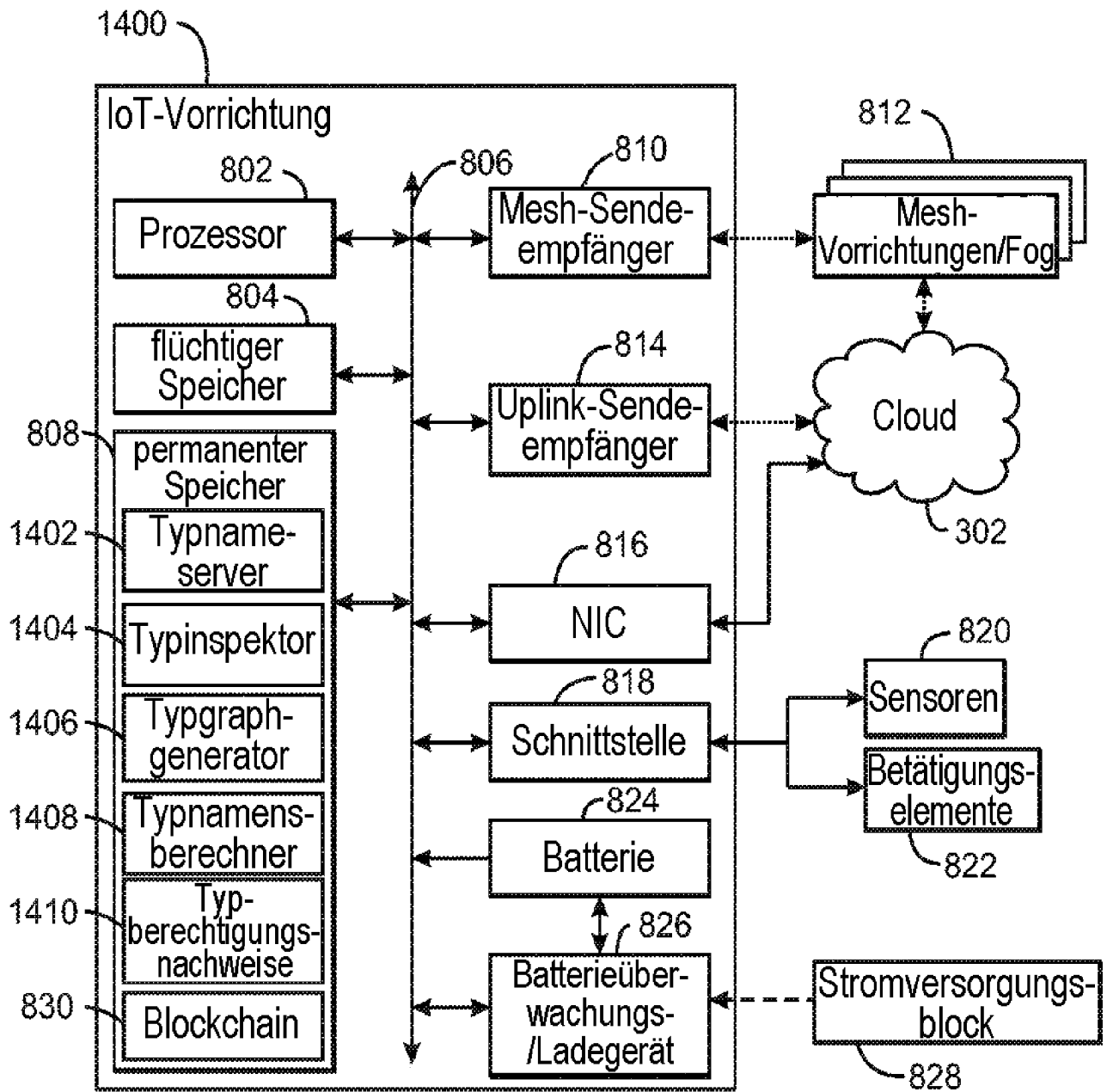


FIG. 14

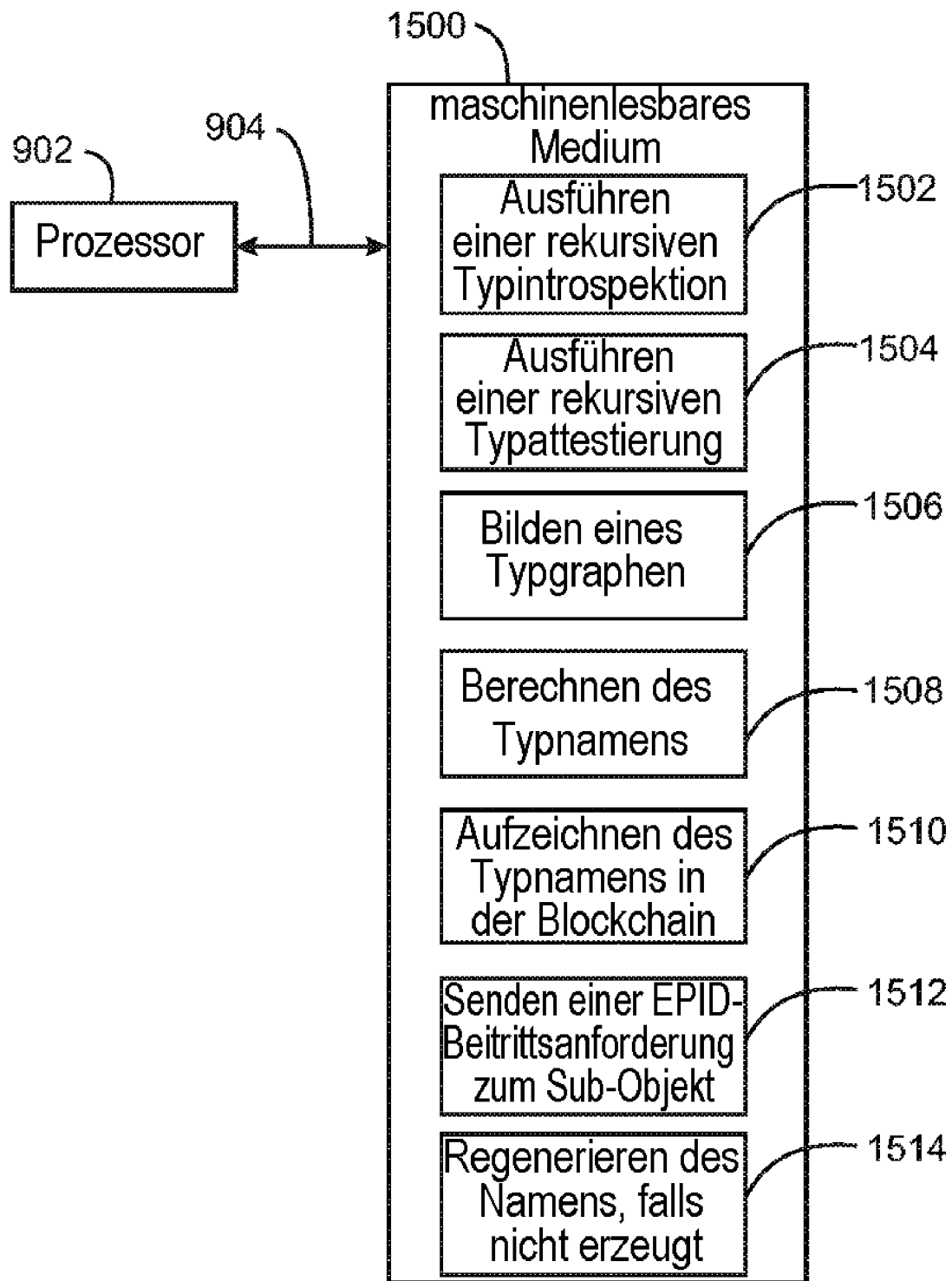
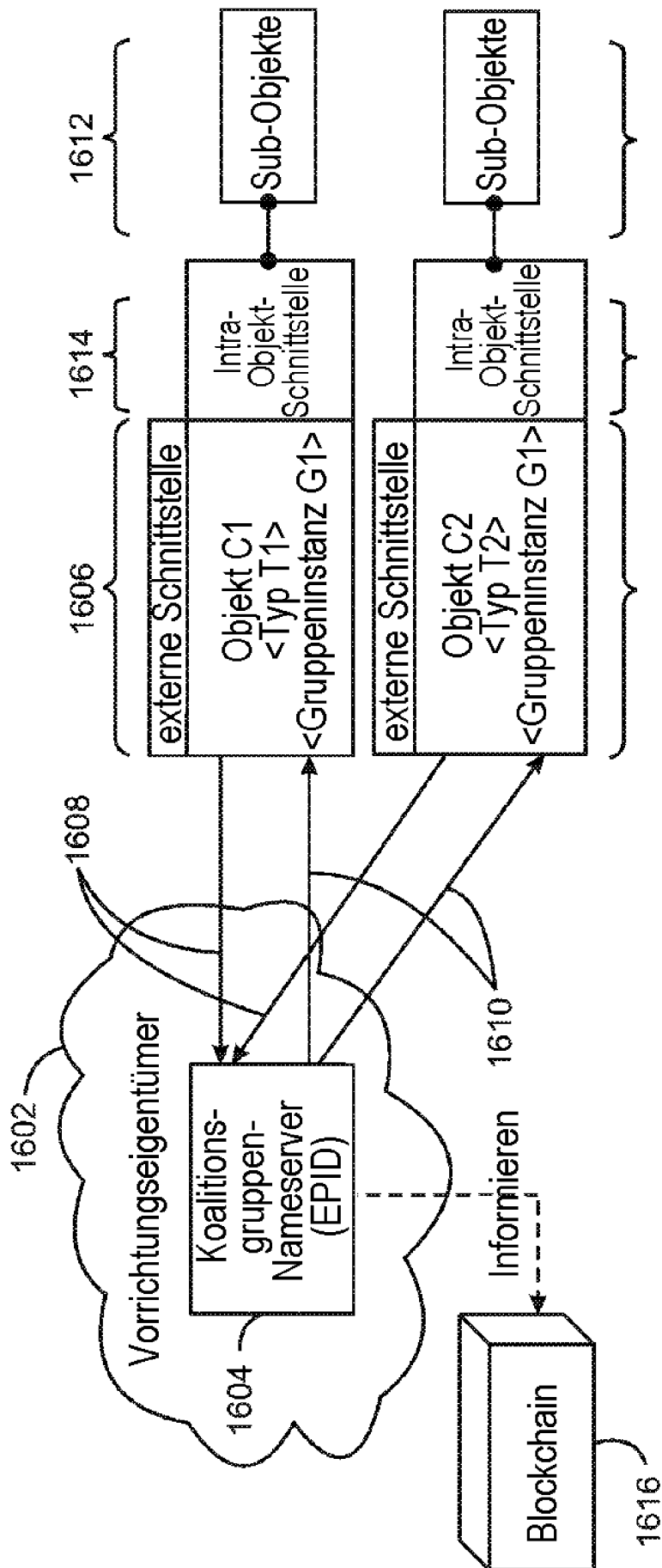
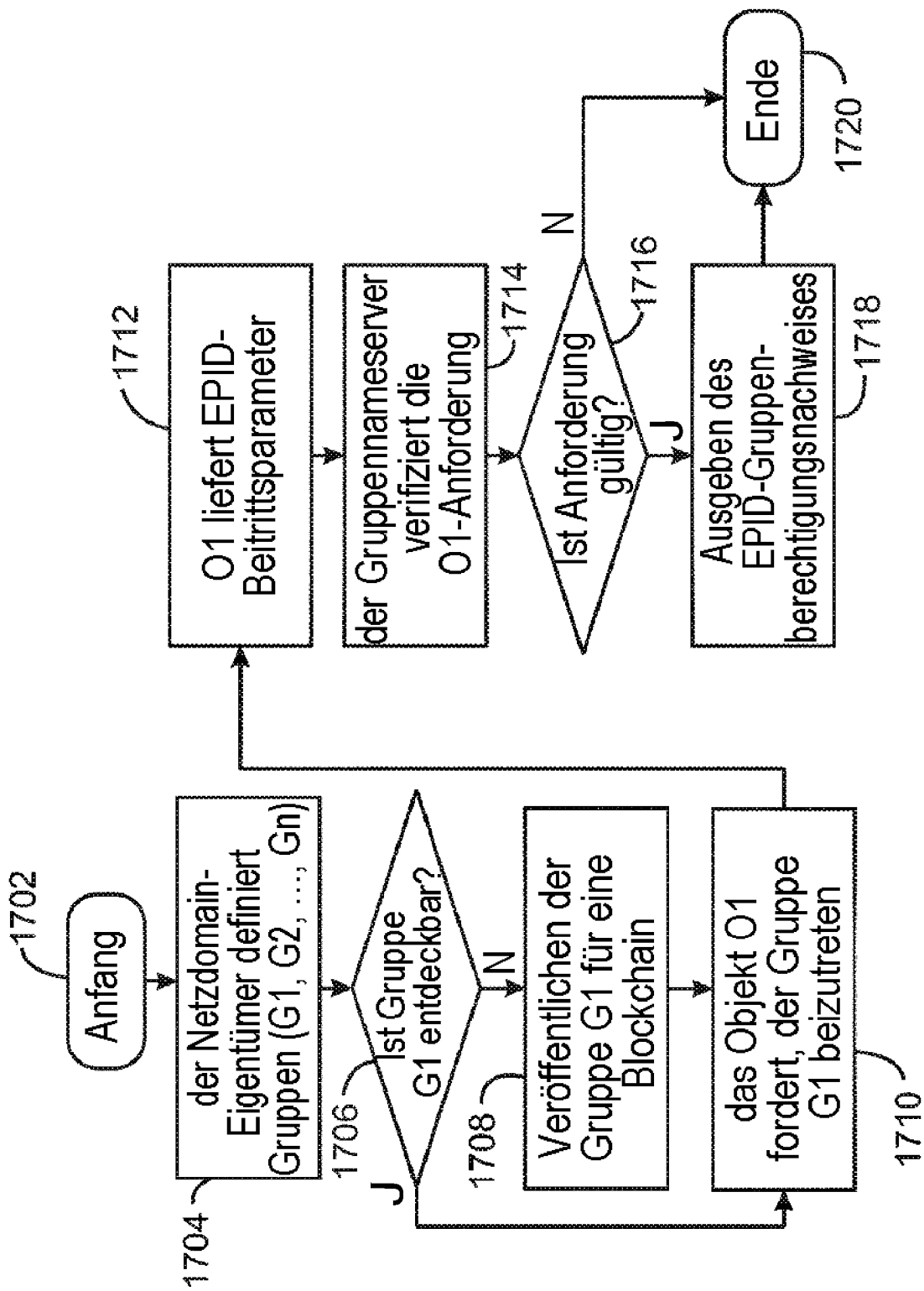


FIG. 15



1600

FIG. 16



1700

FIG. 17

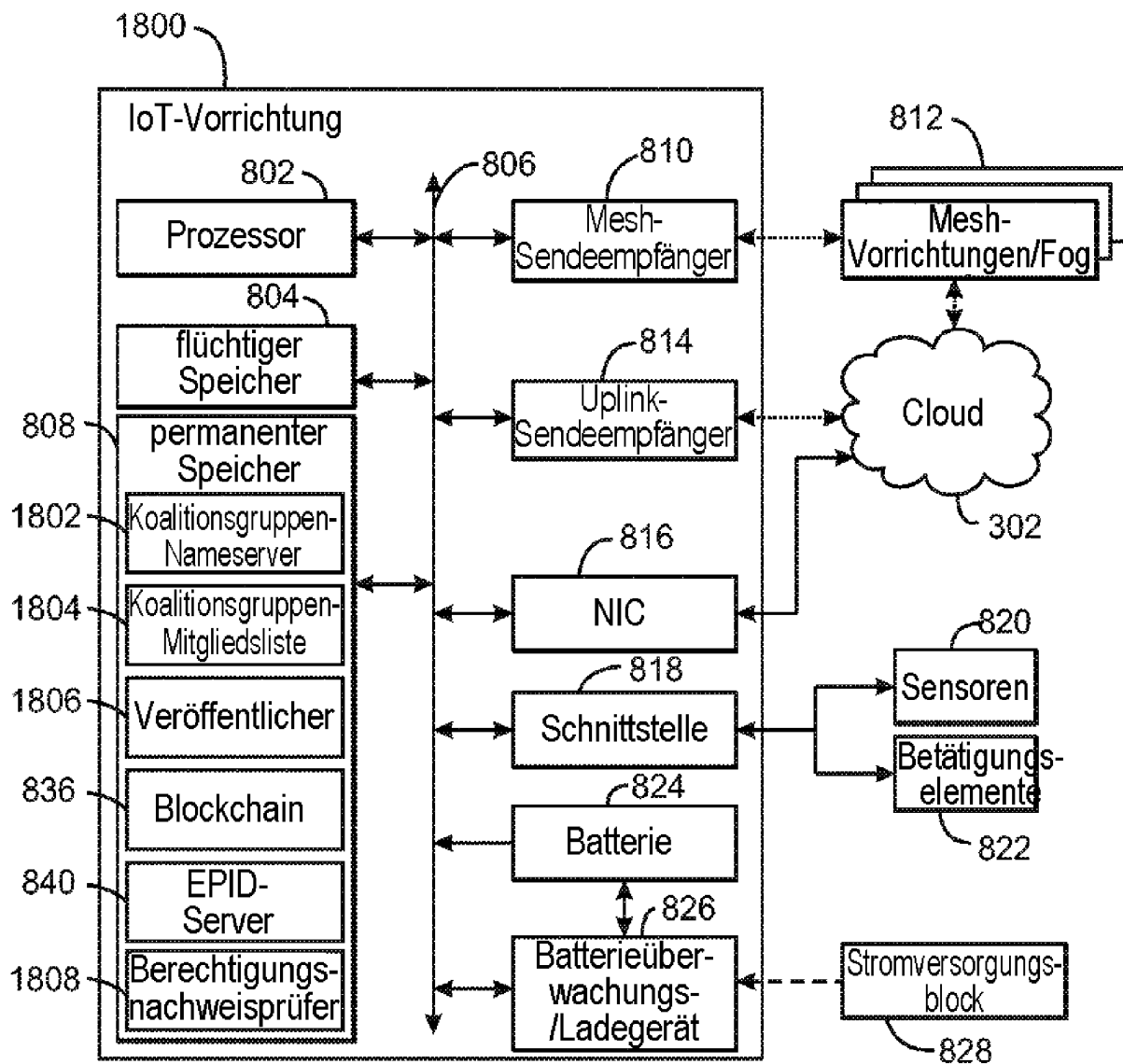


FIG. 18

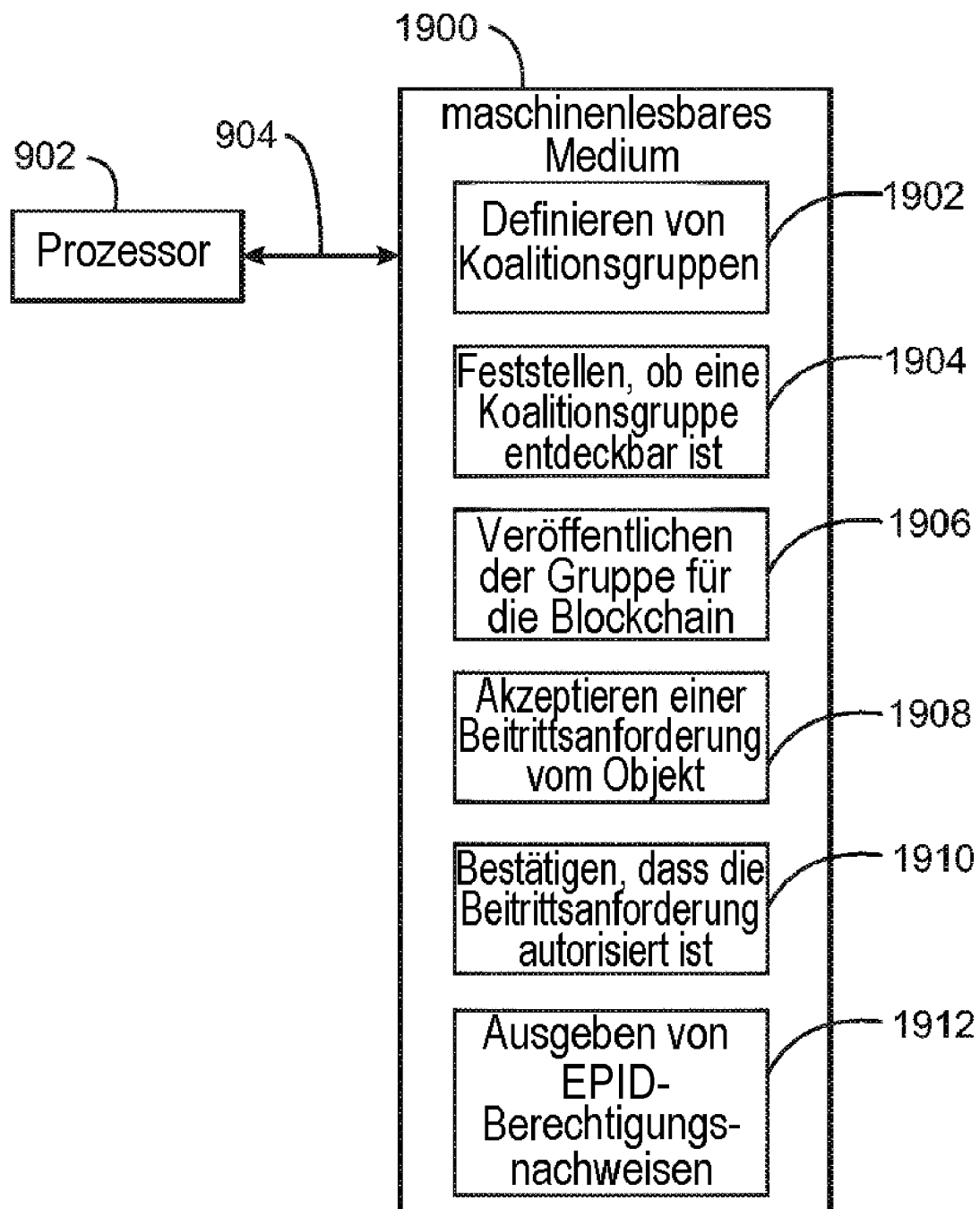
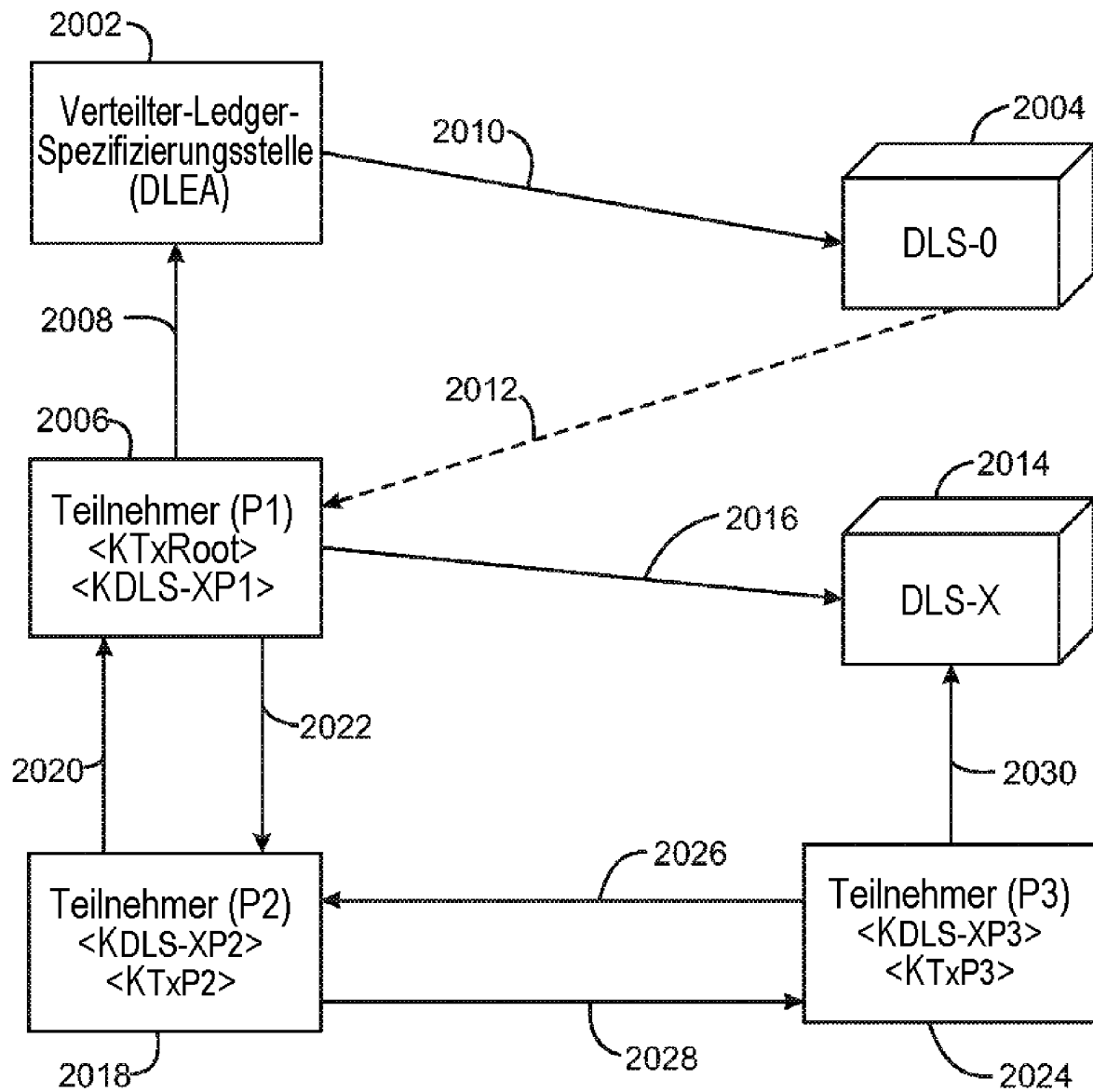
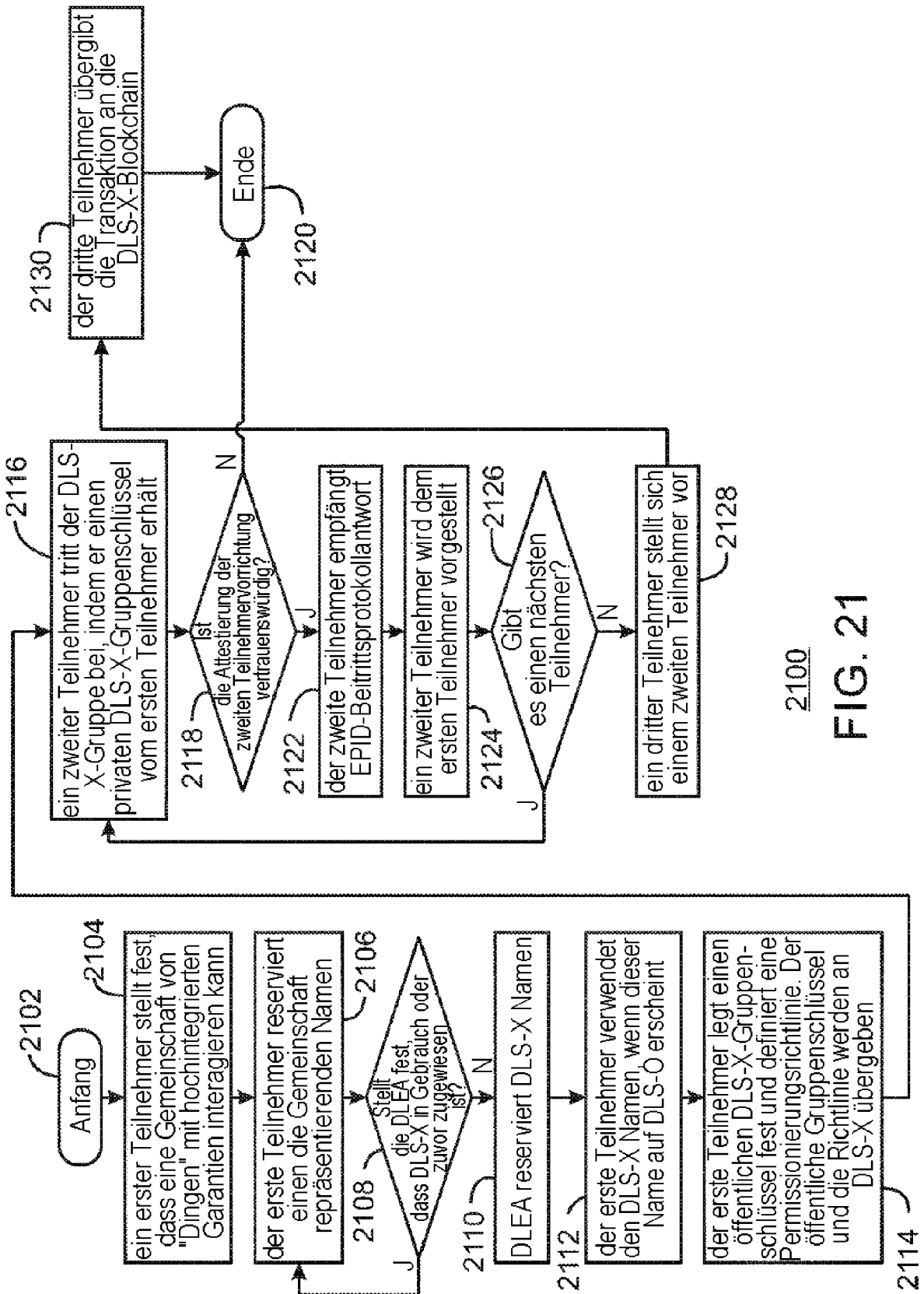


FIG. 19



2000
FIG. 20



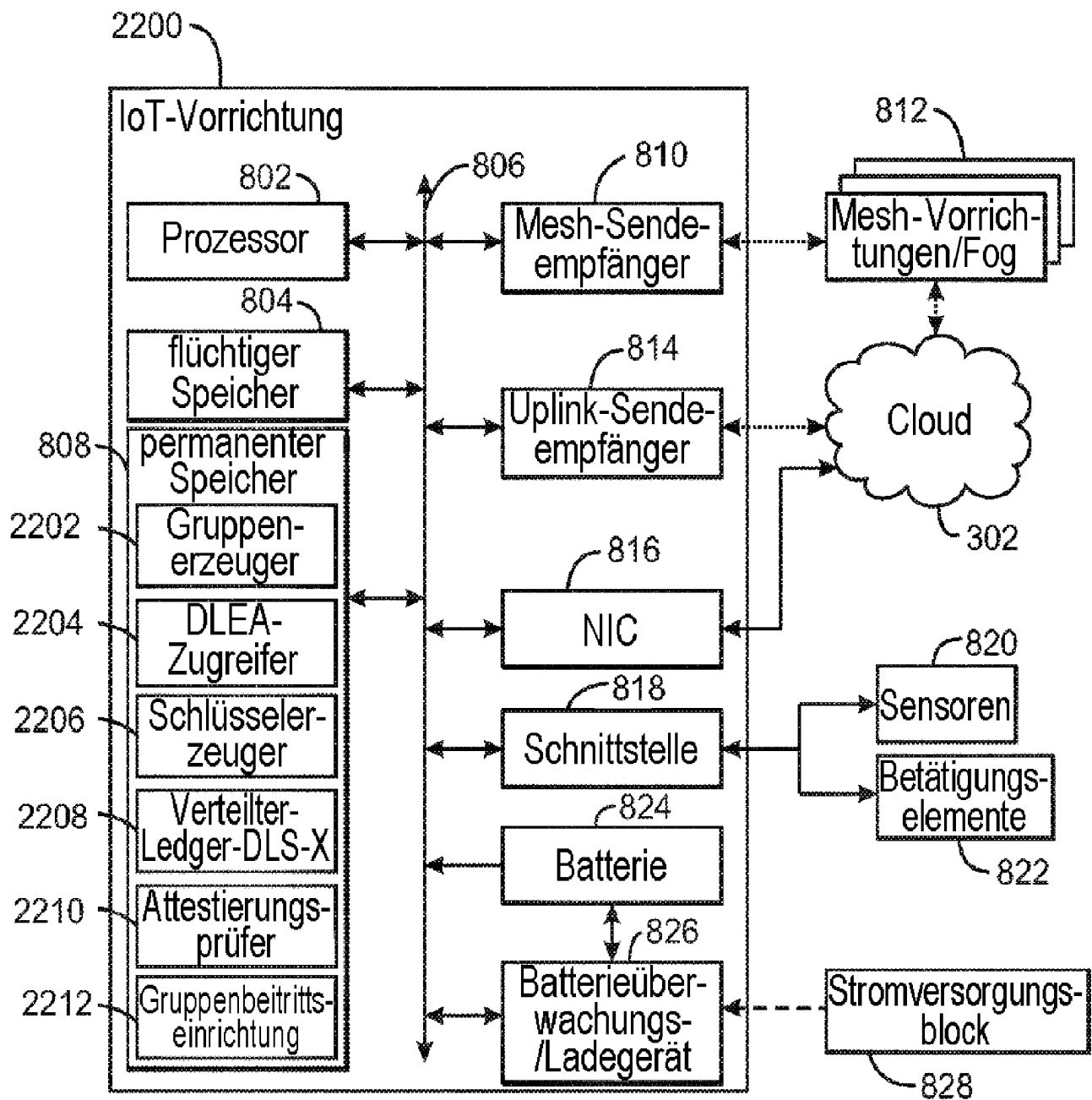


FIG. 22

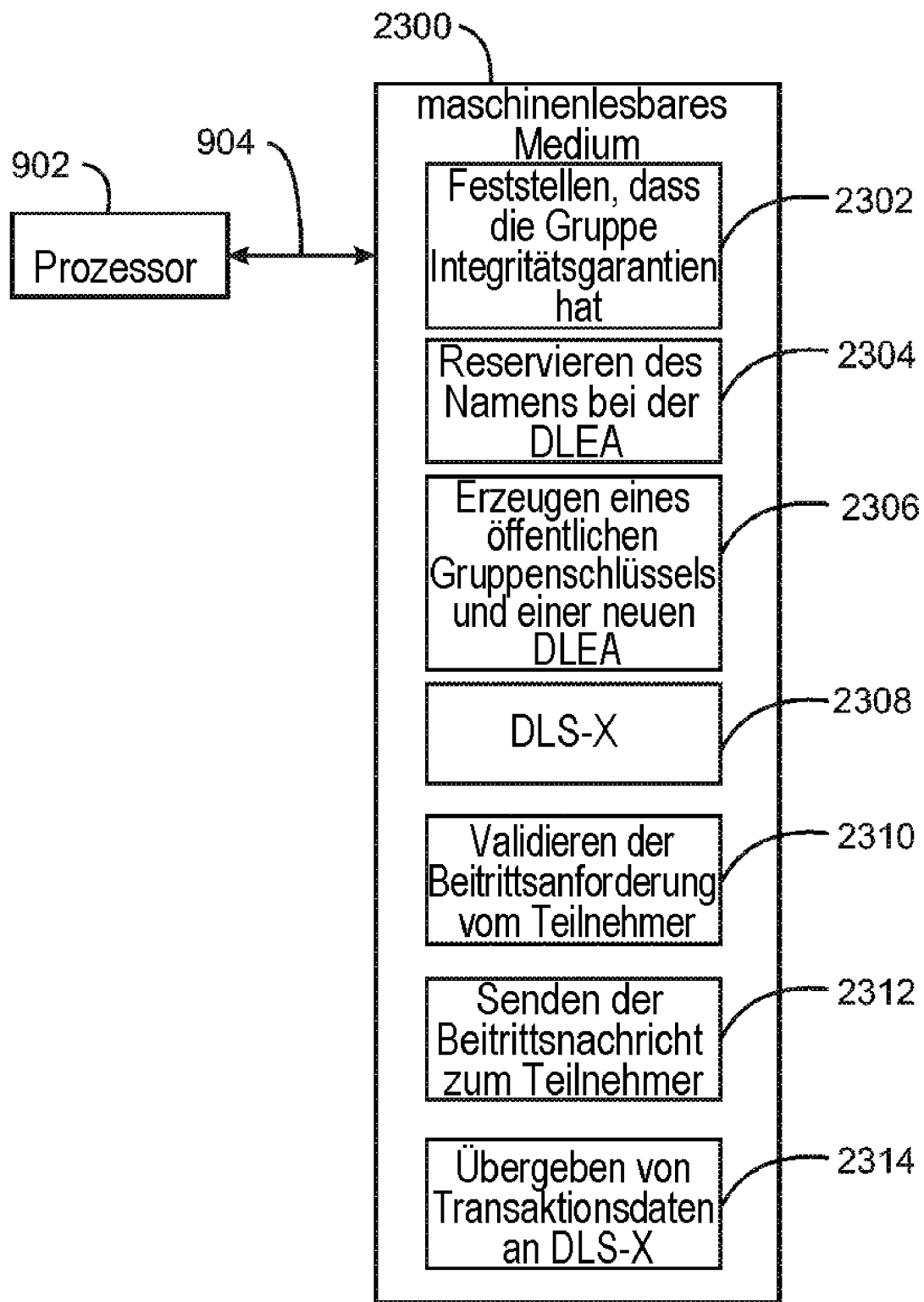
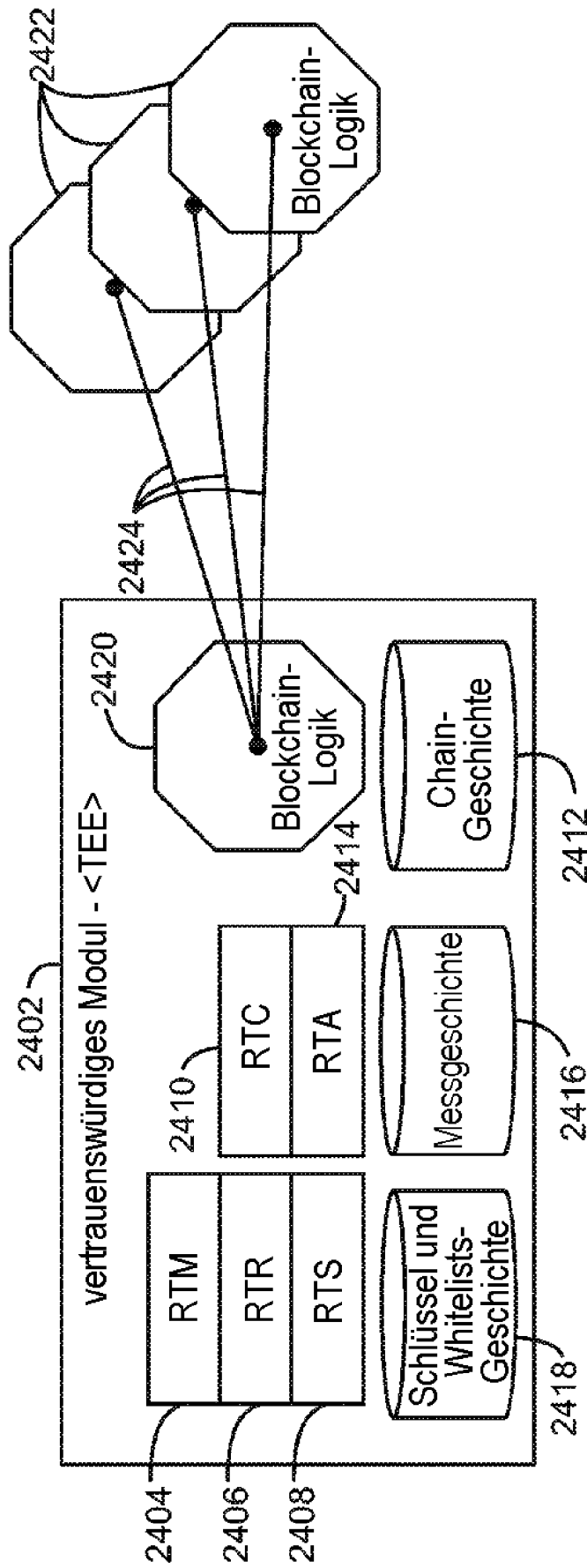
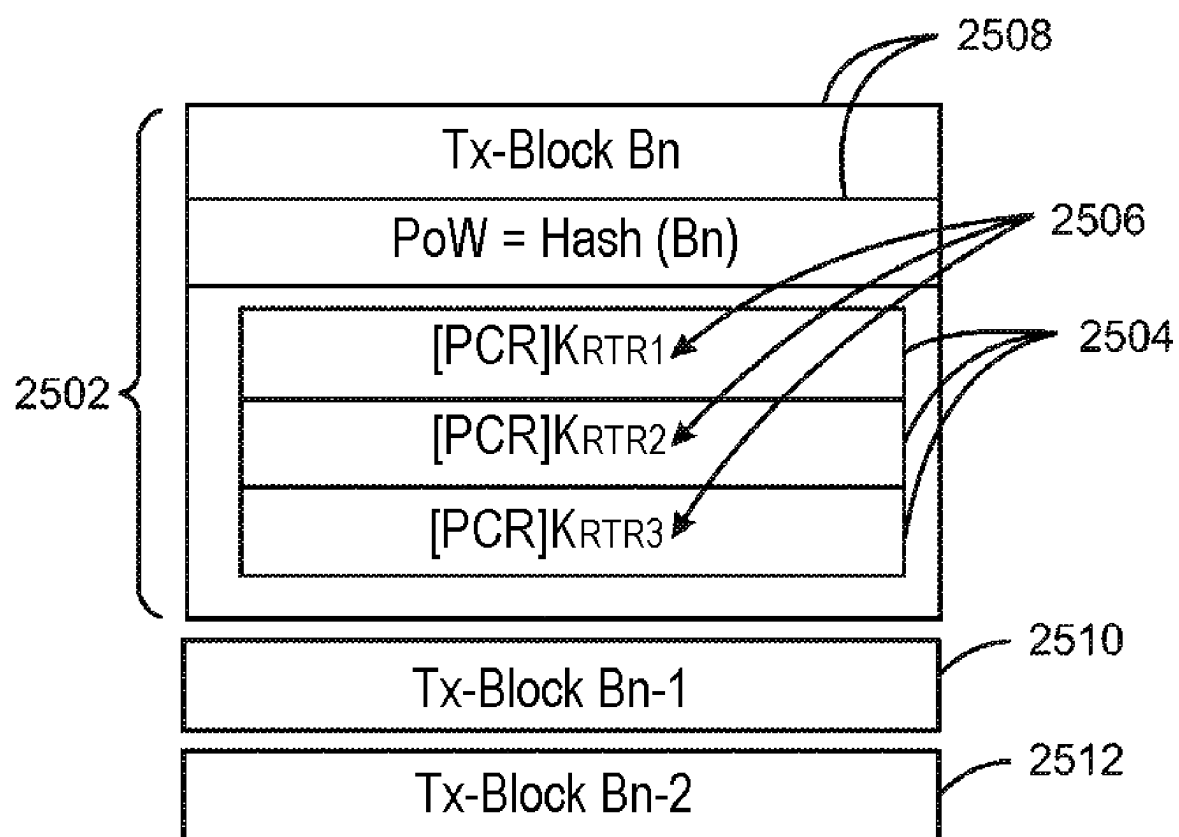


FIG. 23



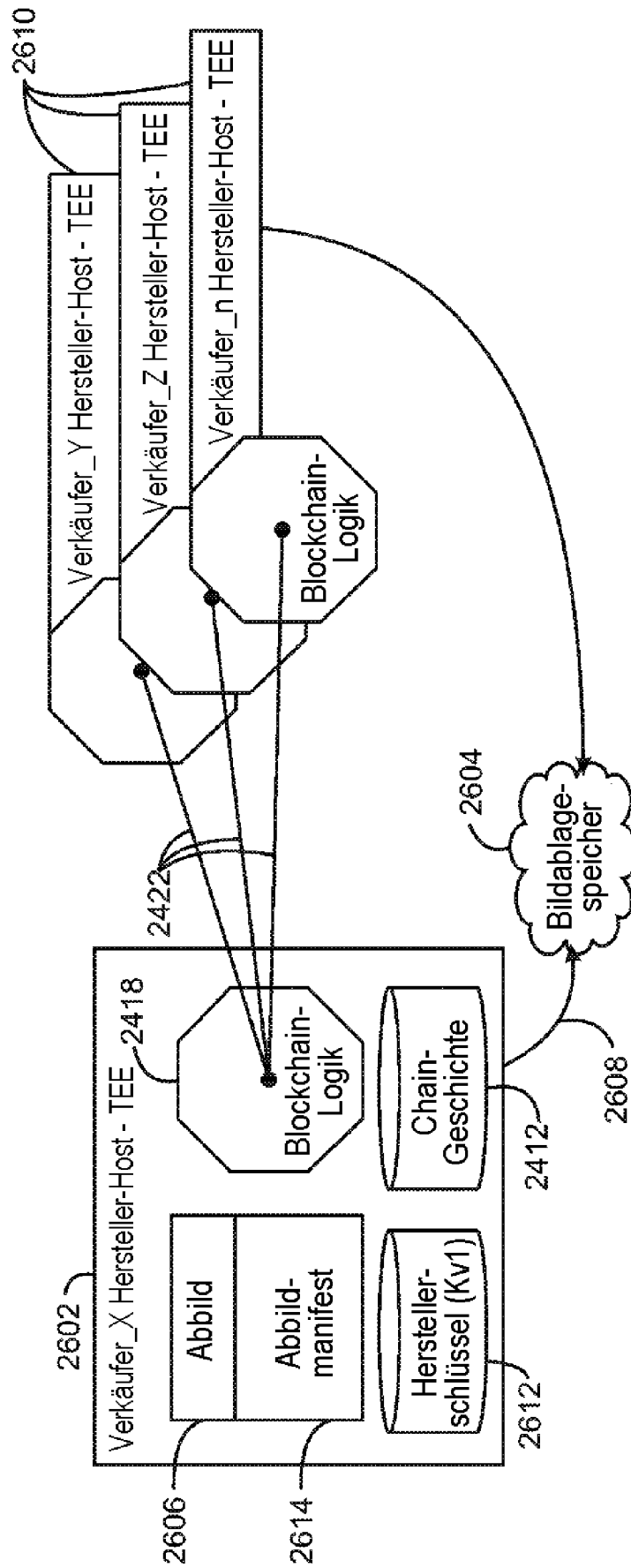
2400

FIG. 24



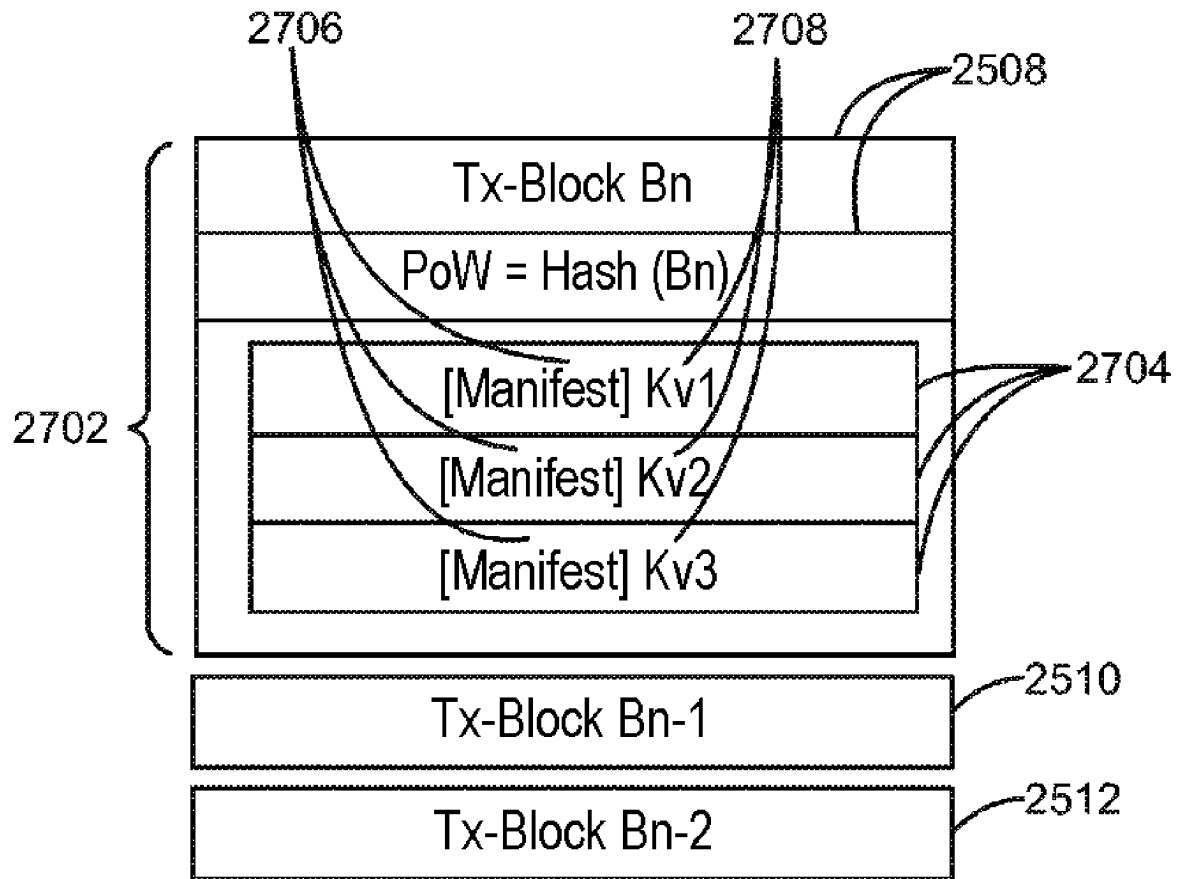
2500

FIG. 25



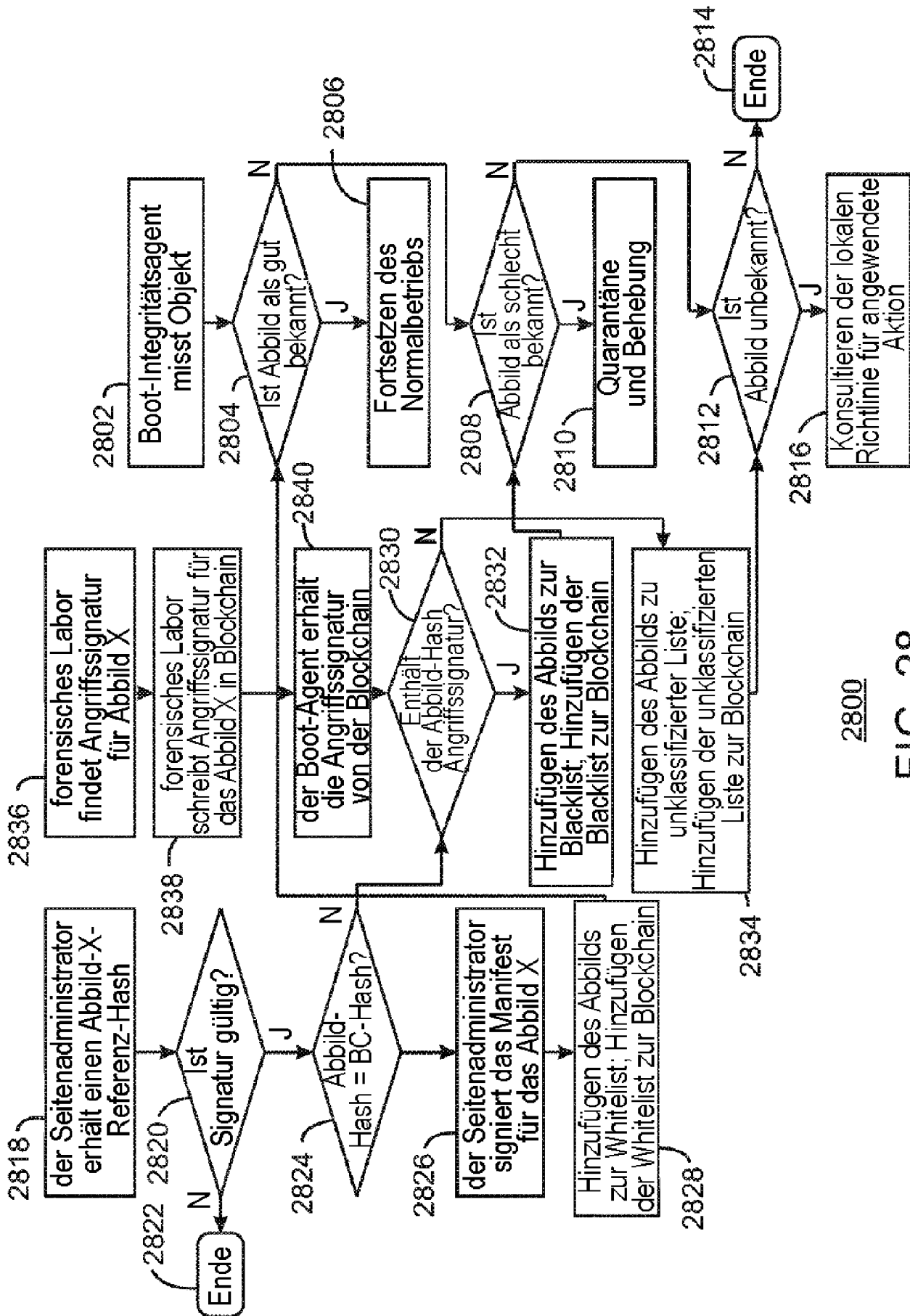
2600

FIG. 26



2700

FIG. 27



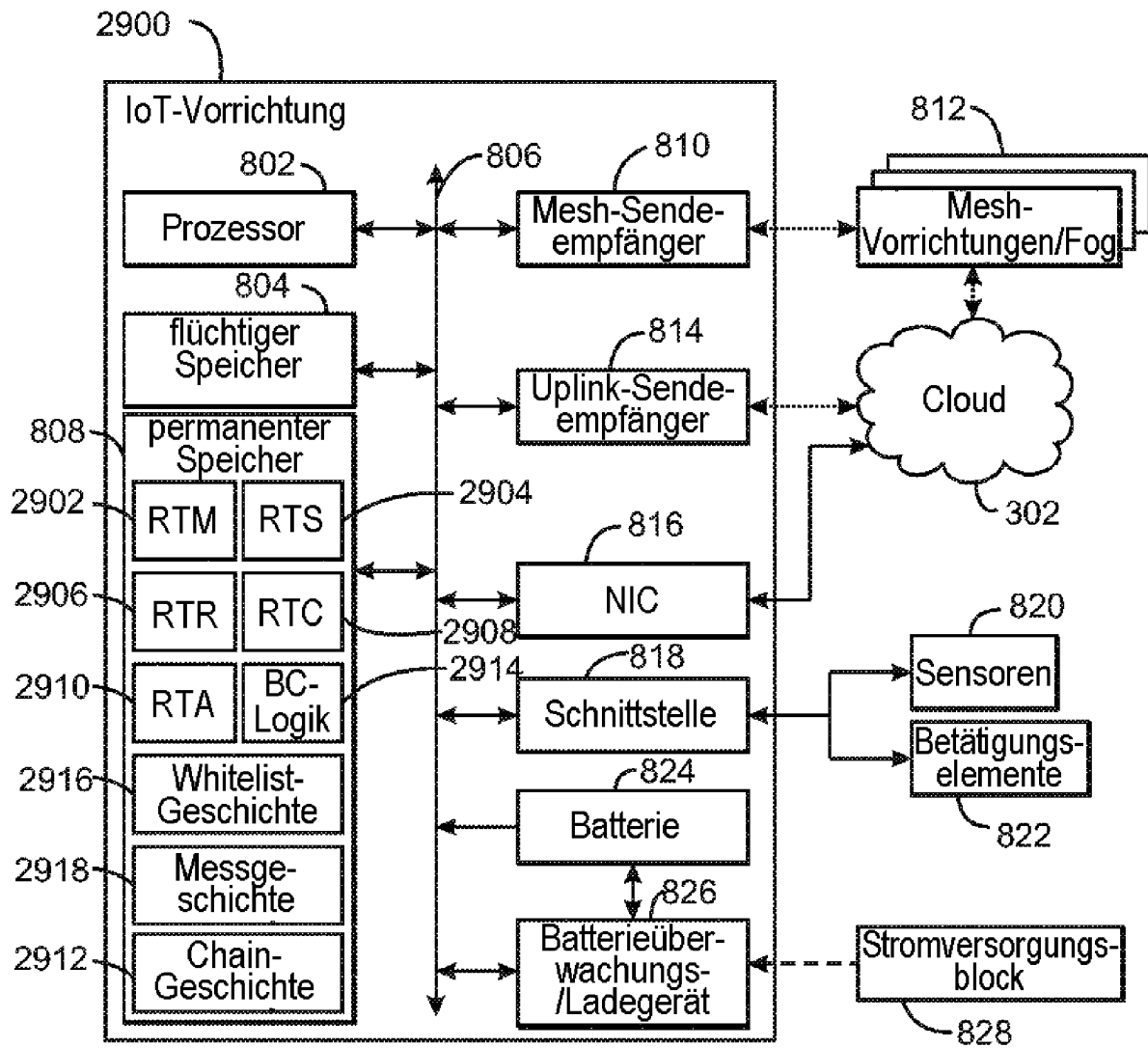


FIG. 29

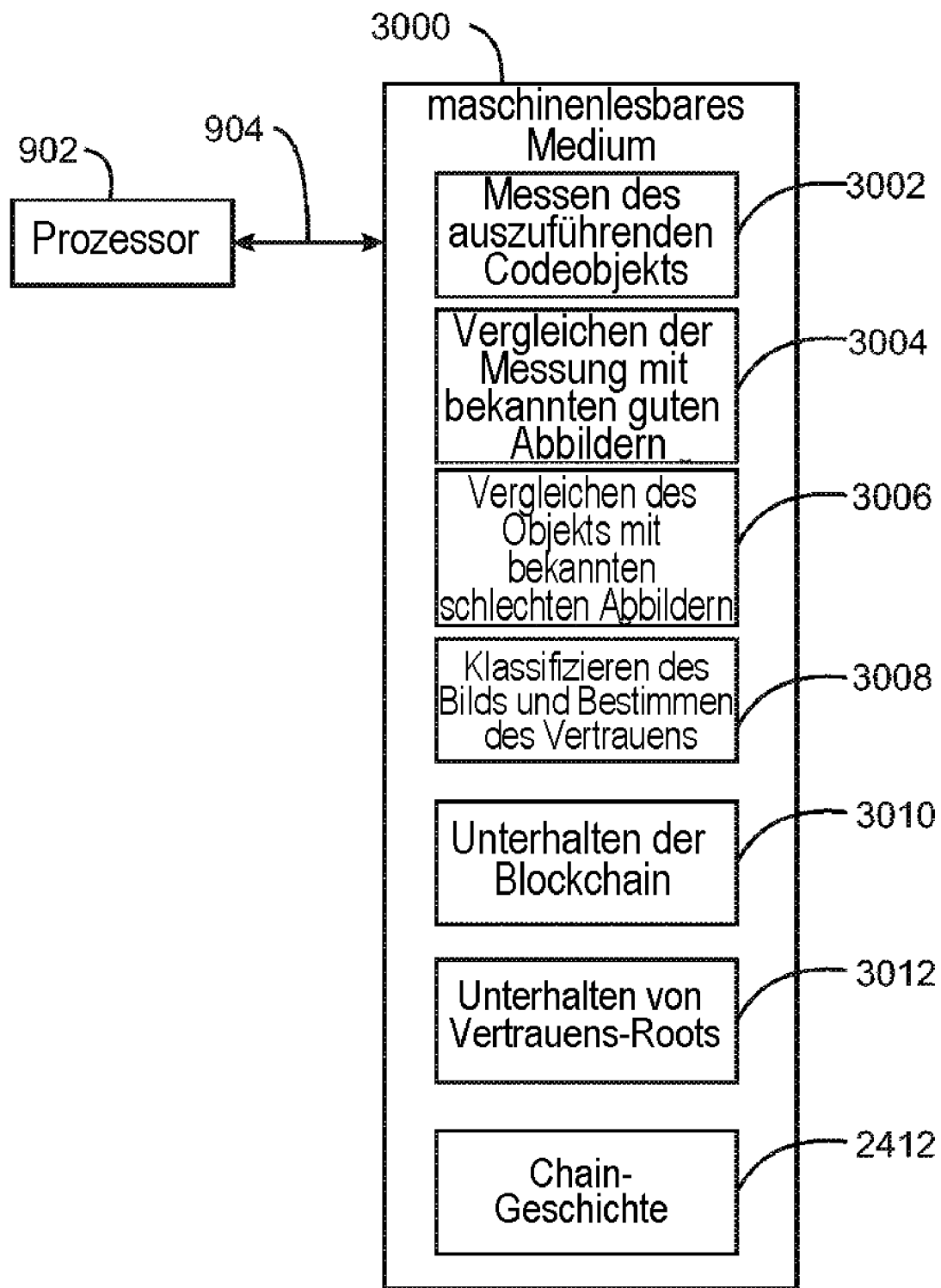


FIG. 30

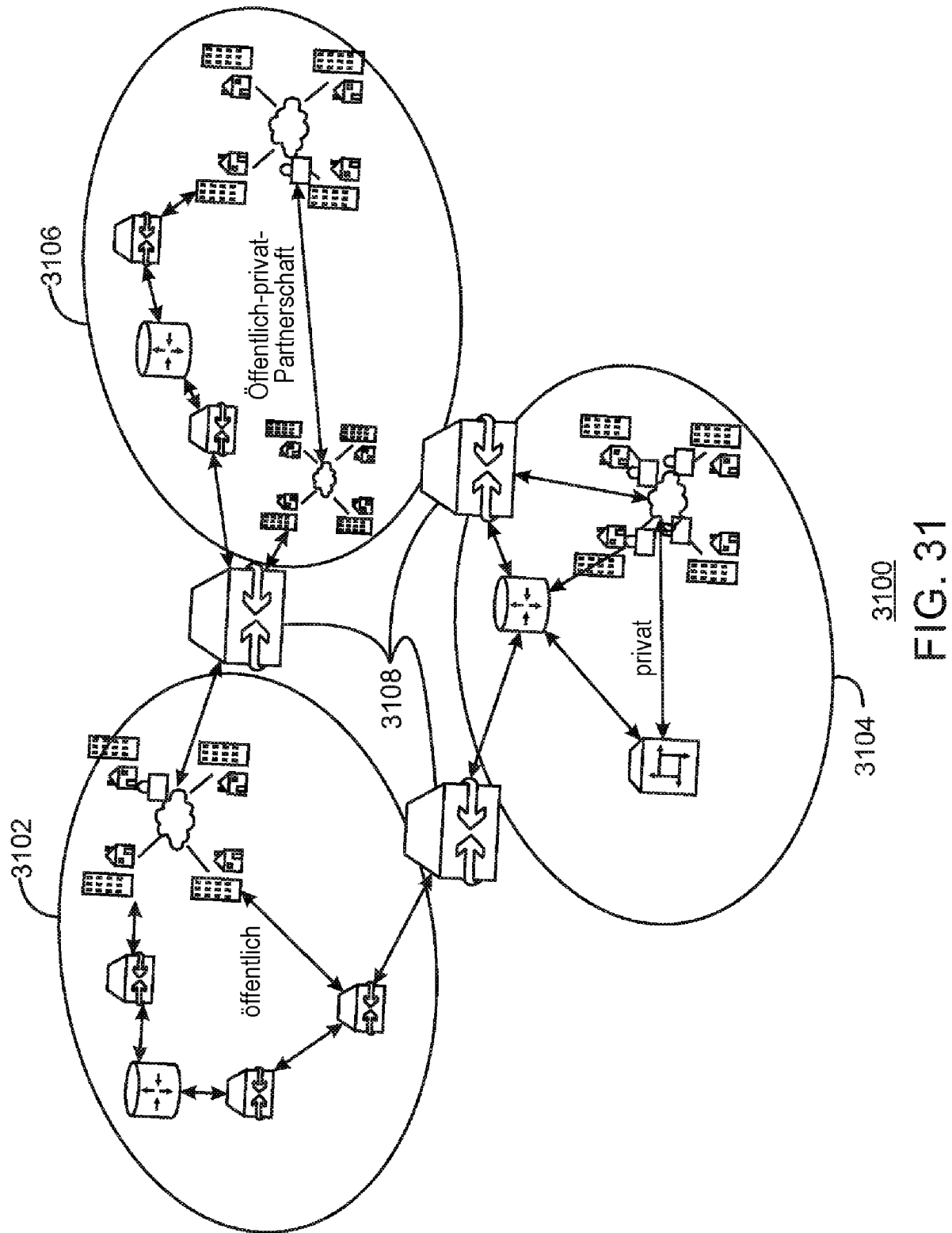
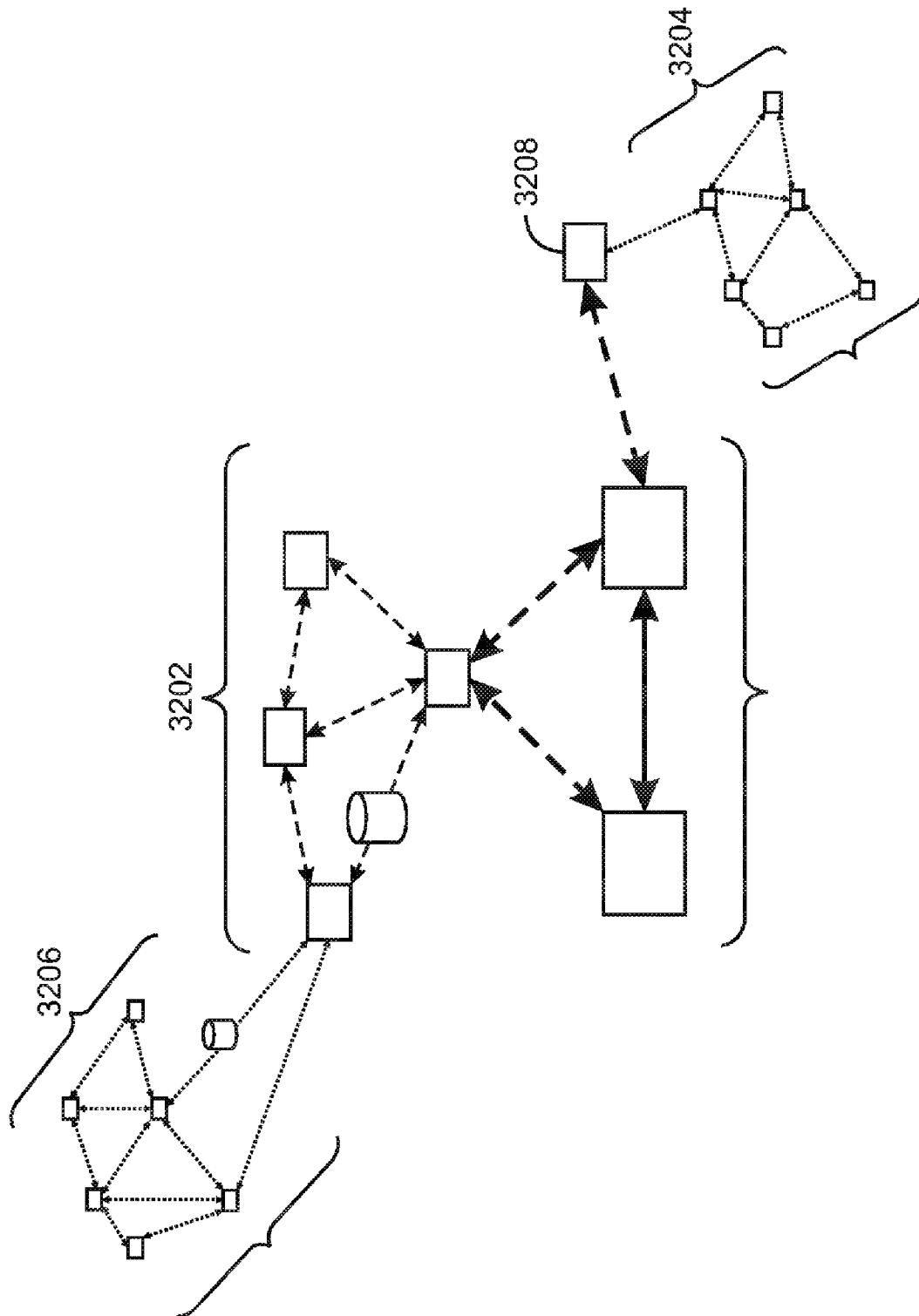
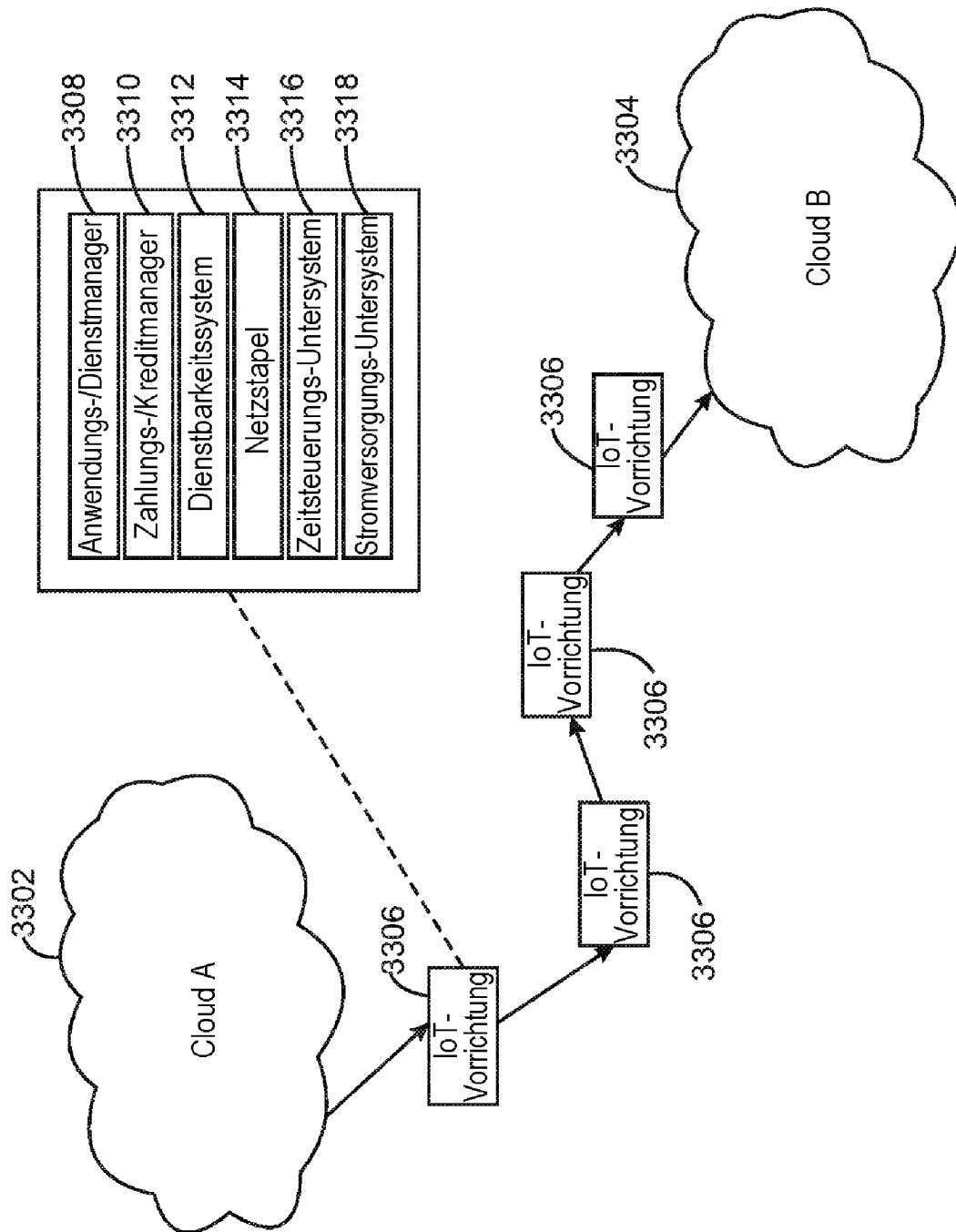


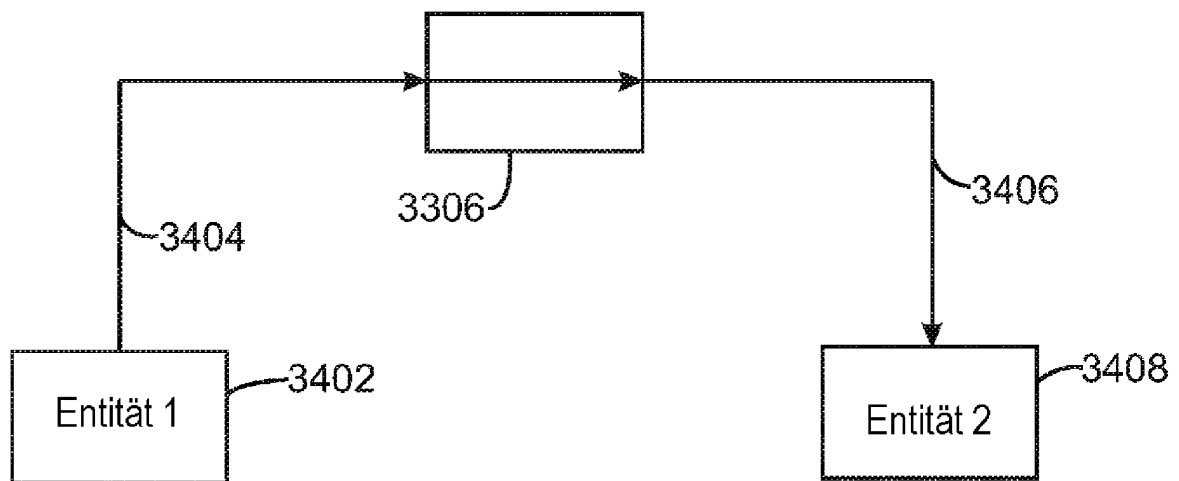
FIG. 31



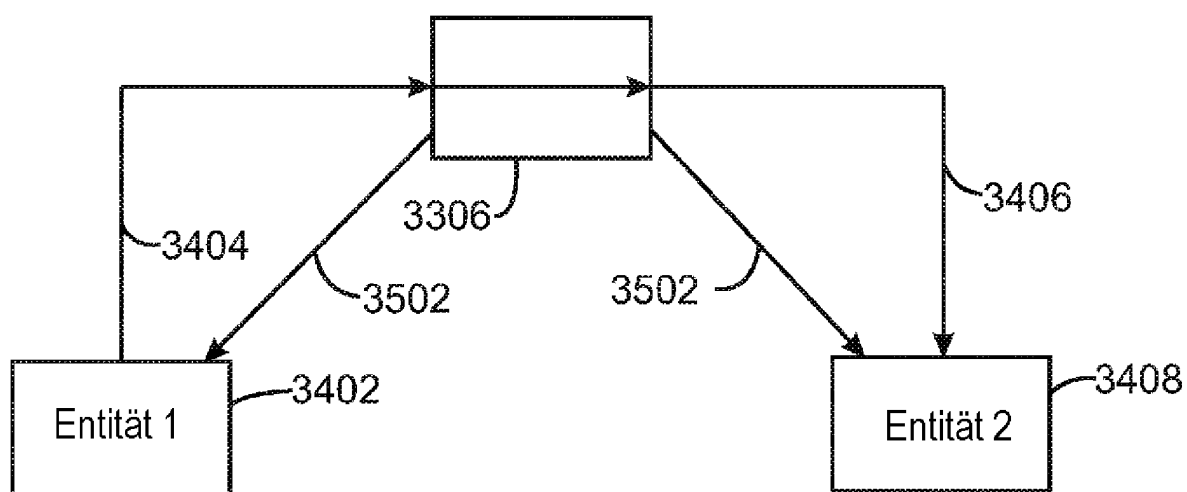
3200
FIG. 32



3300
FIG. 33



3400
FIG. 34



3500
FIG. 35

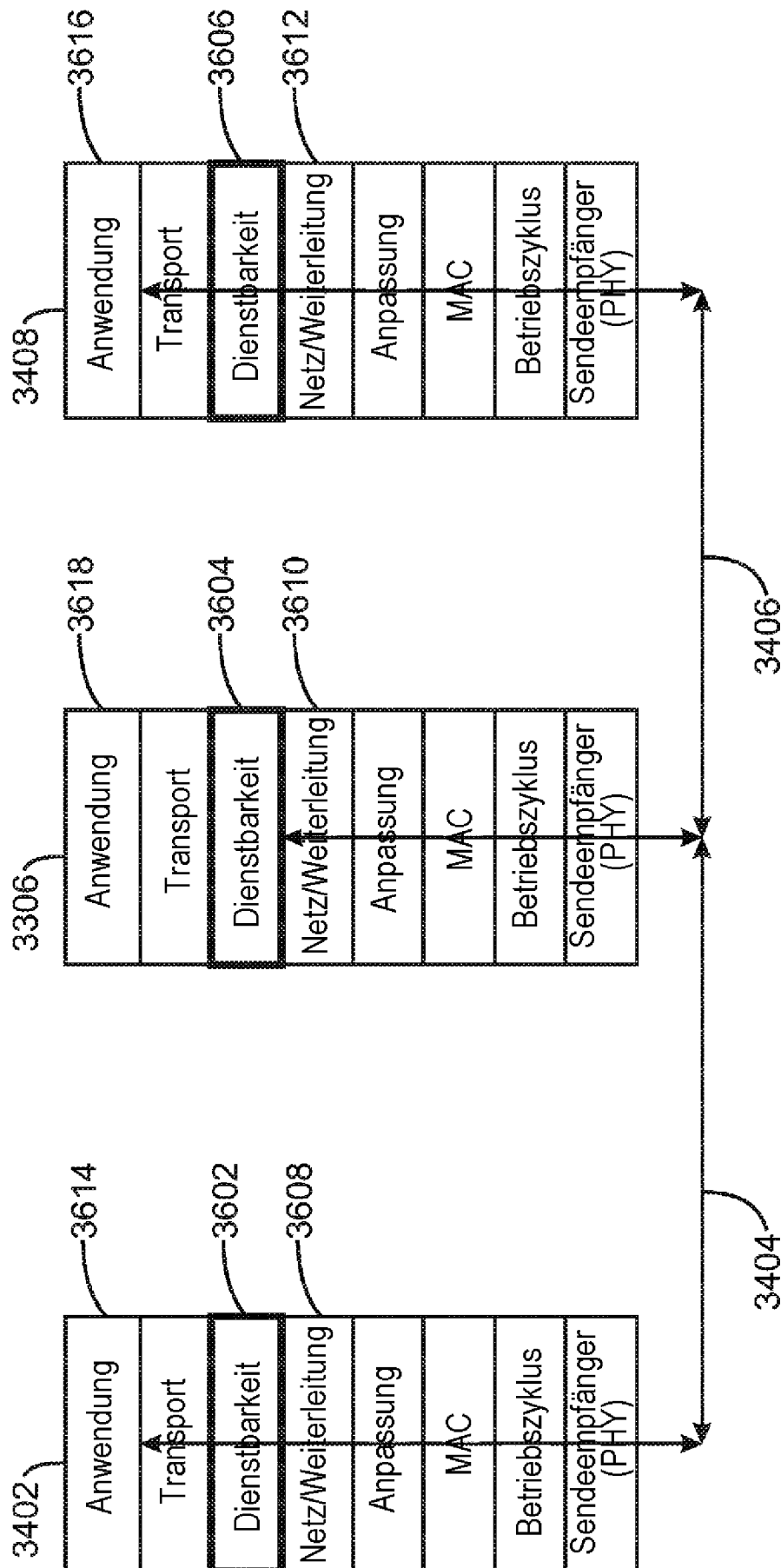
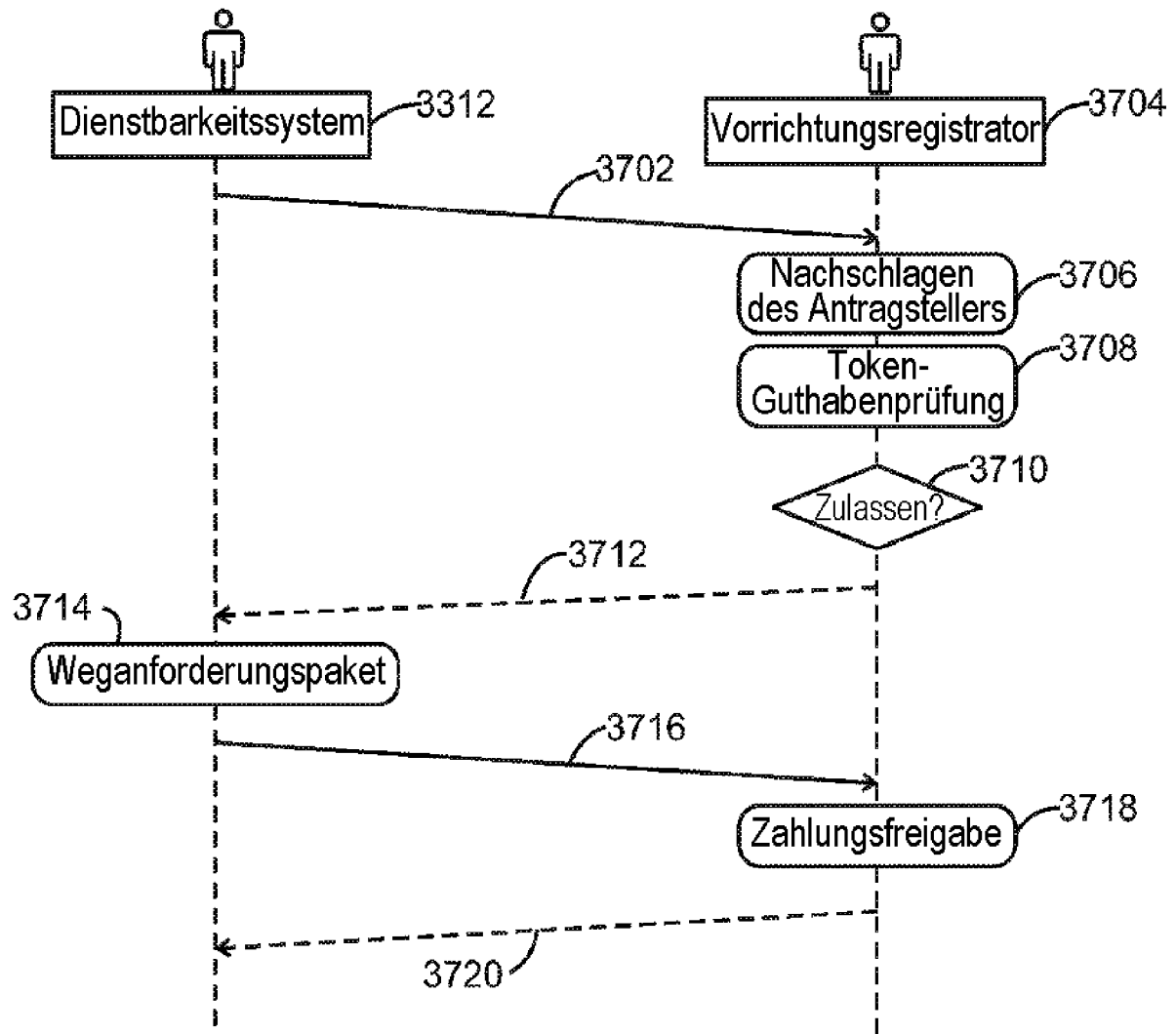
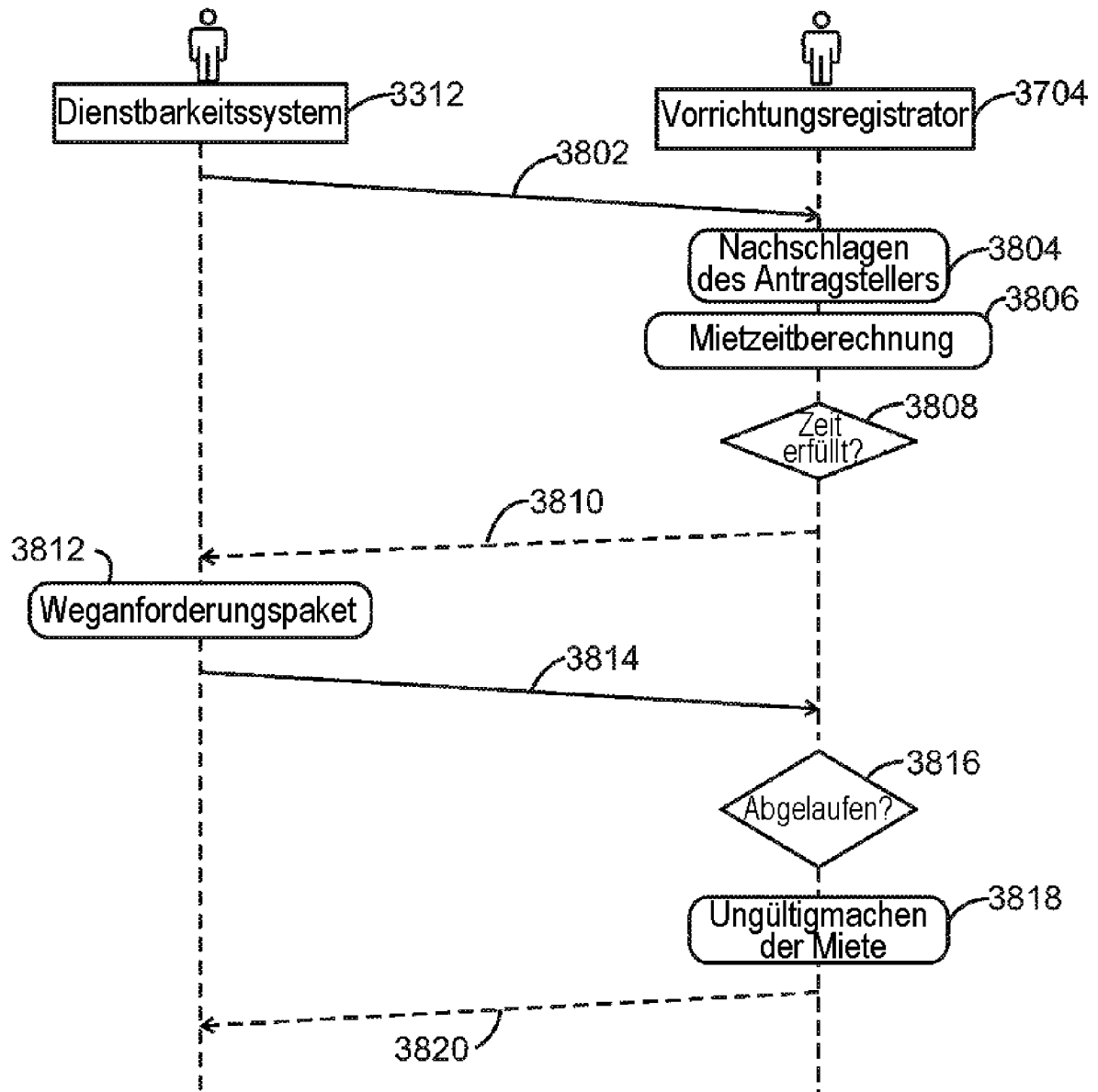
3600

FIG. 36



3700

FIG. 37



3800
FIG.38

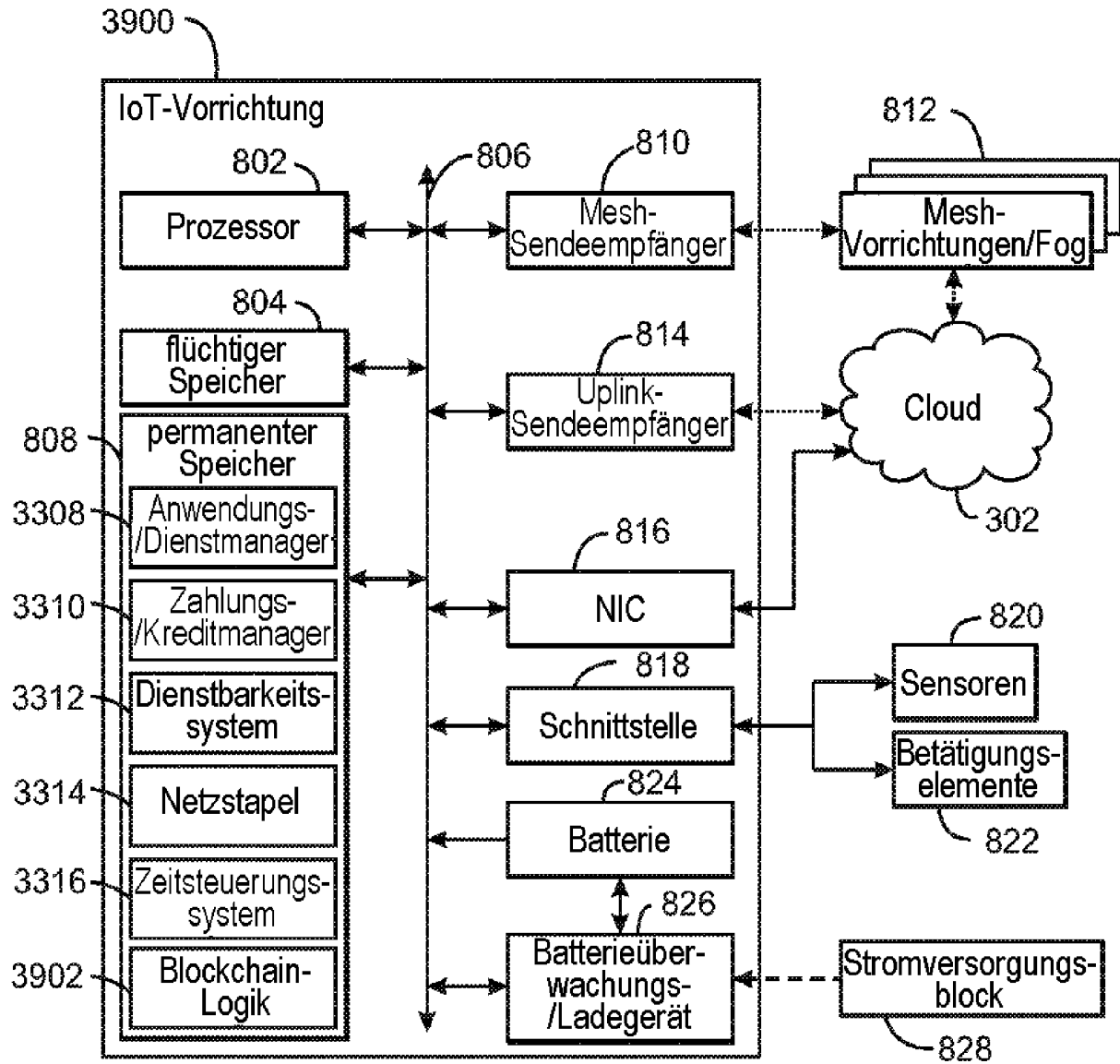


FIG. 39

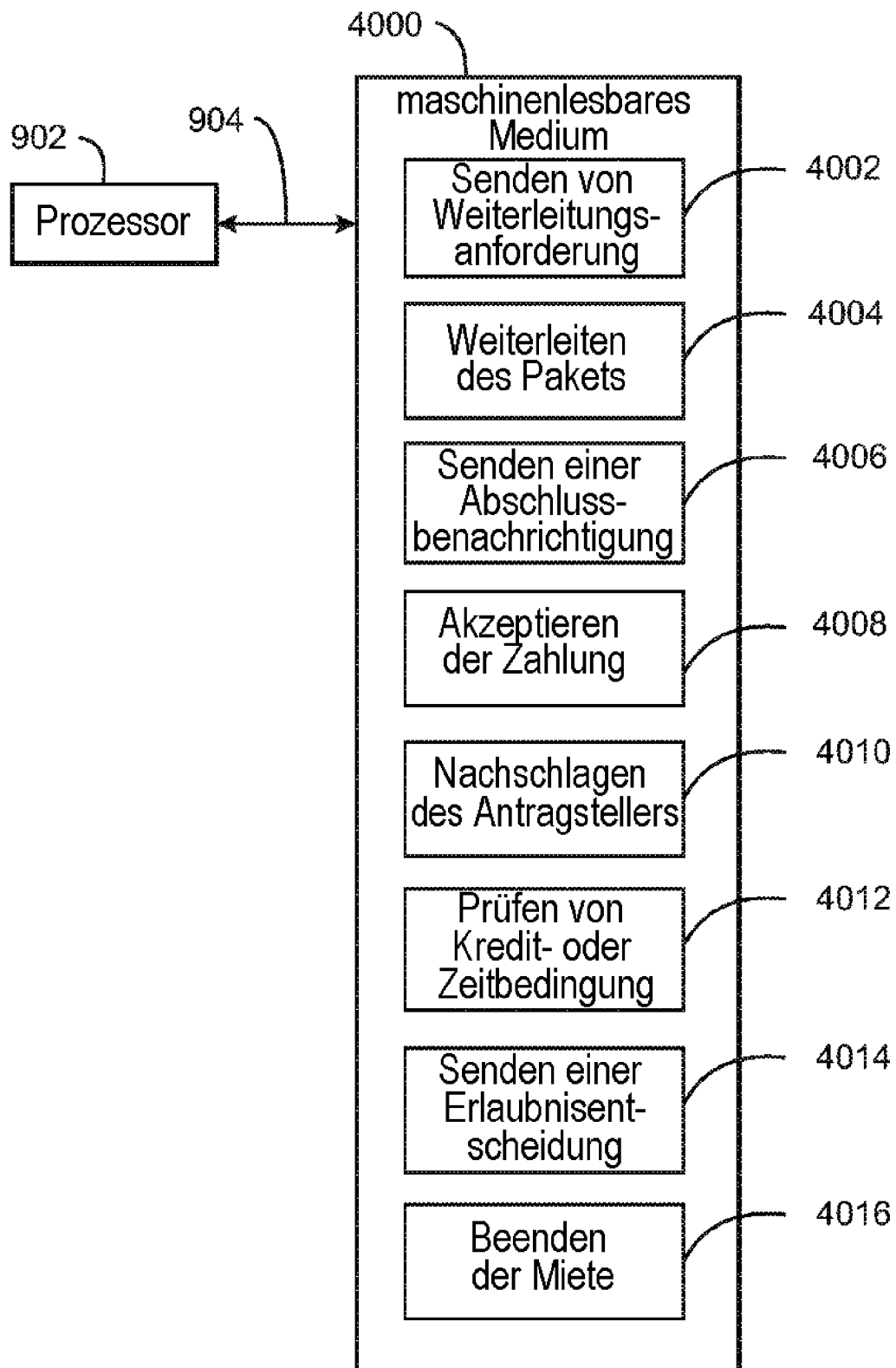
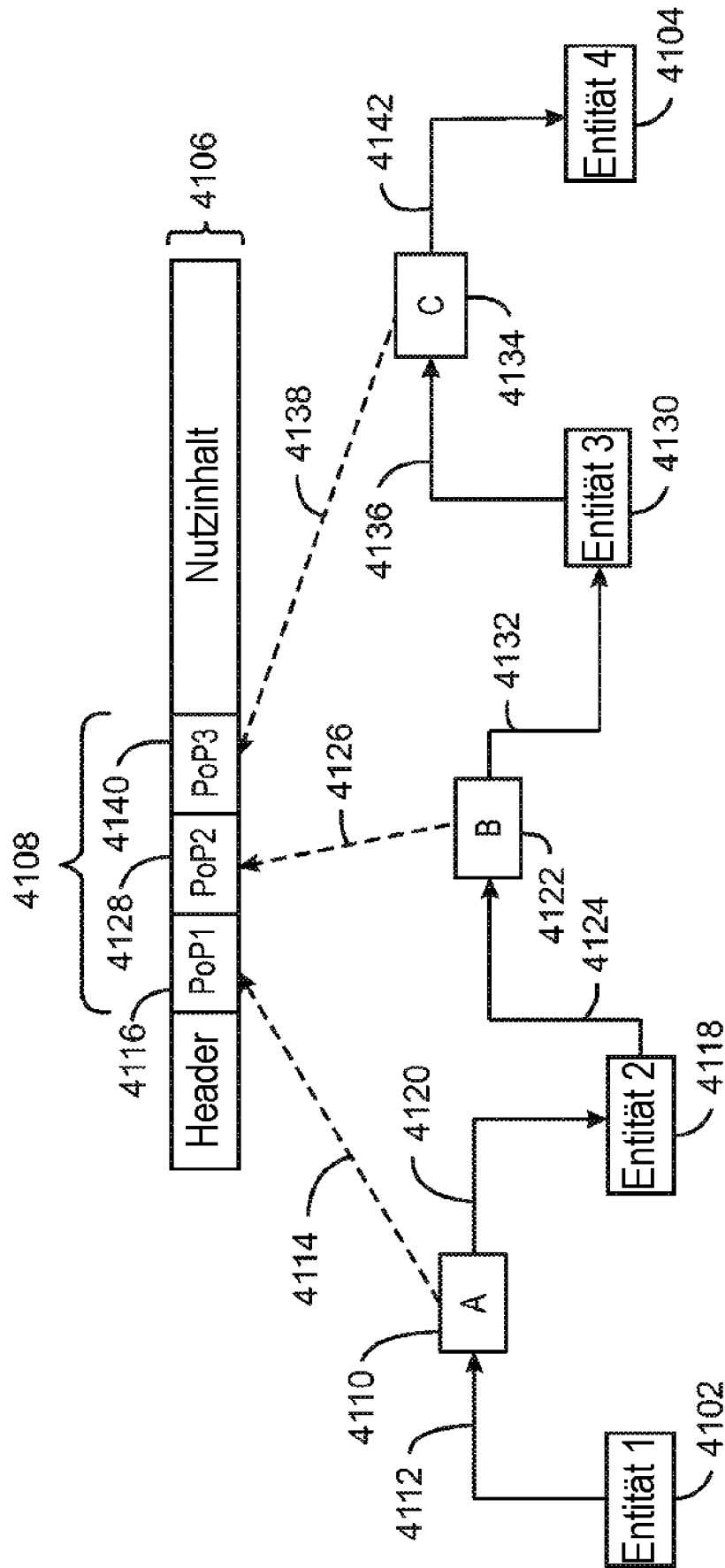


FIG. 40



4100

FIG. 41

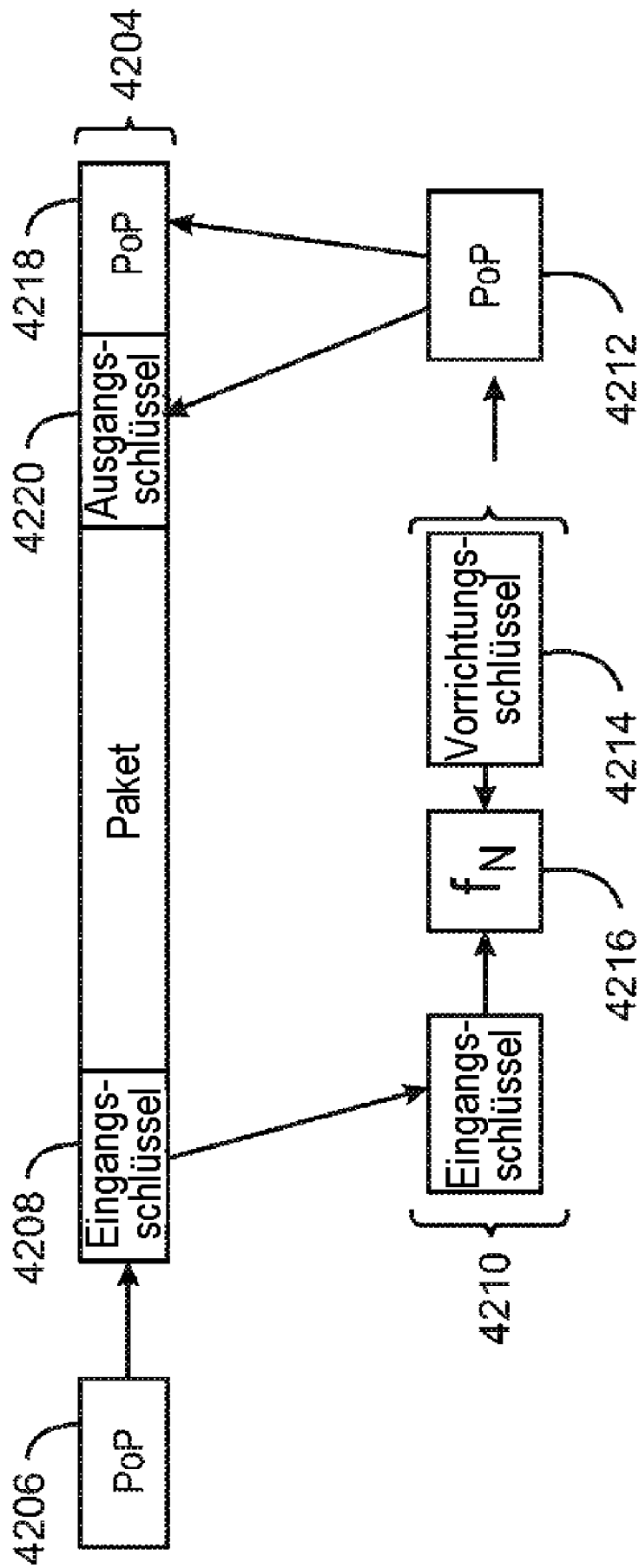
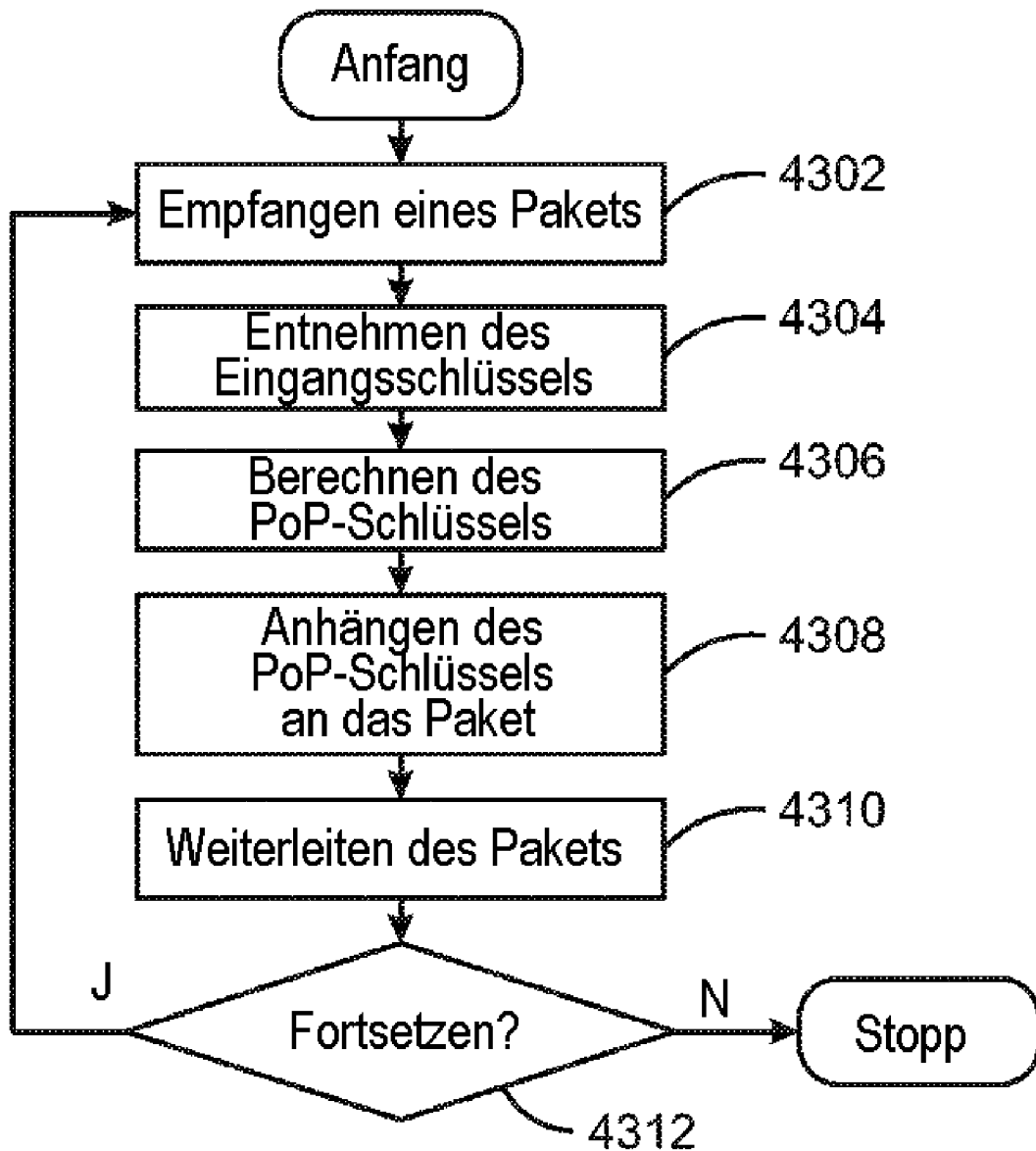
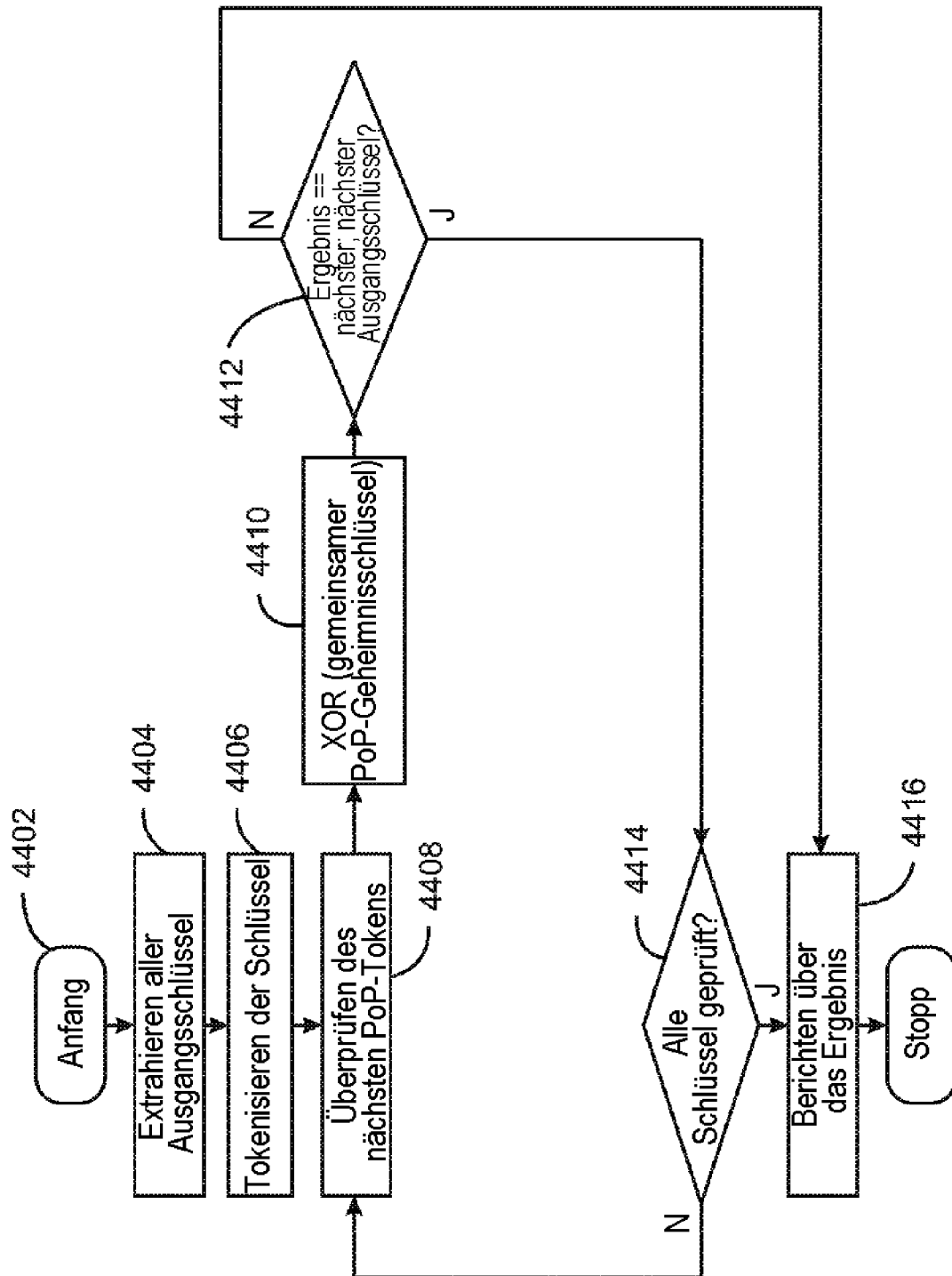
4200

FIG. 42



4300

FIG. 43



4400

FIG. 44

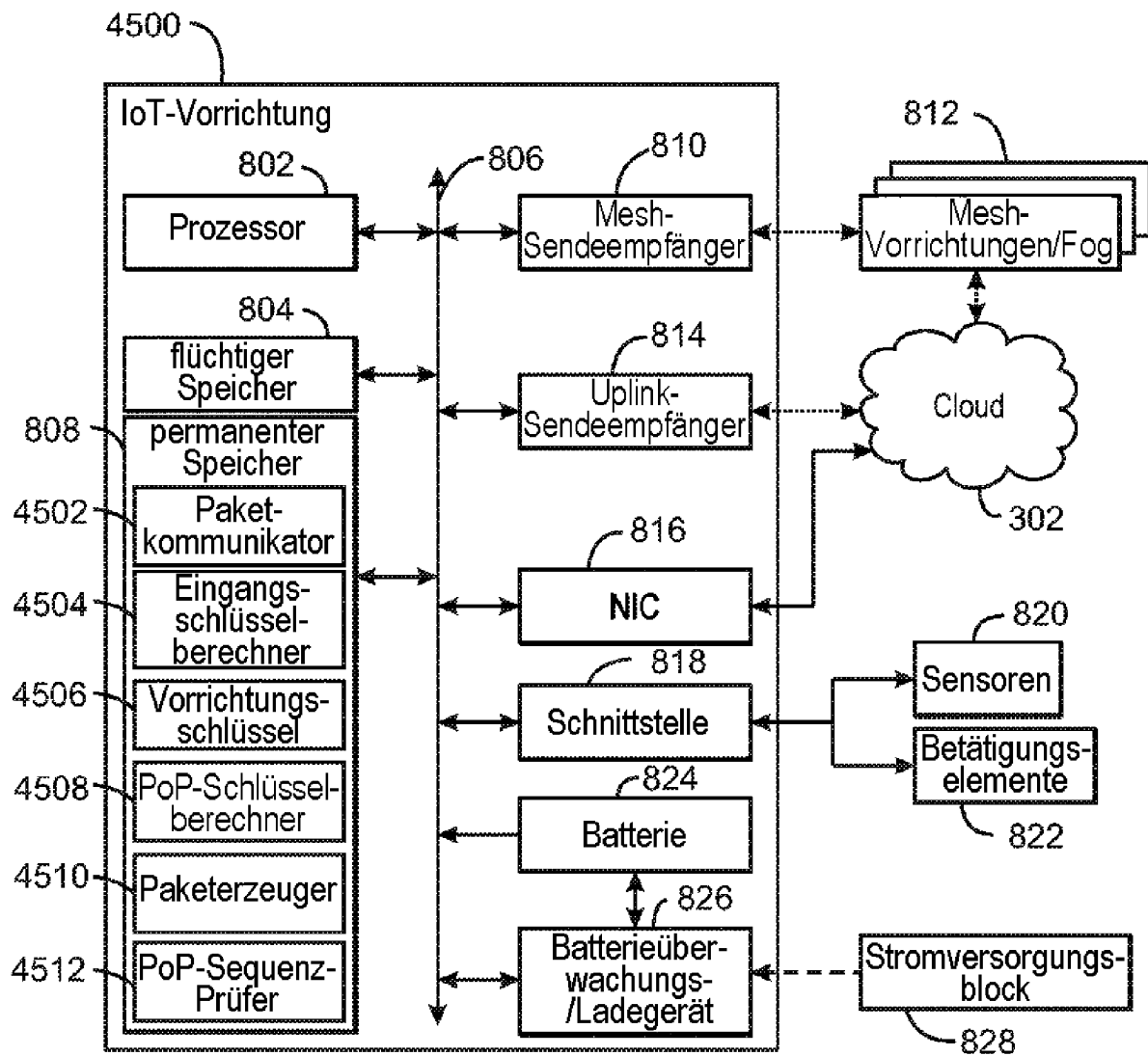


FIG. 45

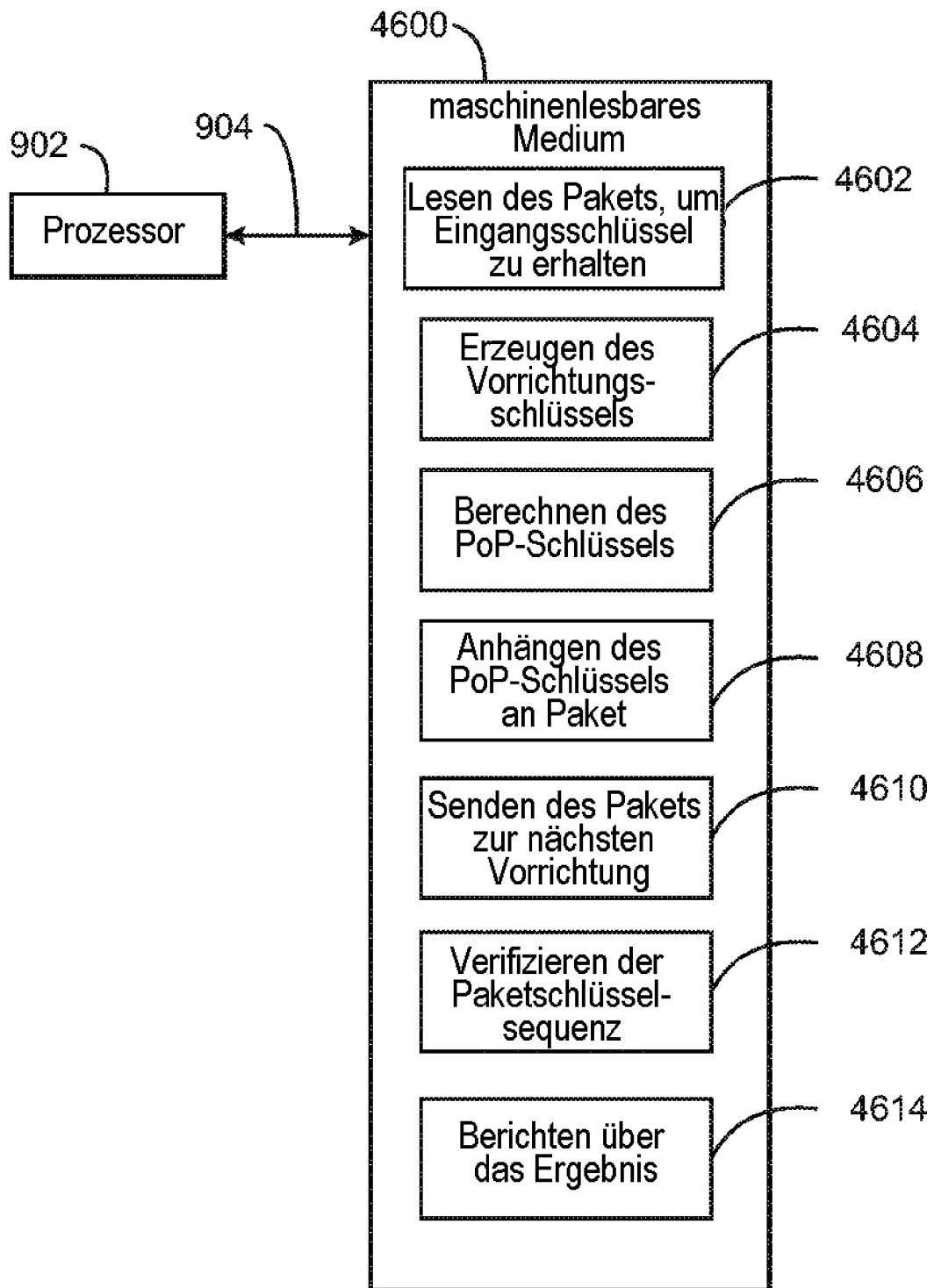
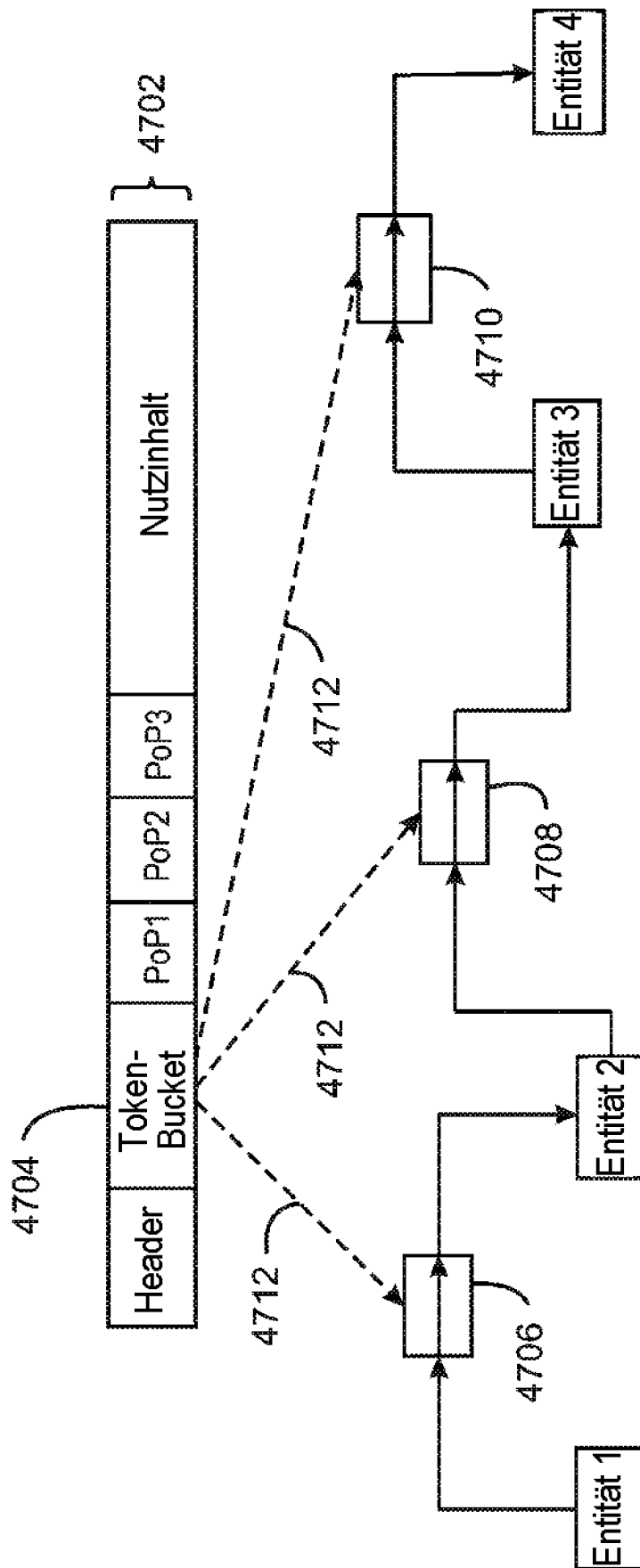
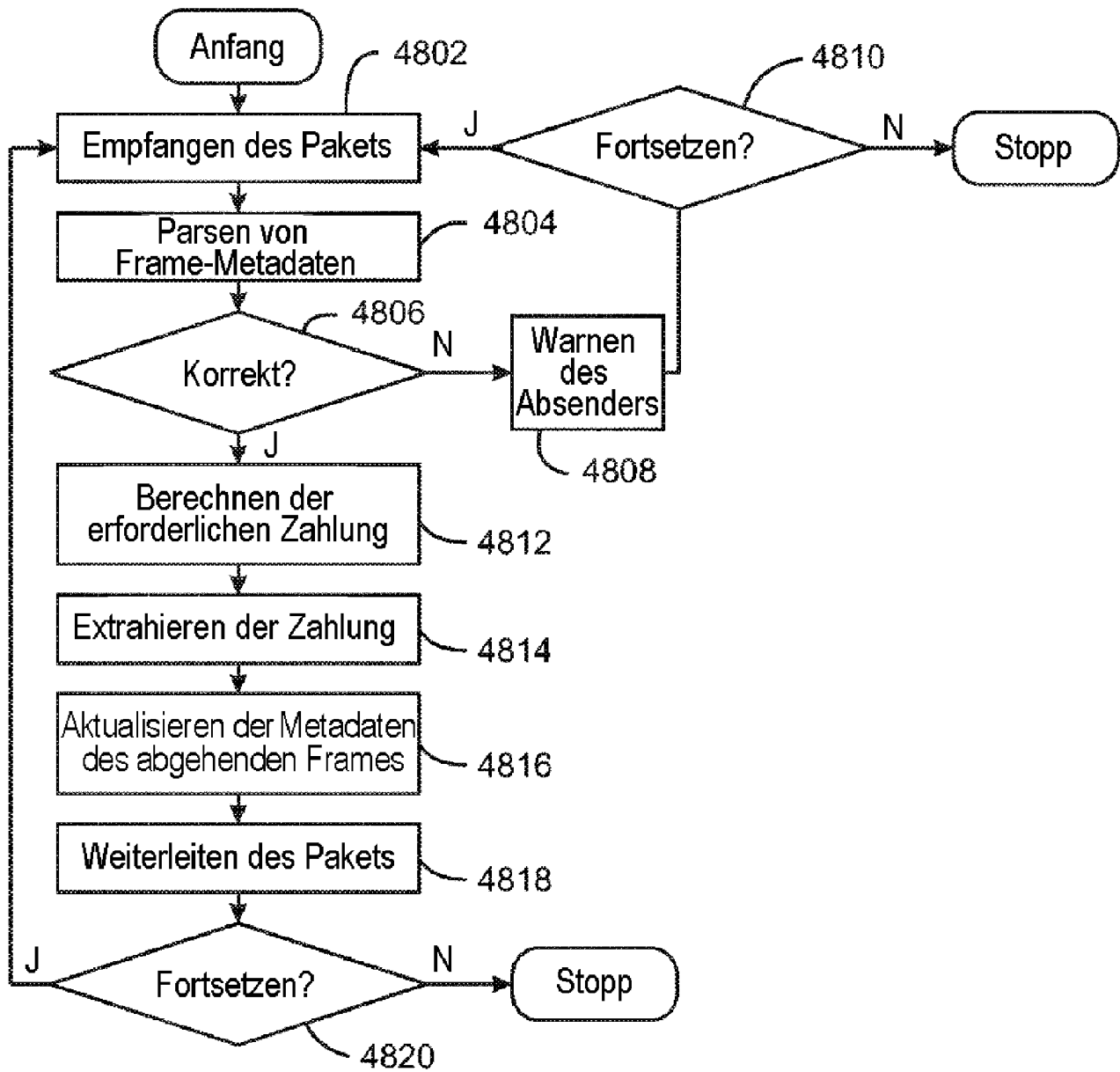


FIG. 46



4700
FIG. 47



4800
FIG. 48

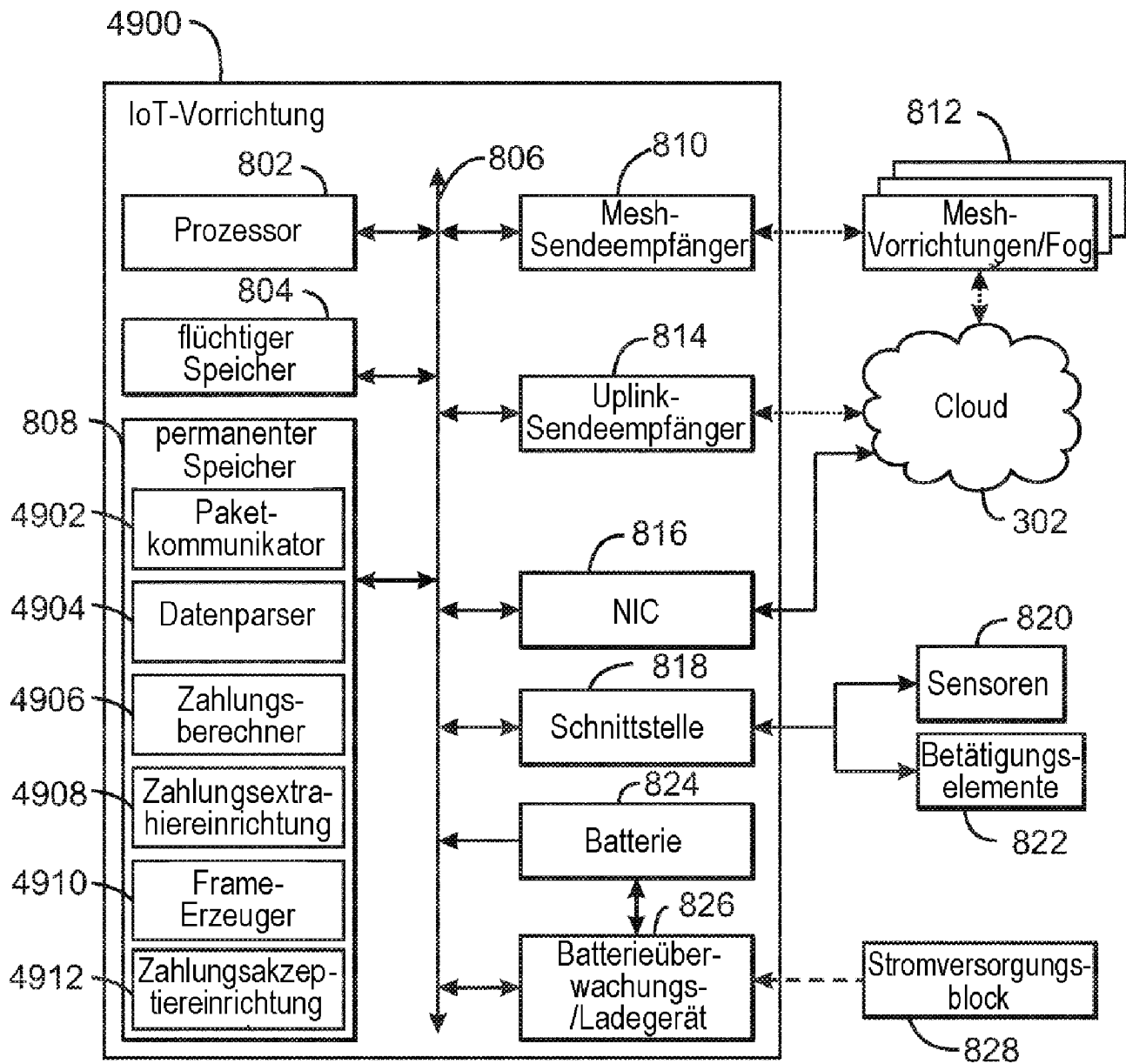


FIG. 49

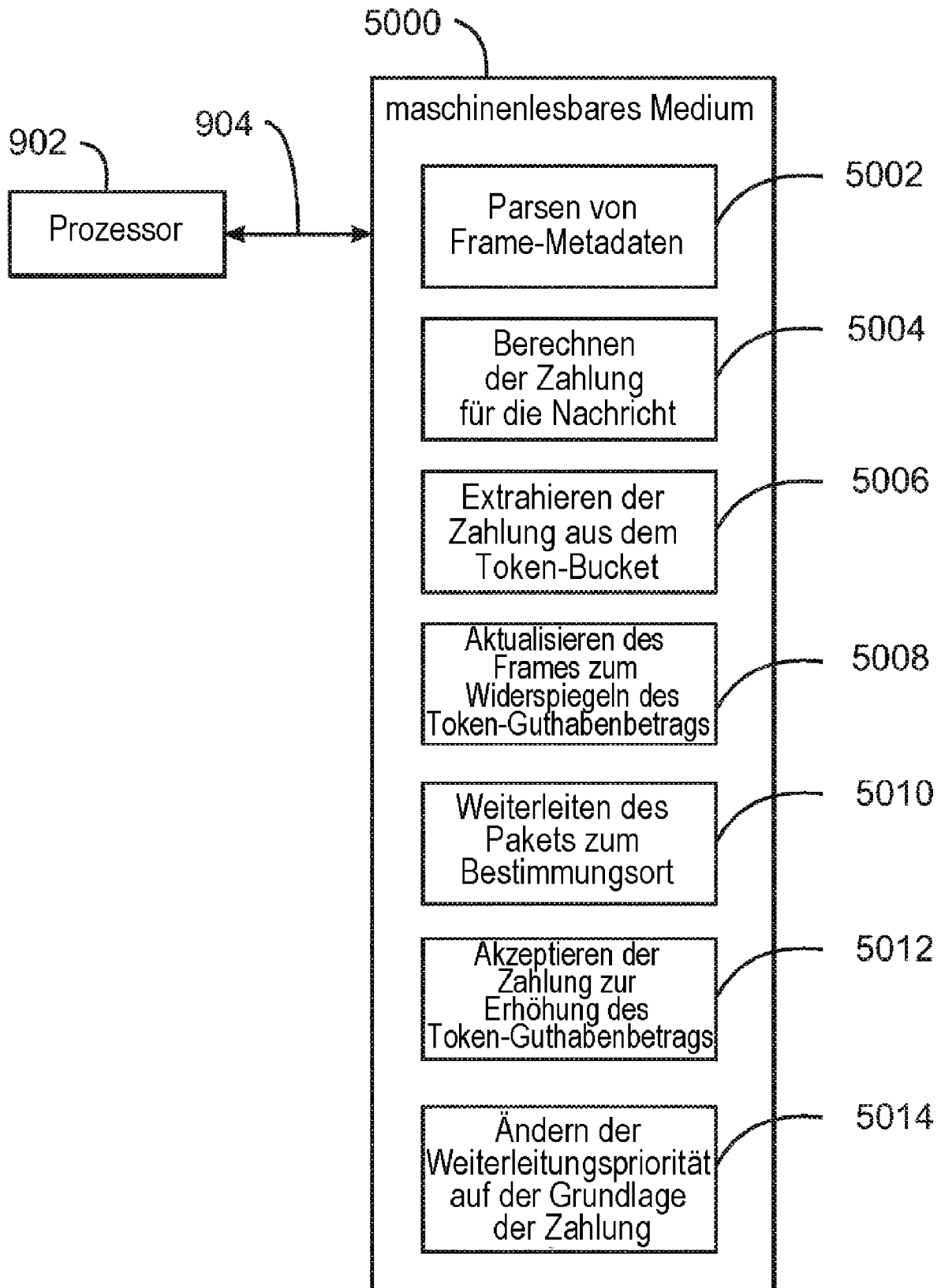
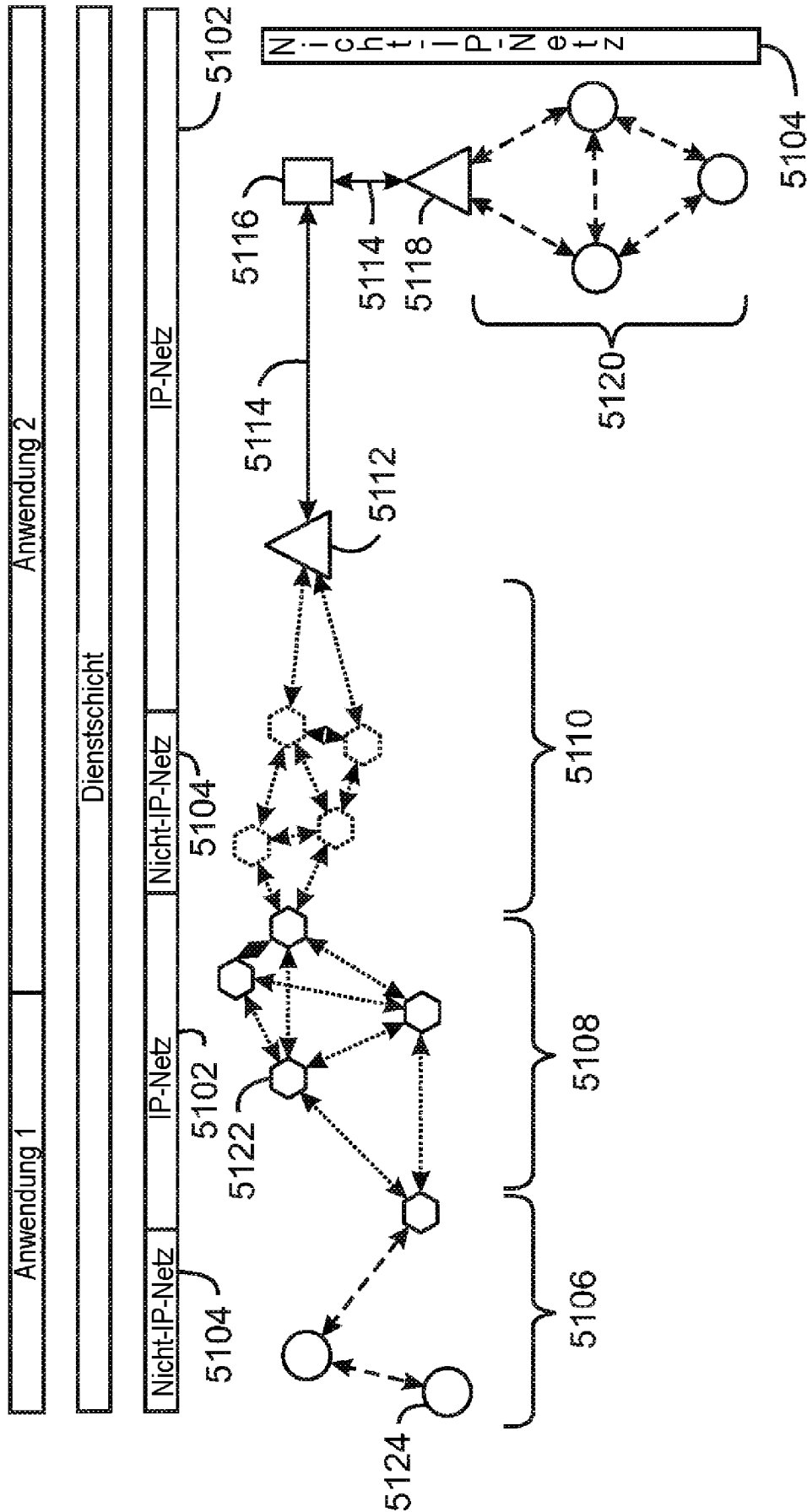
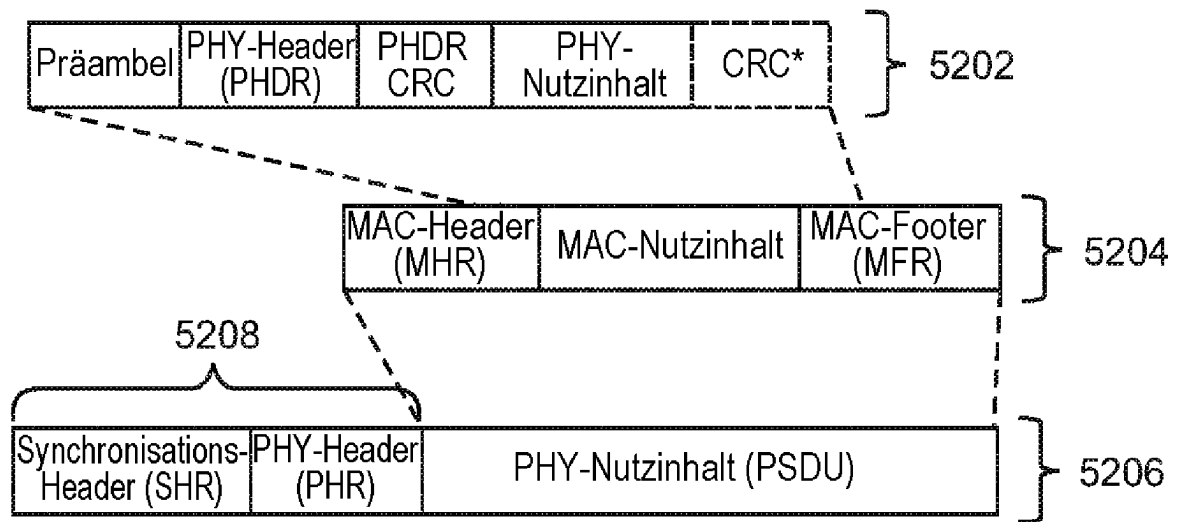


FIG. 50

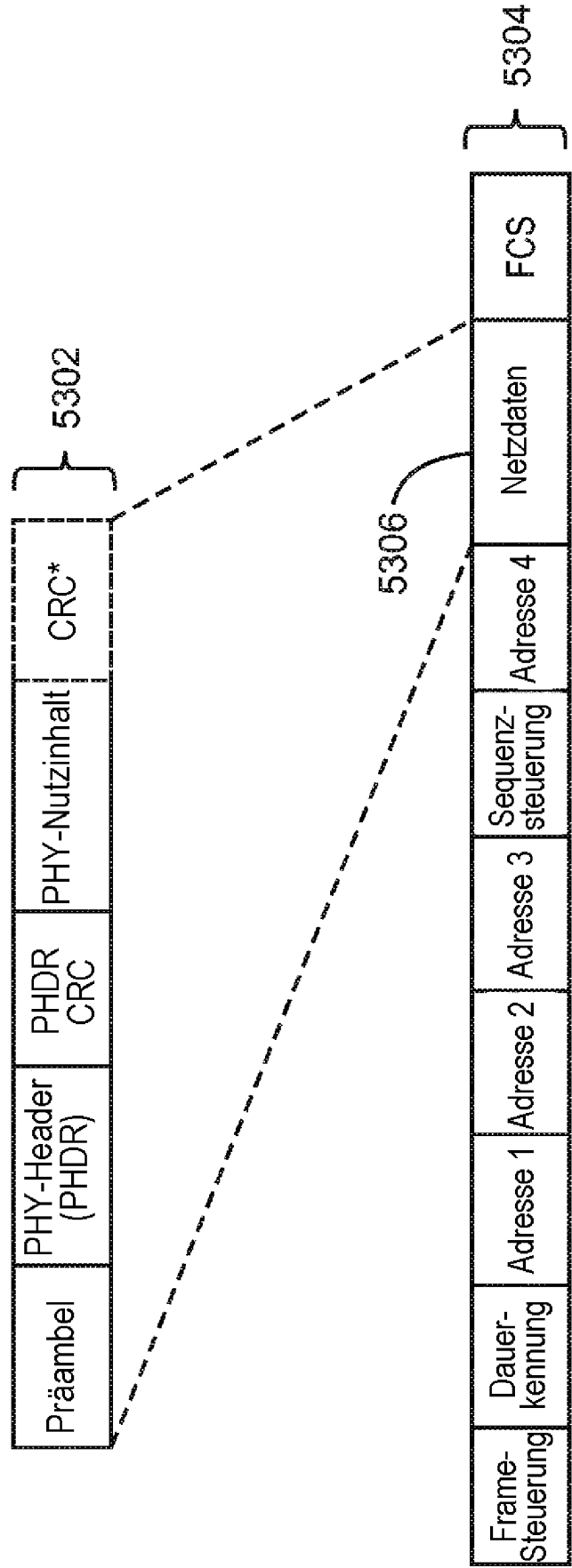


5100

FIG. 51

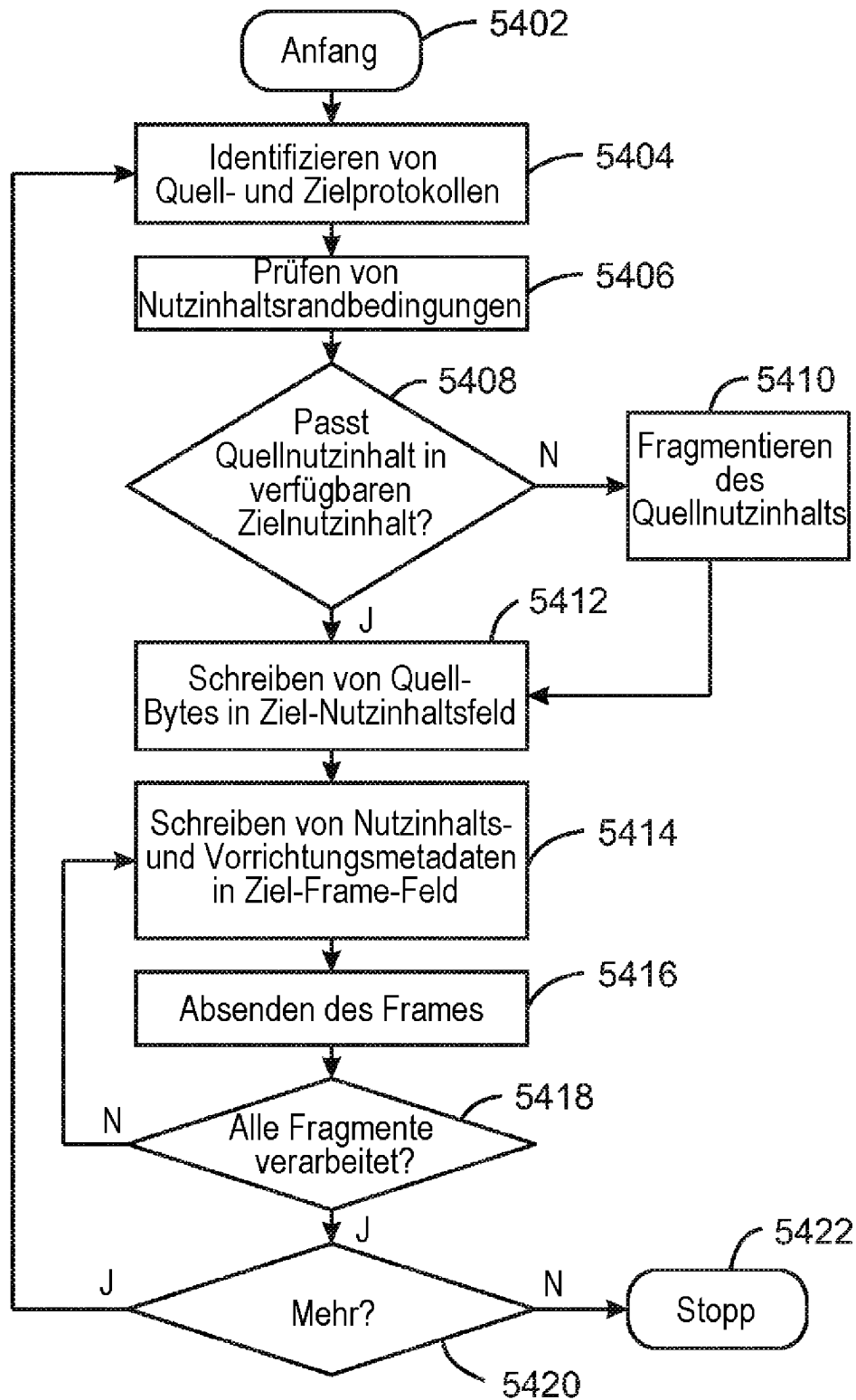


5200
FIG. 52



5300

FIG. 53



5400
FIG. 54

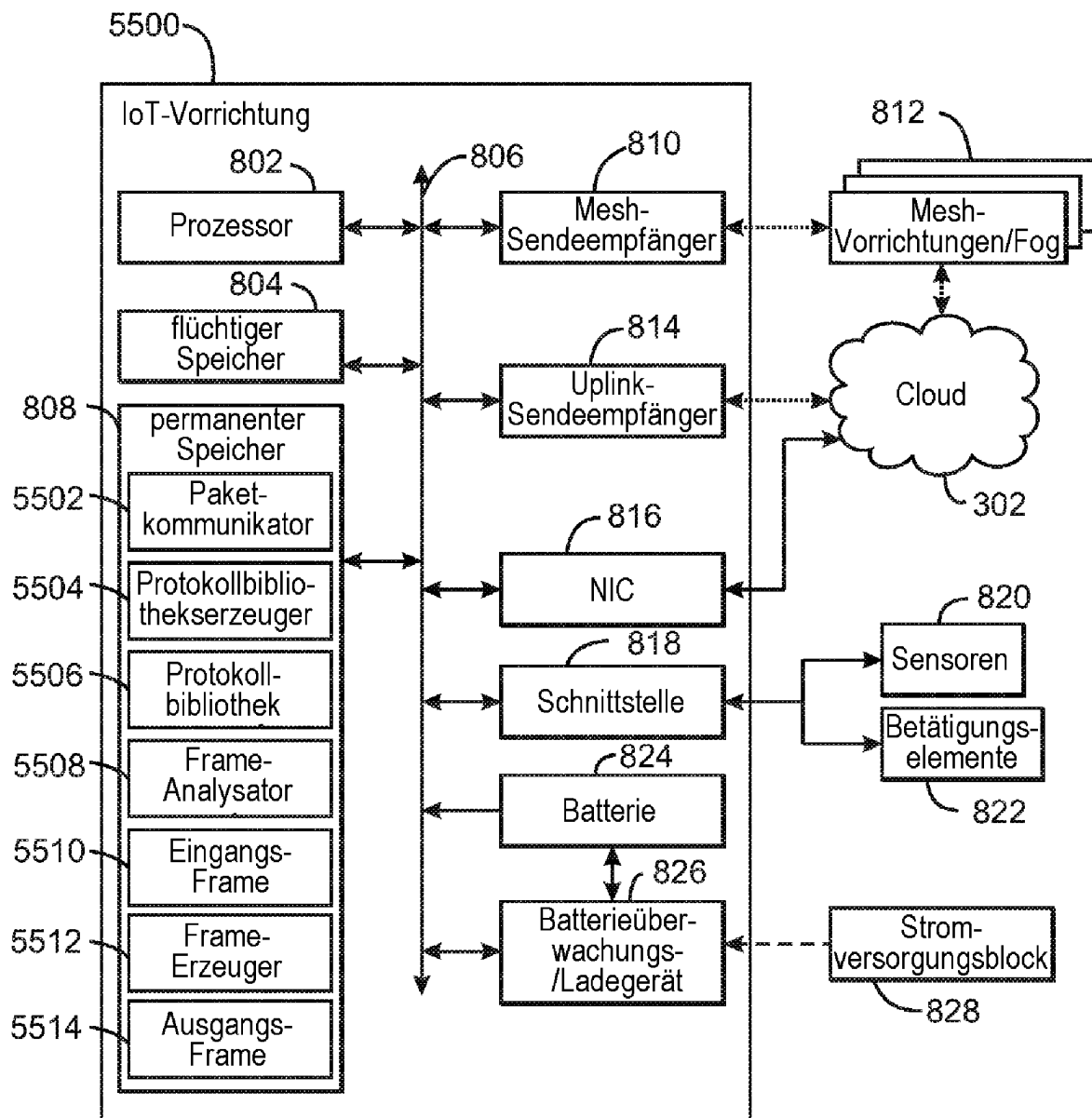


FIG. 55

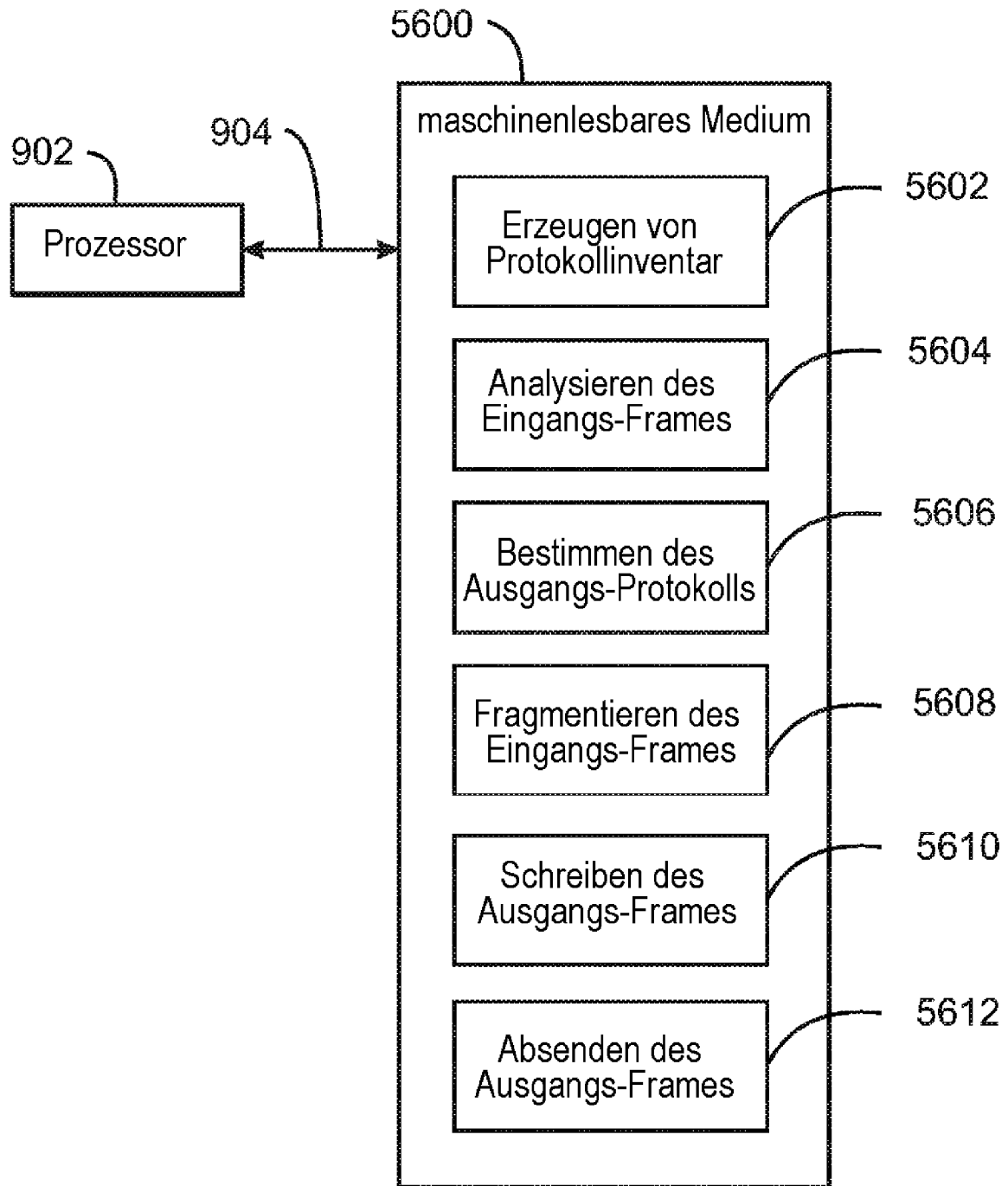


FIG. 56

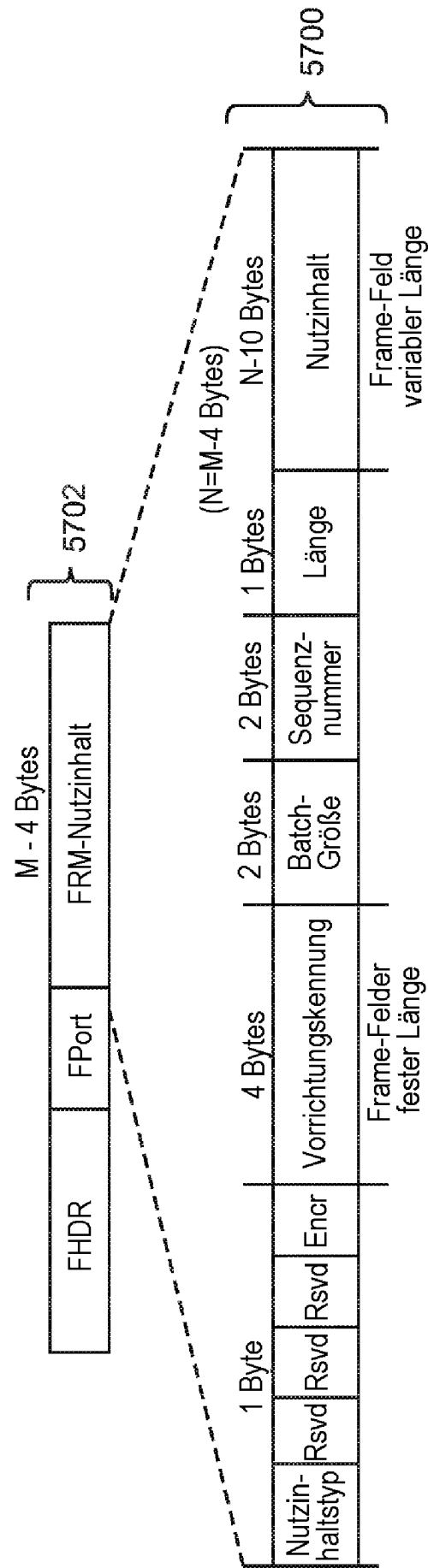
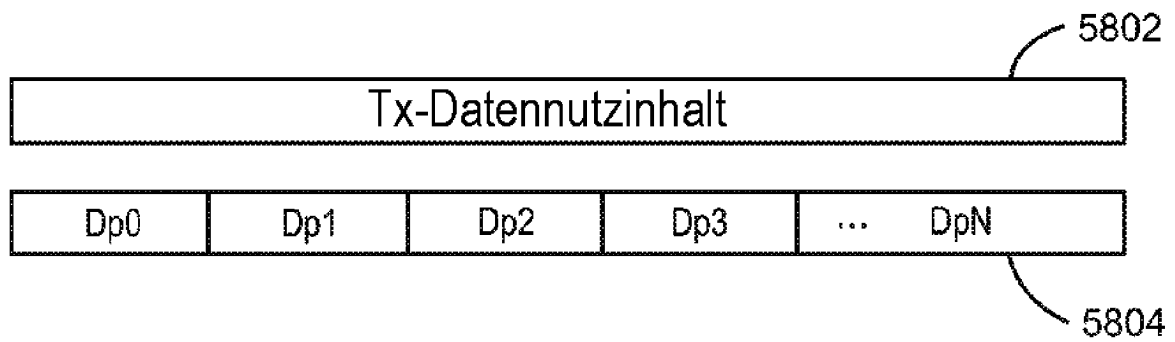
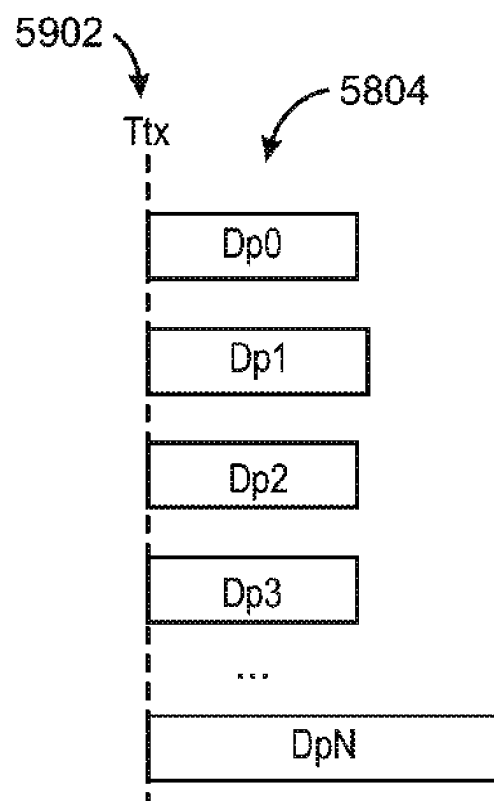


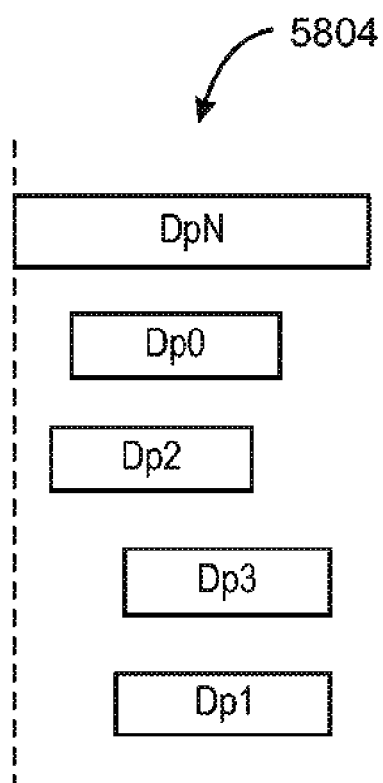
FIG. 57



5800
FIG. 58

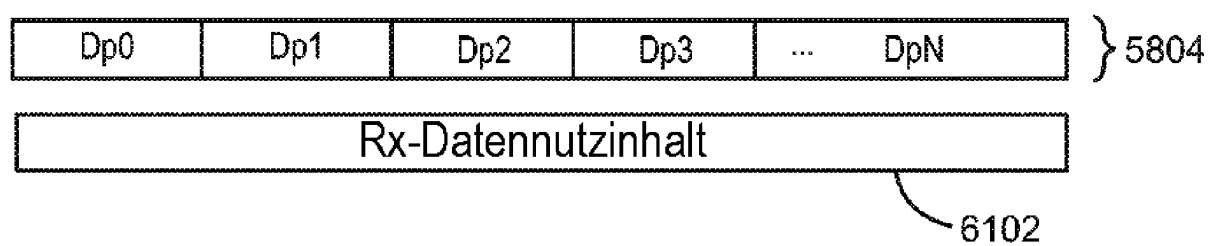


5900
FIG. 59

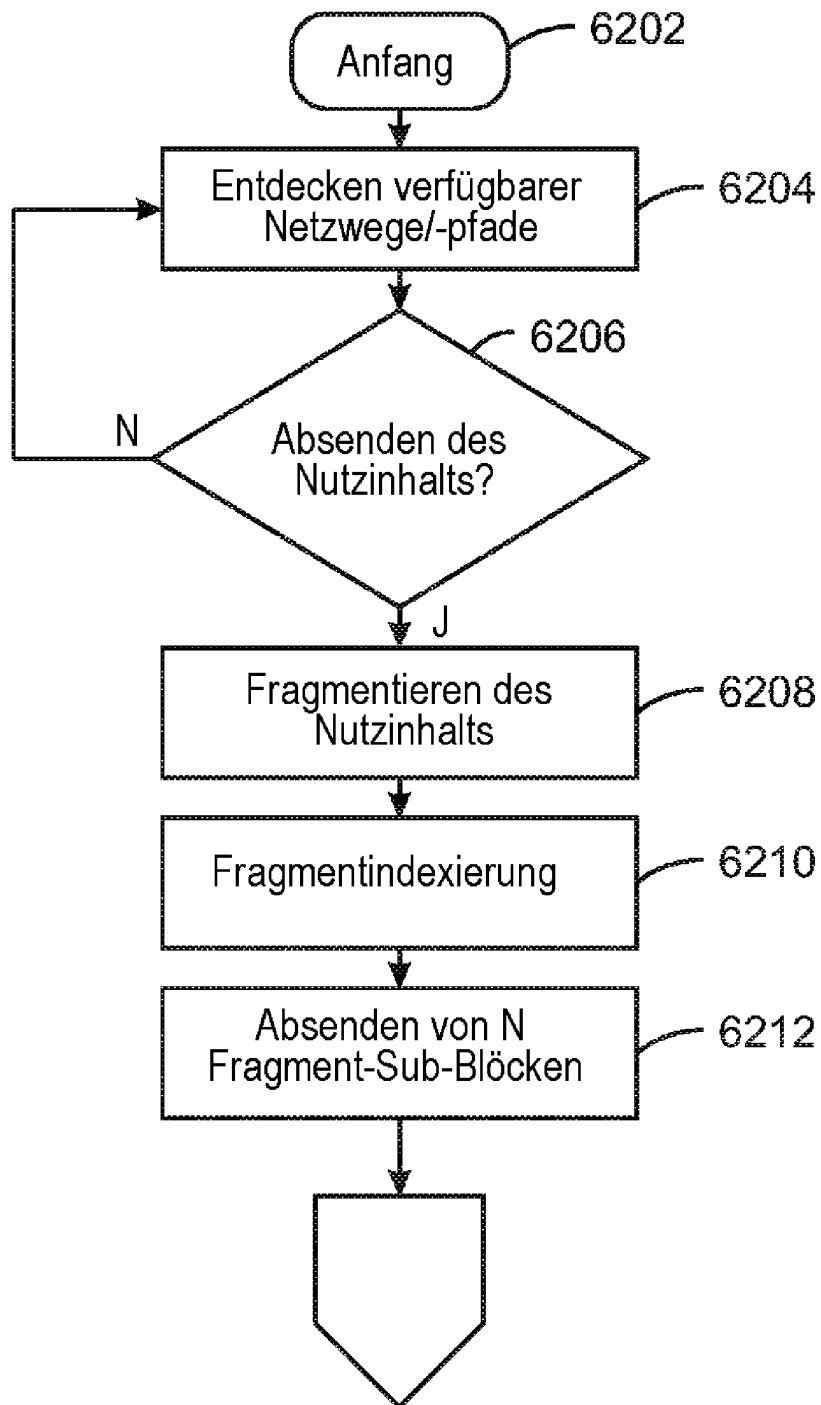


6000

FIG. 60

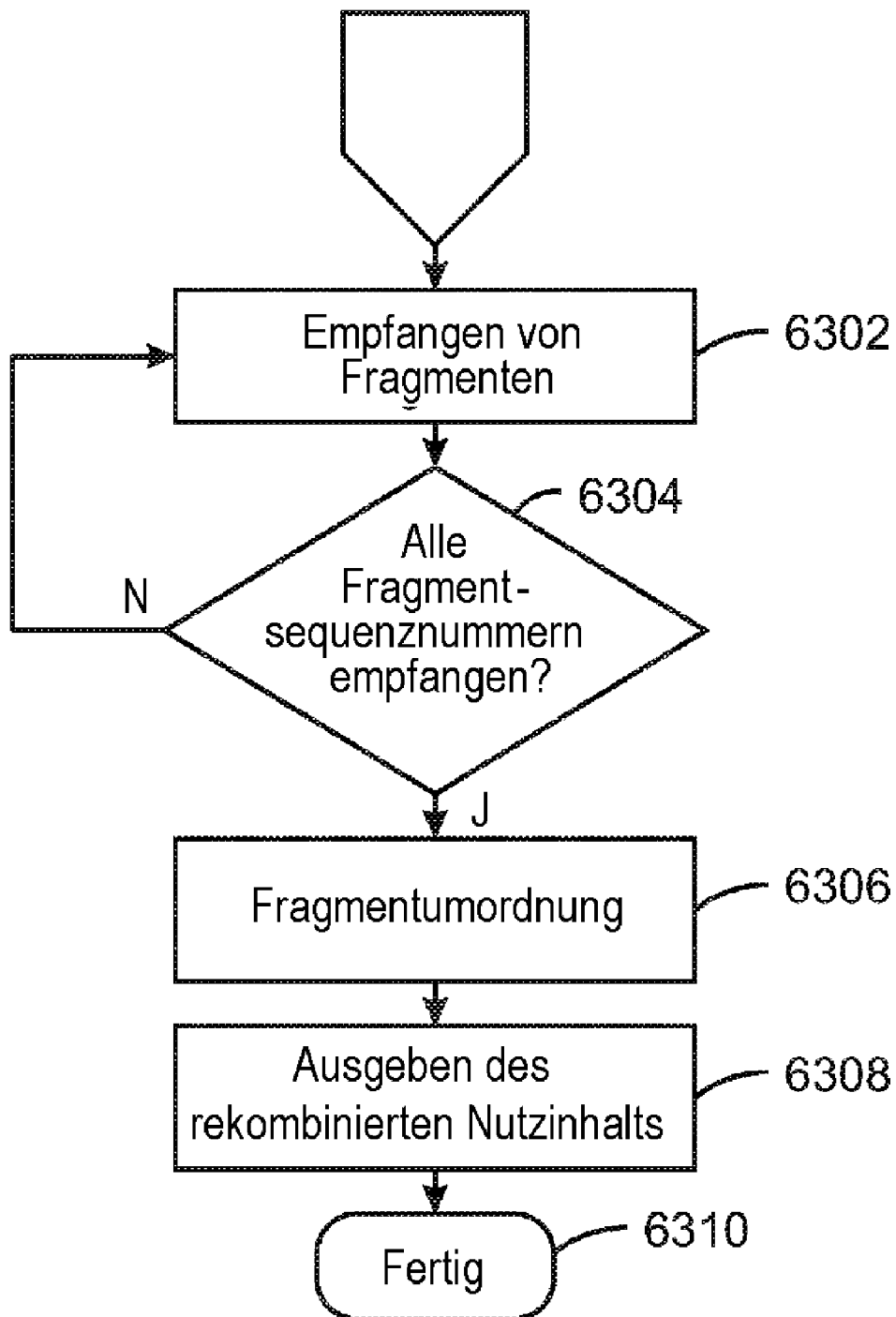


6100
FIG. 61



6200

FIG. 62



6300

FIG. 63

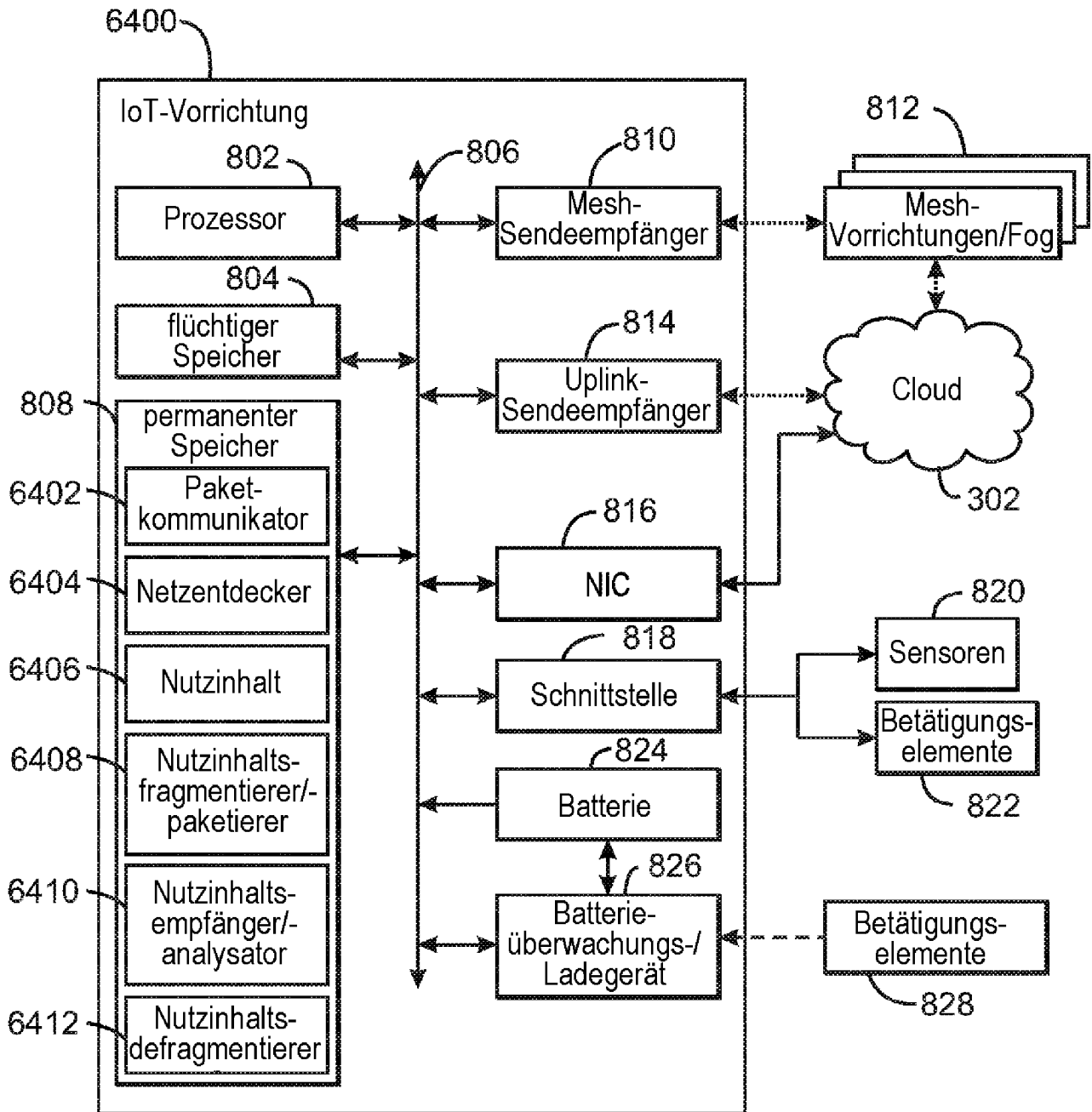


FIG. 64

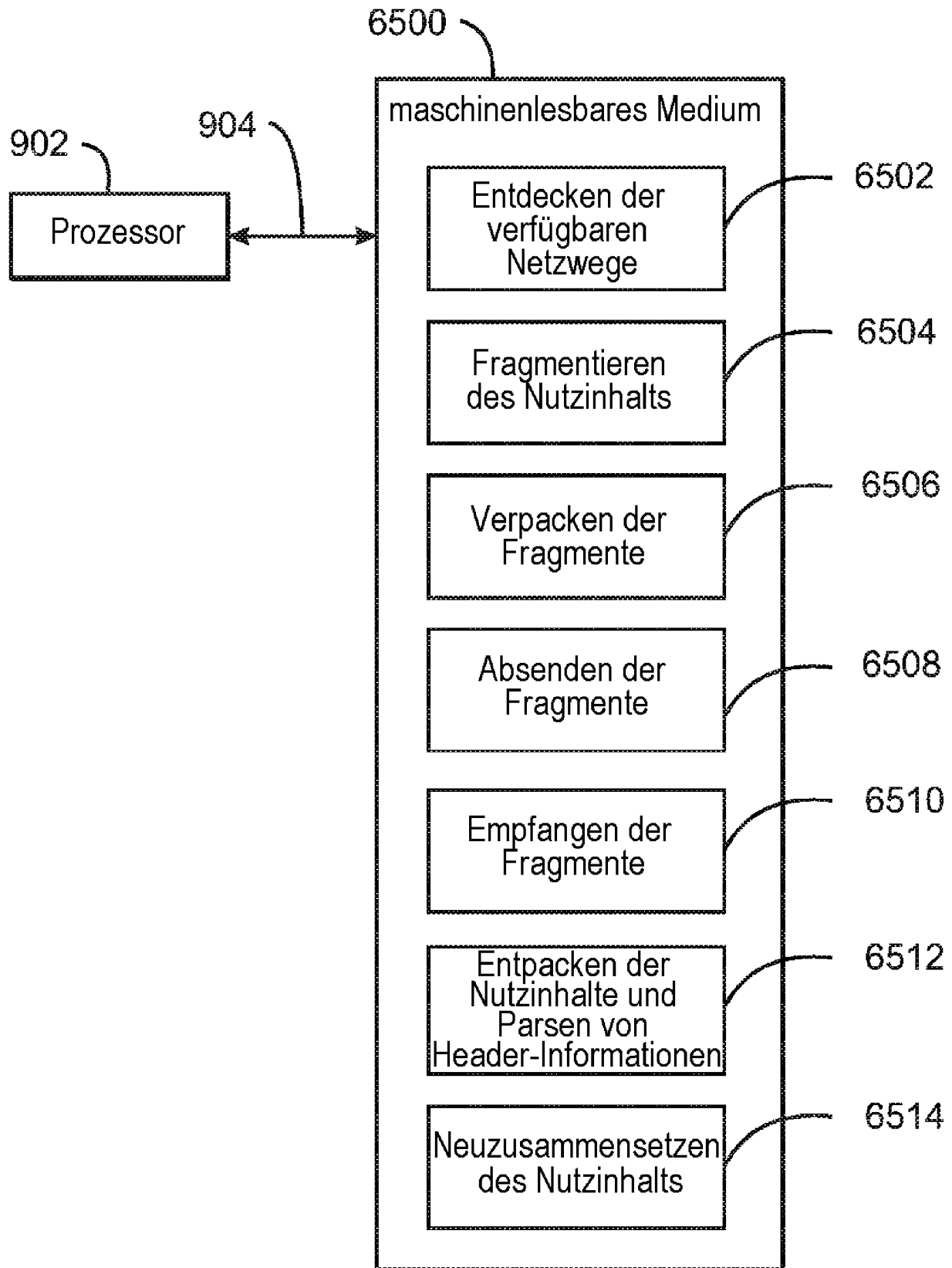
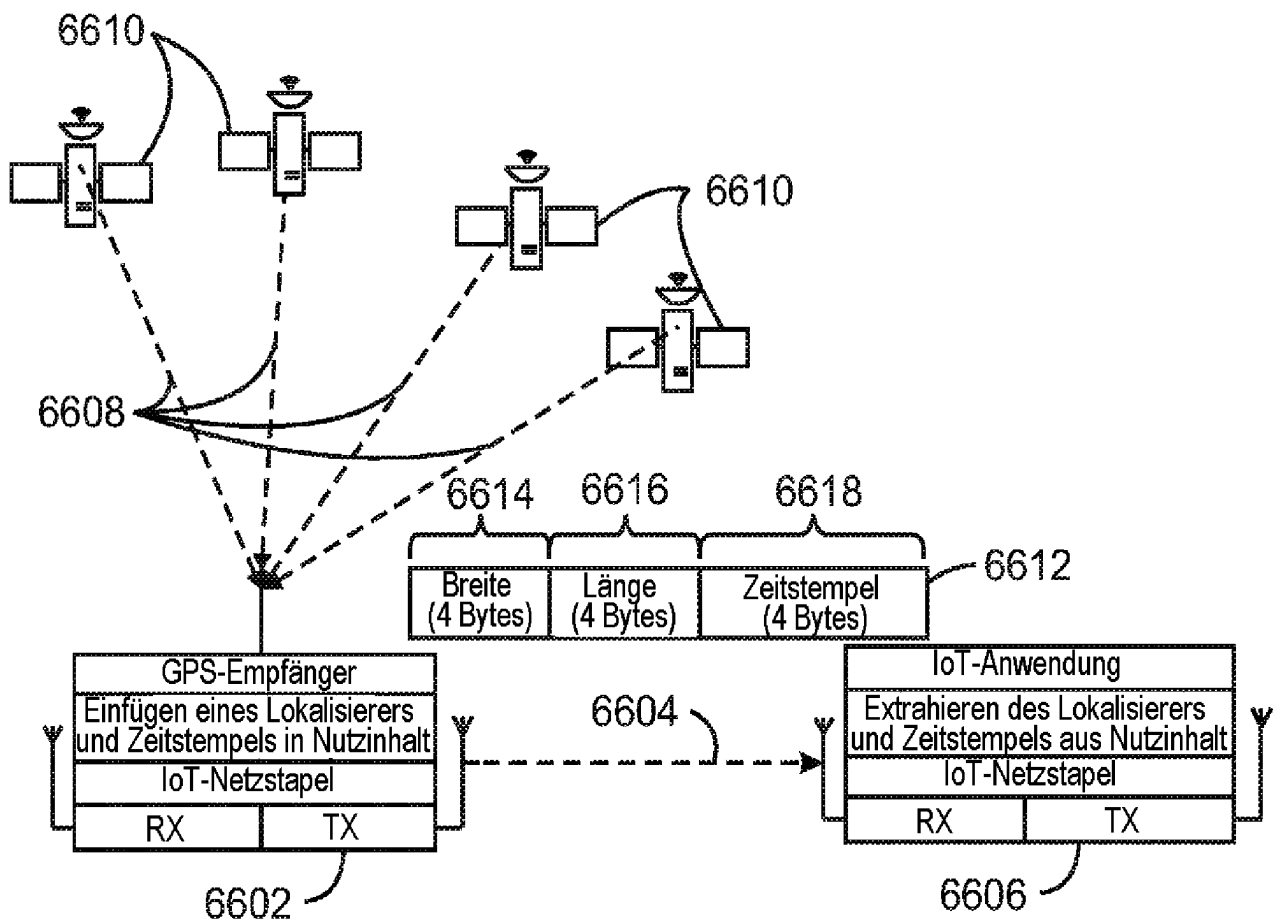
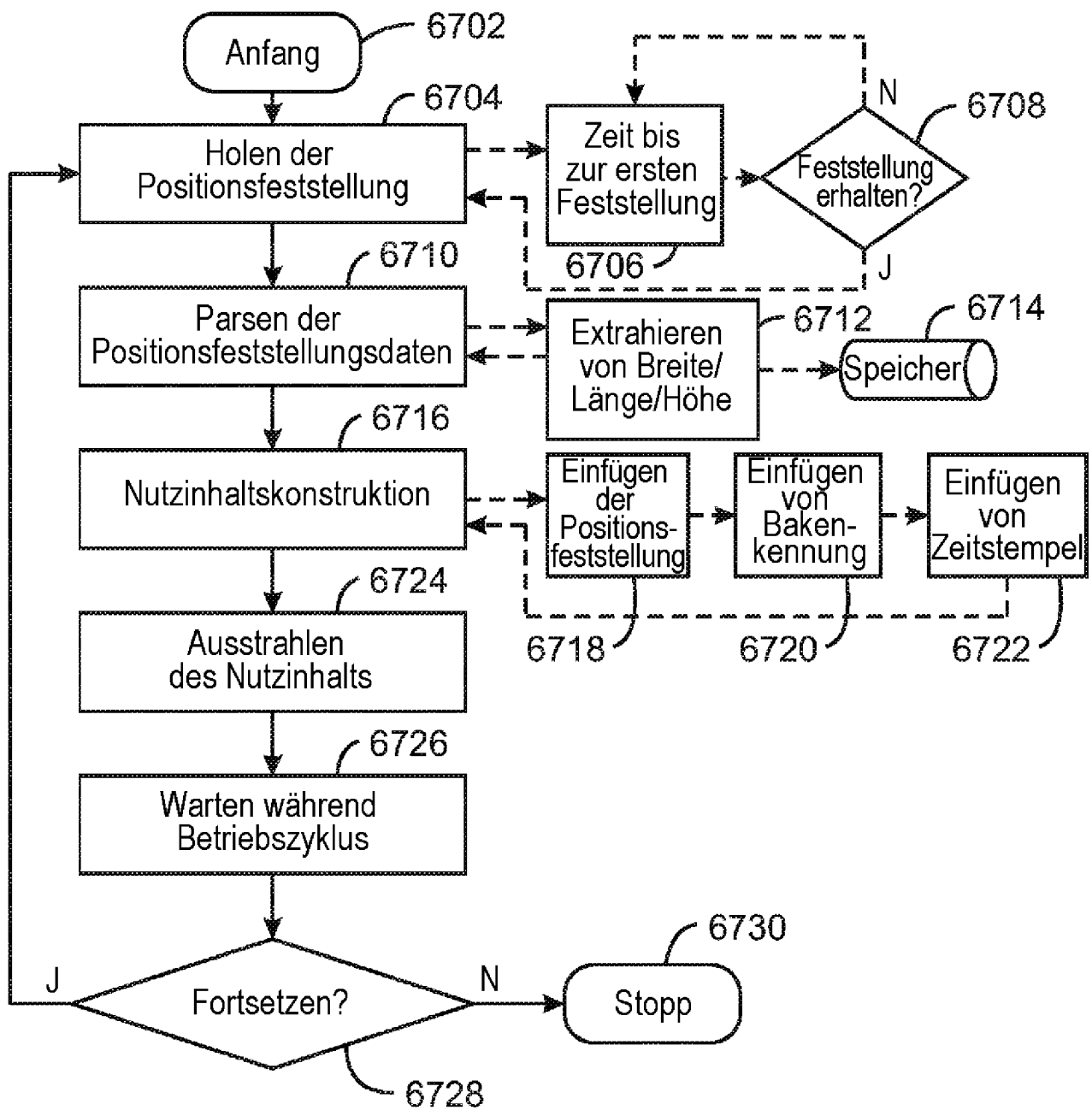


FIG. 65



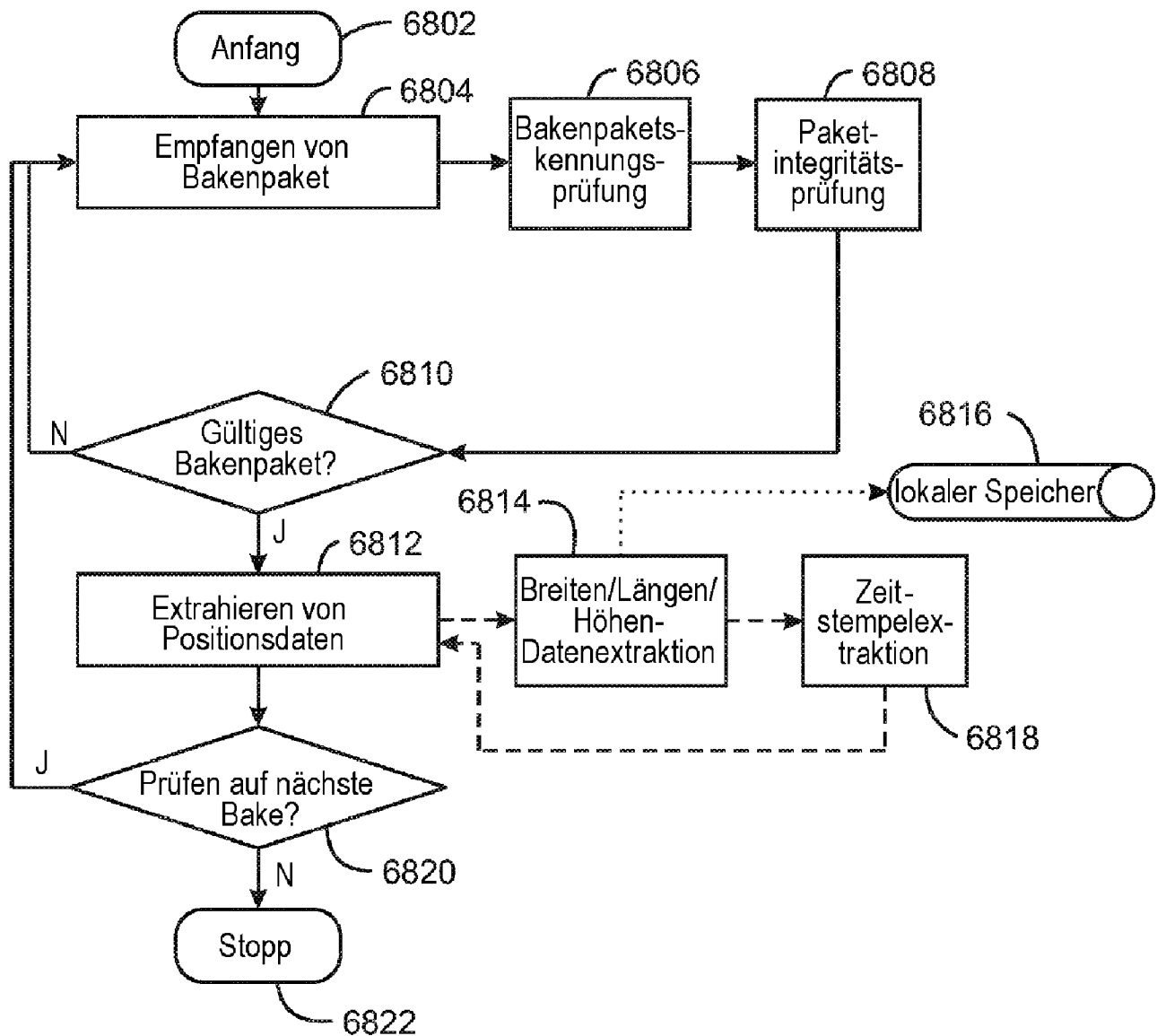
6600

FIG. 66



6700

FIG. 67



6800
FIG. 68

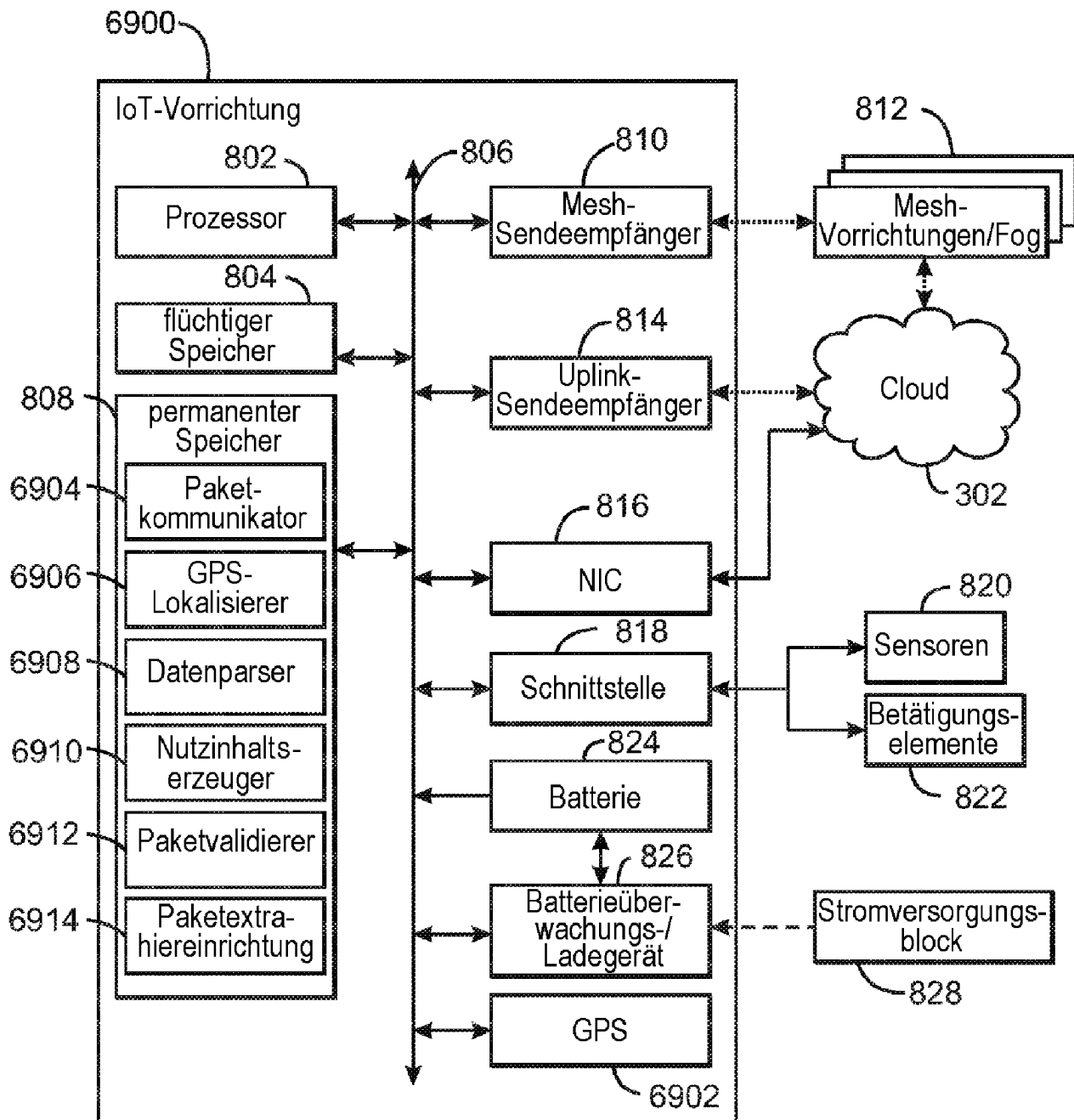


FIG. 69

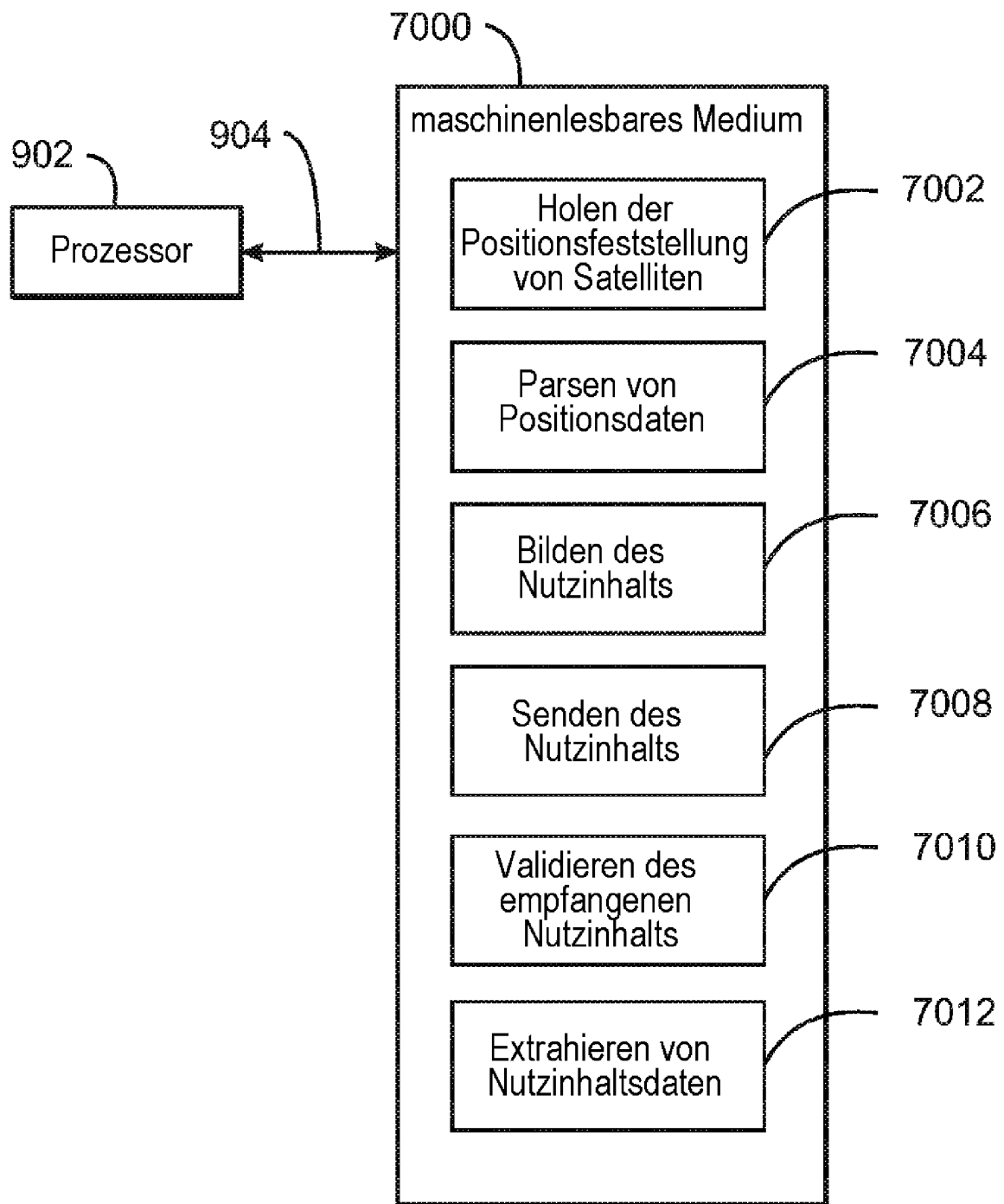
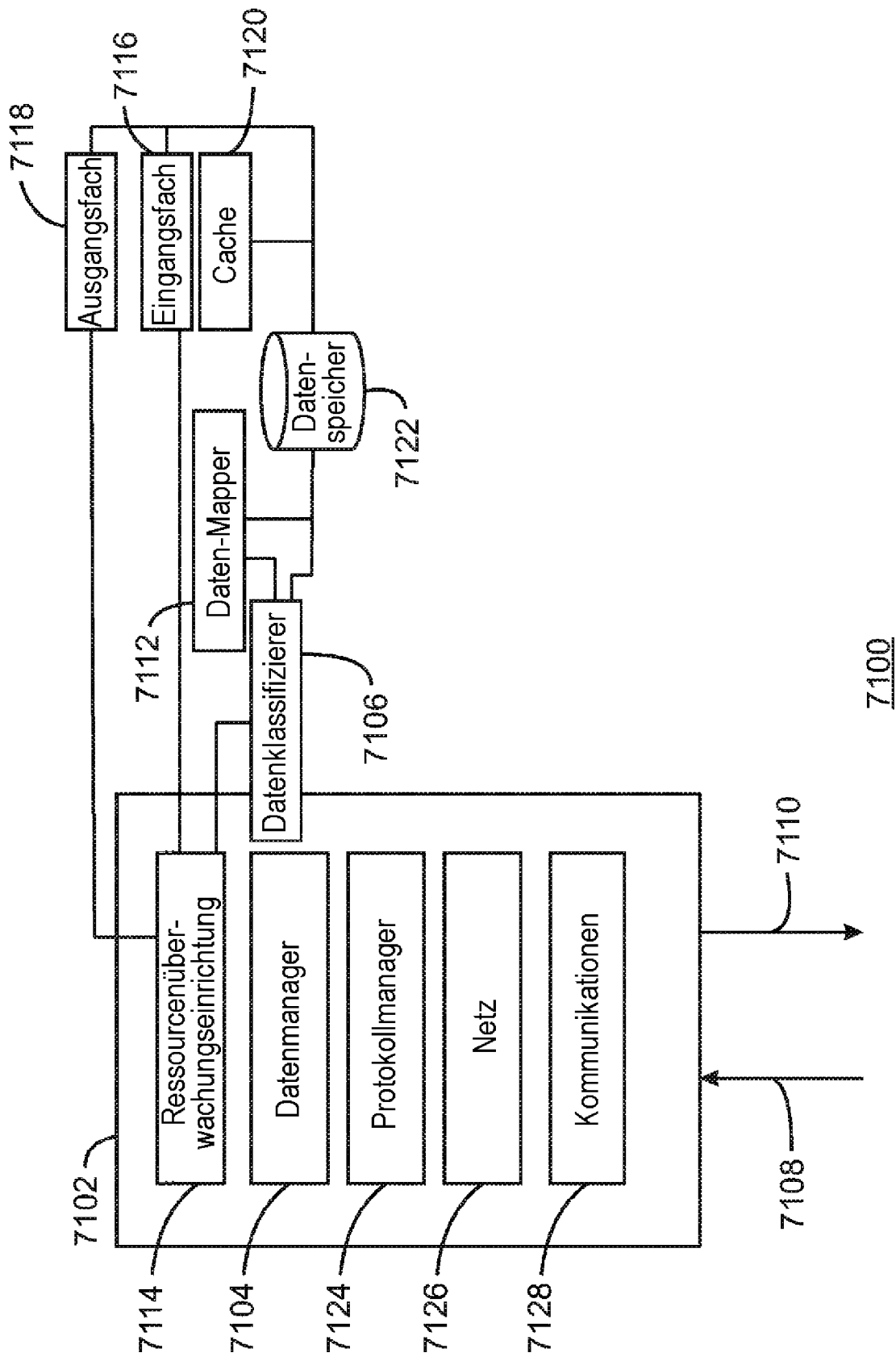
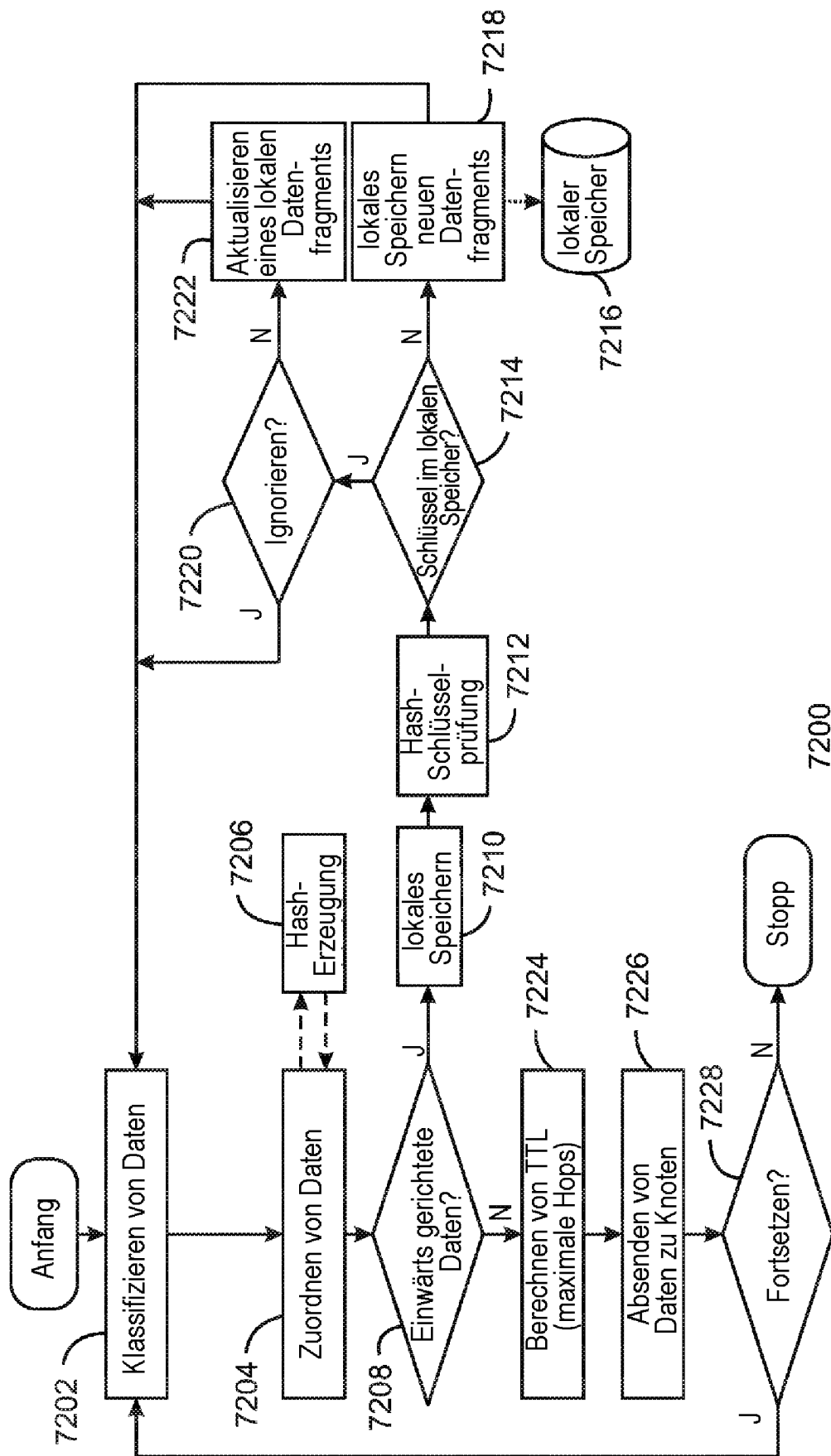


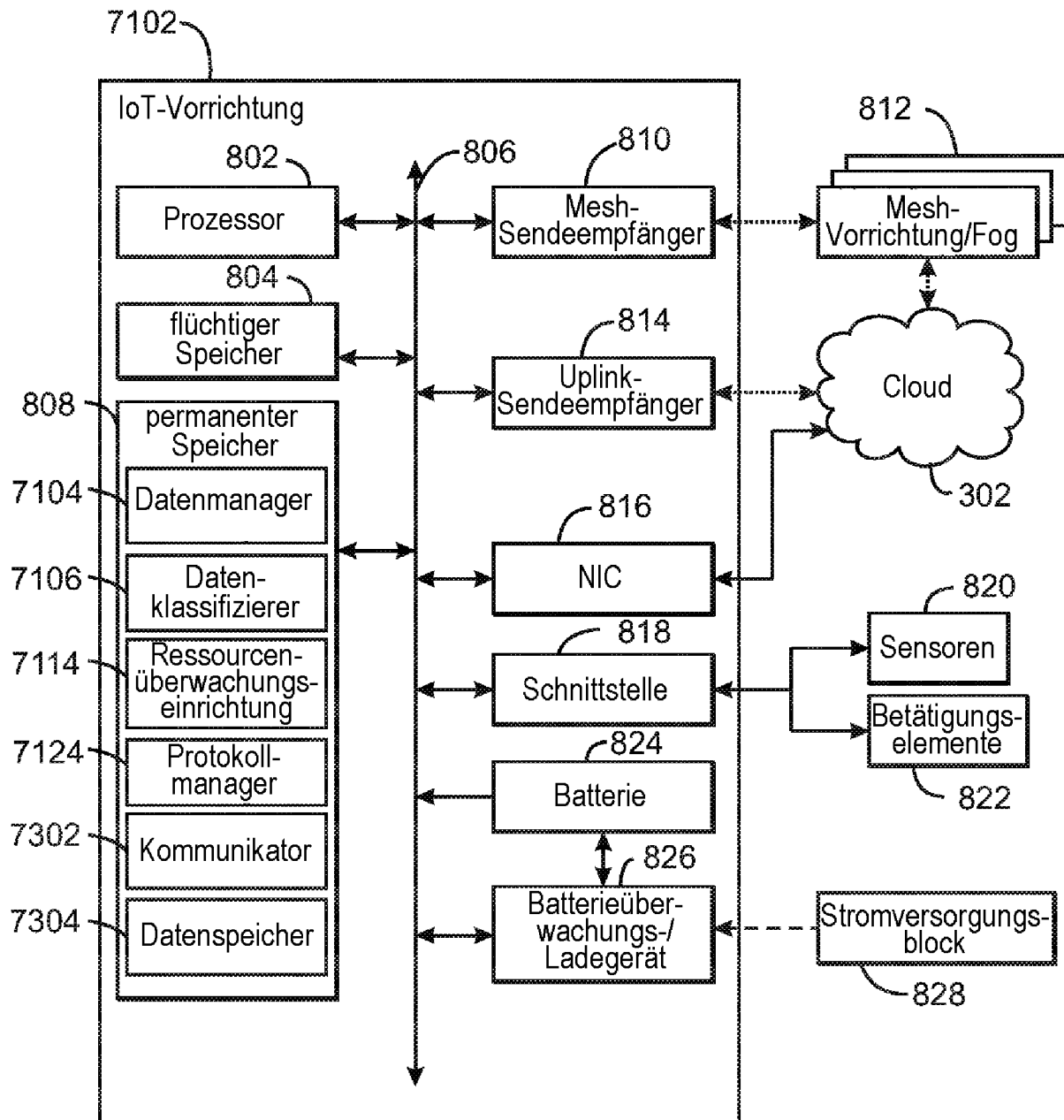
FIG. 70



7100
FIG. 71



7200
FIG. 72



7300

FIG. 73

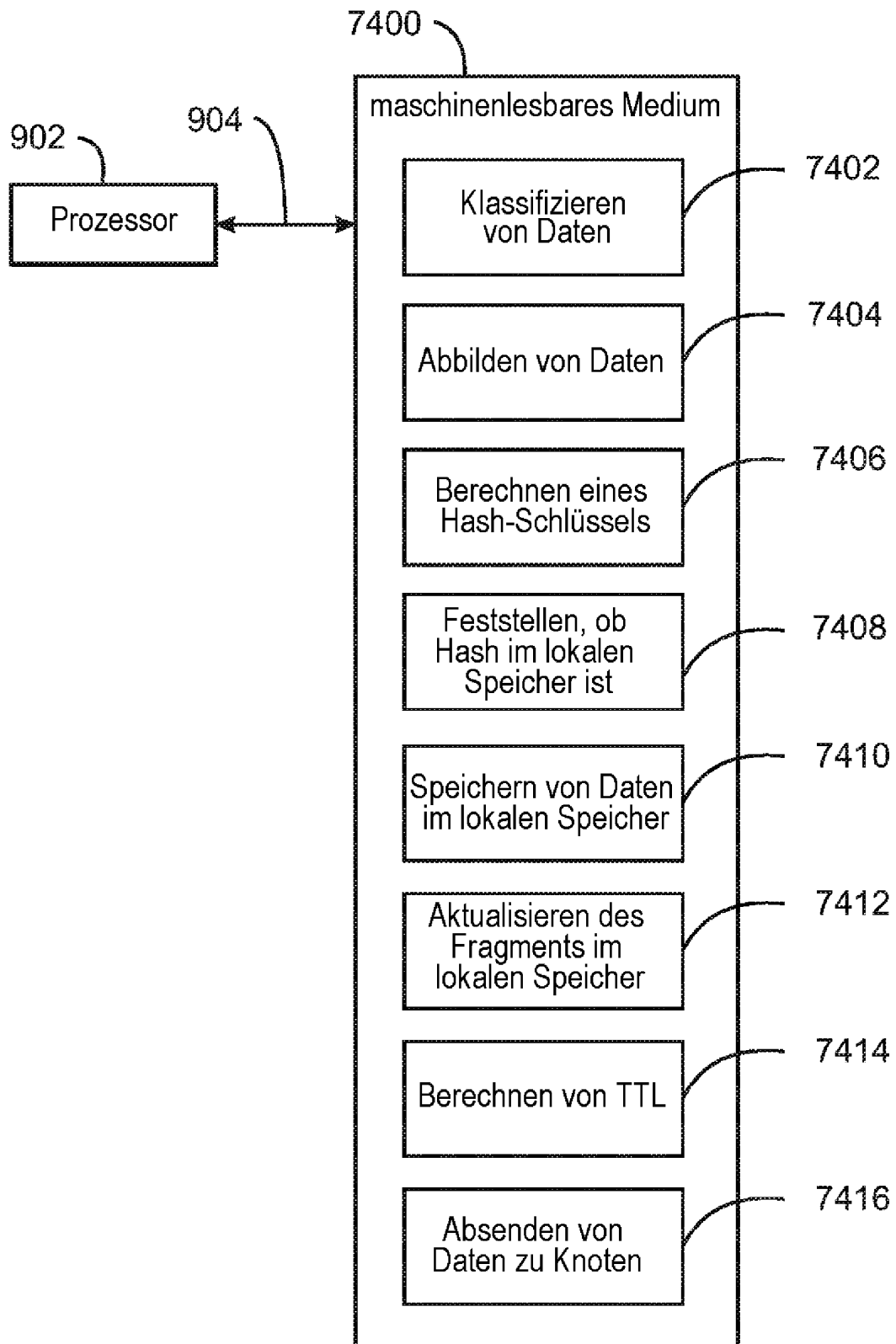
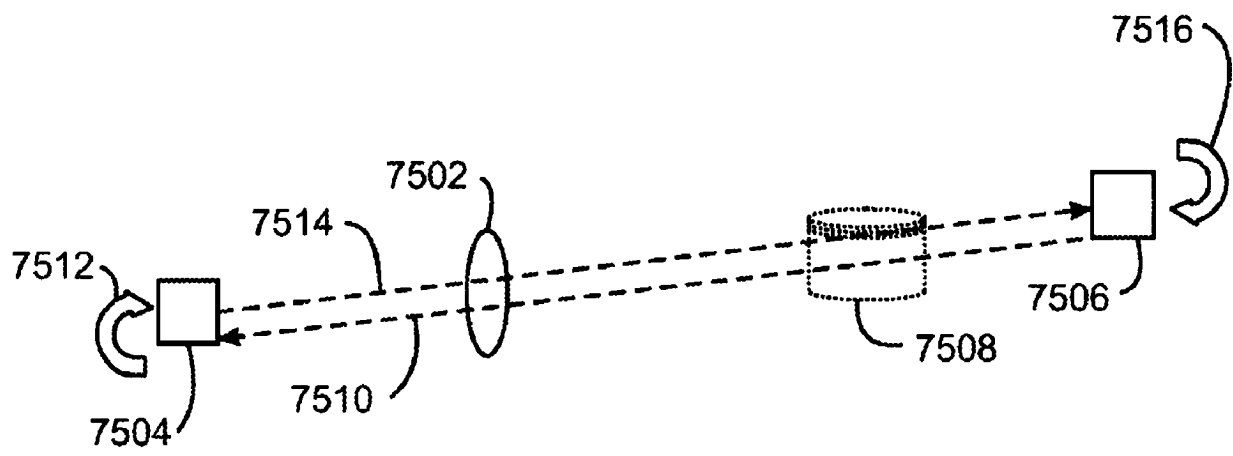
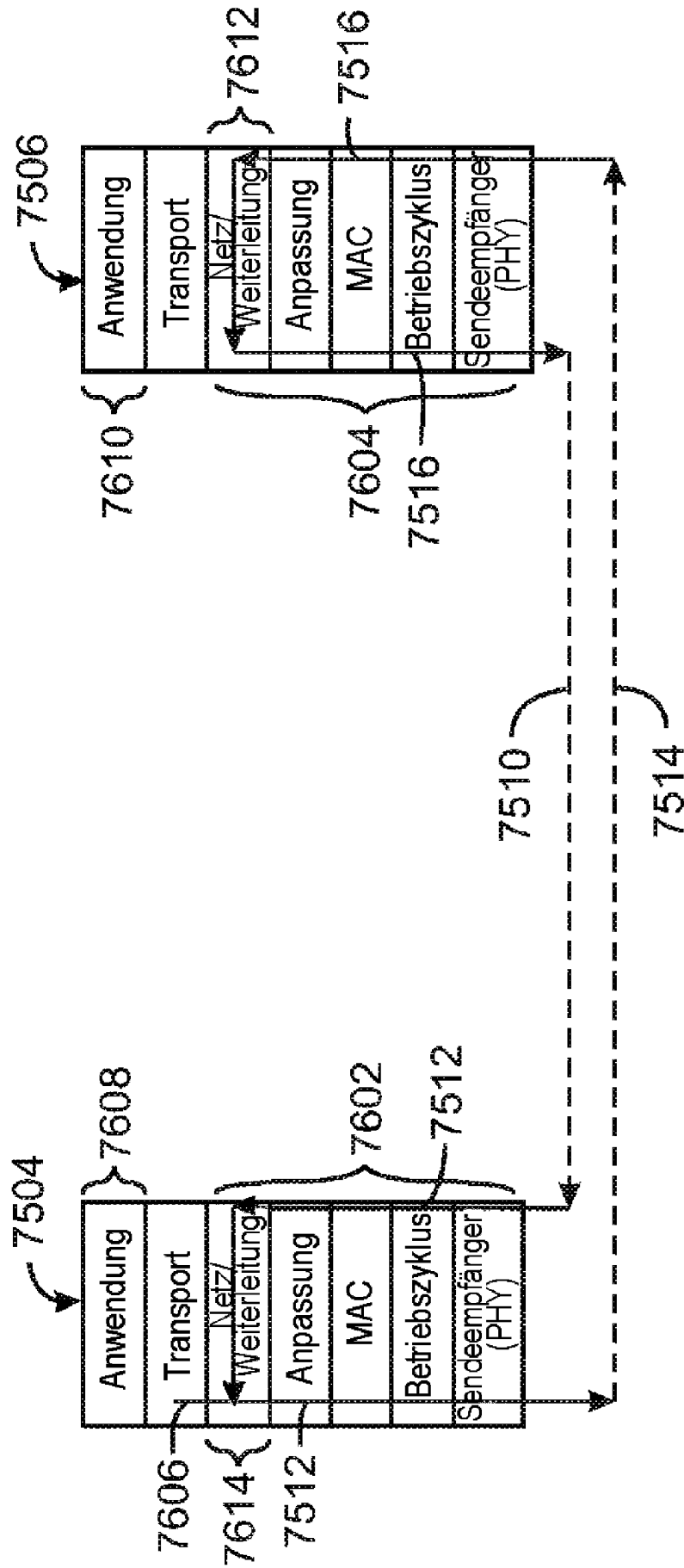


FIG. 74

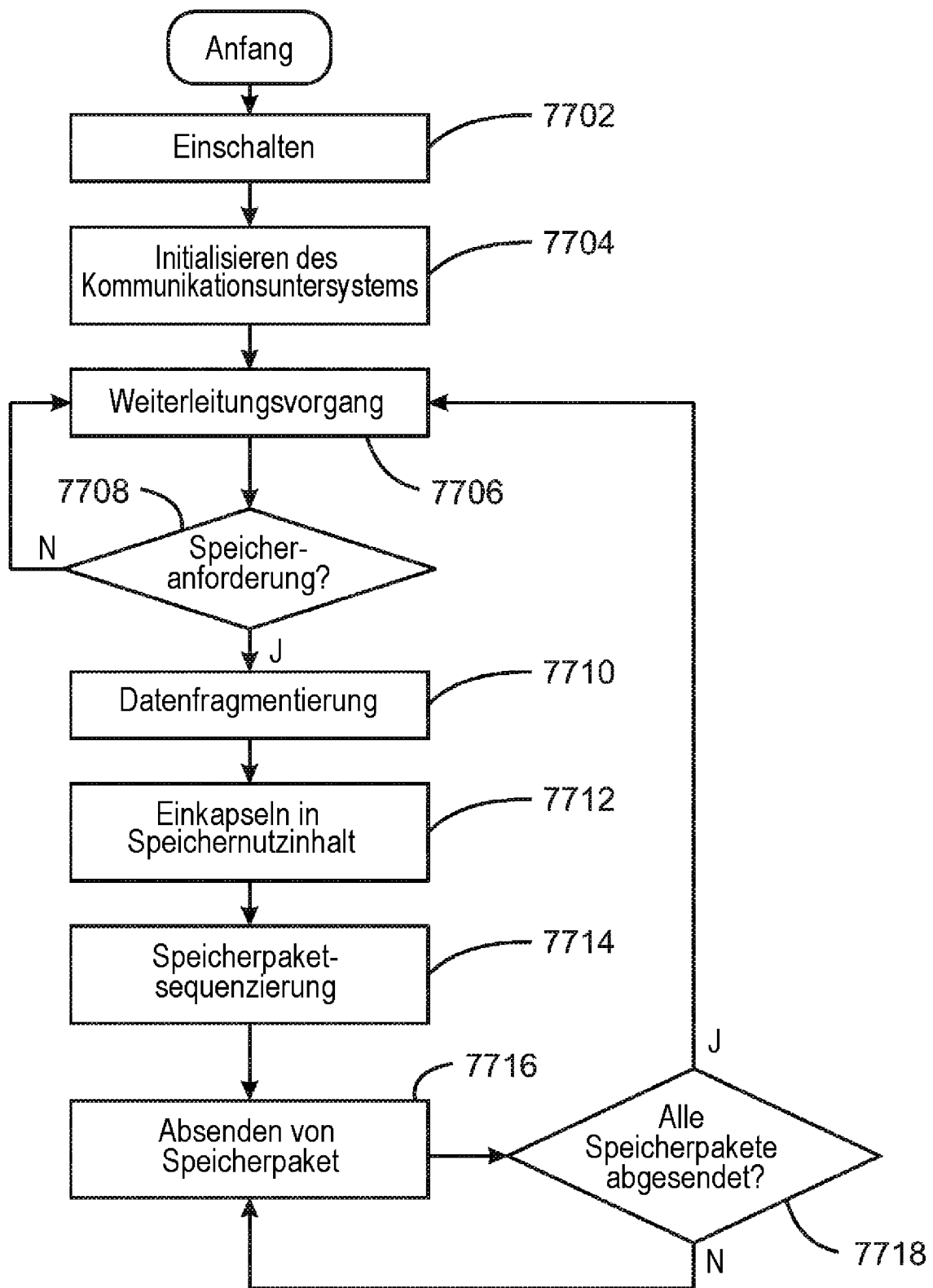


7500
FIG. 75



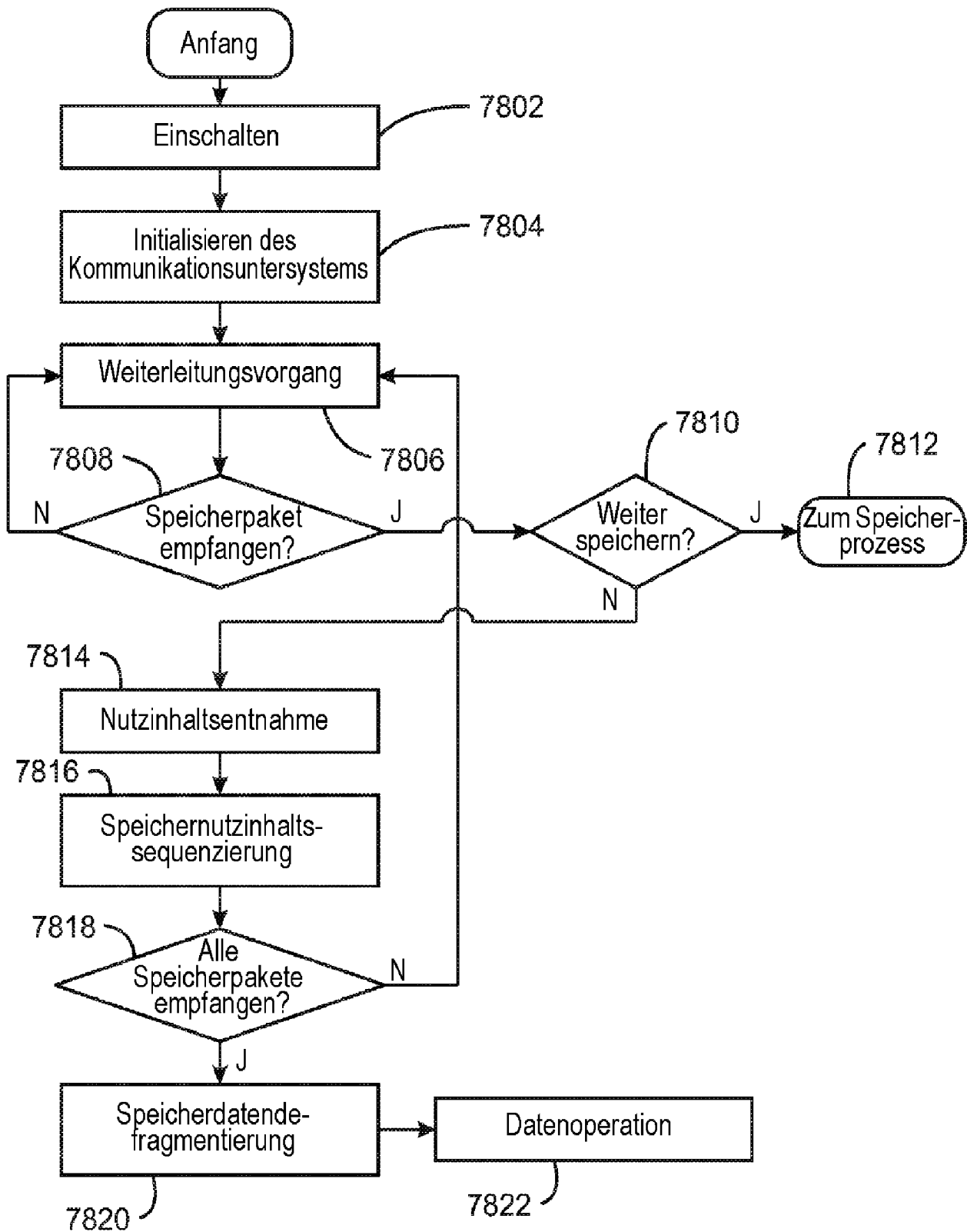
7600

FIG. 76



7700

FIG. 77



7800
FIG. 78

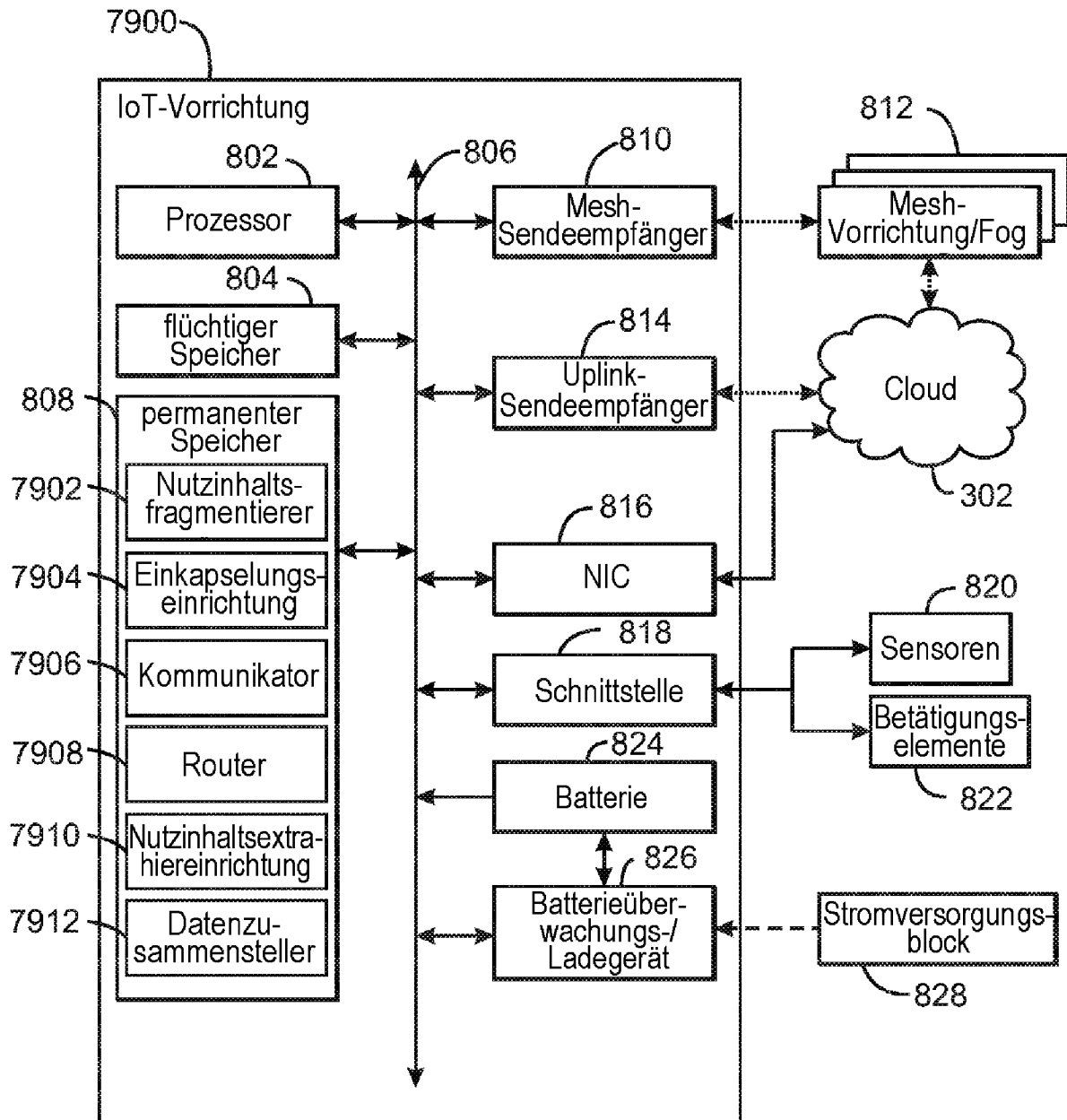


FIG. 79

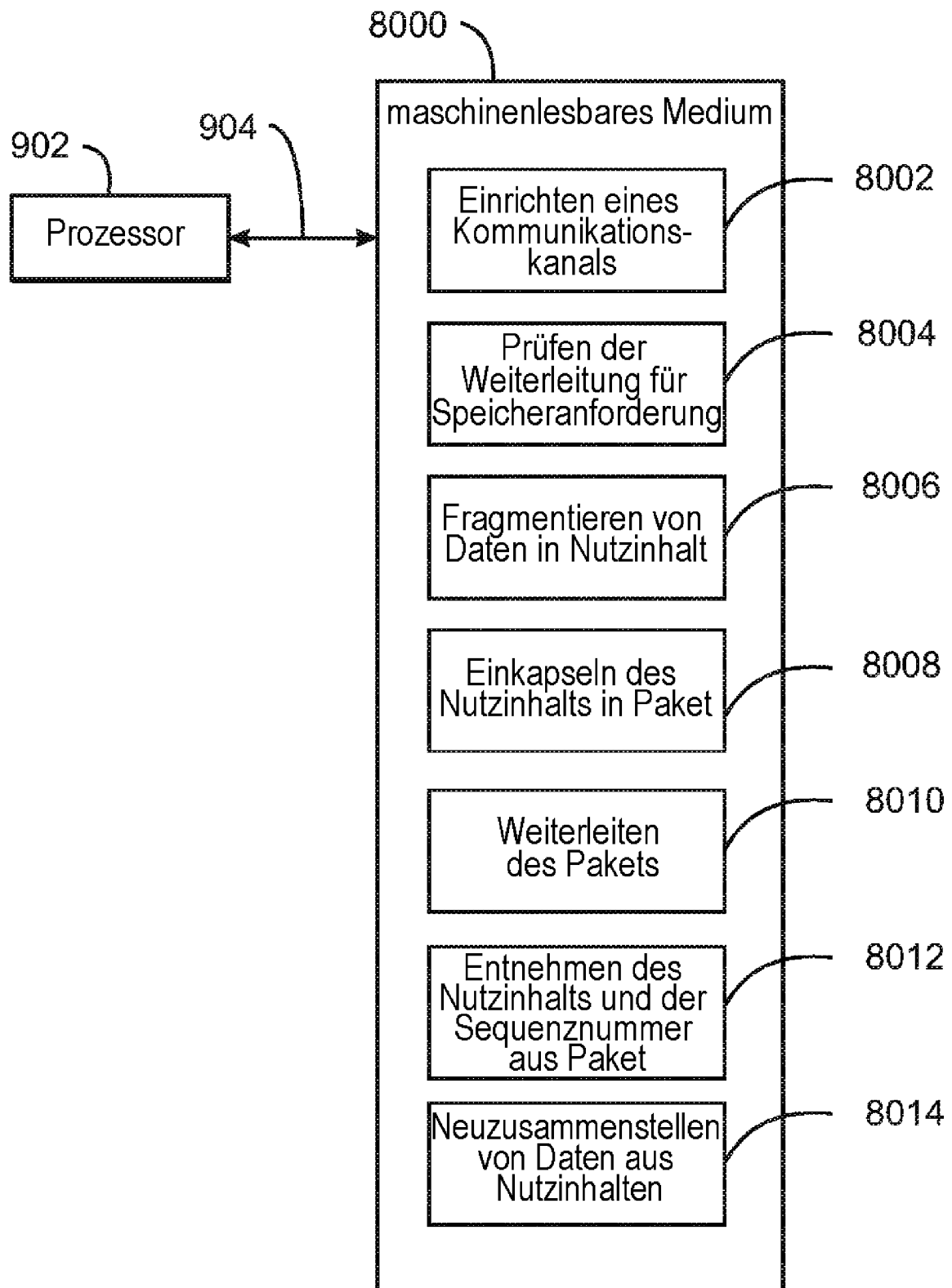
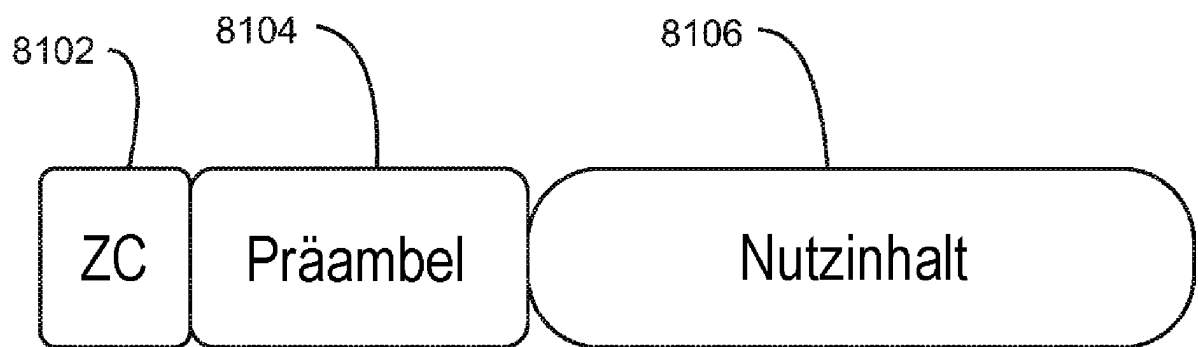


FIG. 80



8100
FIG. 81

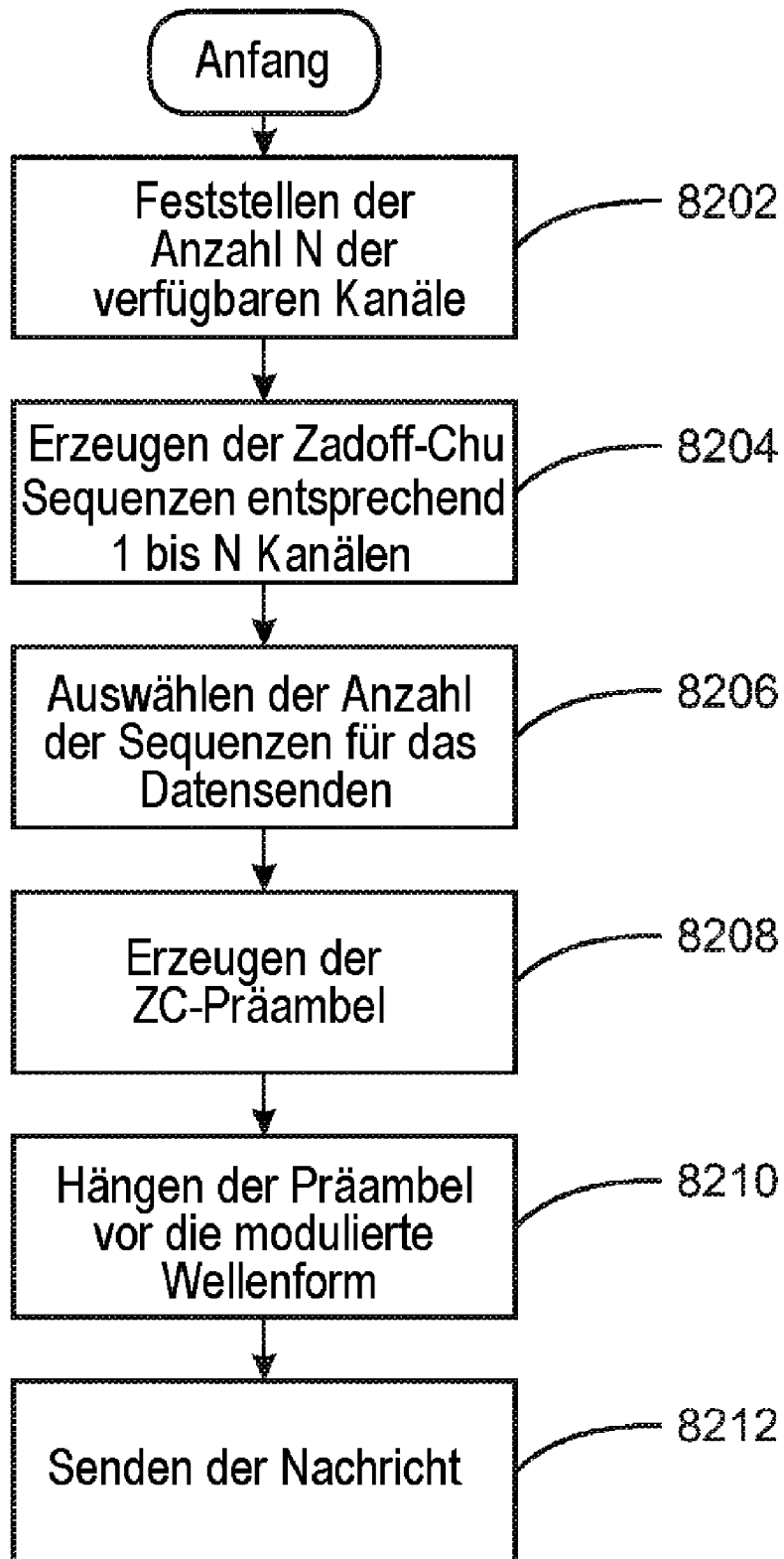
8200

FIG. 82

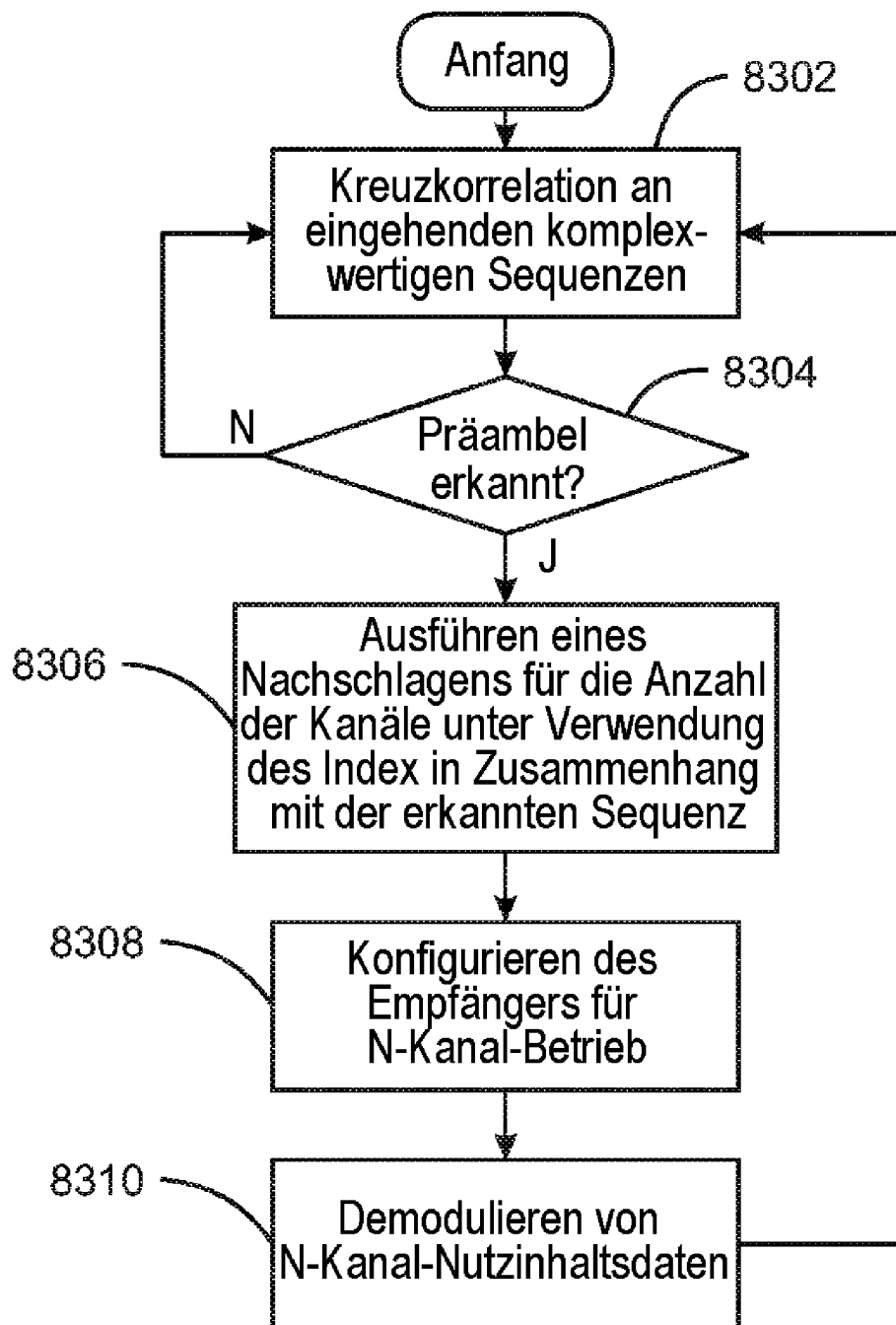
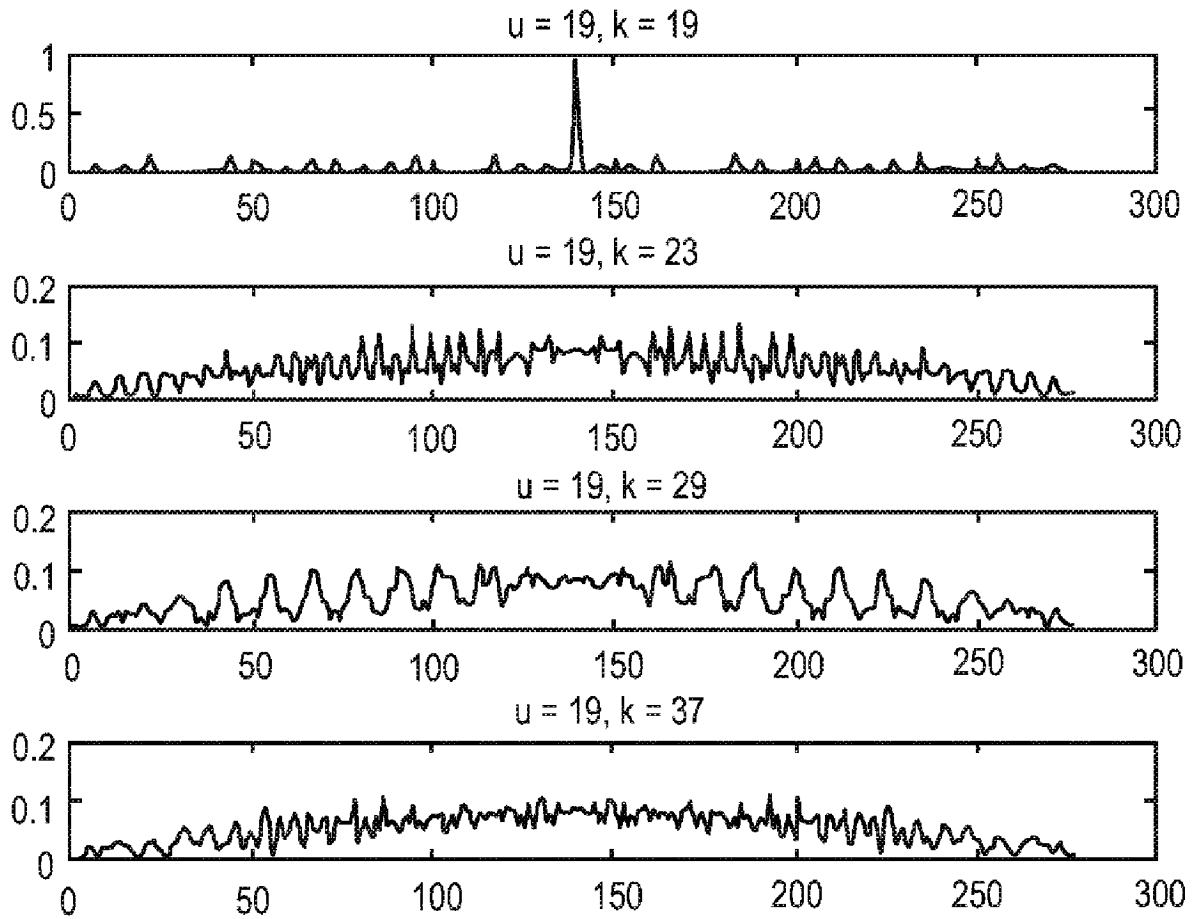
8300

FIG. 83



8400
FIG. 84

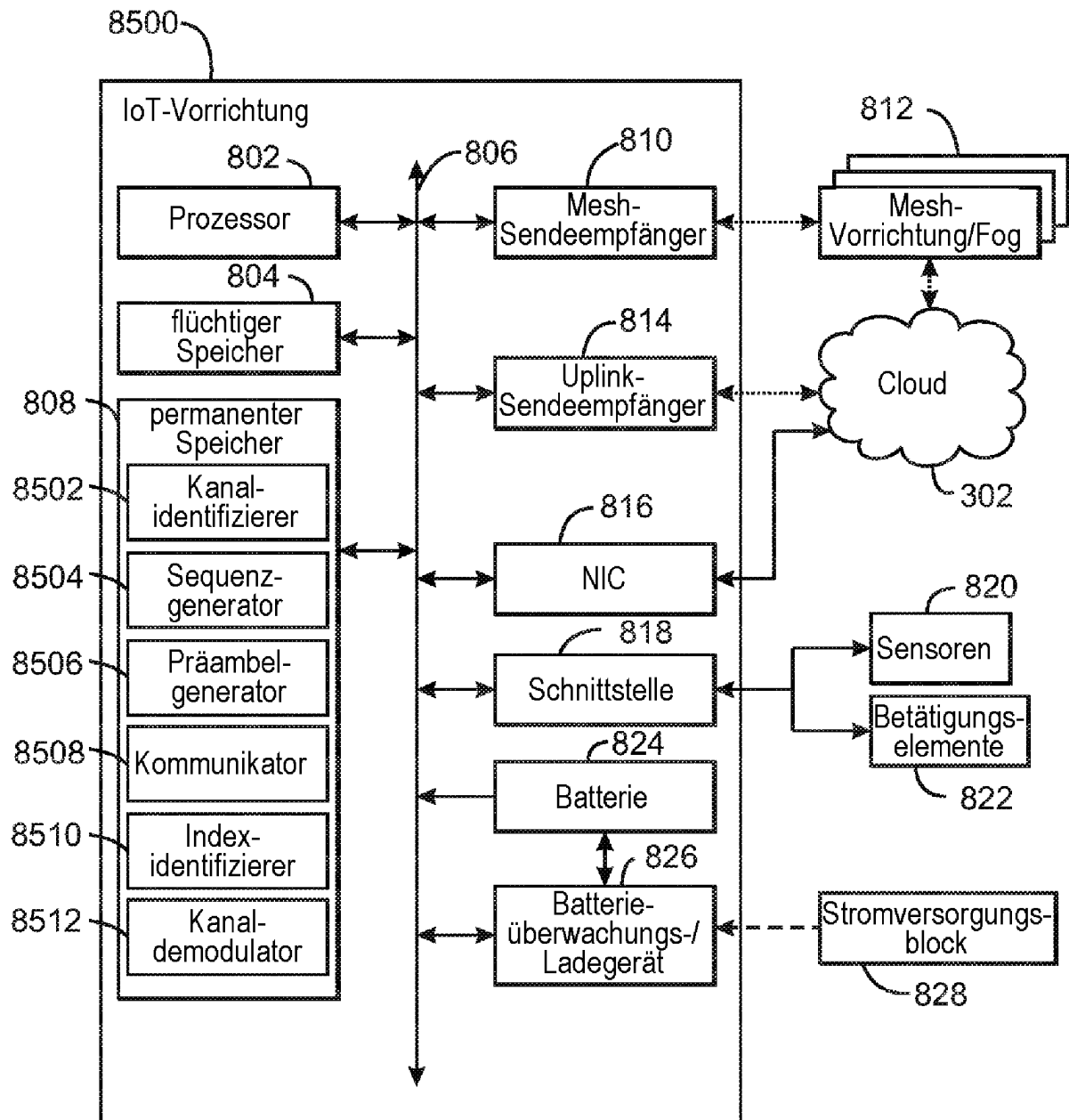


FIG. 85

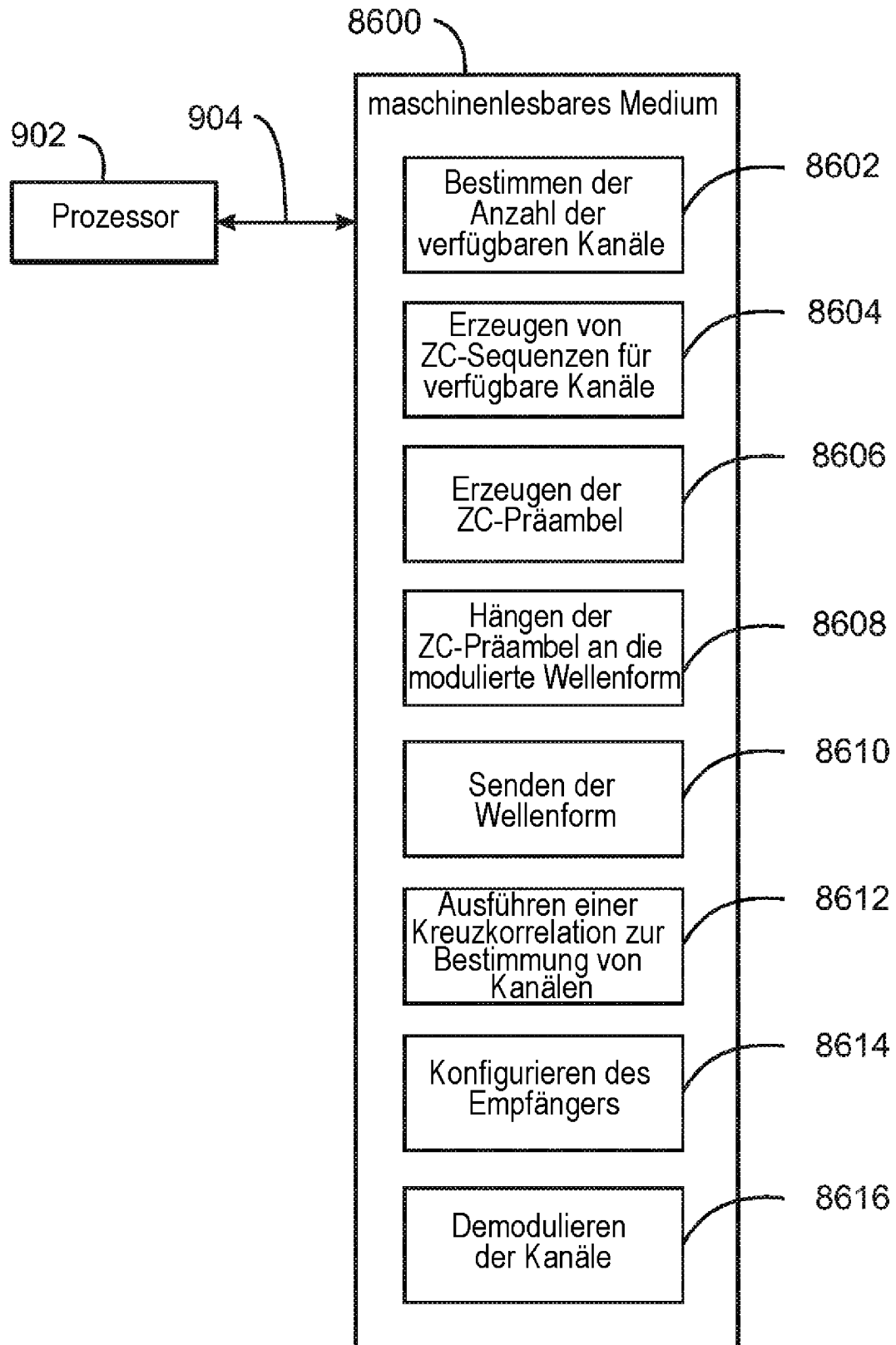
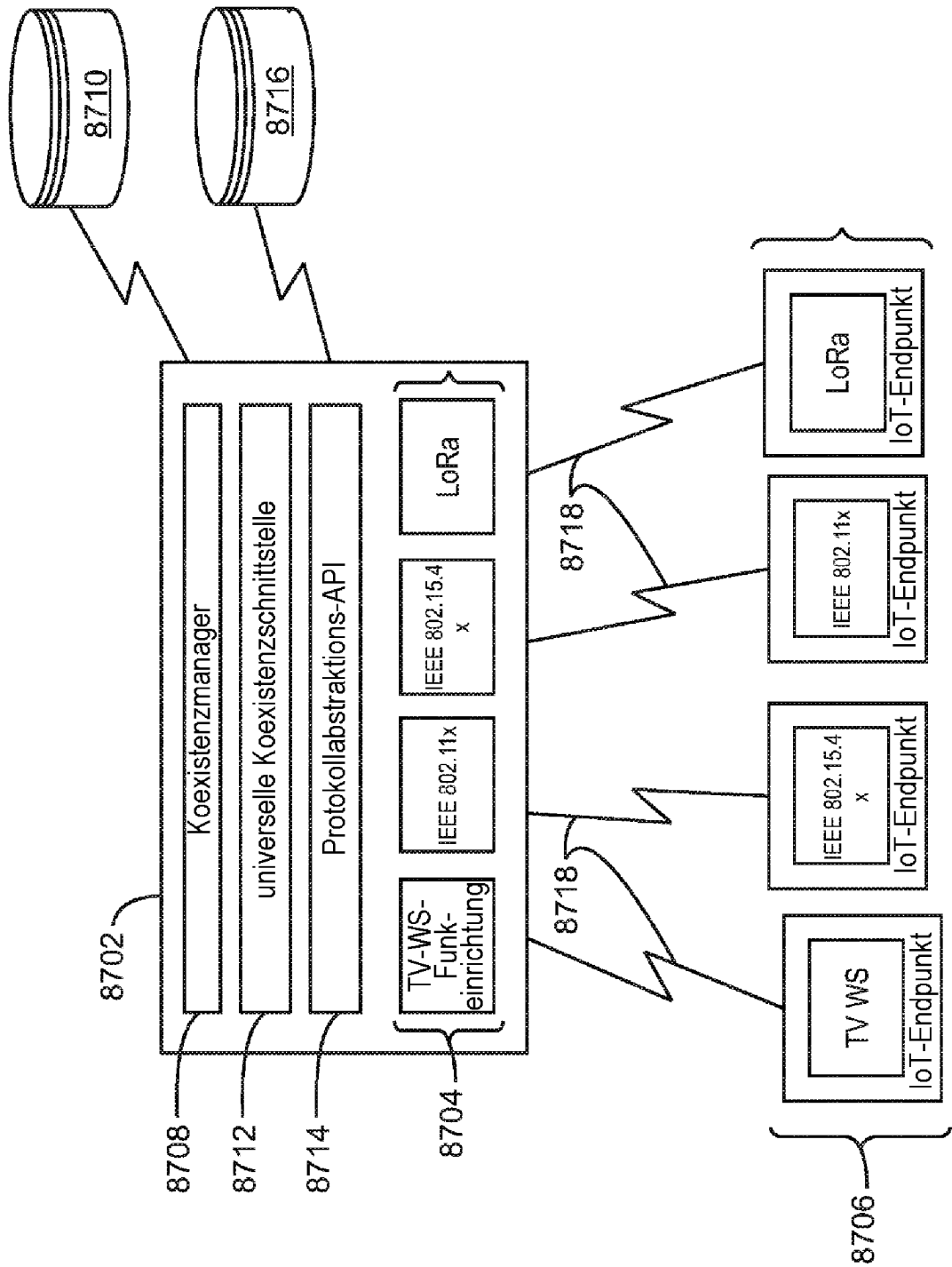


FIG. 86



8700
FIG. 87

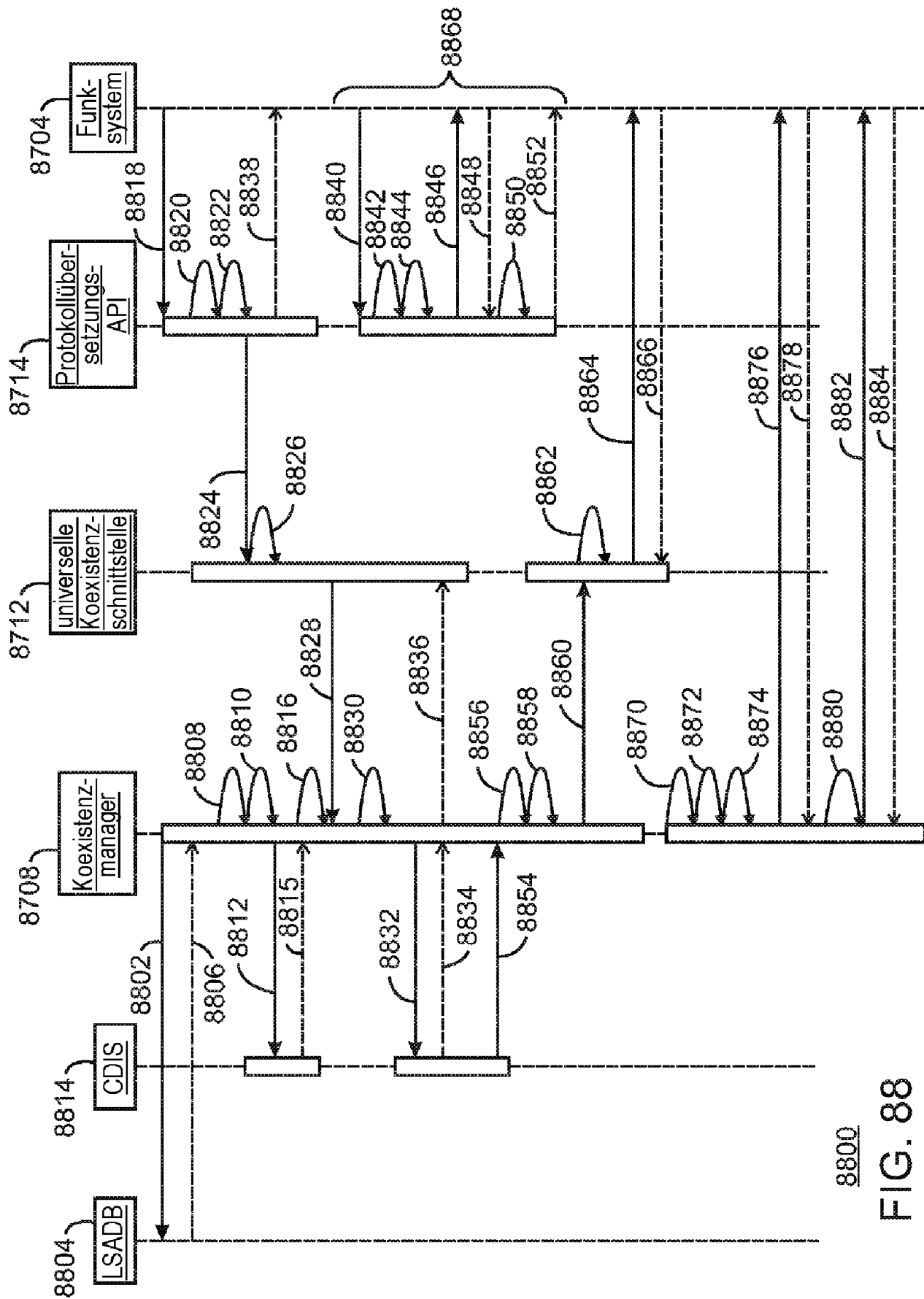
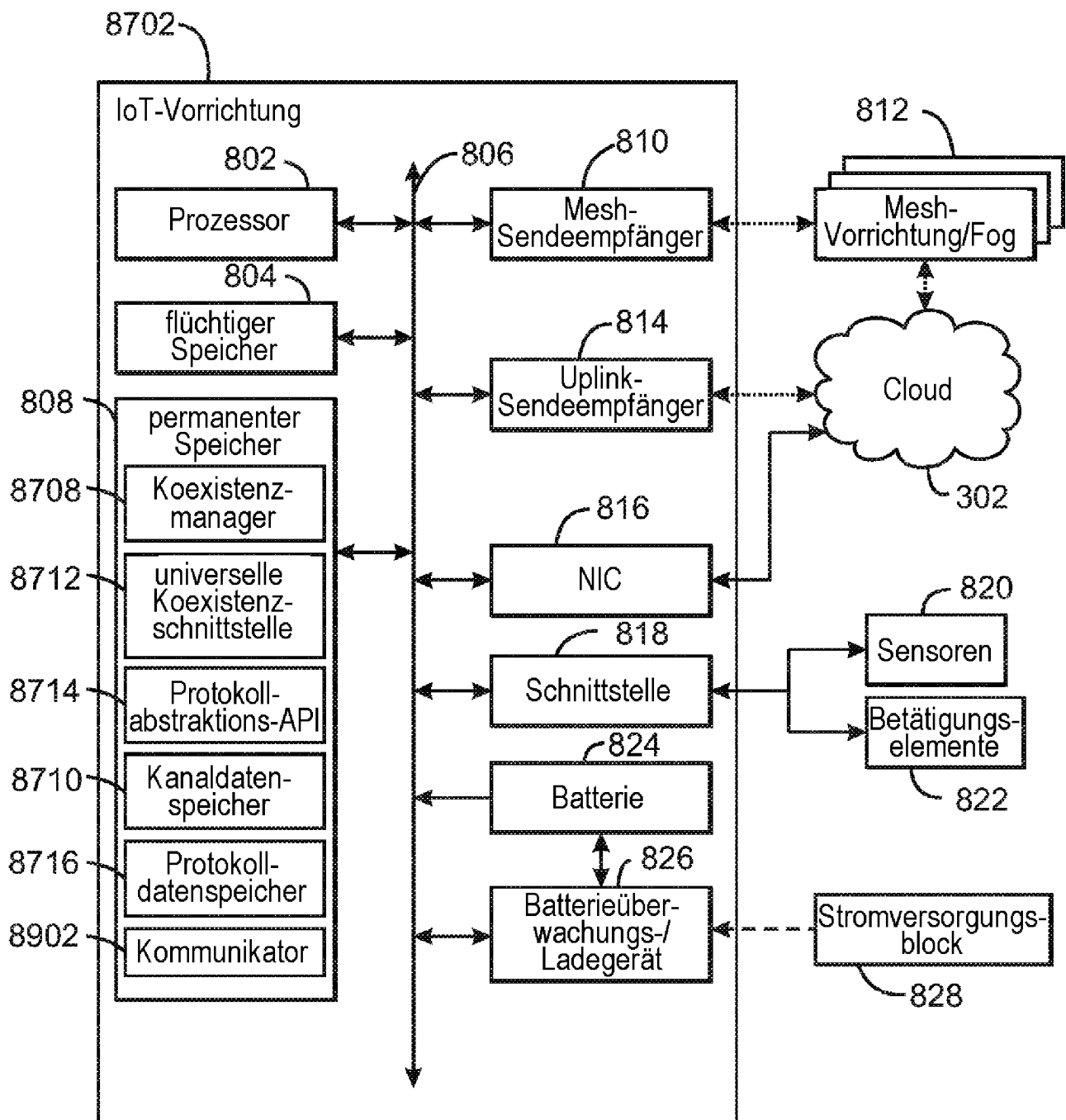


FIG. 88



8900
FIG. 89

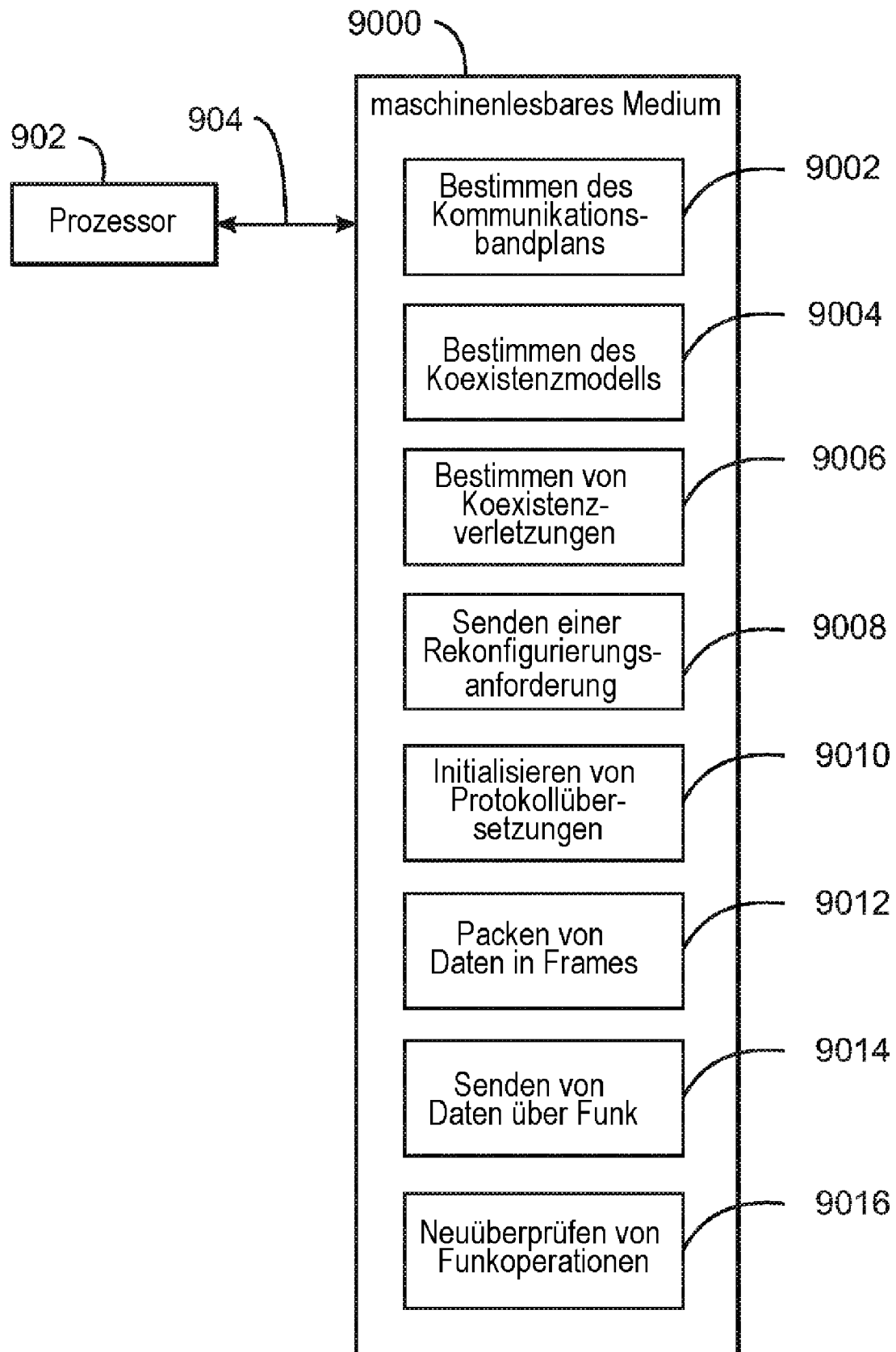
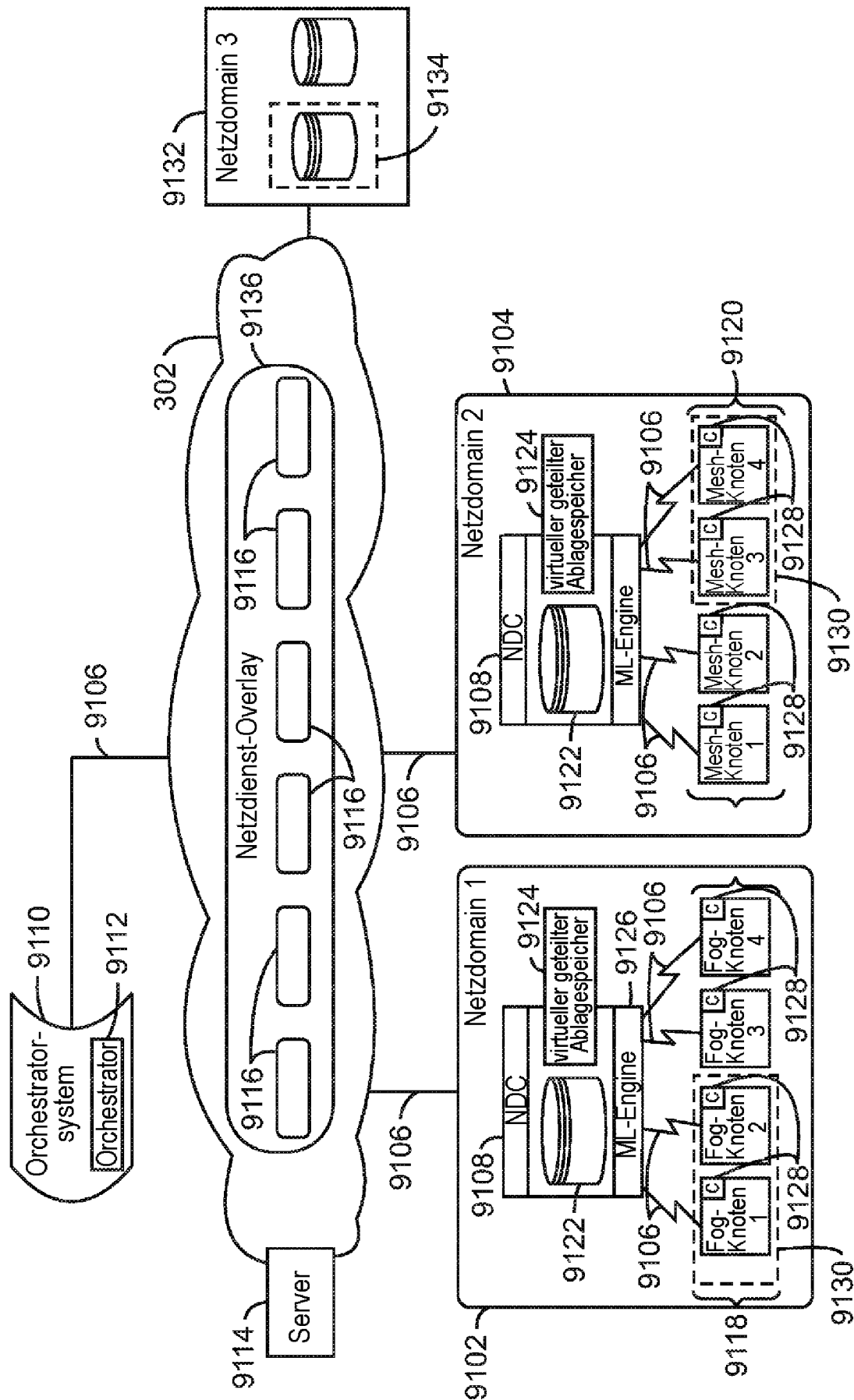


FIG. 90



9100

50

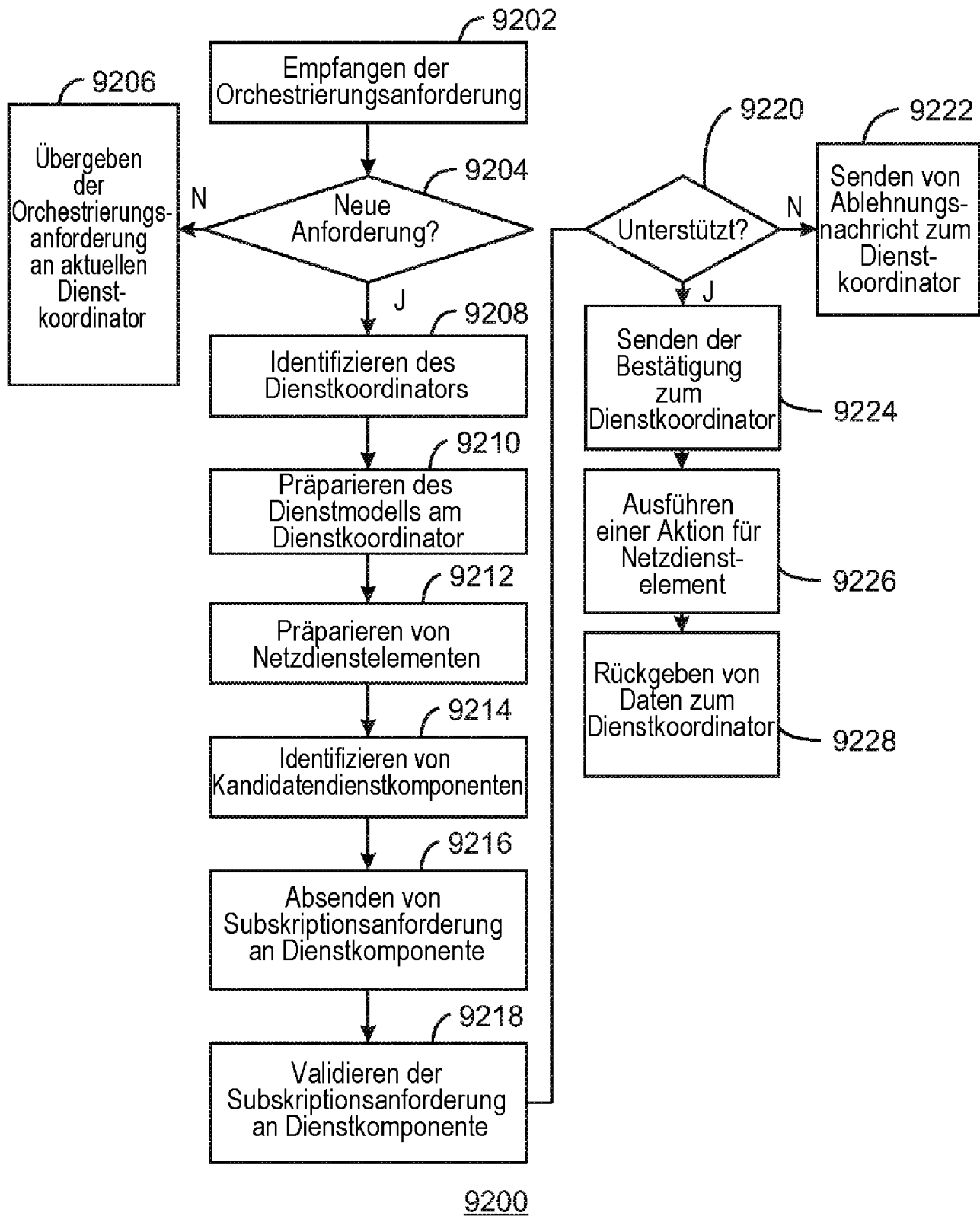
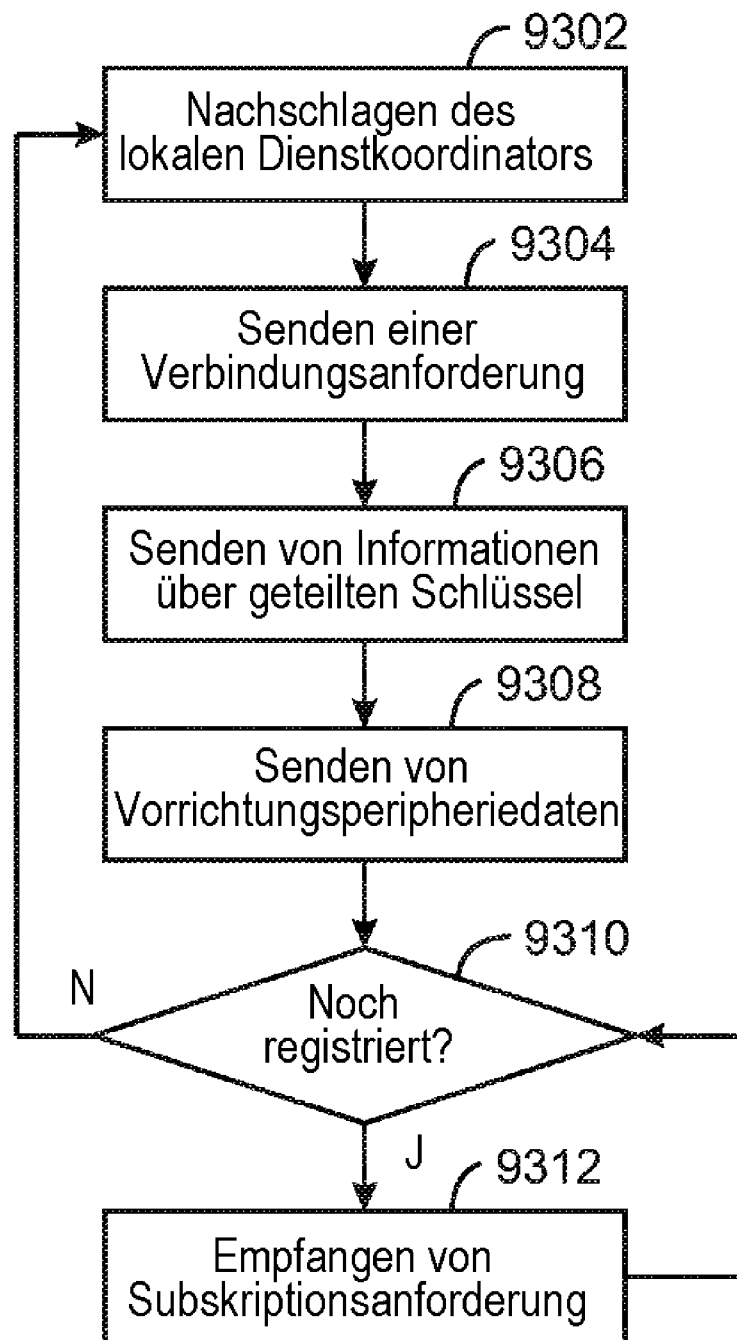


FIG. 92

9300**FIG. 93**

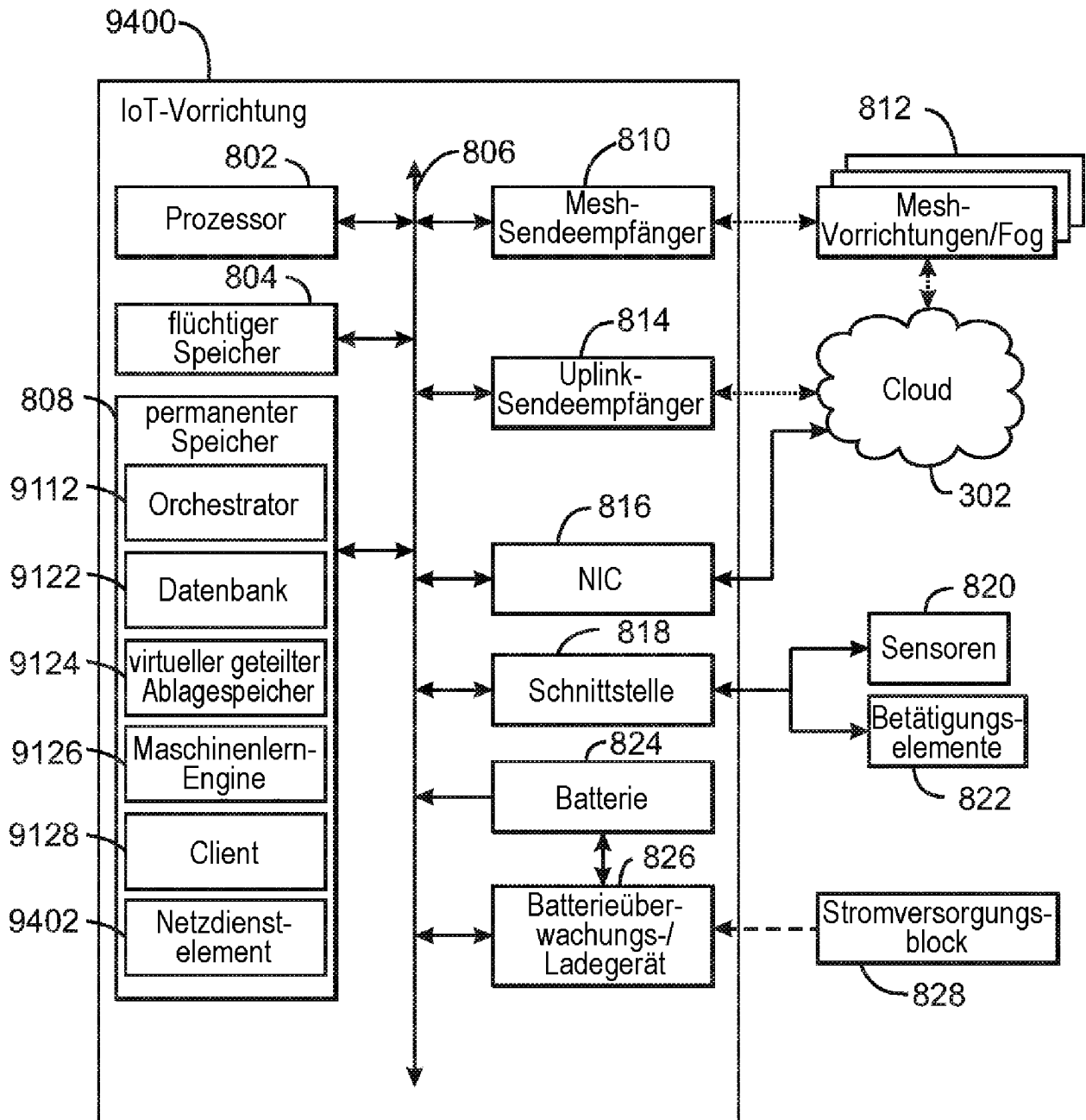


FIG. 94

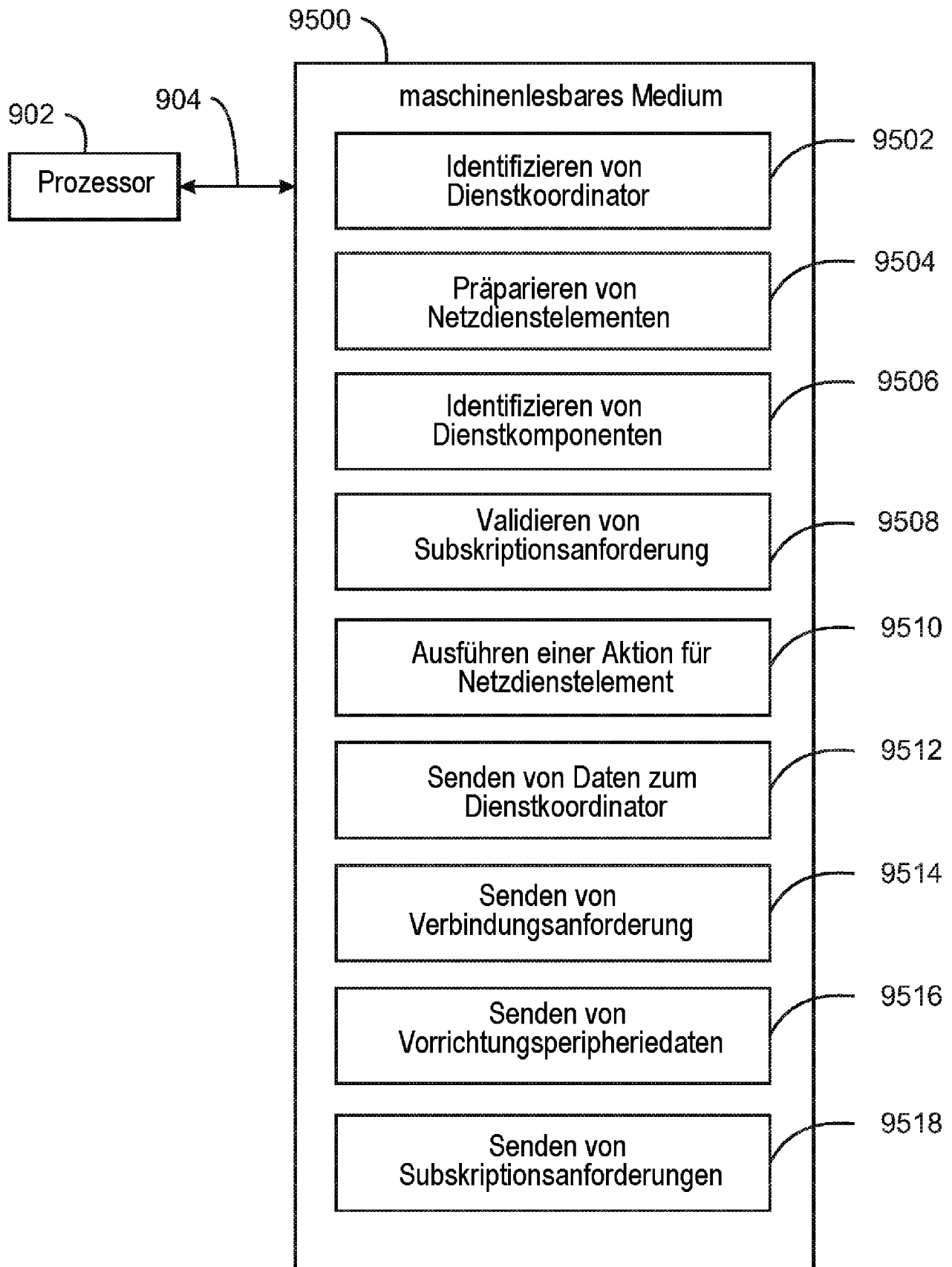
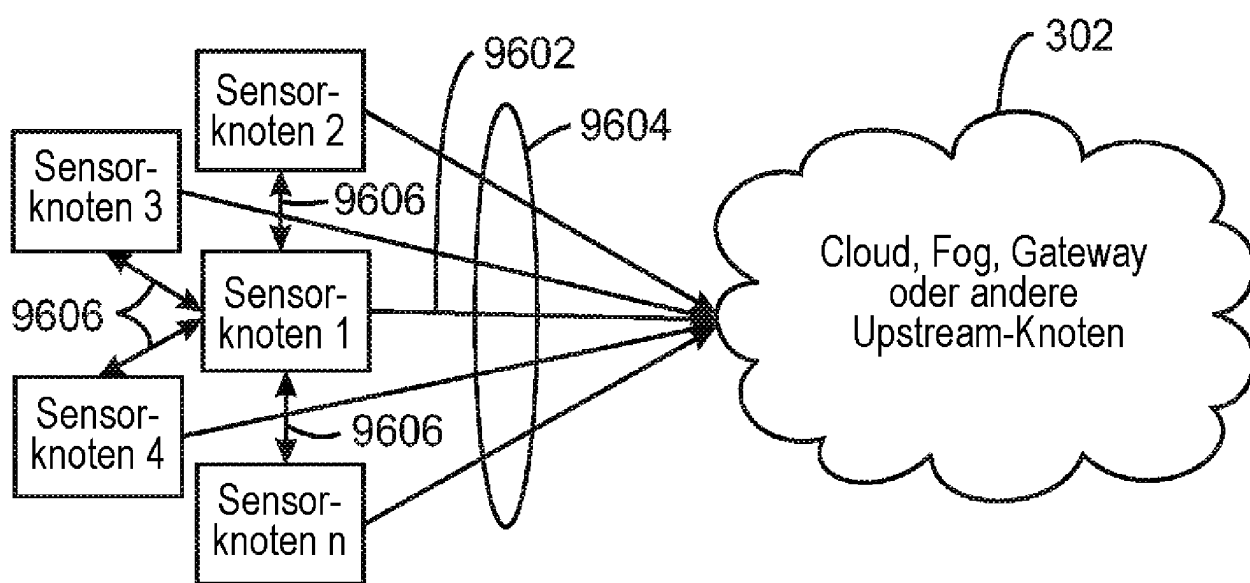
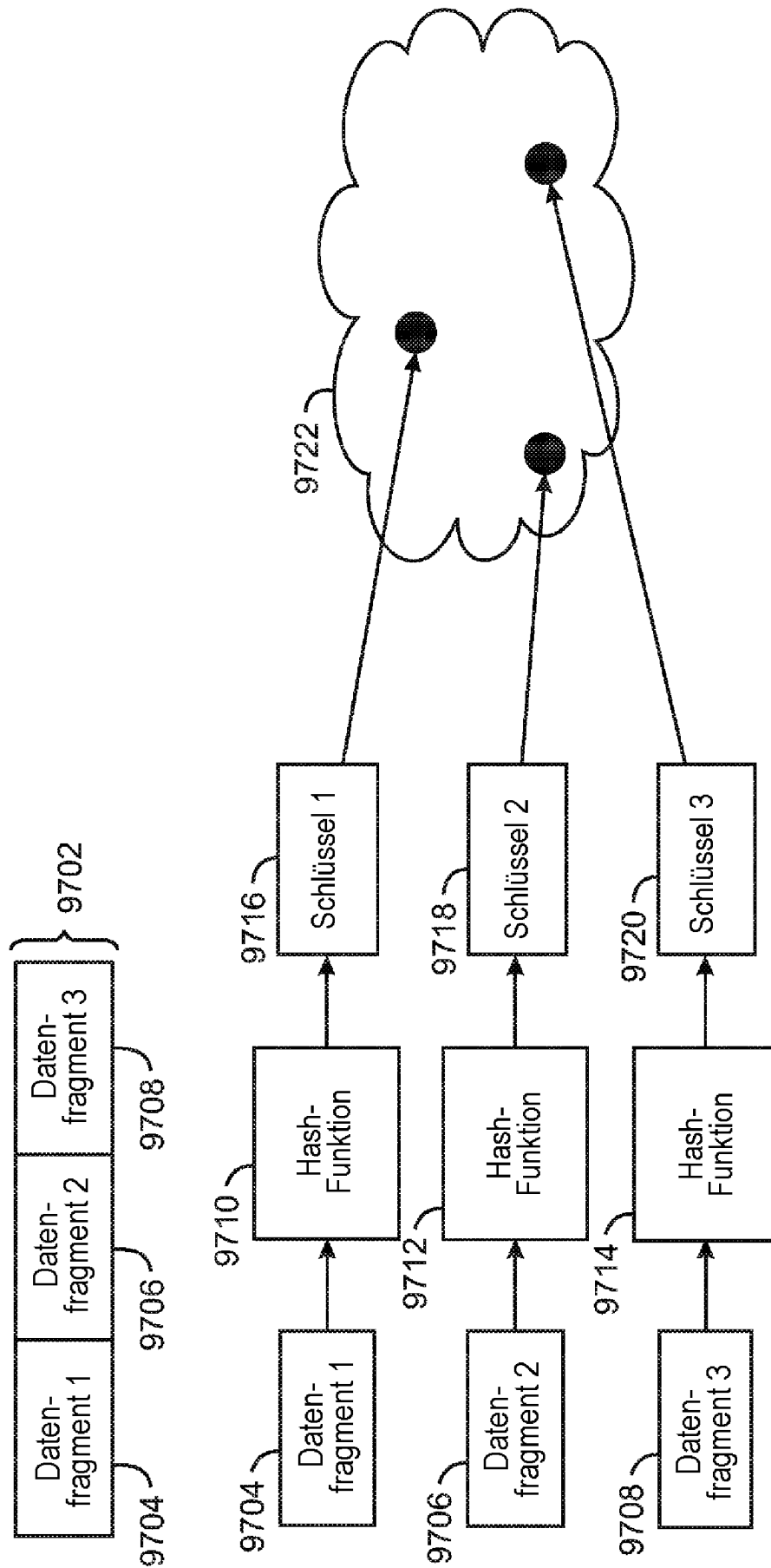


FIG. 95

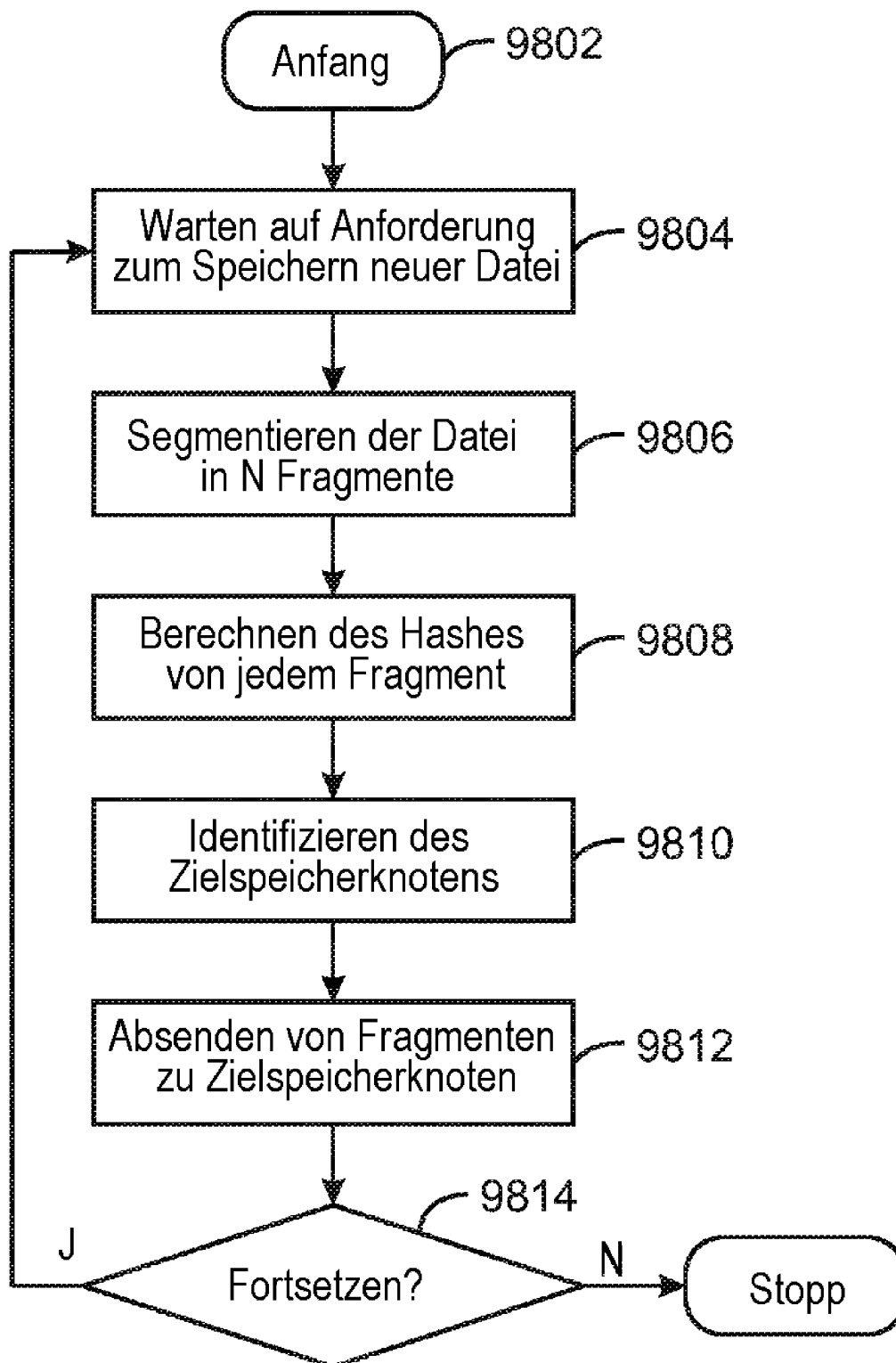


9600
FIG. 96



9700

FIG. 97

9800**FIG. 98**

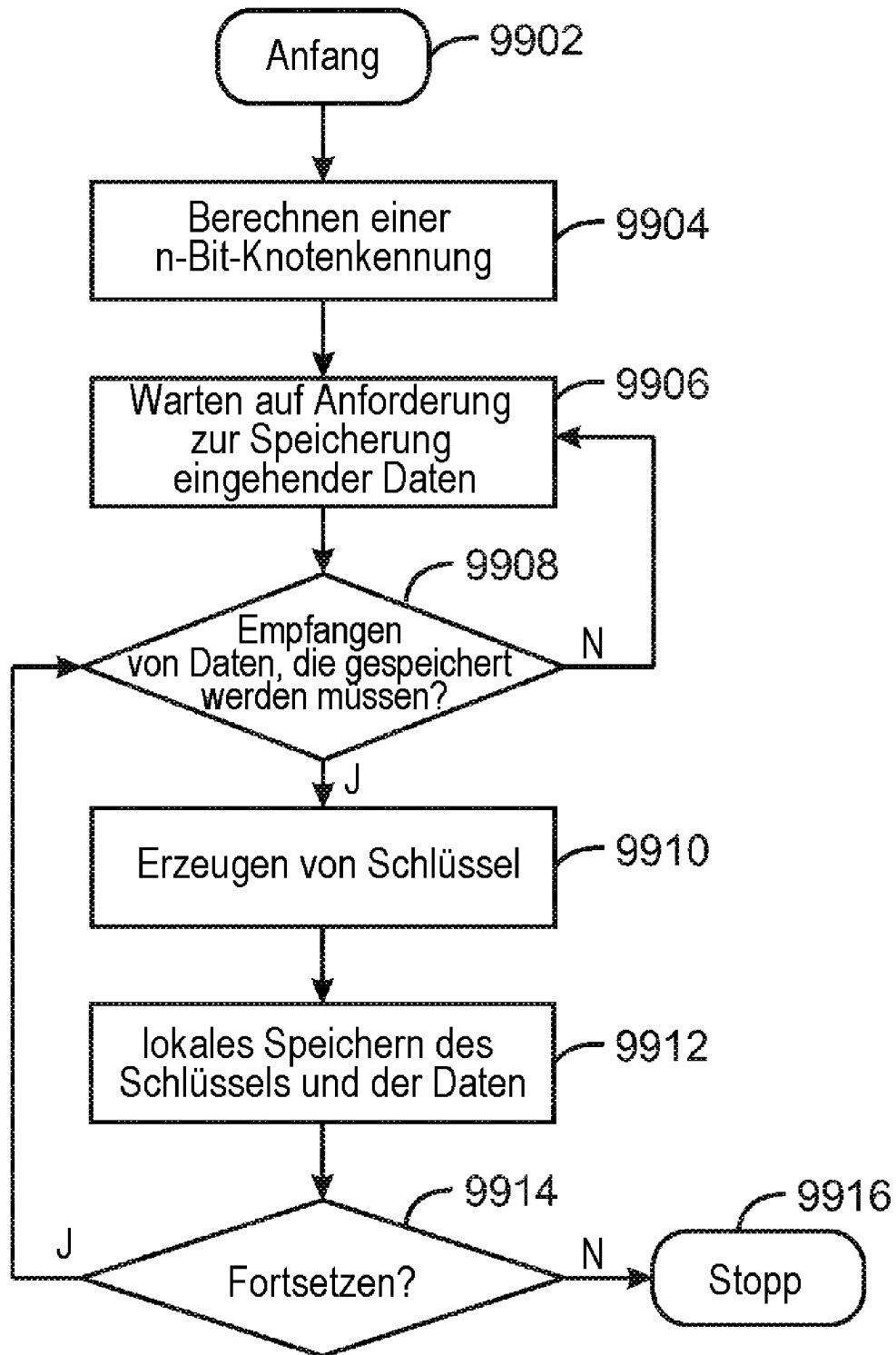
9900

FIG. 99

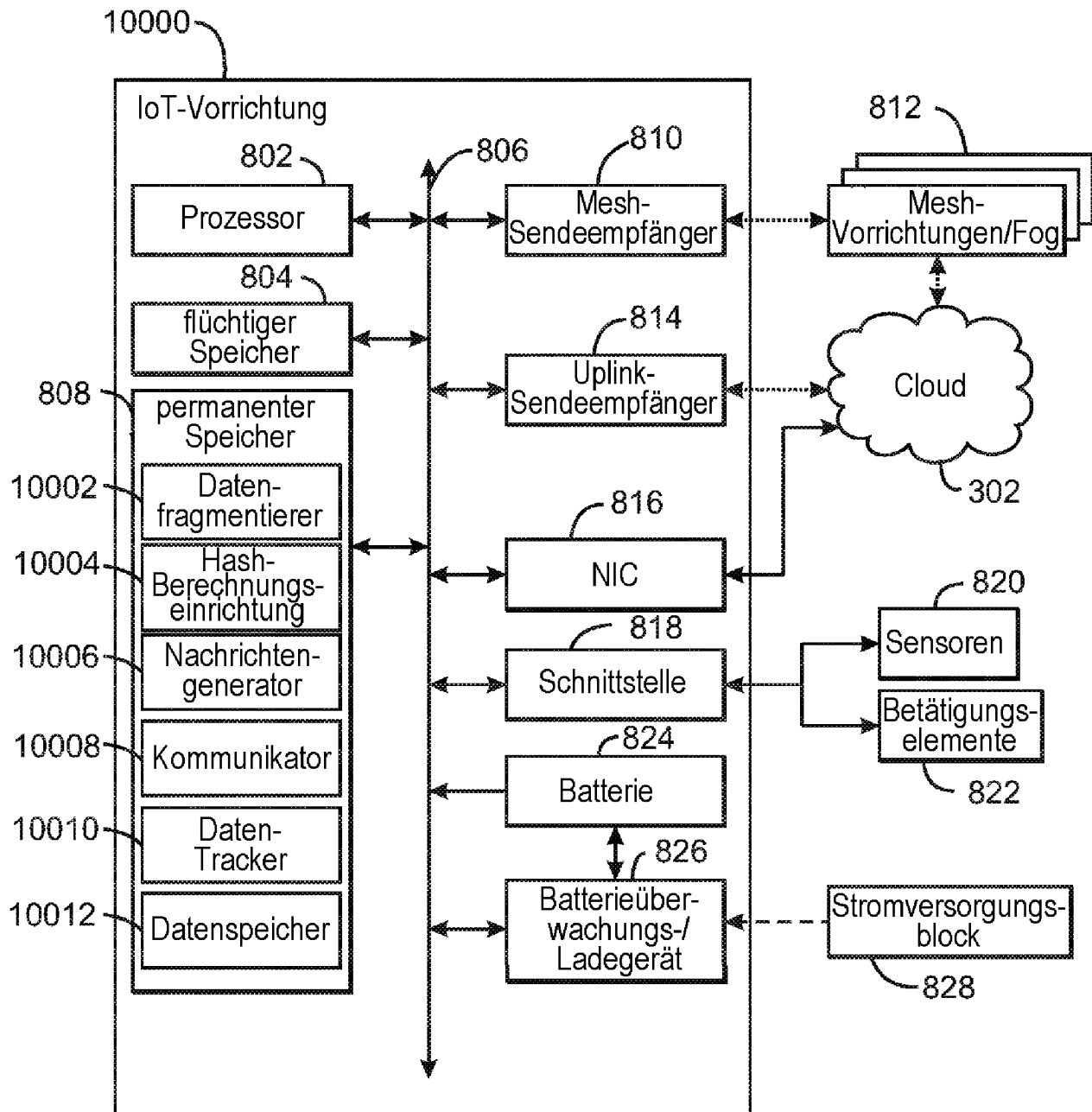


FIG. 100

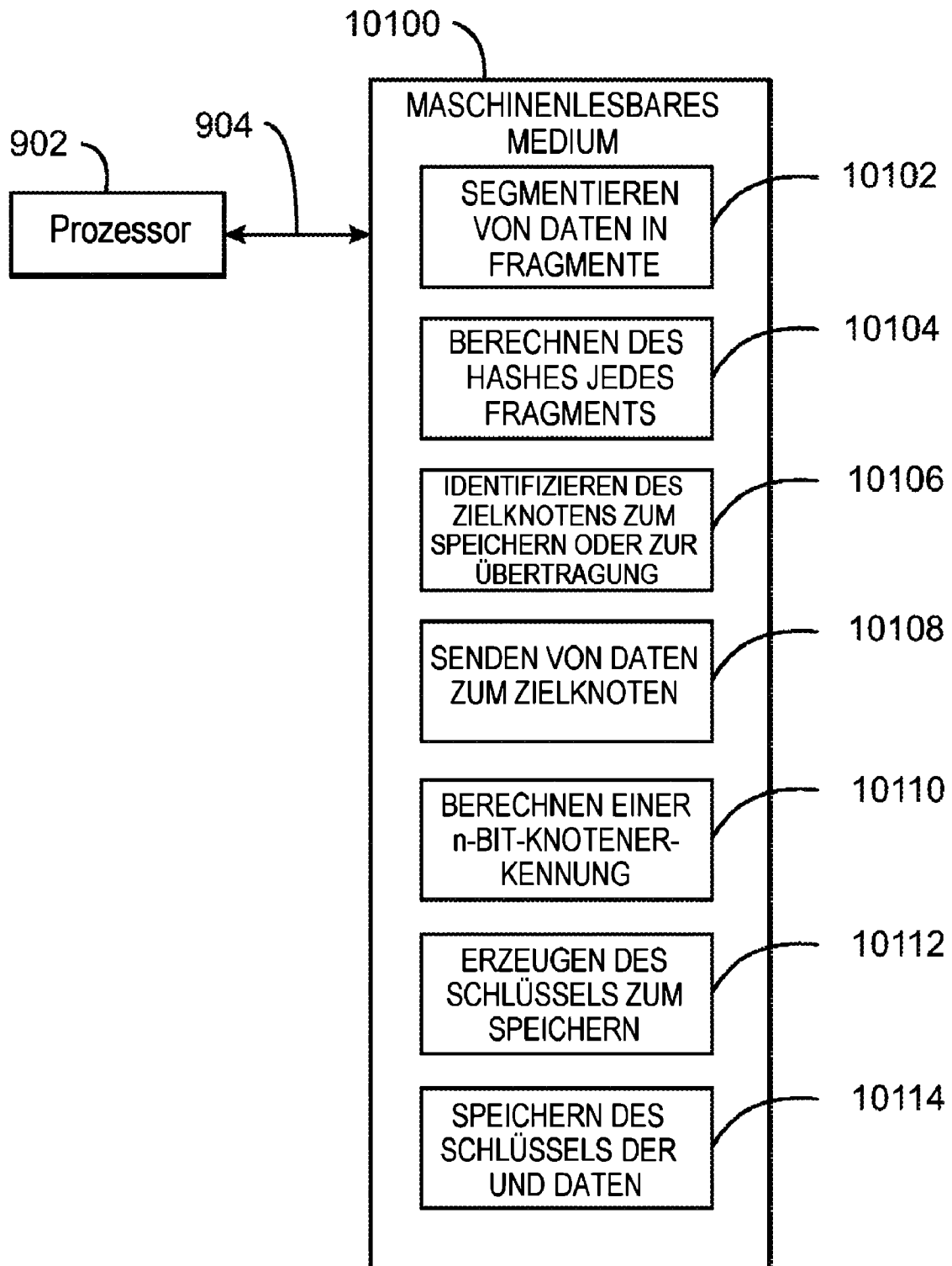
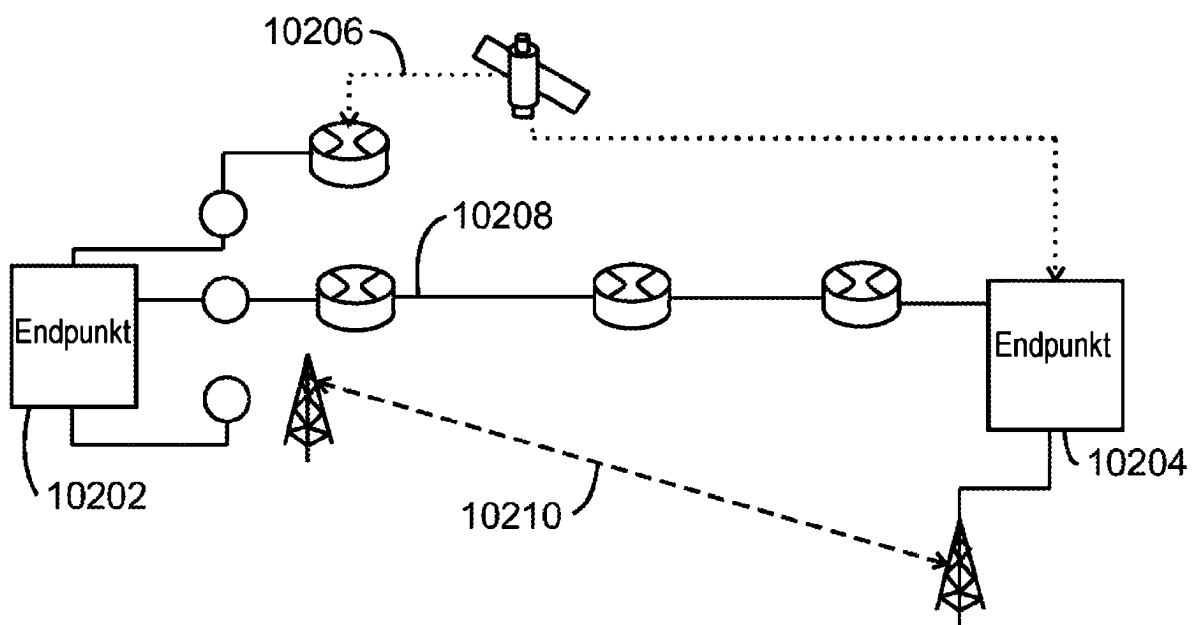
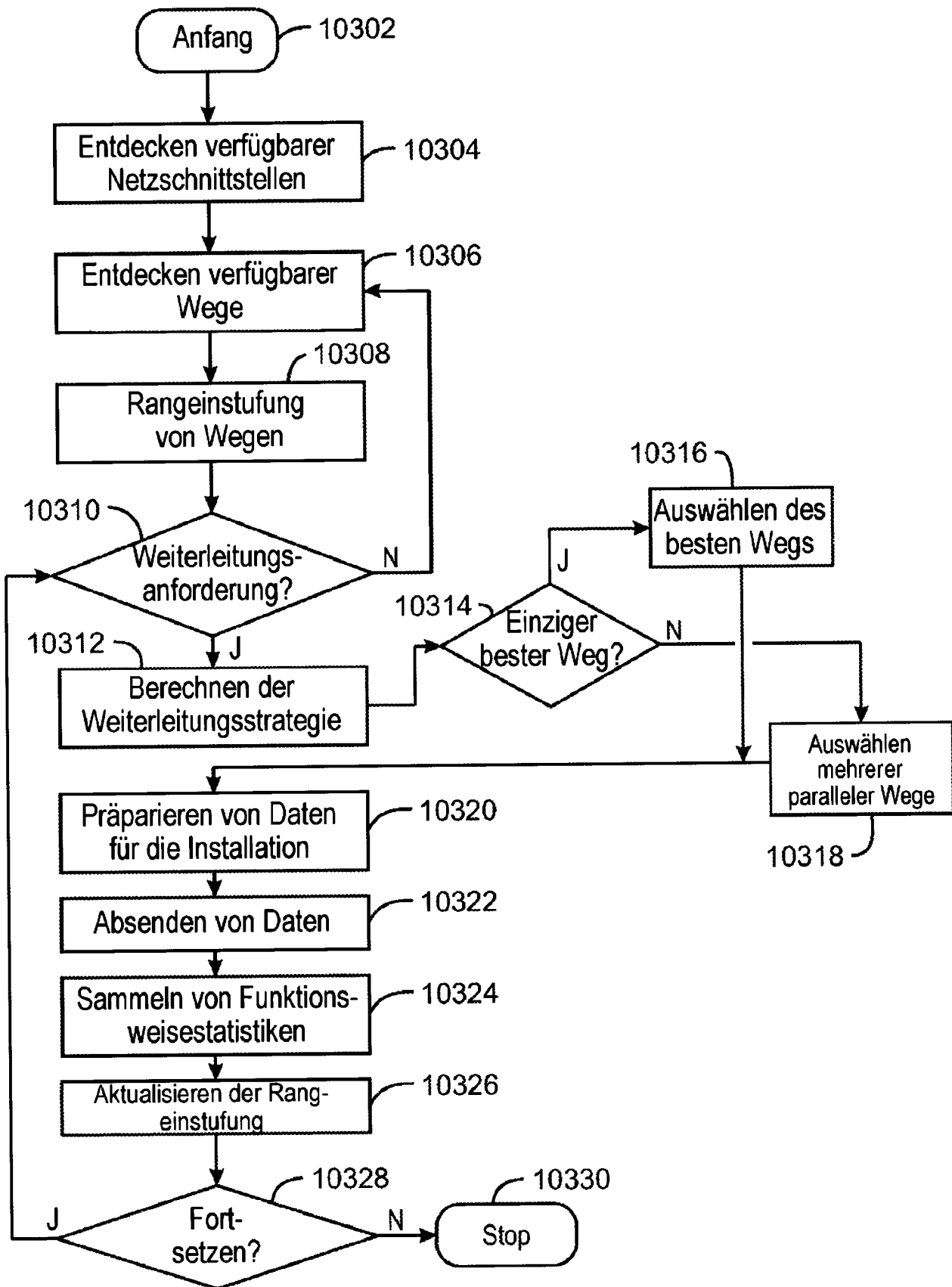


FIG. 101



10200
FIG. 102



10300

FIG. 103

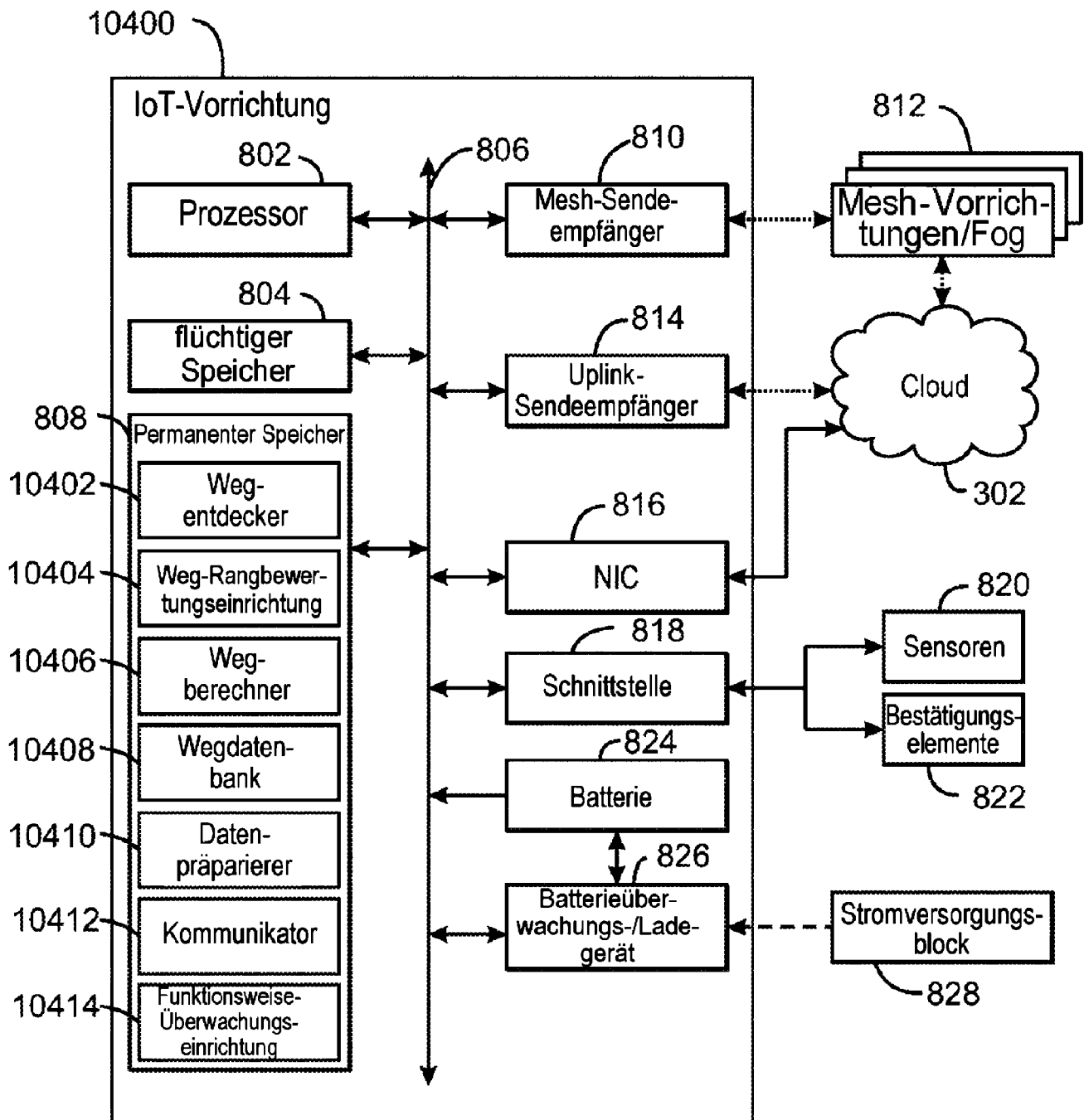


FIG. 104

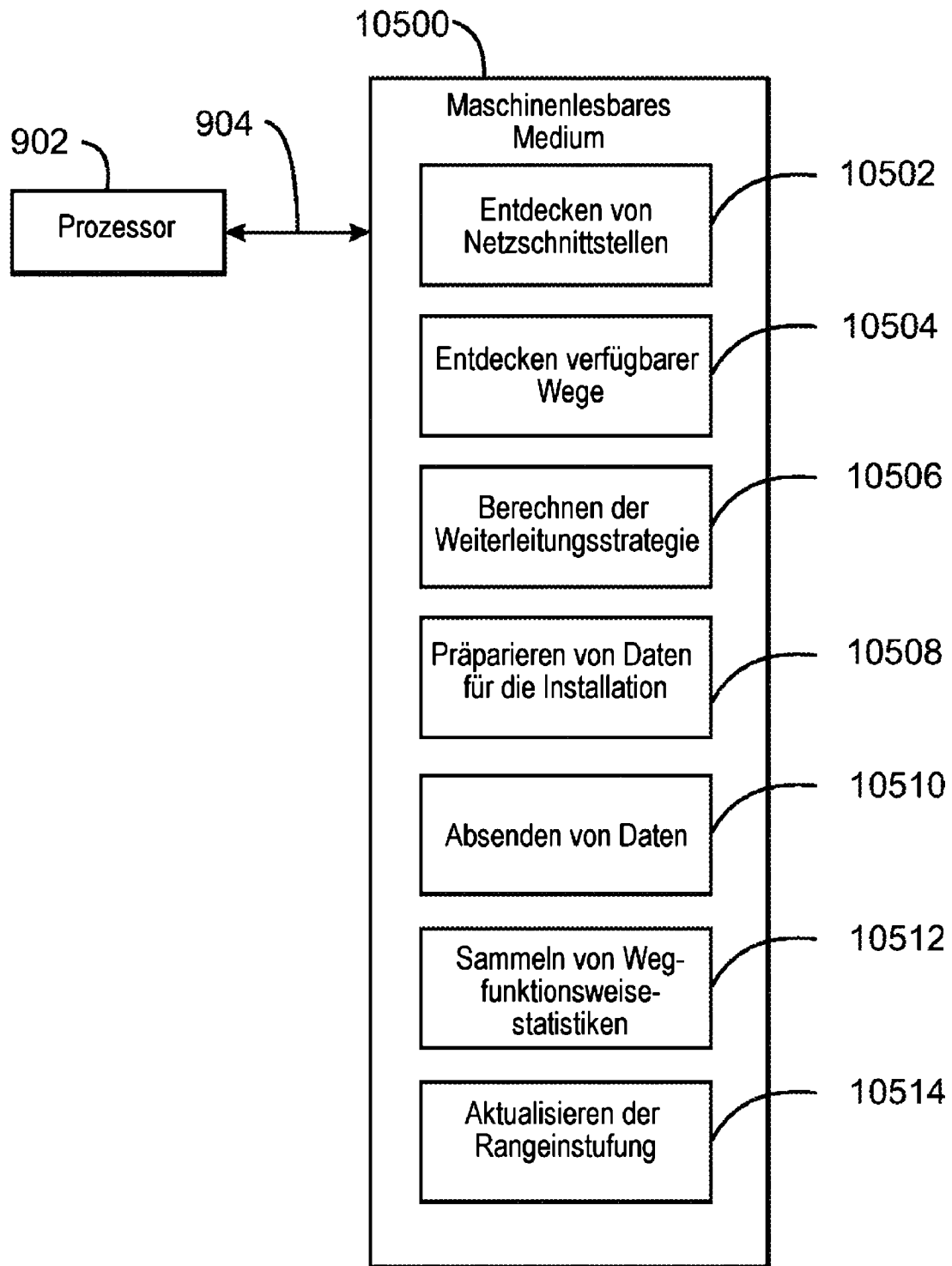
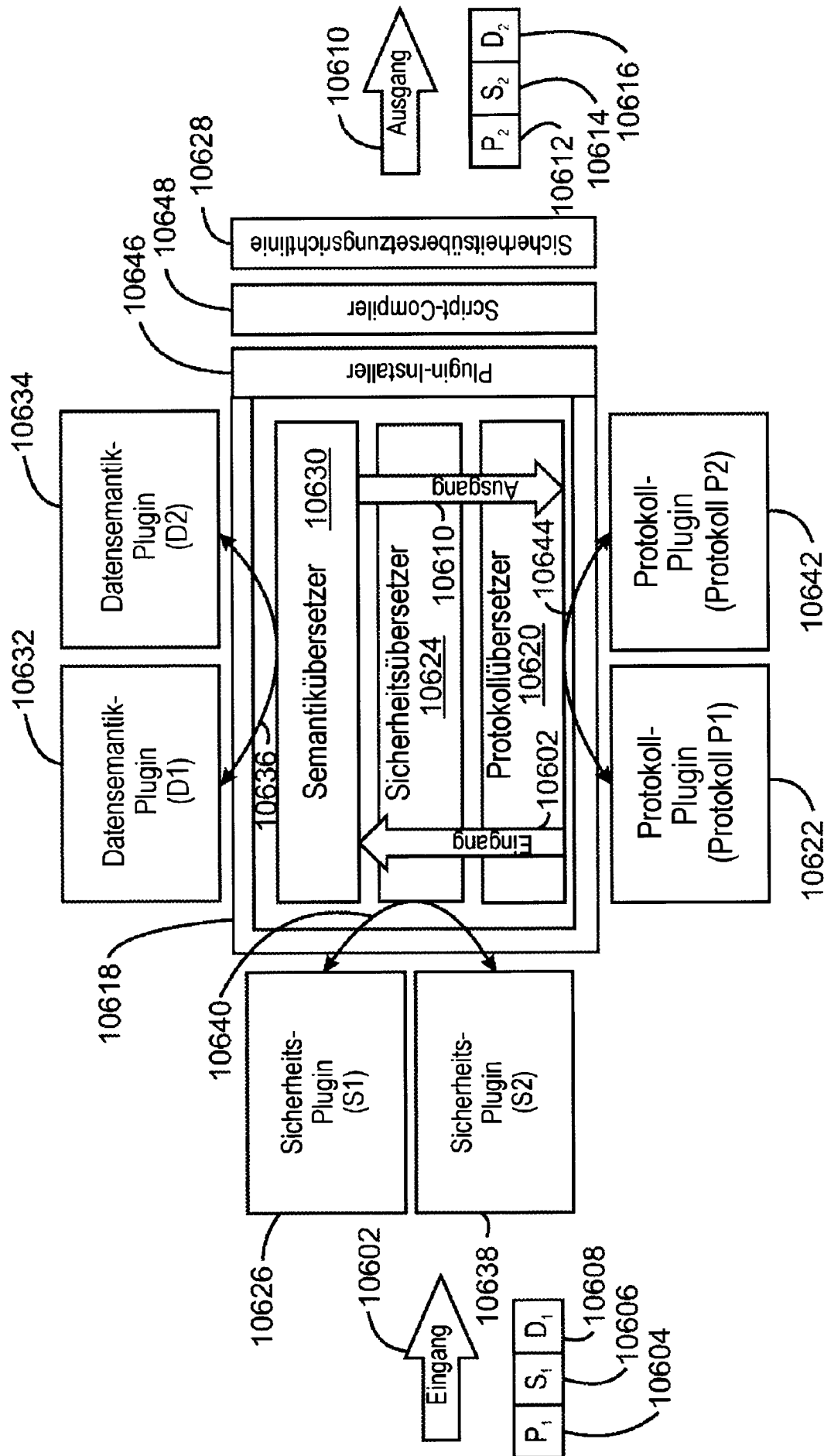
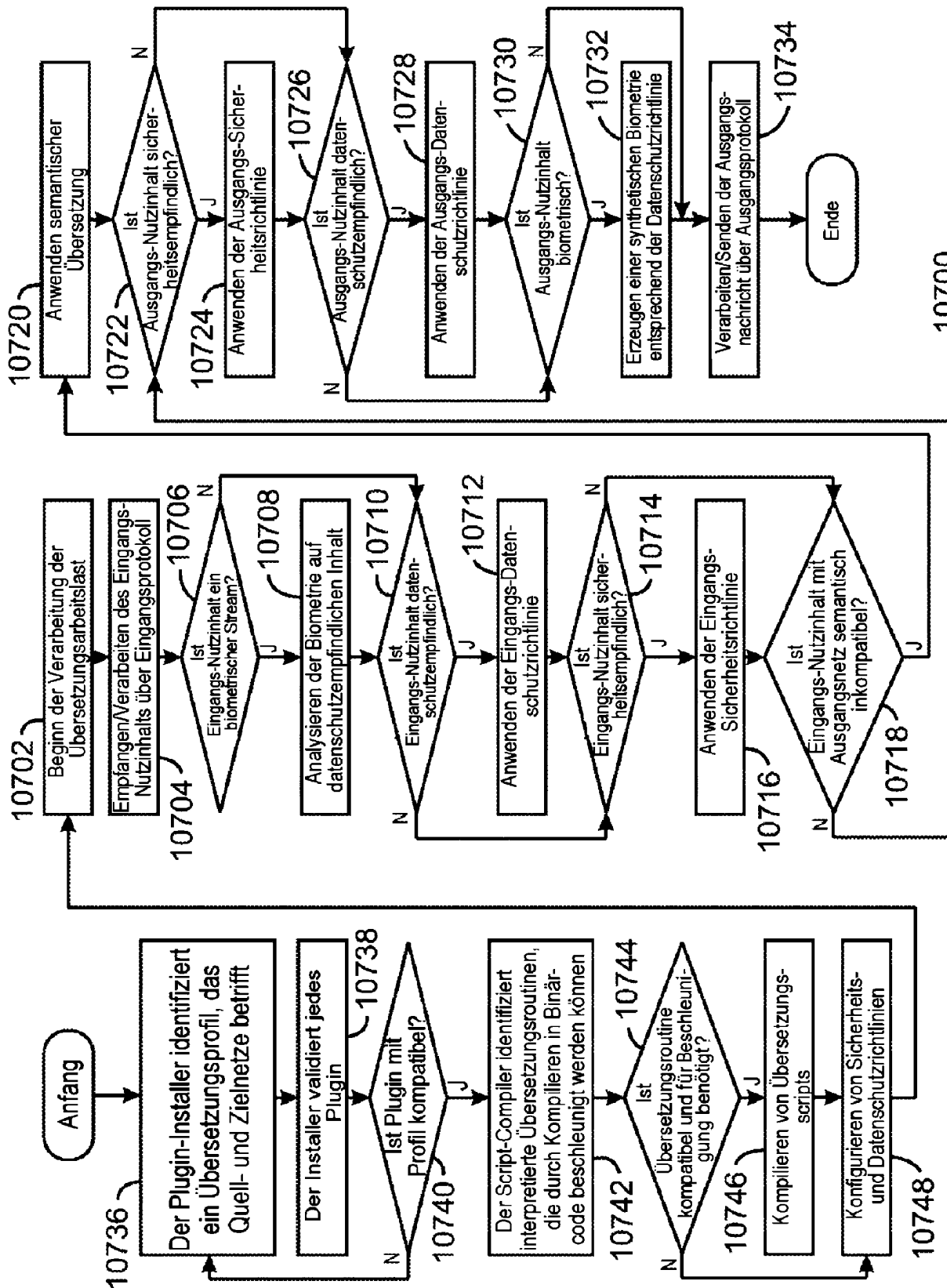


FIG. 105



10600

FIG. 106



10700

FIG. 107

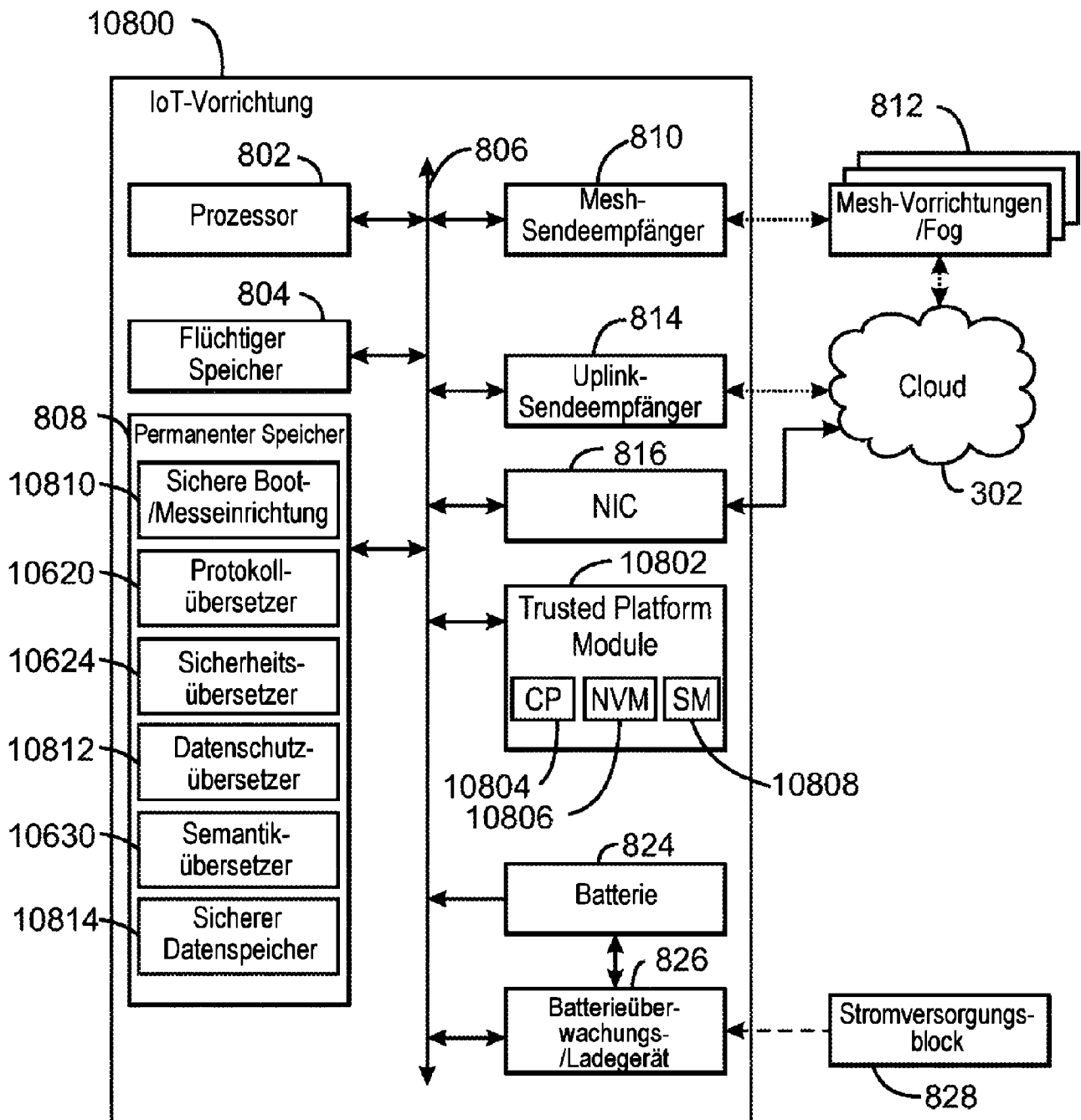


FIG. 108

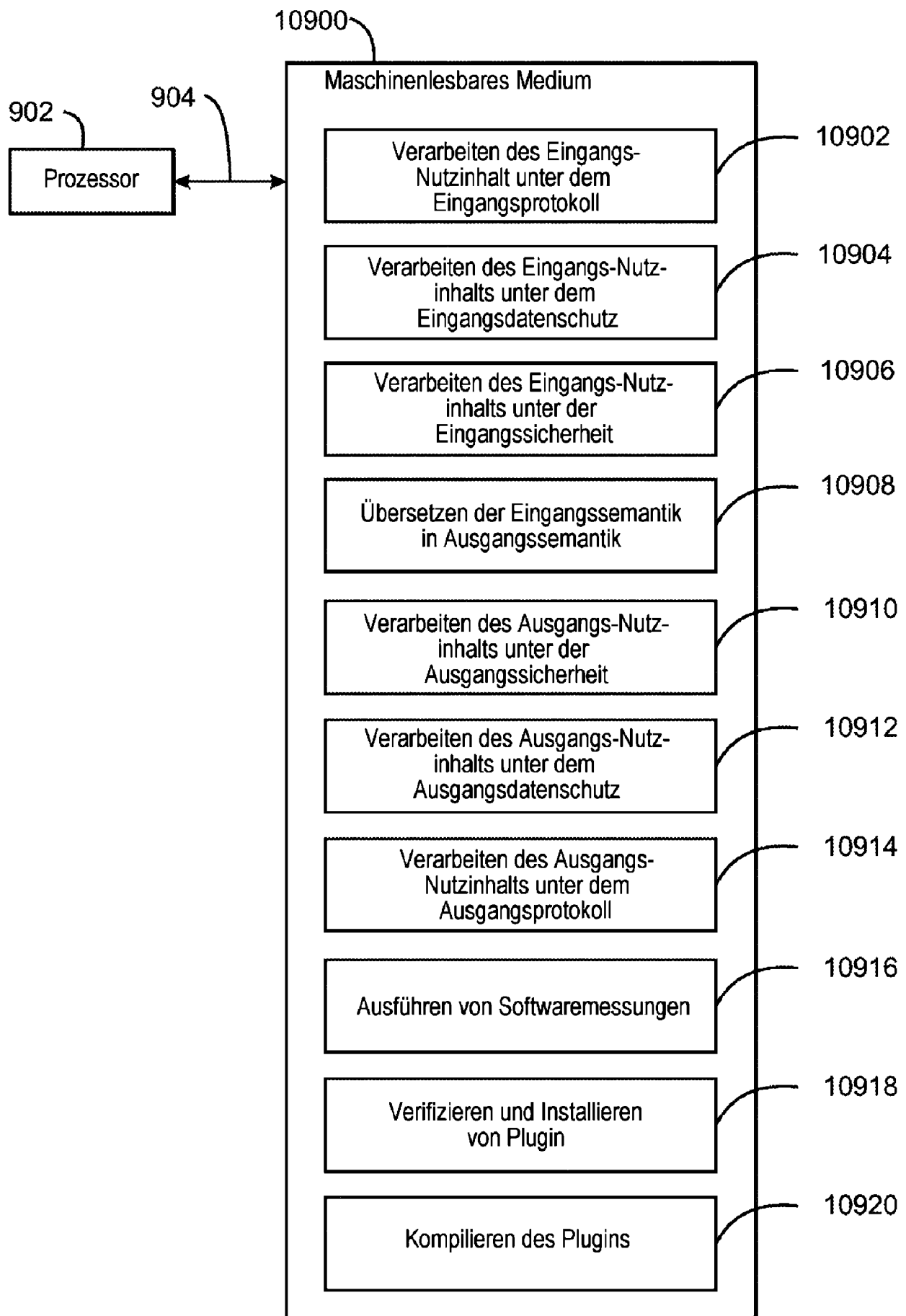
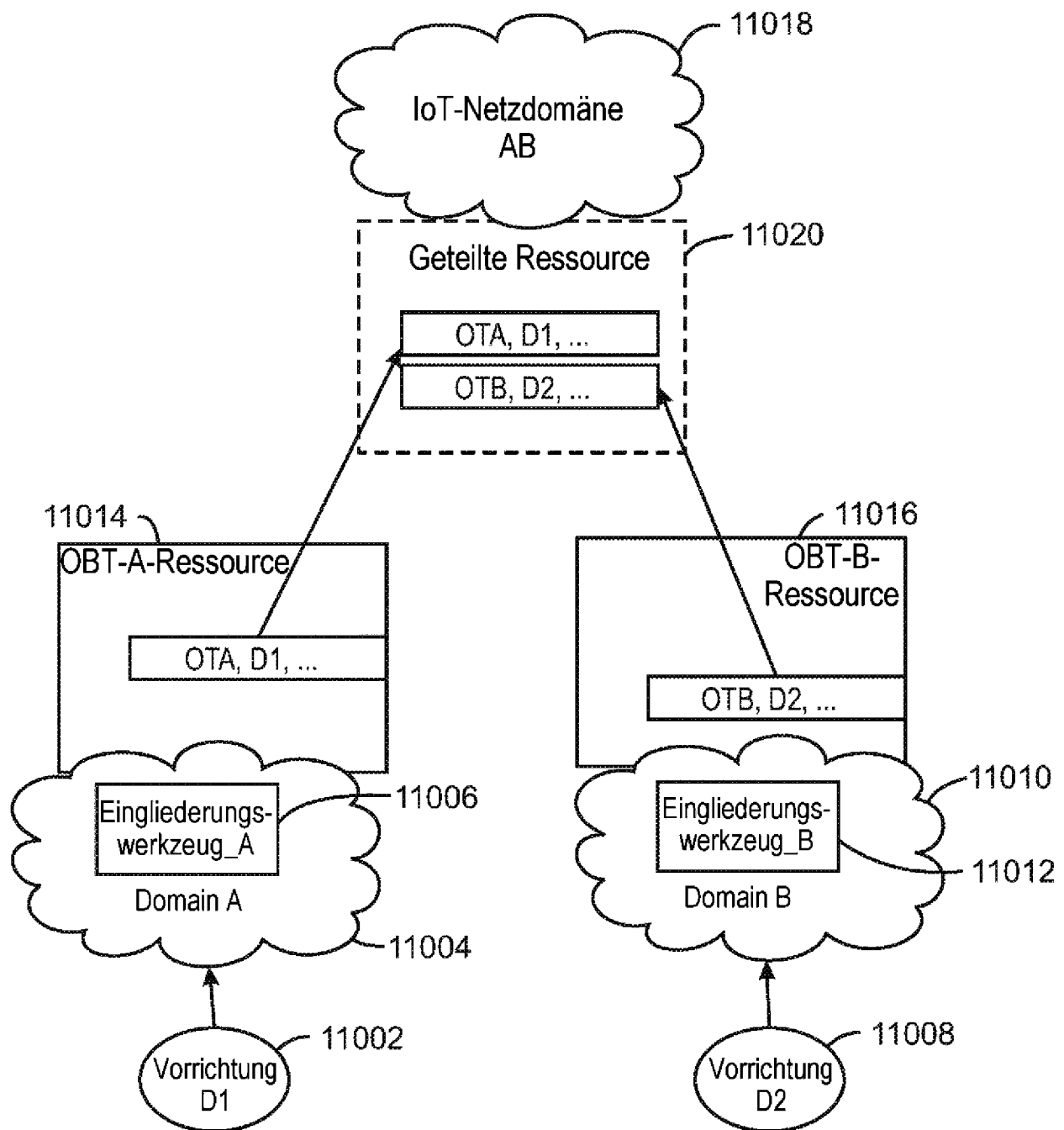
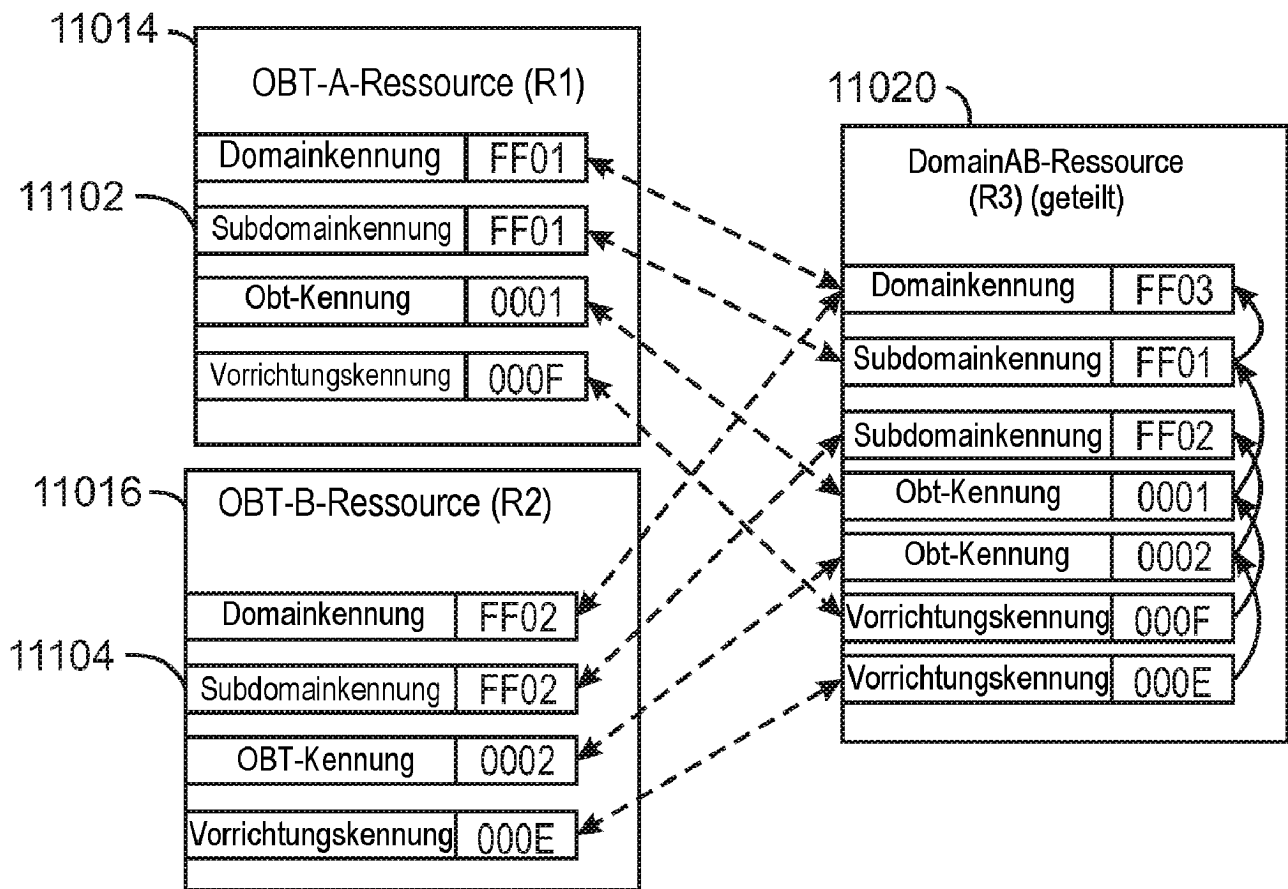


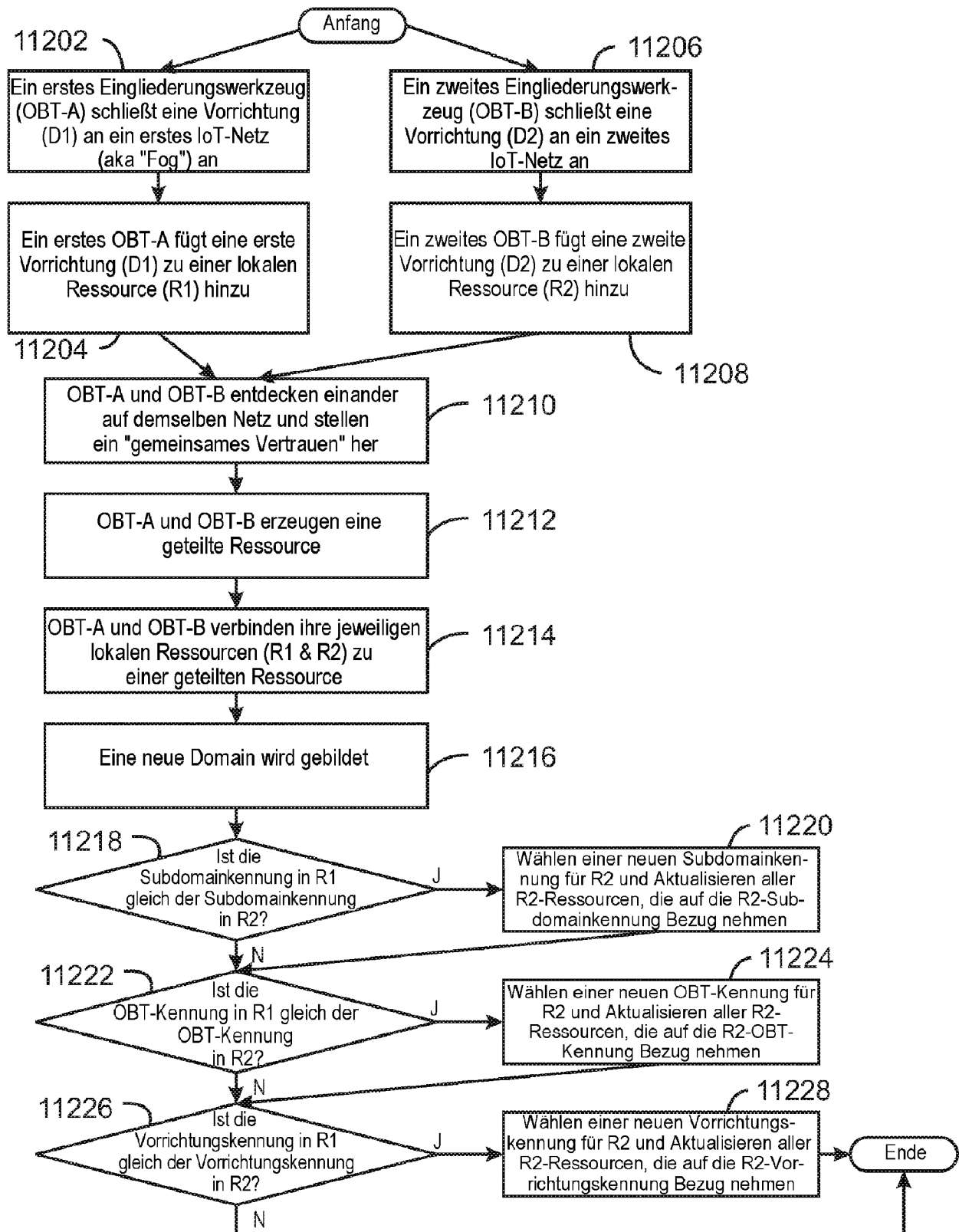
FIG. 109



11000
FIG. 110



11100
FIG. 111



11200
FIG. 112

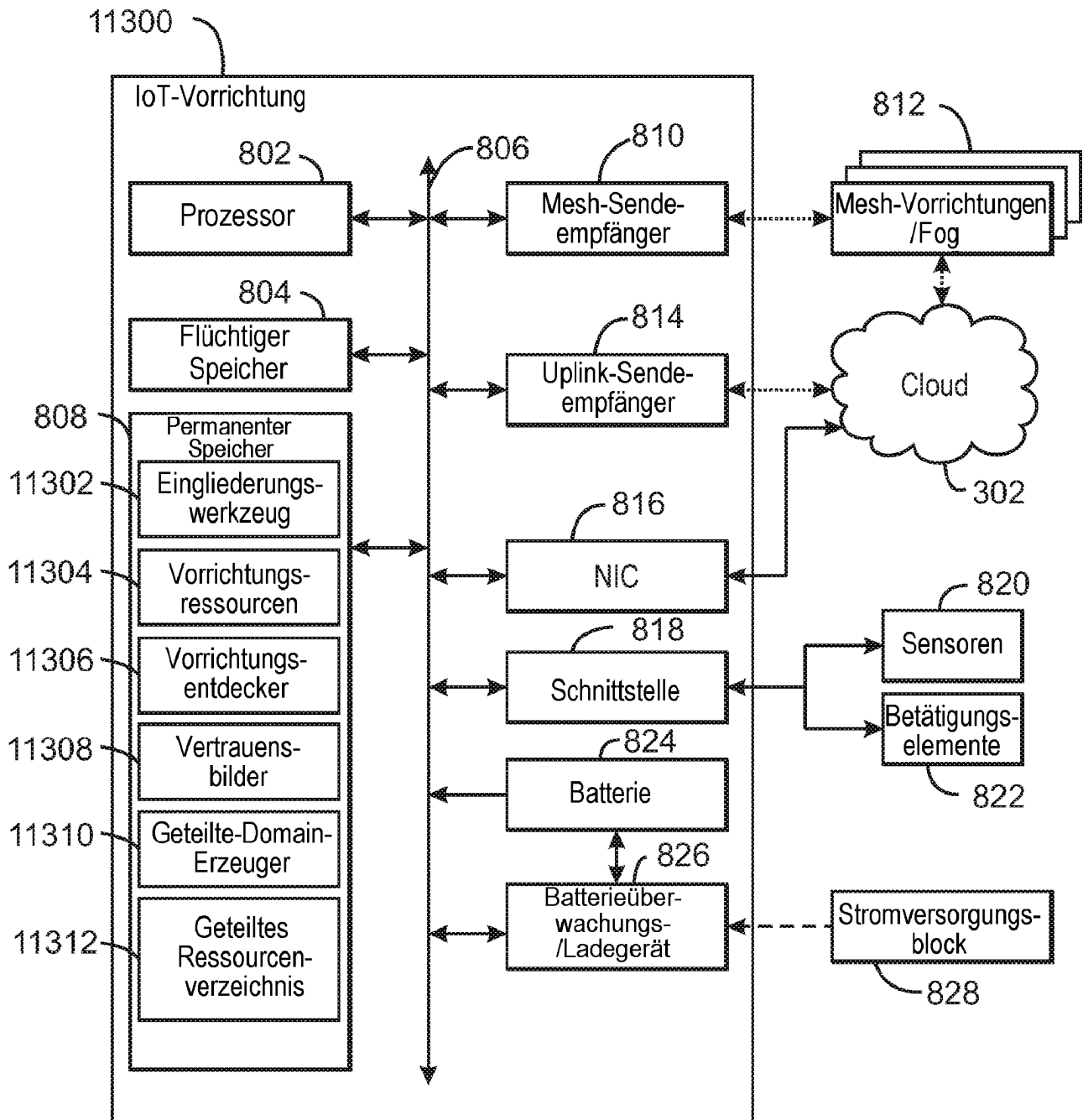


FIG. 113

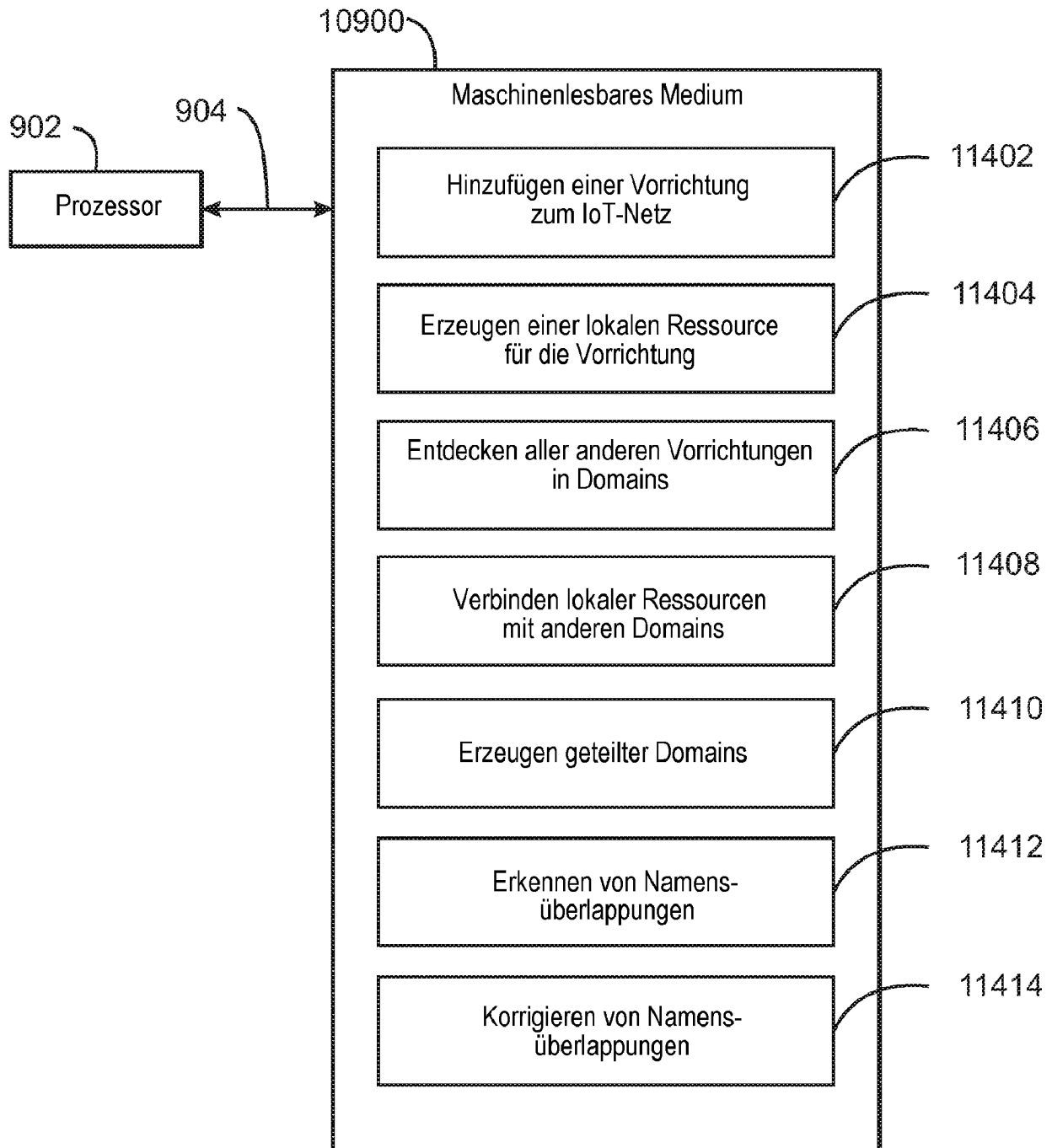
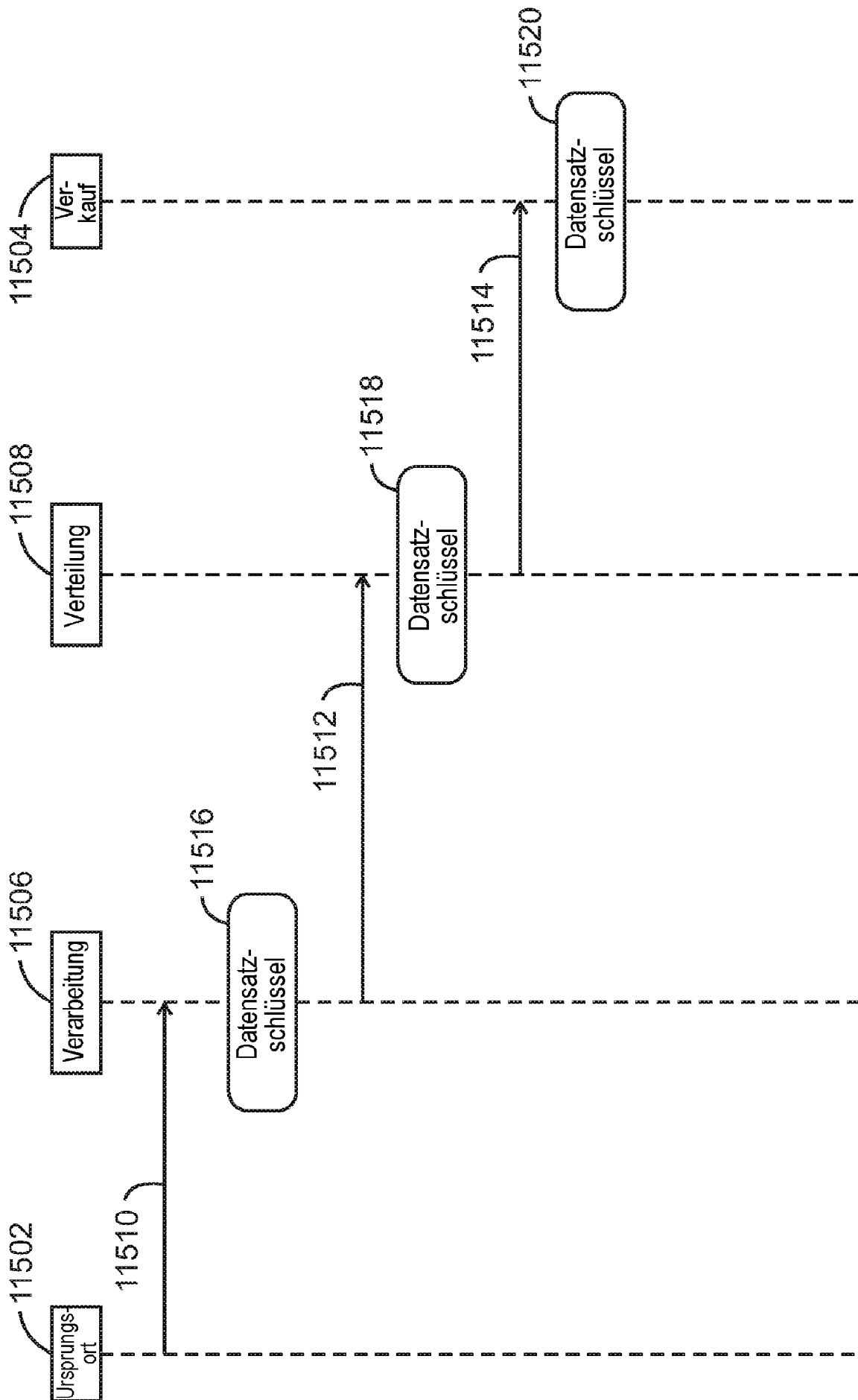
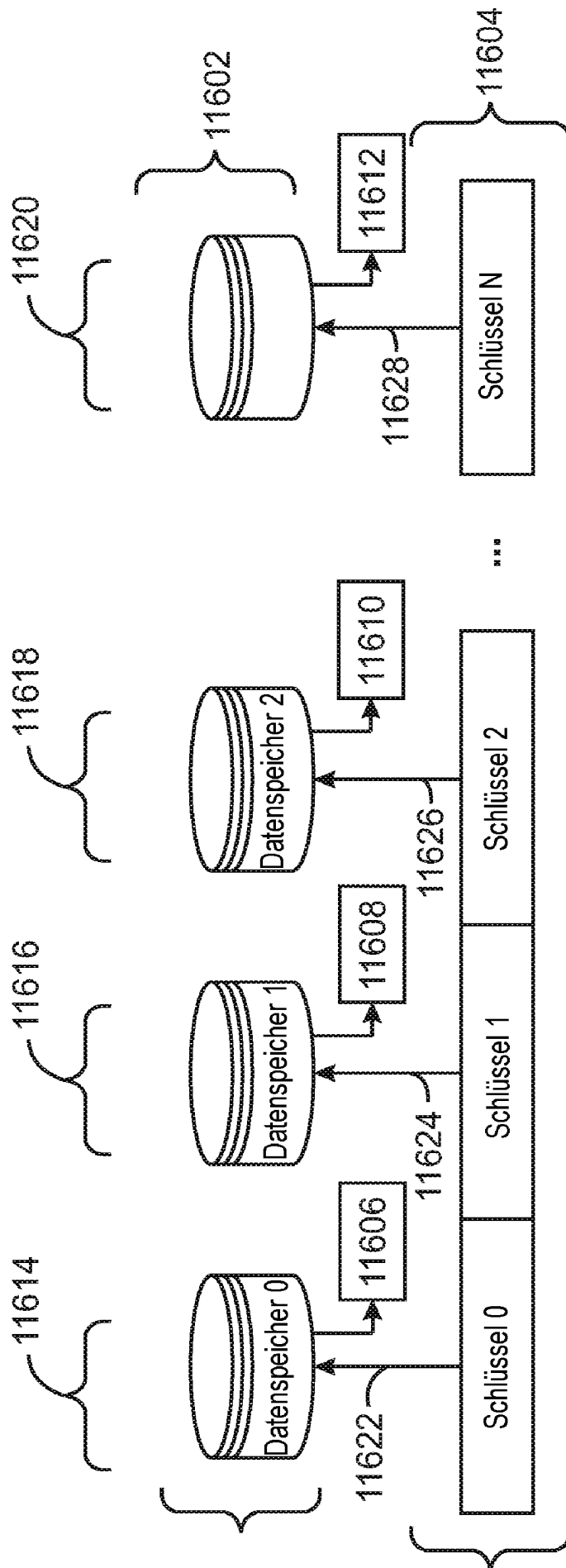


FIG. 114



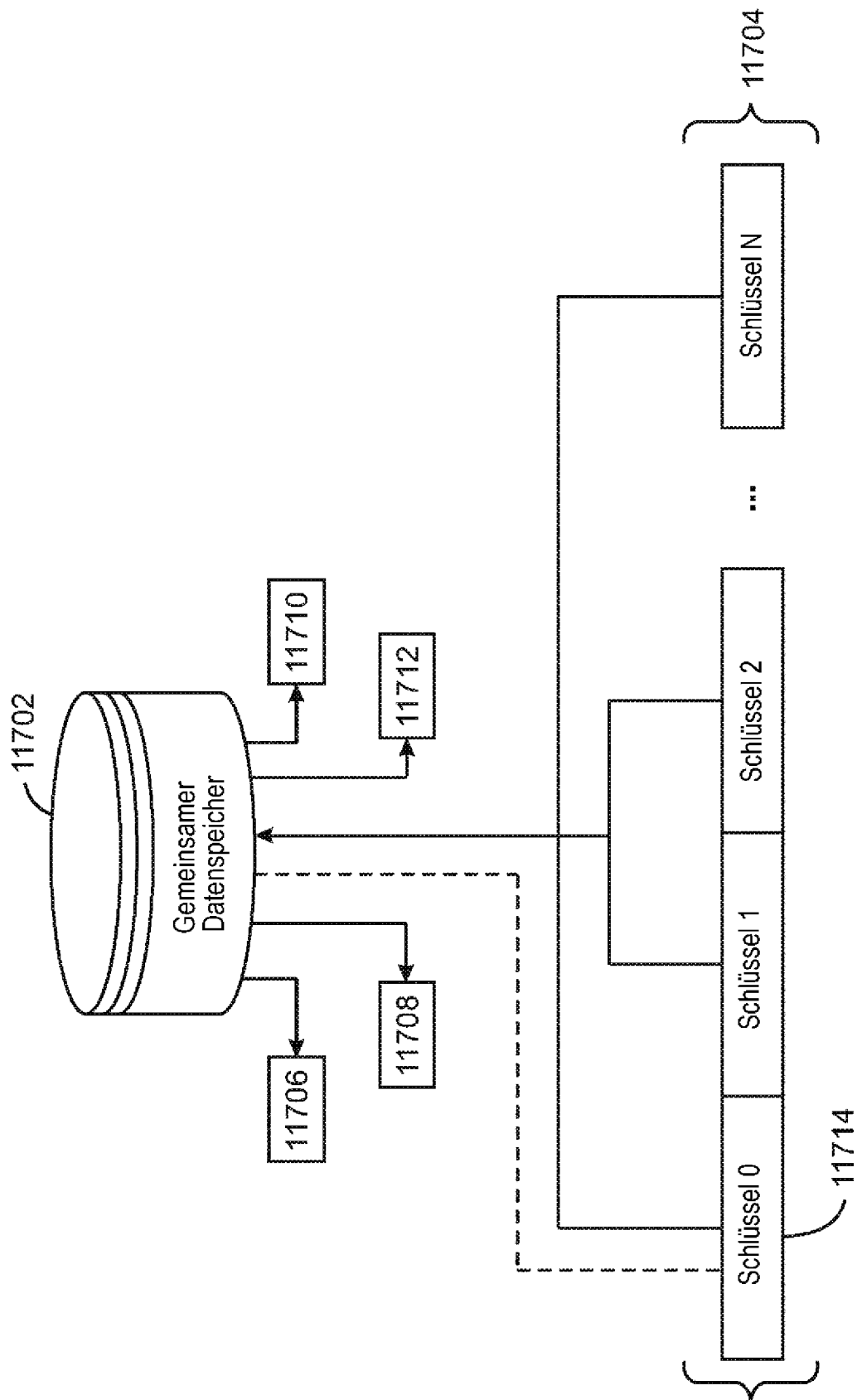
11500

FIG. 115



11600

FIG. 116



11700

FIG. 117

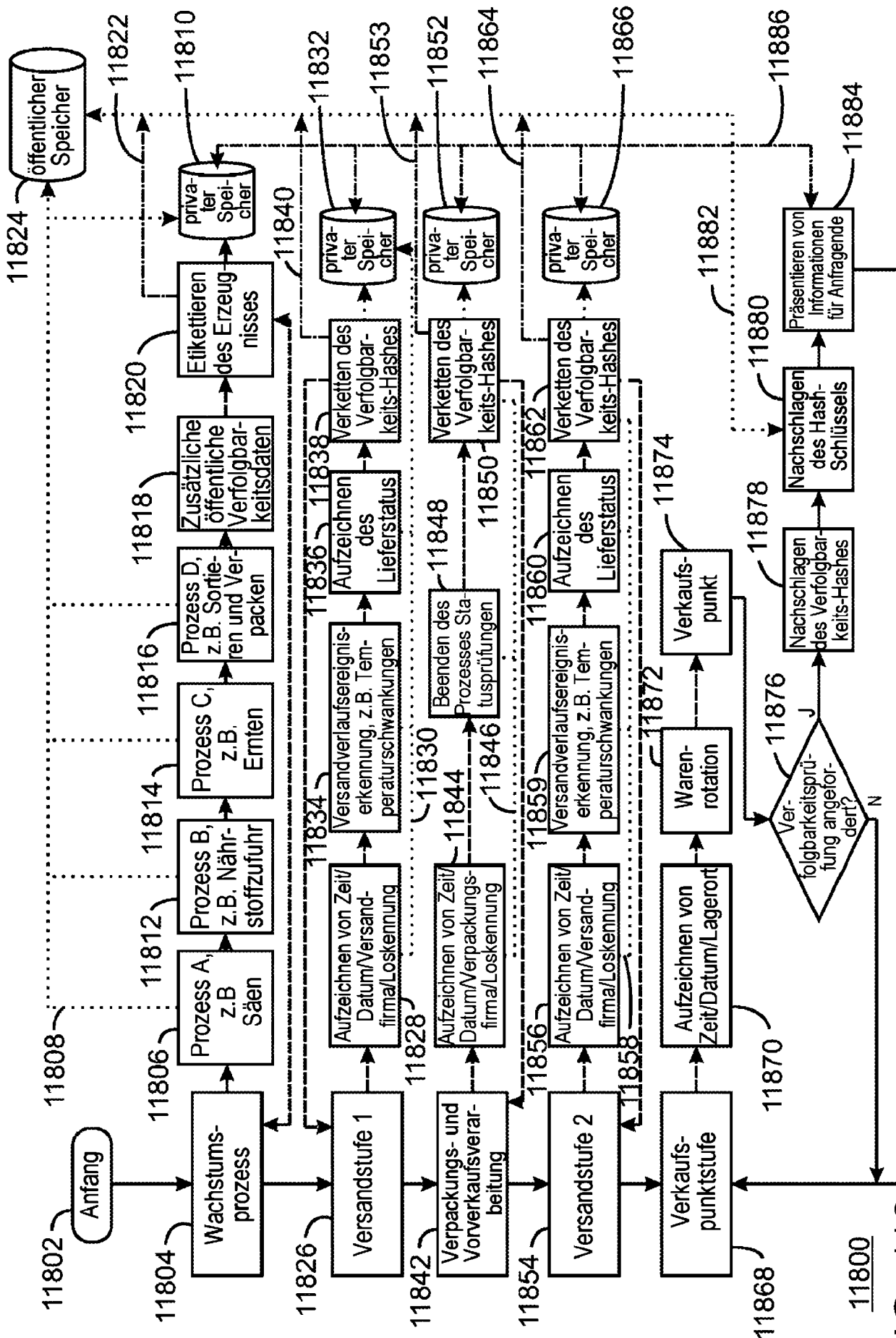


FIG. 118

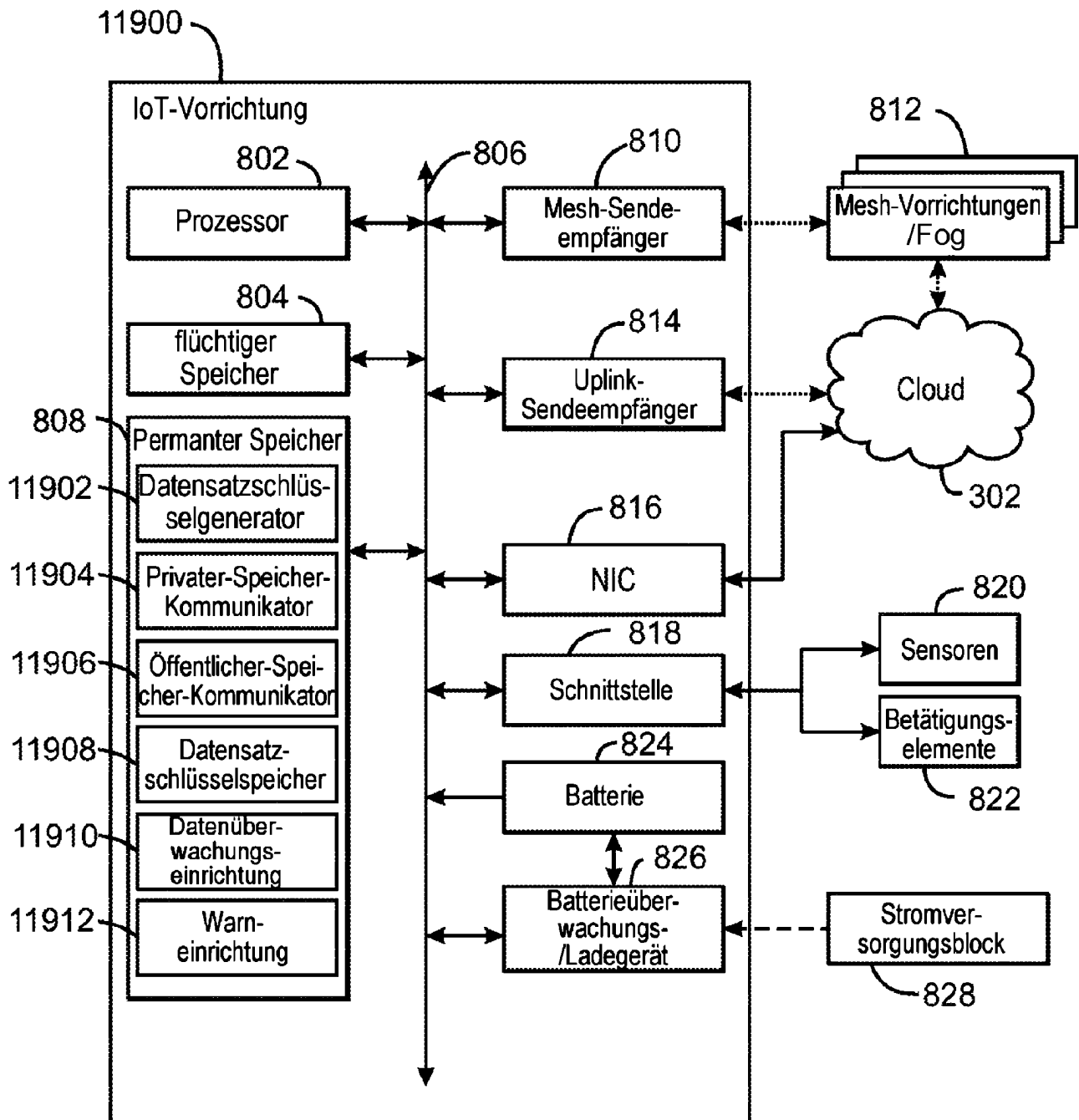


FIG. 119

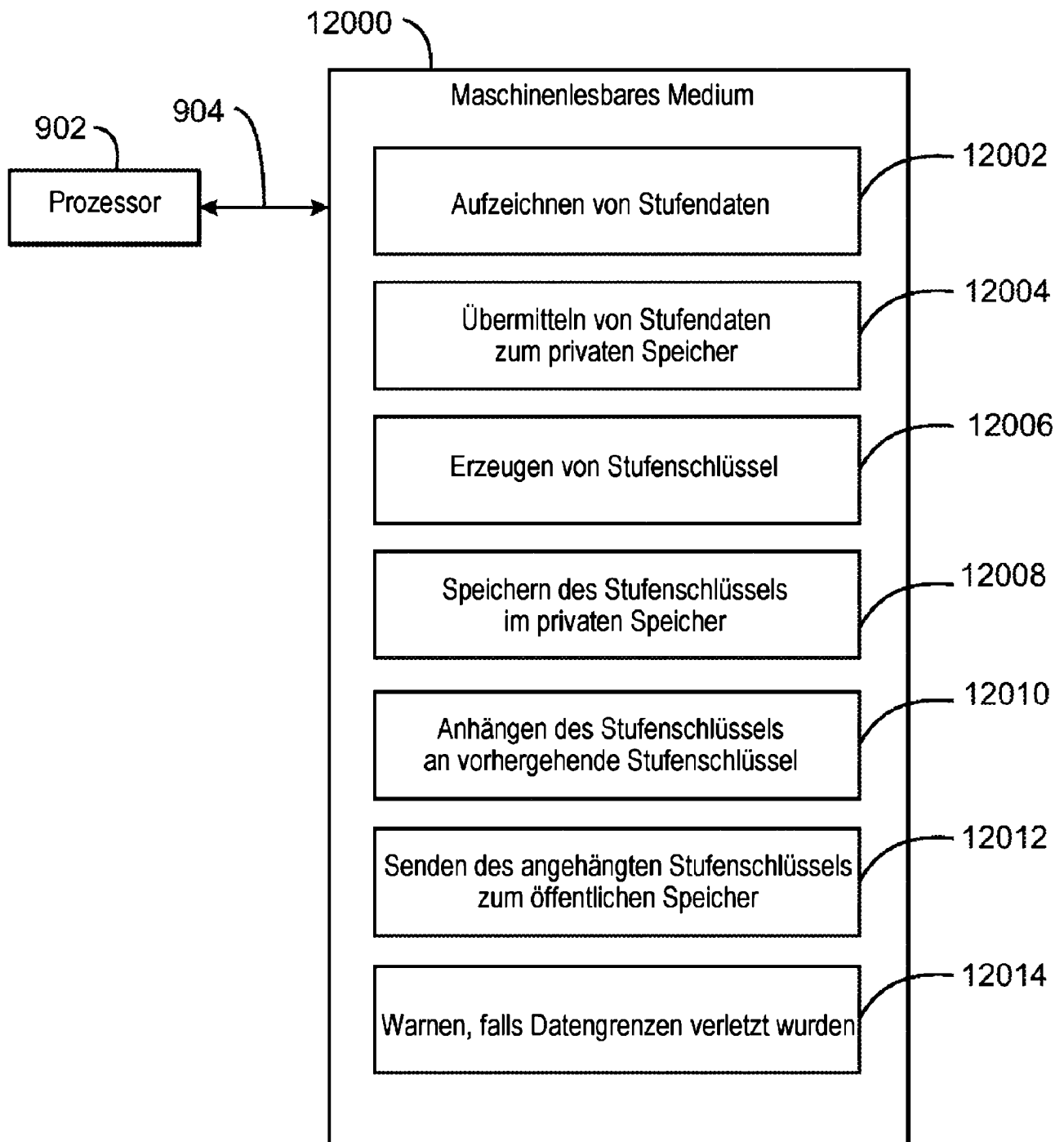
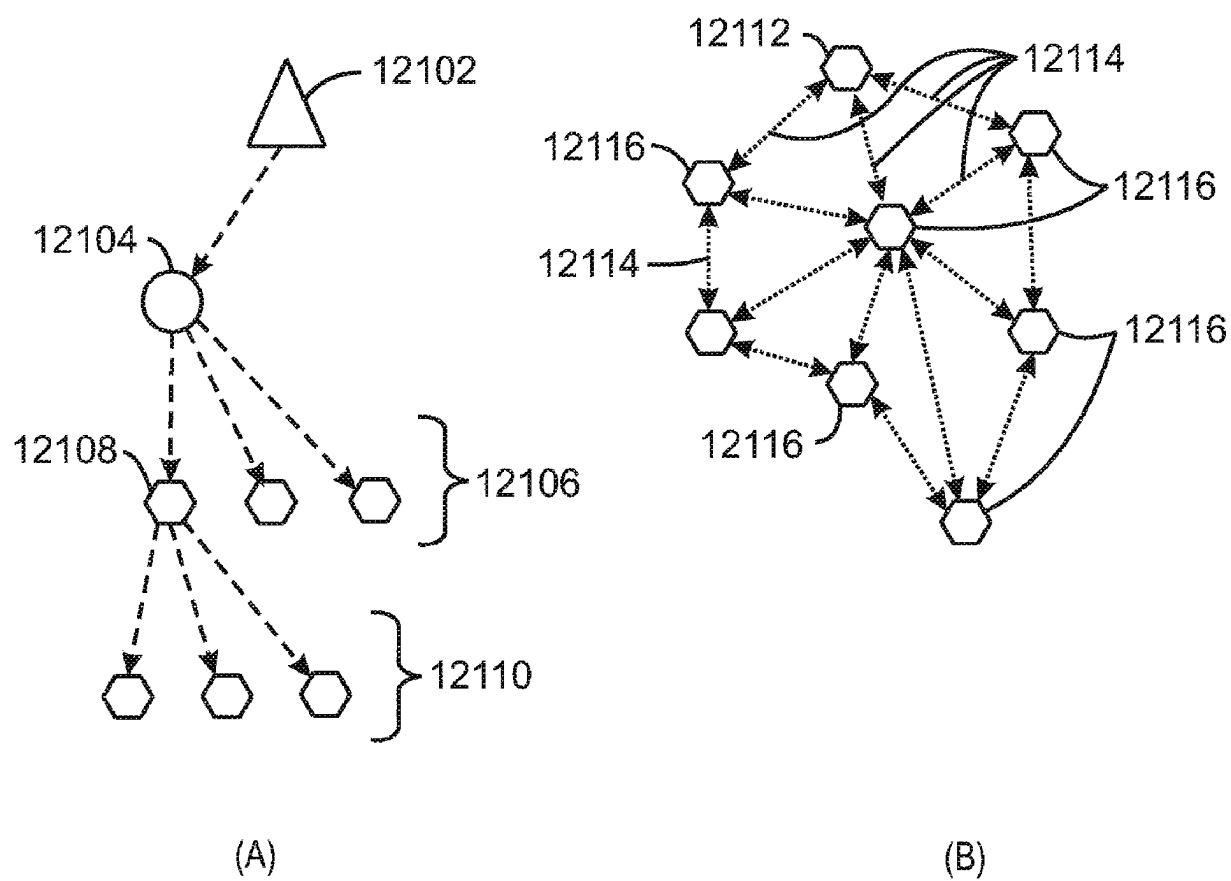
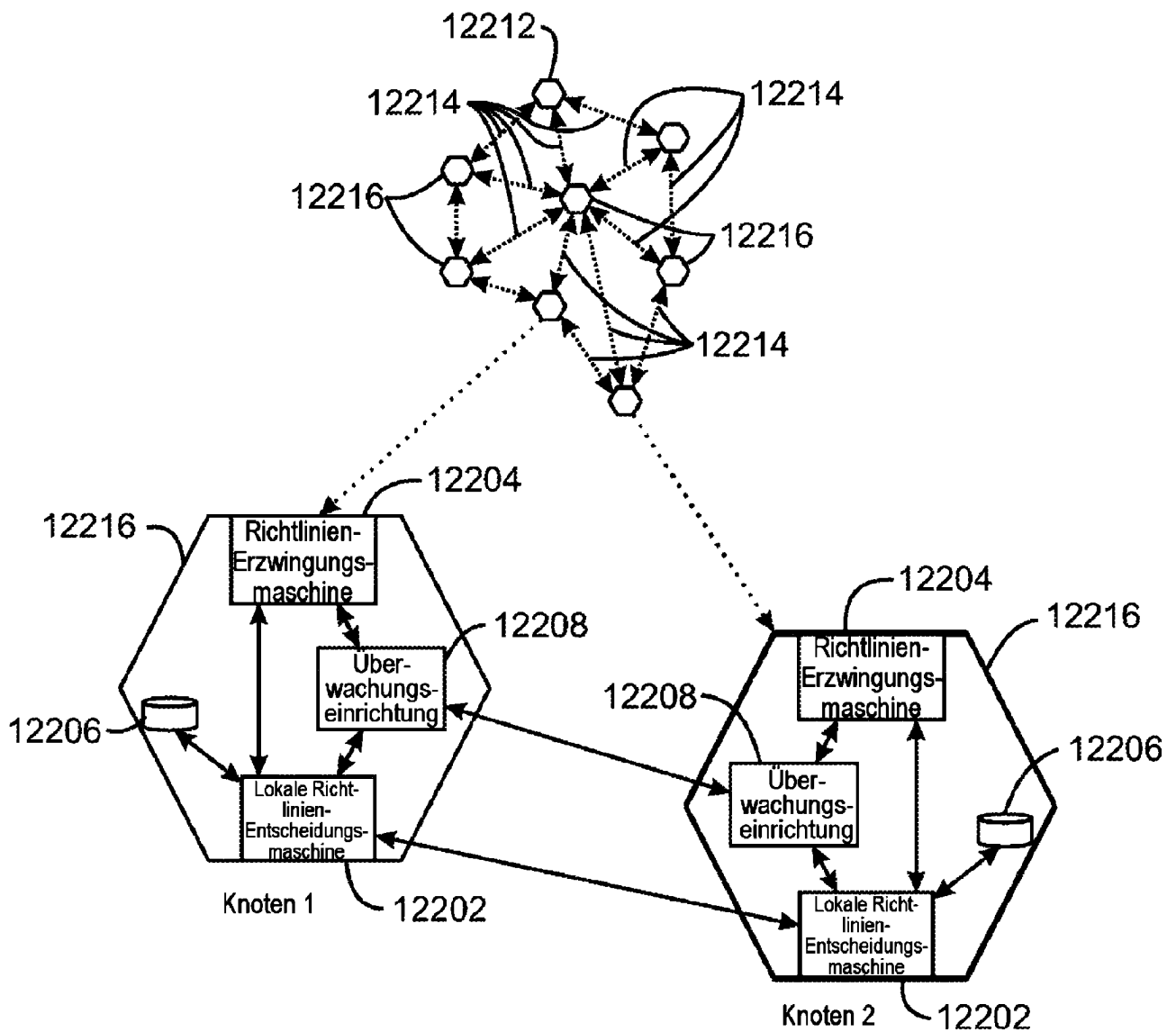


FIG. 120



12100
FIG. 121



12200
FIG. 122

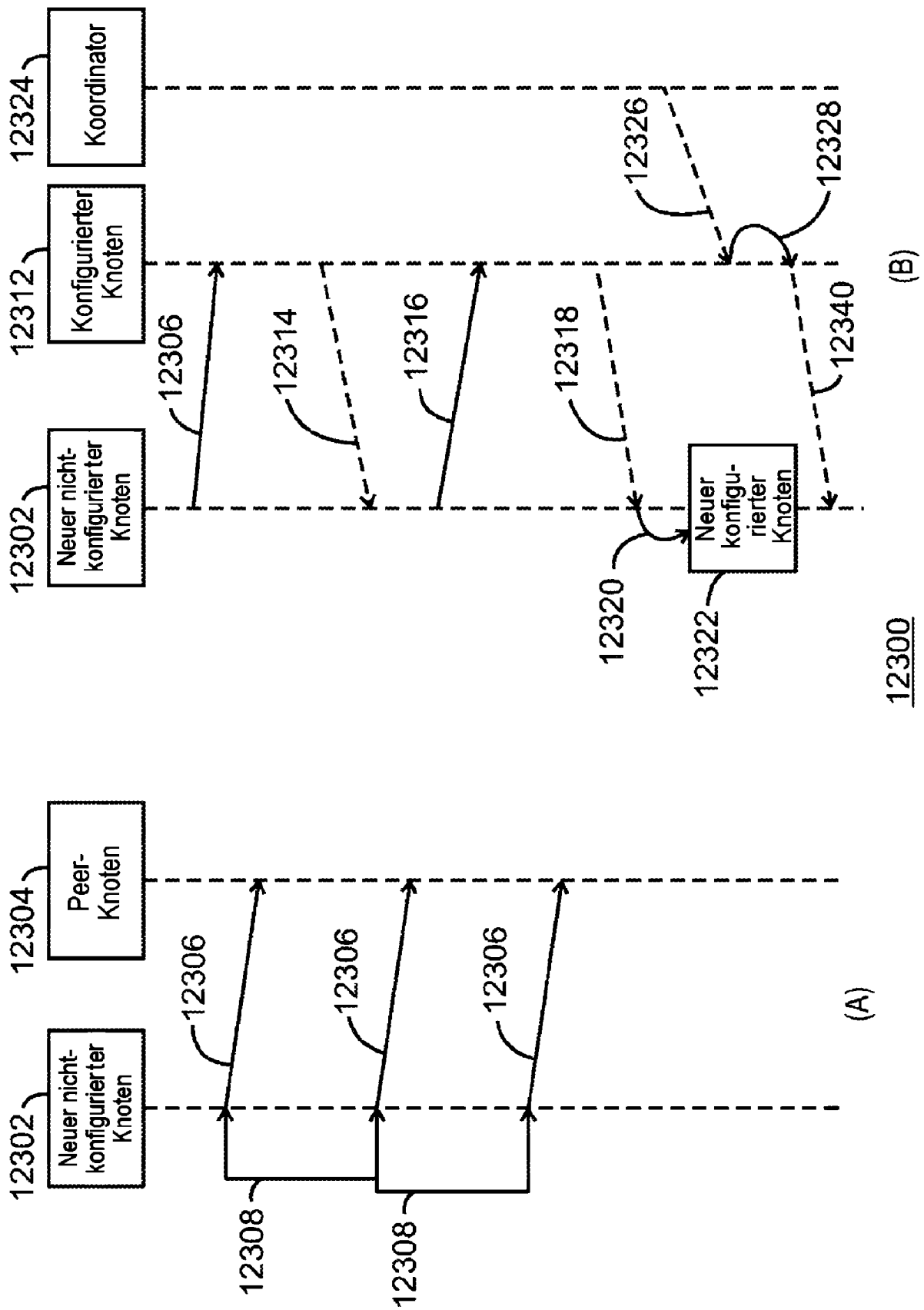
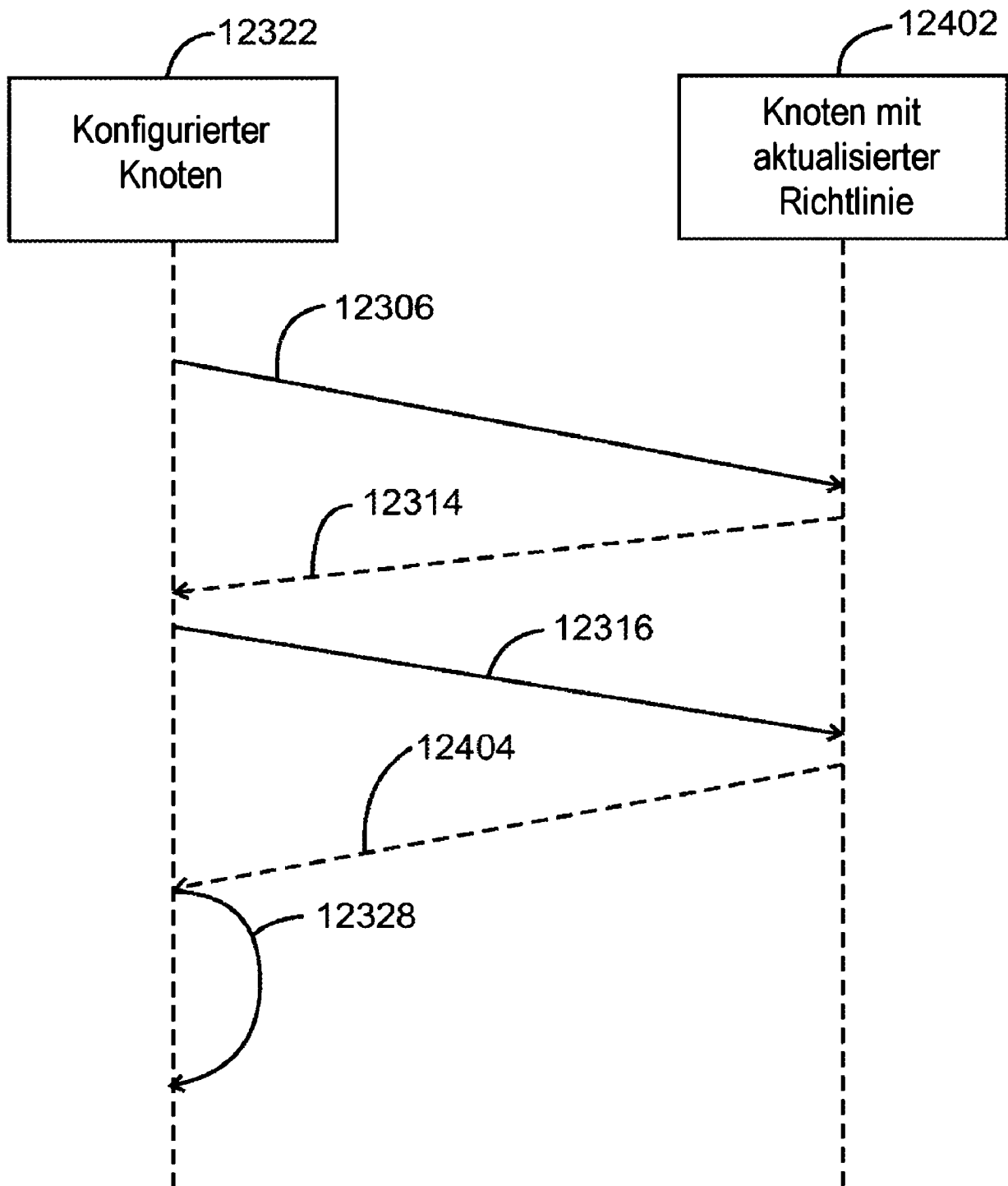
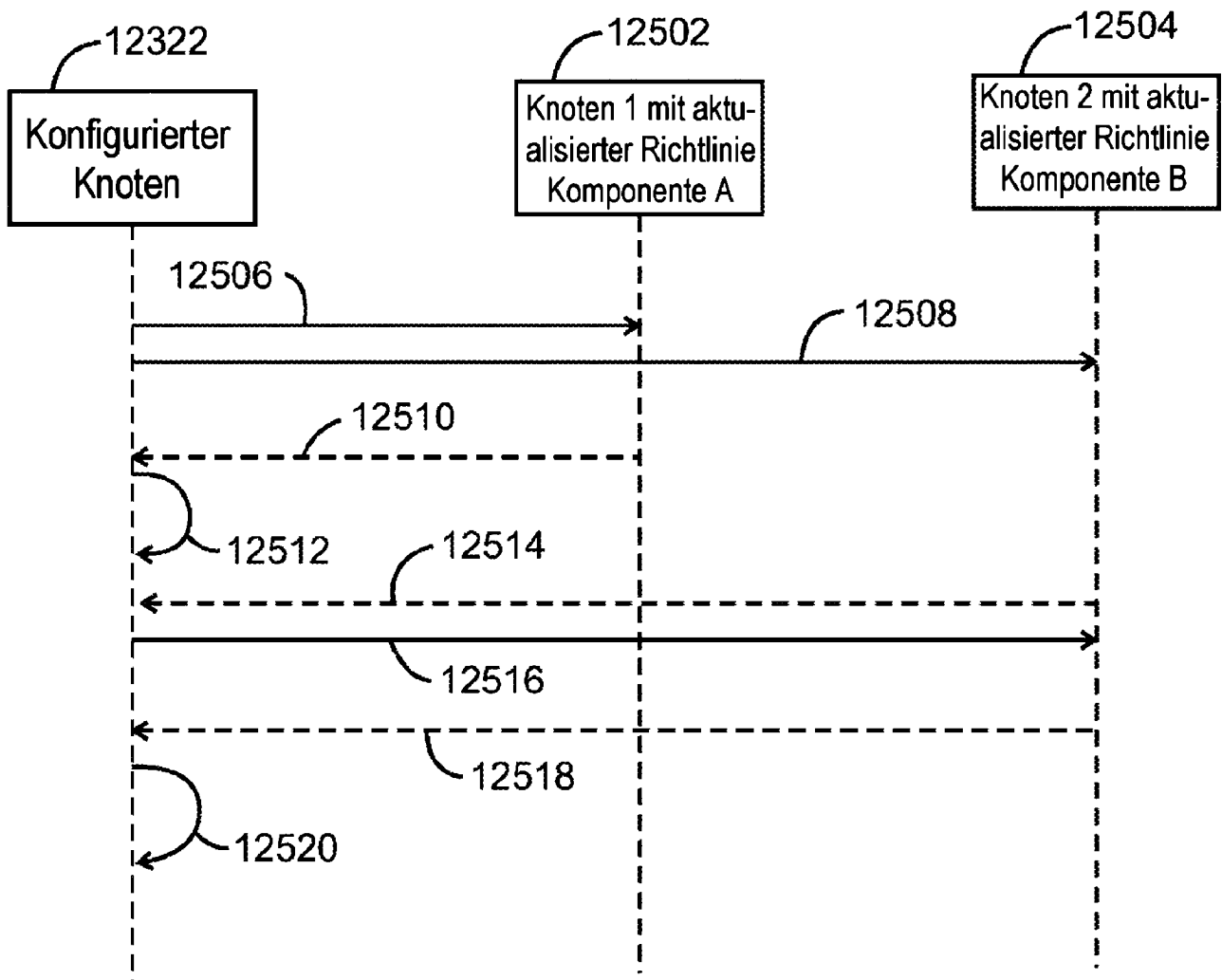


FIG. 123



12400

FIG. 124



12500

FIG. 125

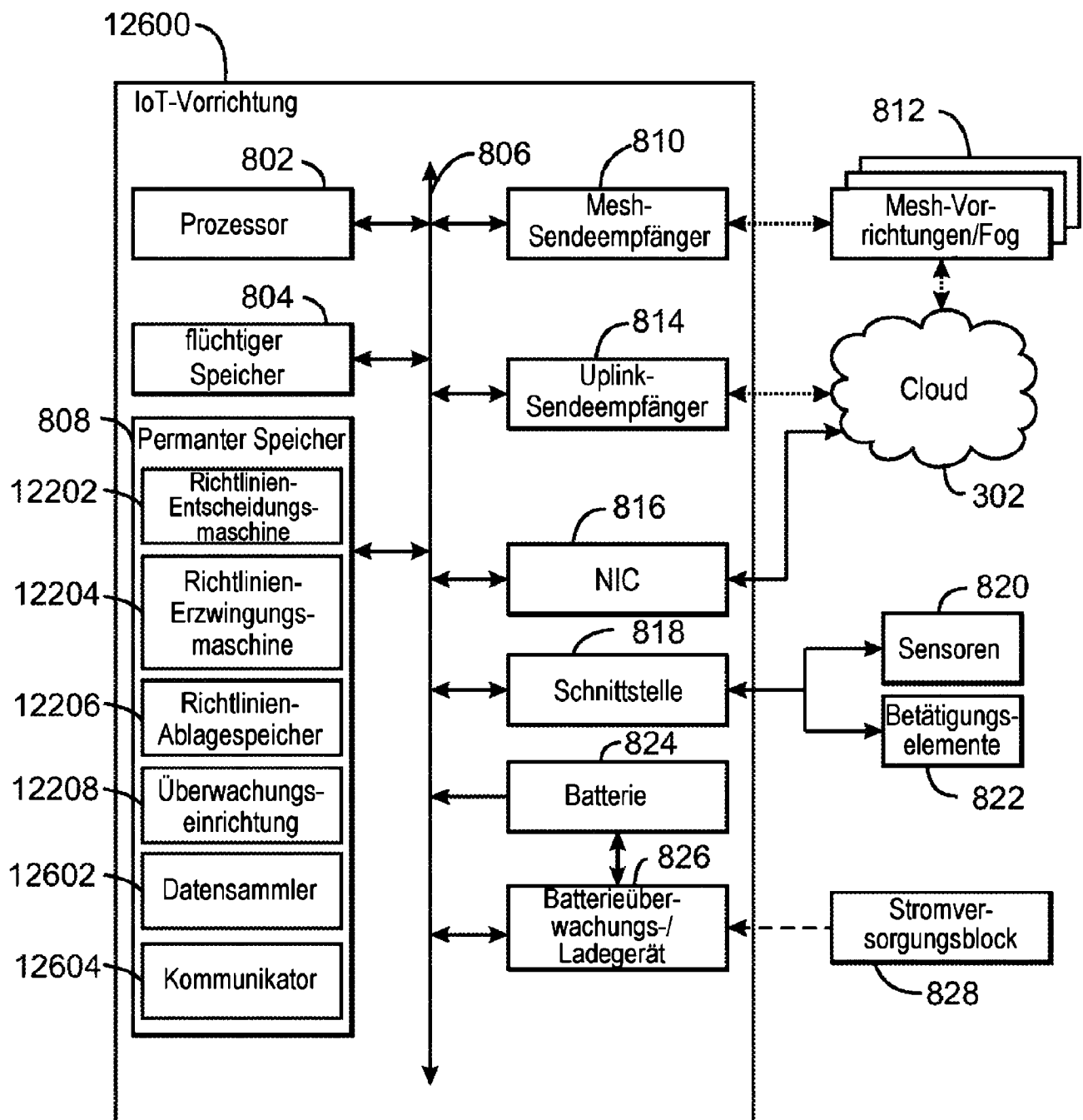


FIG. 126

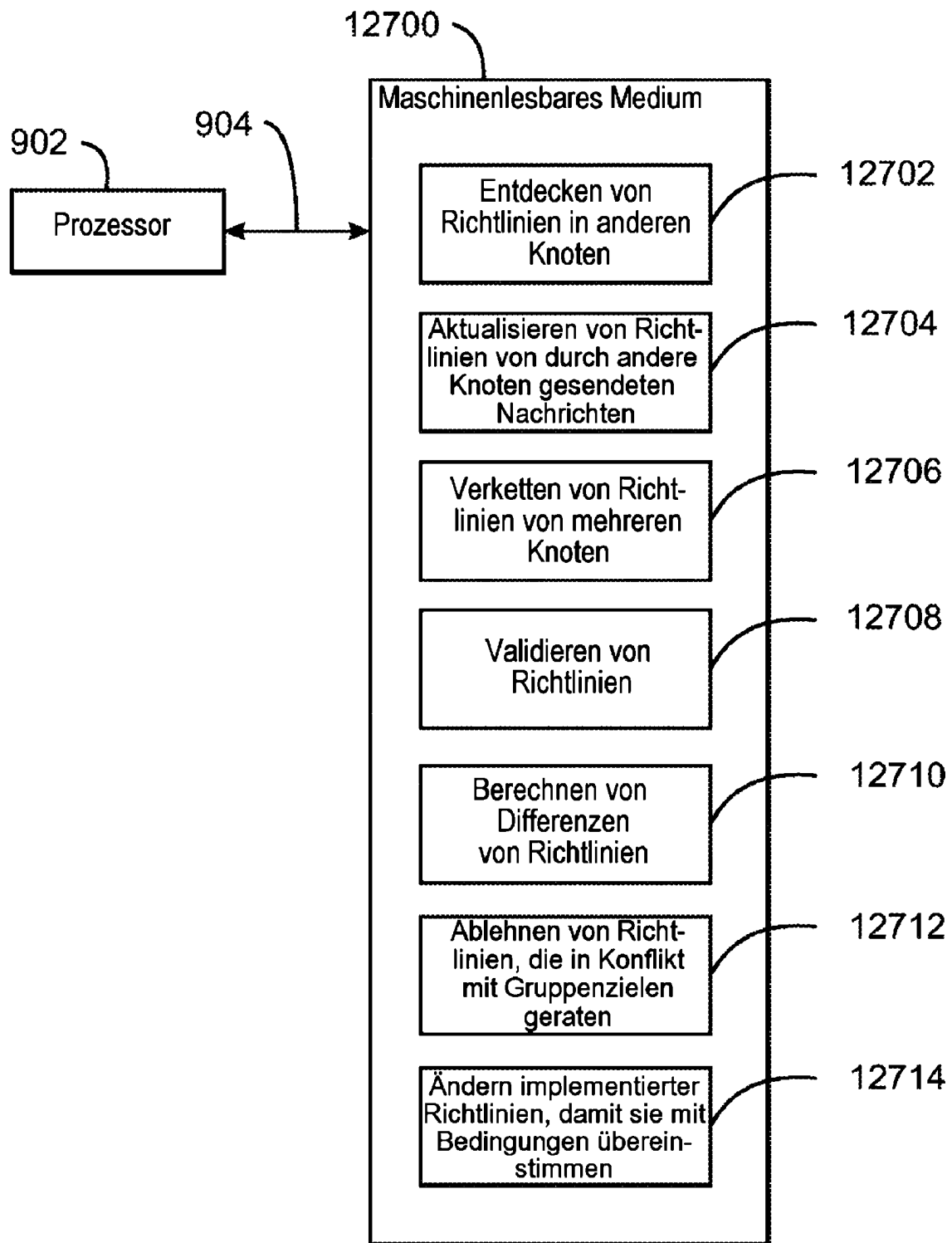
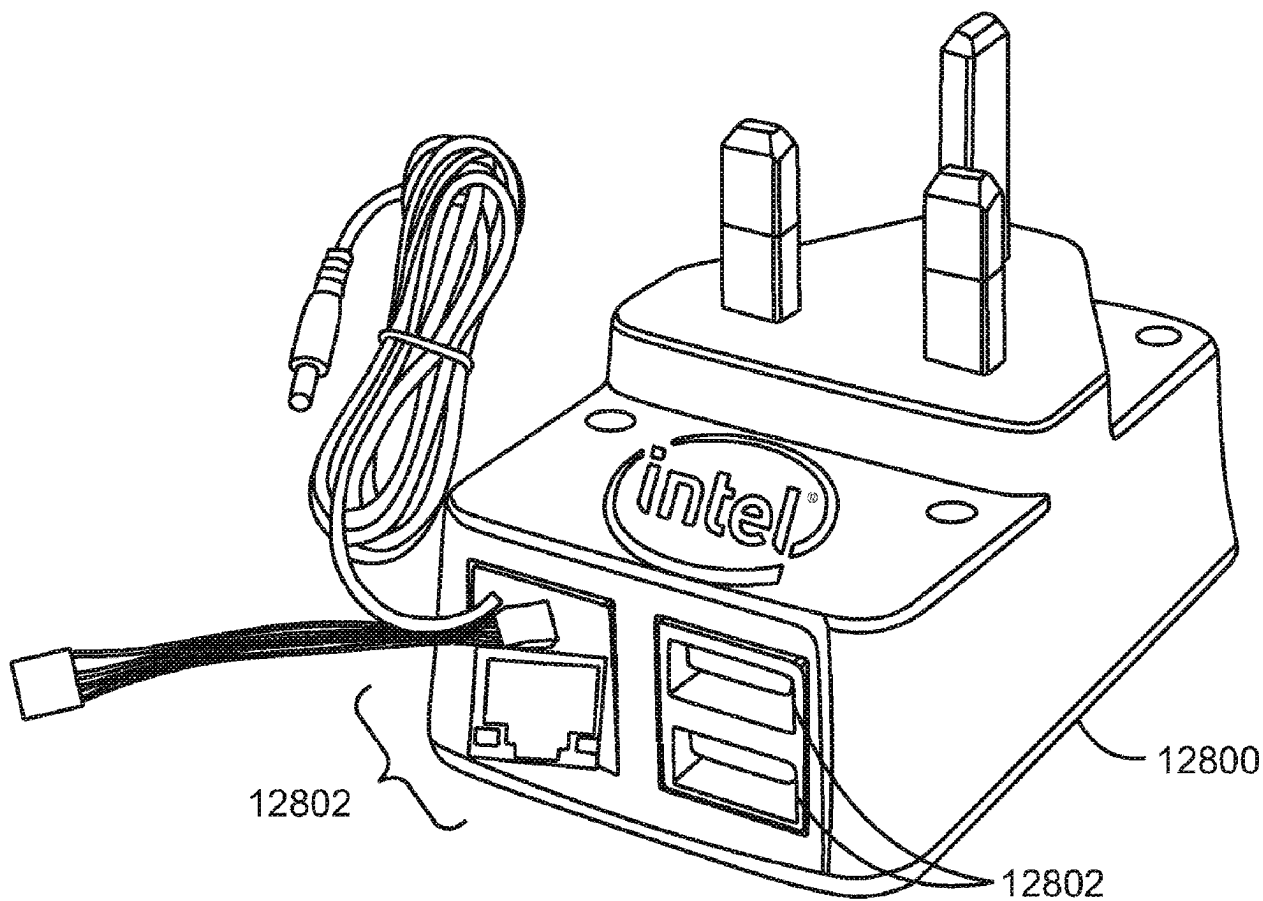
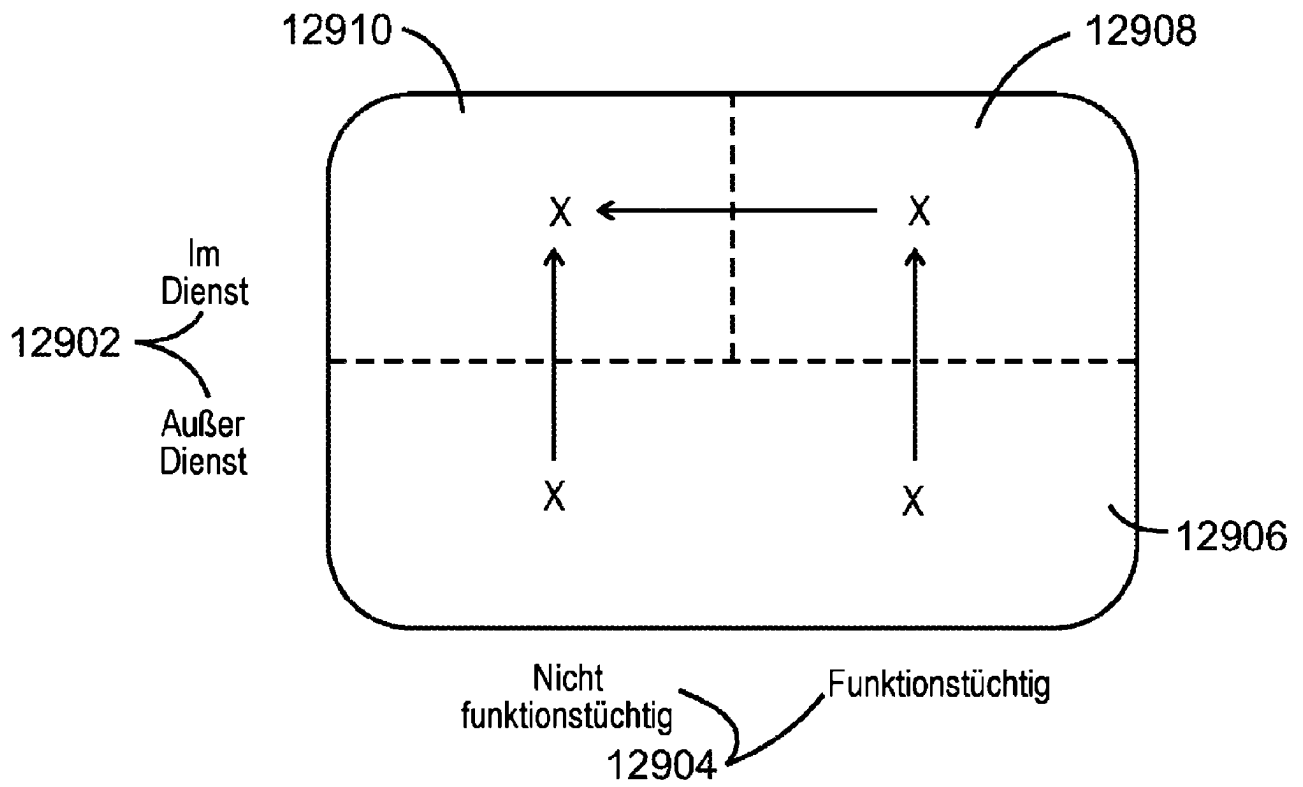


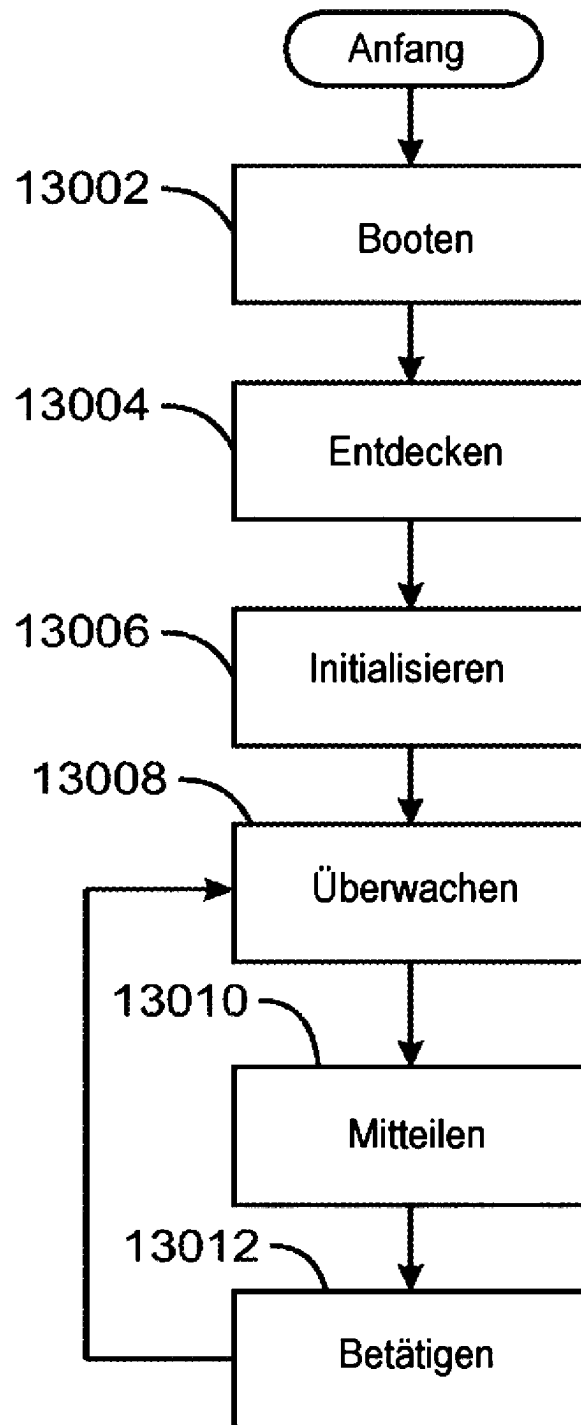
FIG. 127



12800
FIG. 128



12900
FIG. 129



13000
FIG. 130

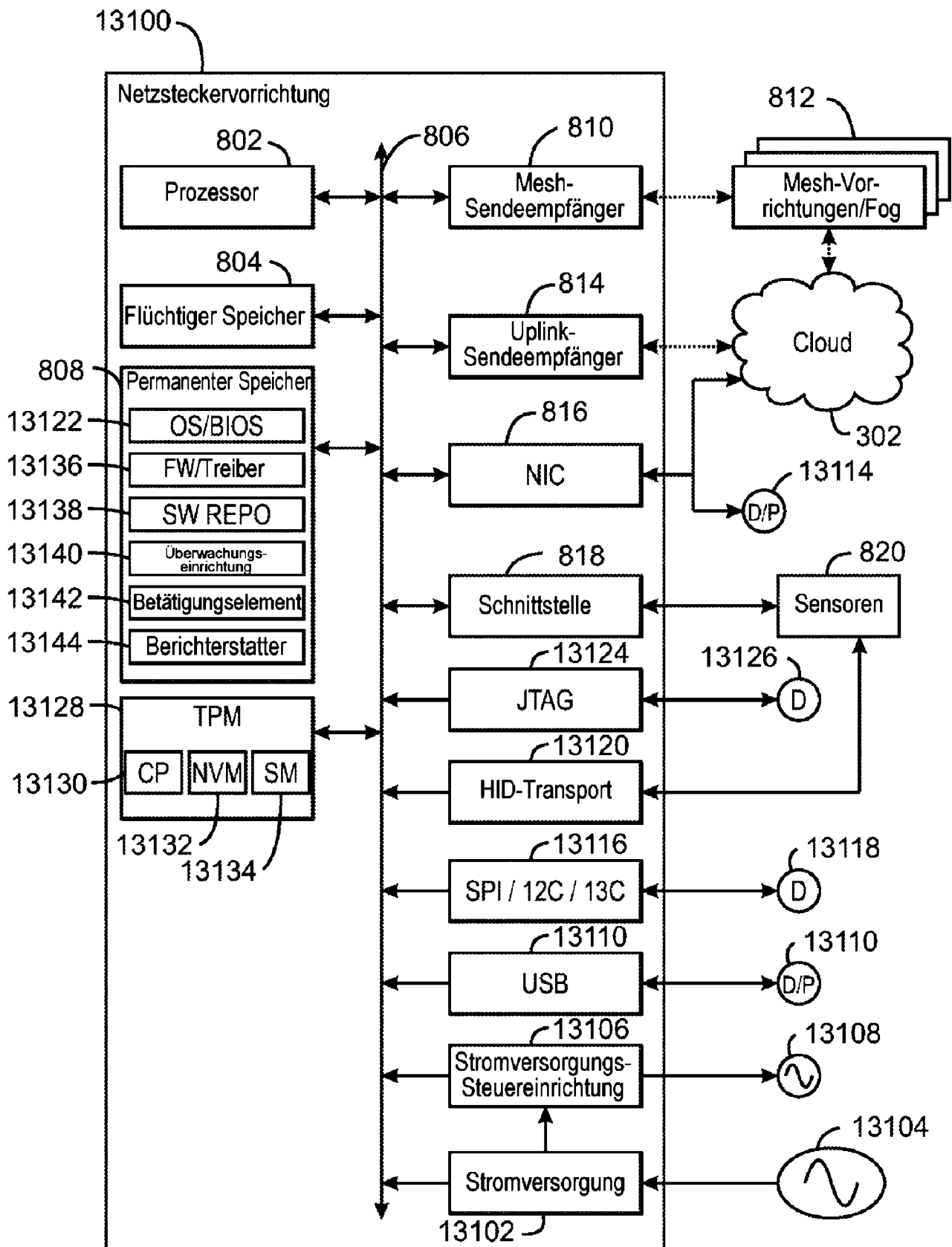


FIG. 131

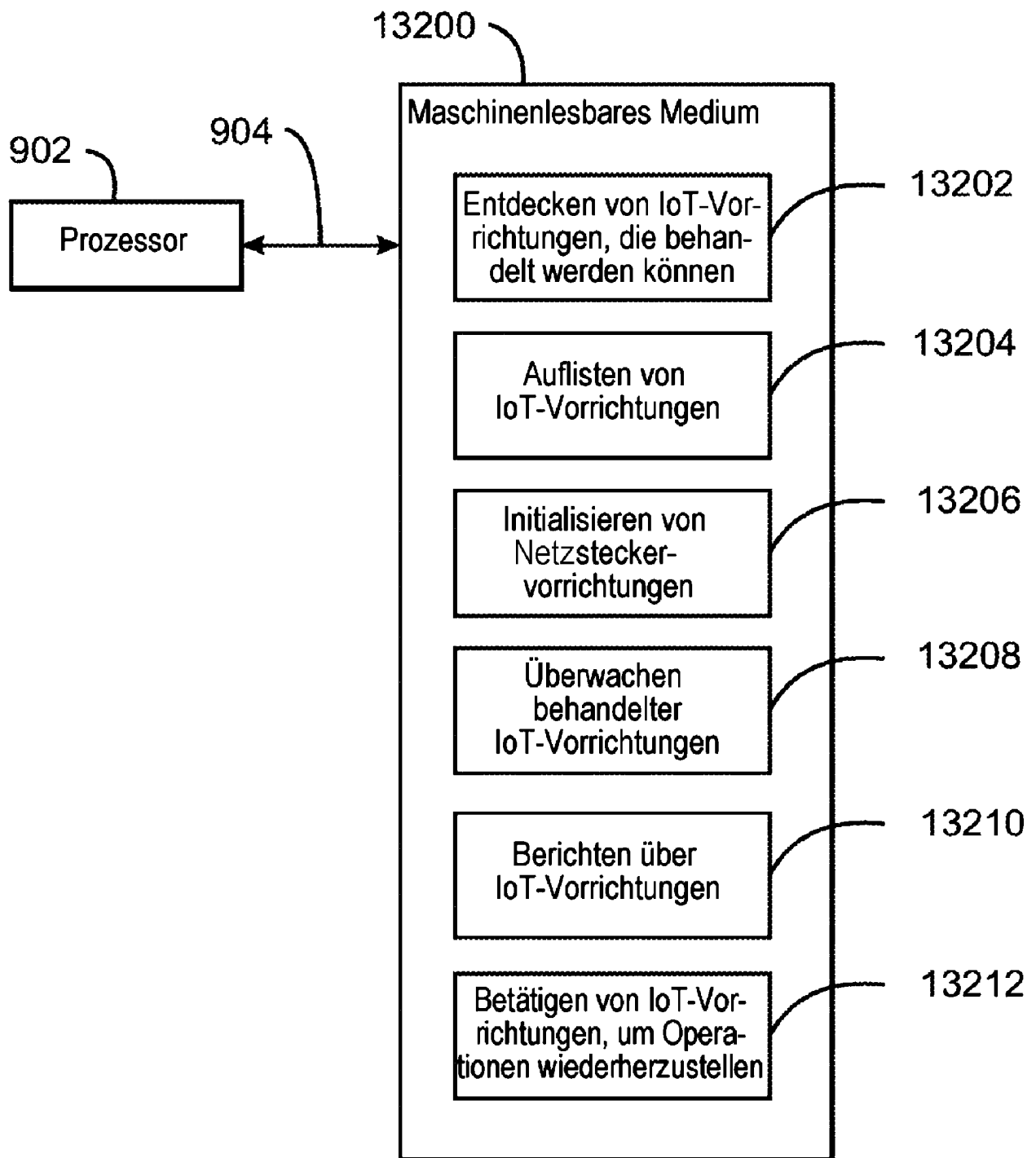
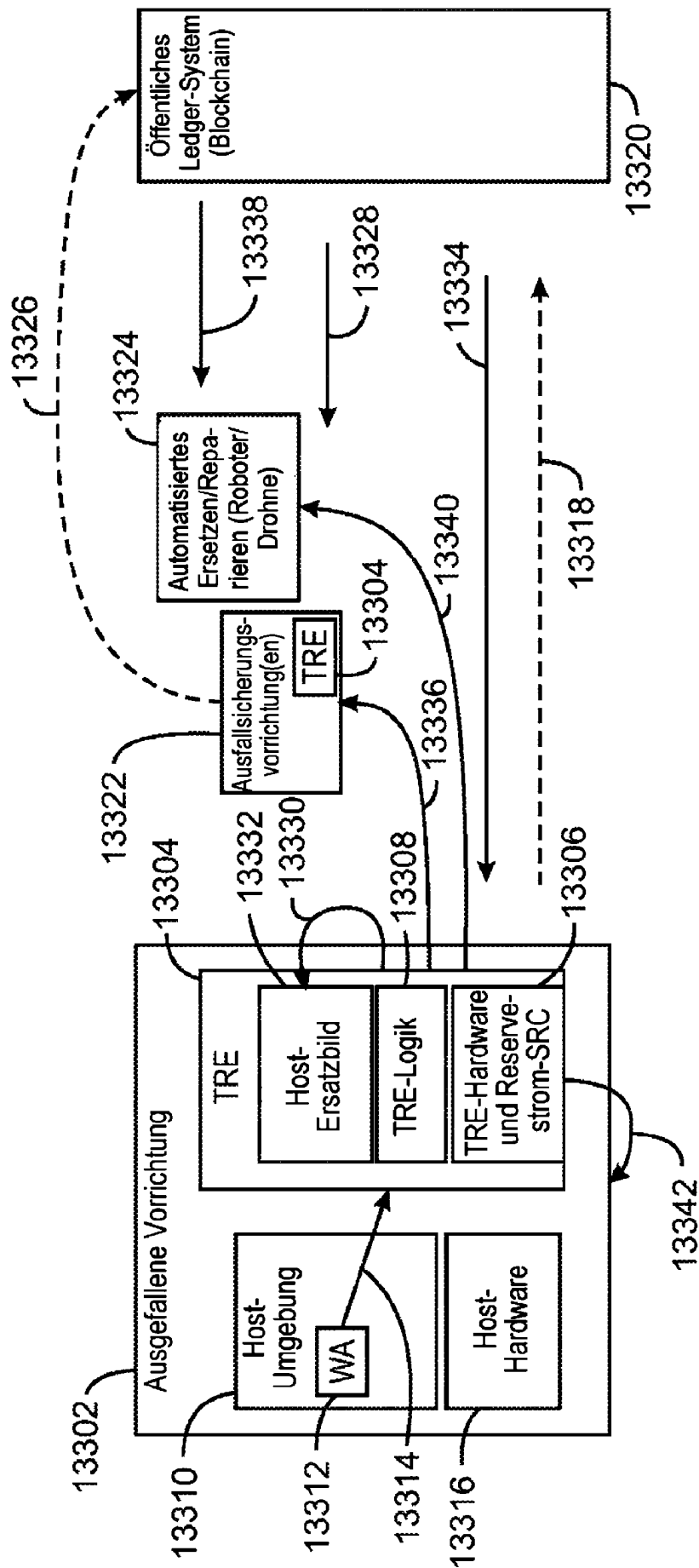
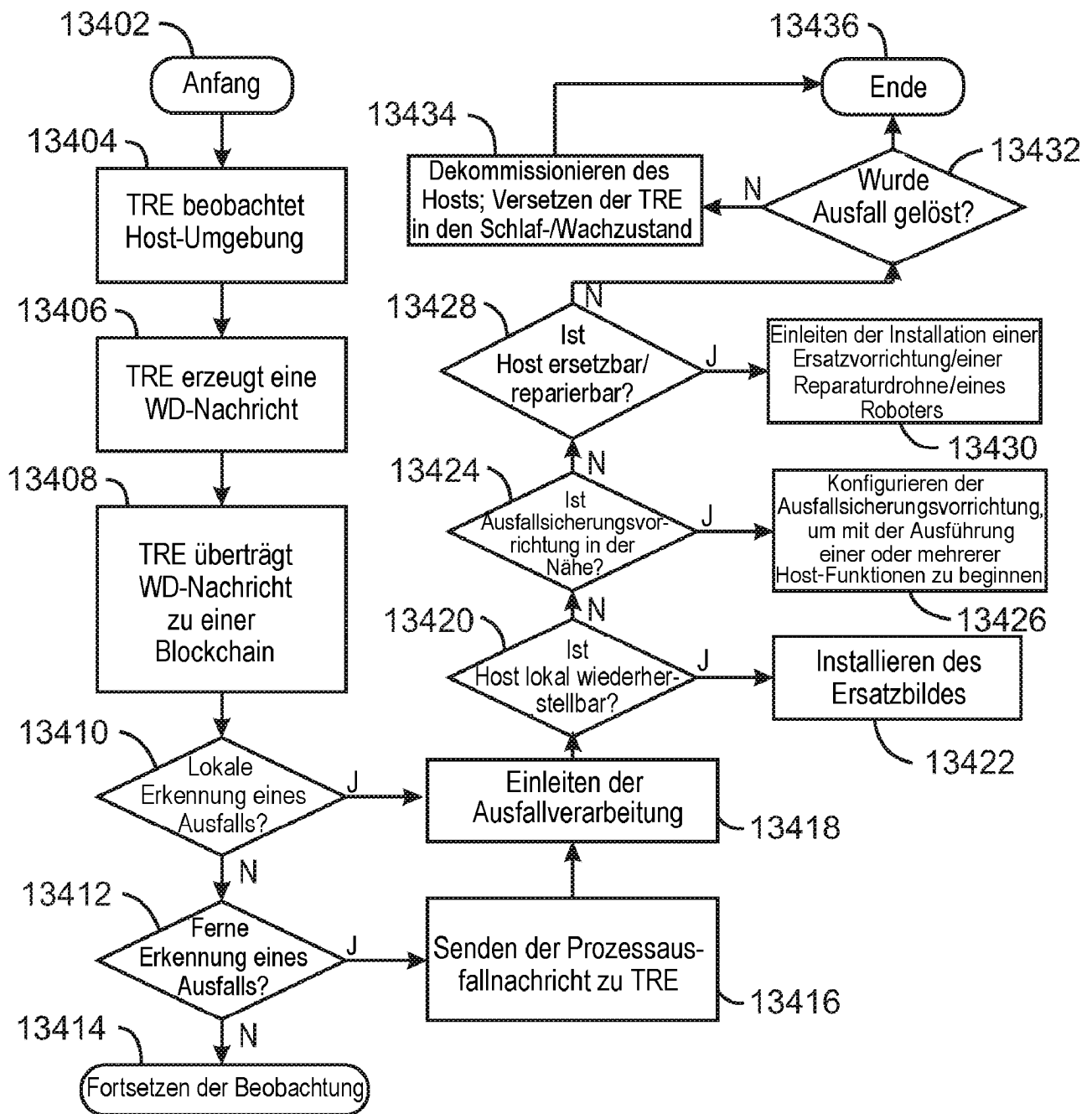


FIG. 132



13300

FIG. 133



13400
FIG. 134

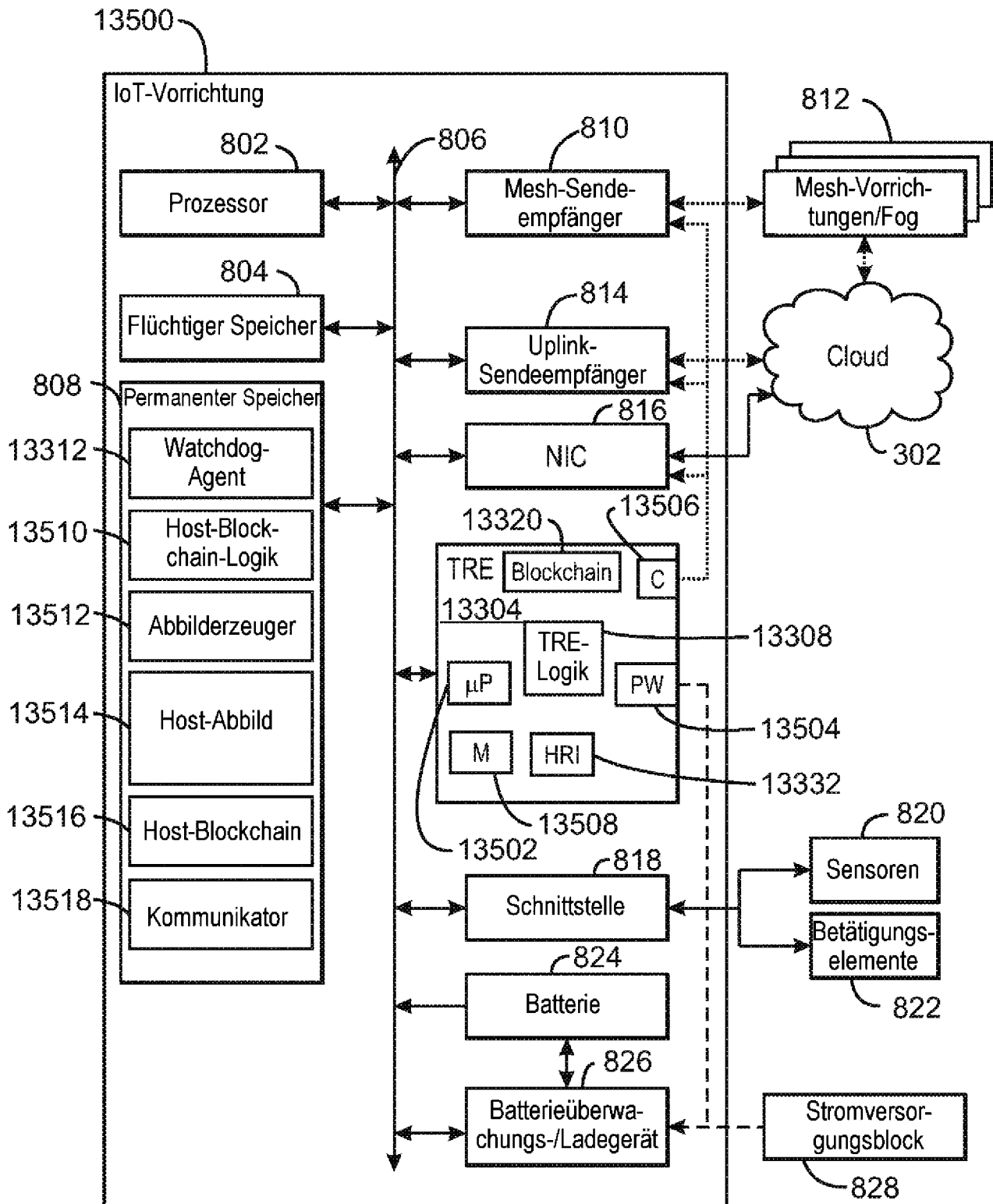


FIG. 135

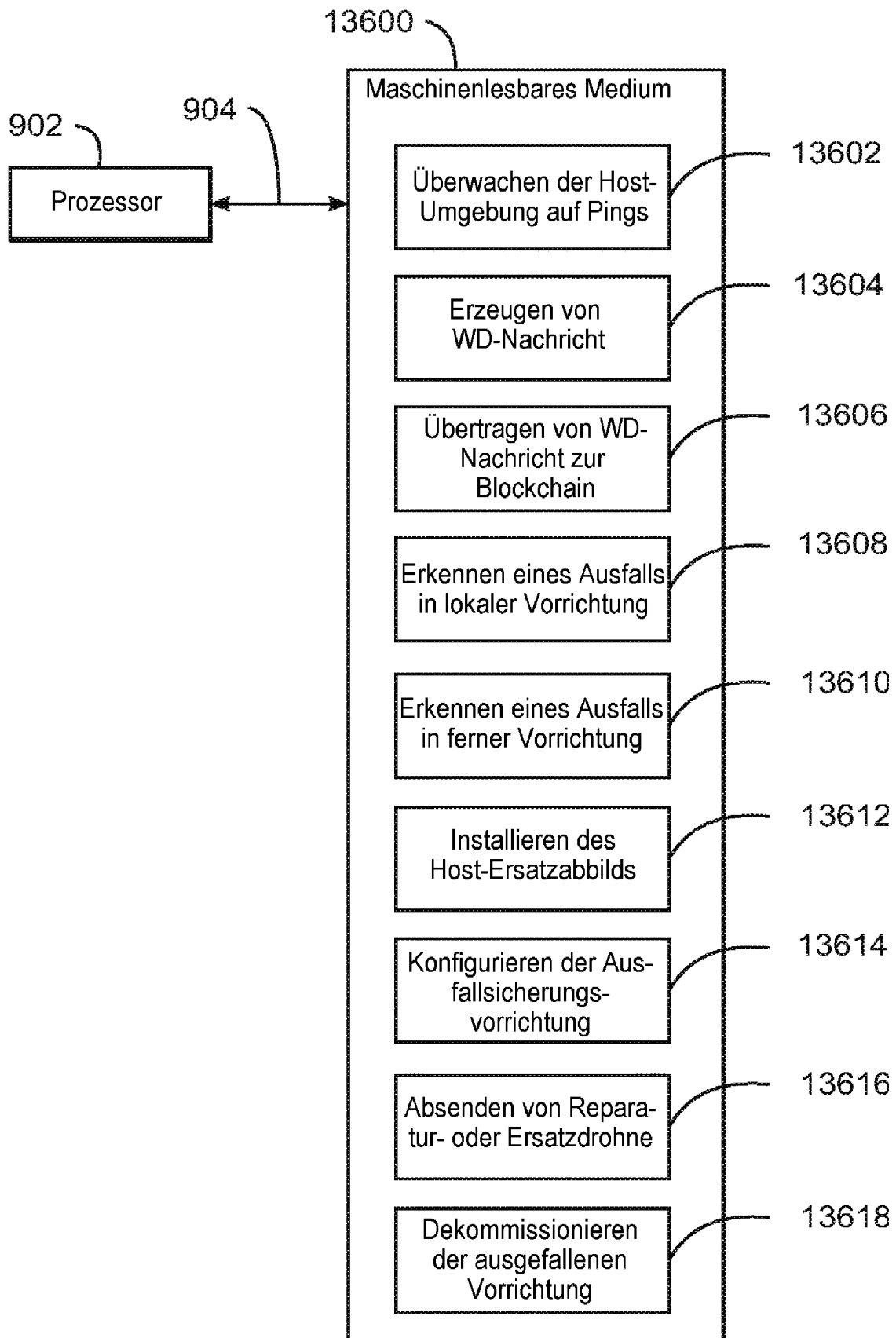


FIG. 136

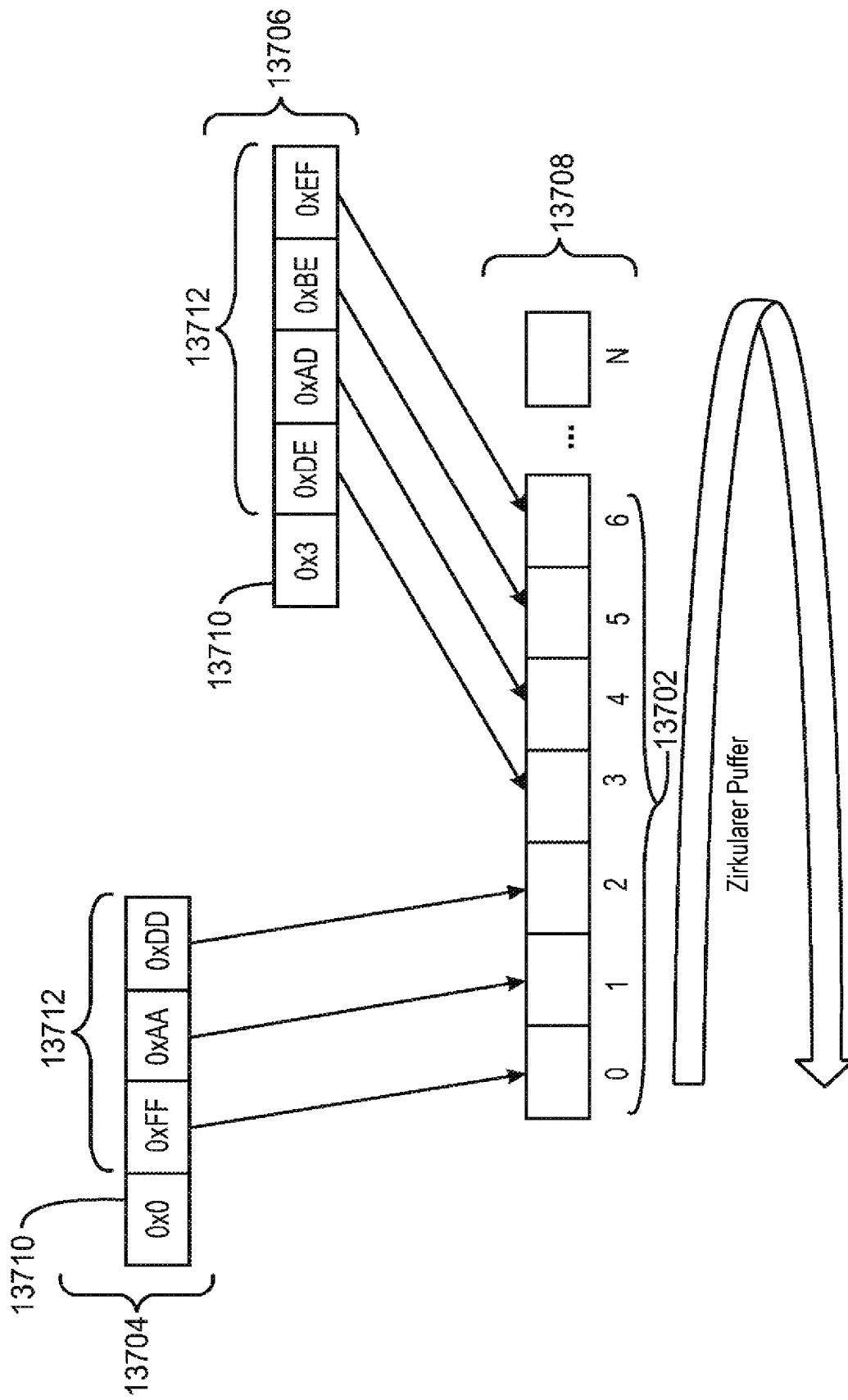


FIG. 137

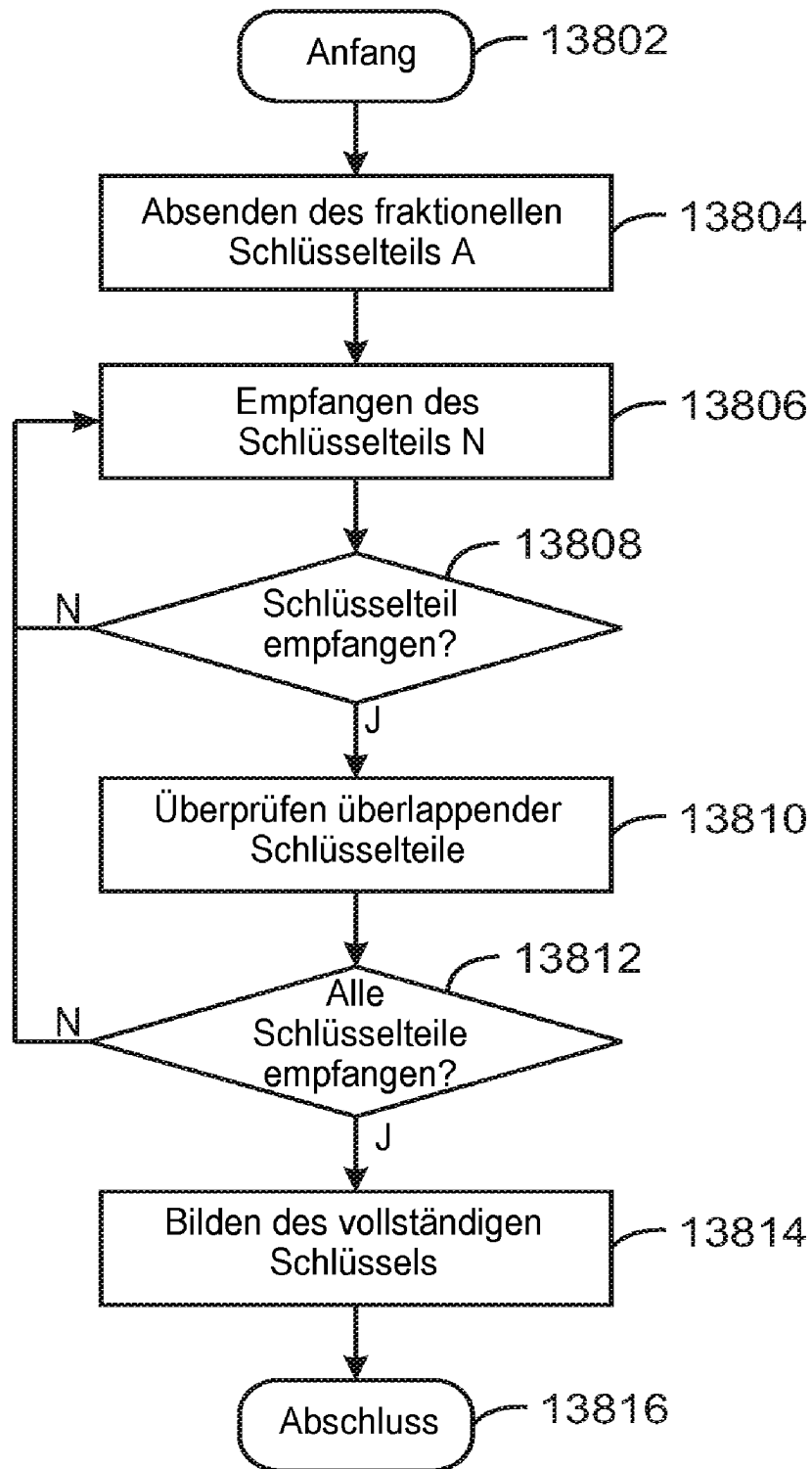
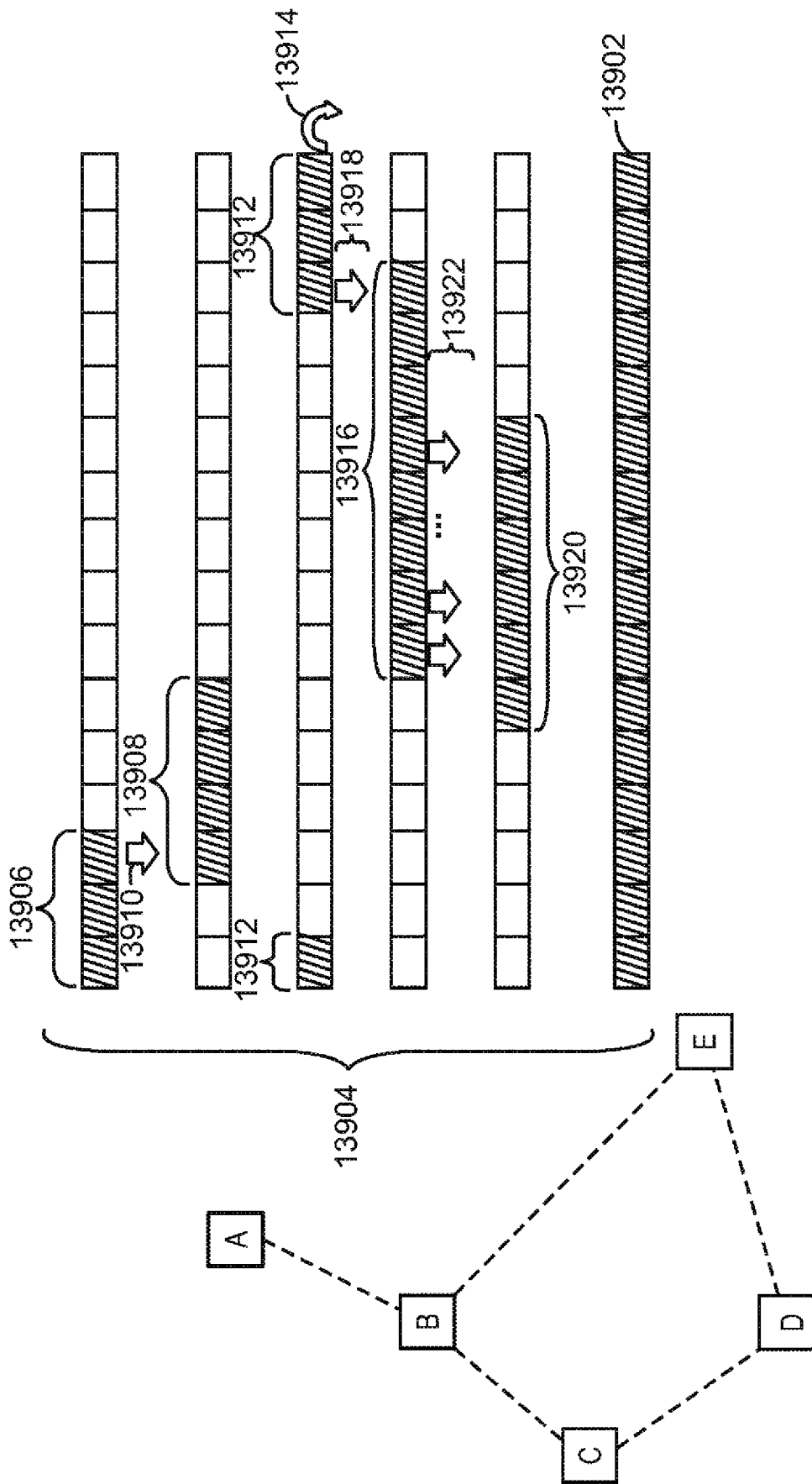
13800

FIG. 138



13900
FIG. 139

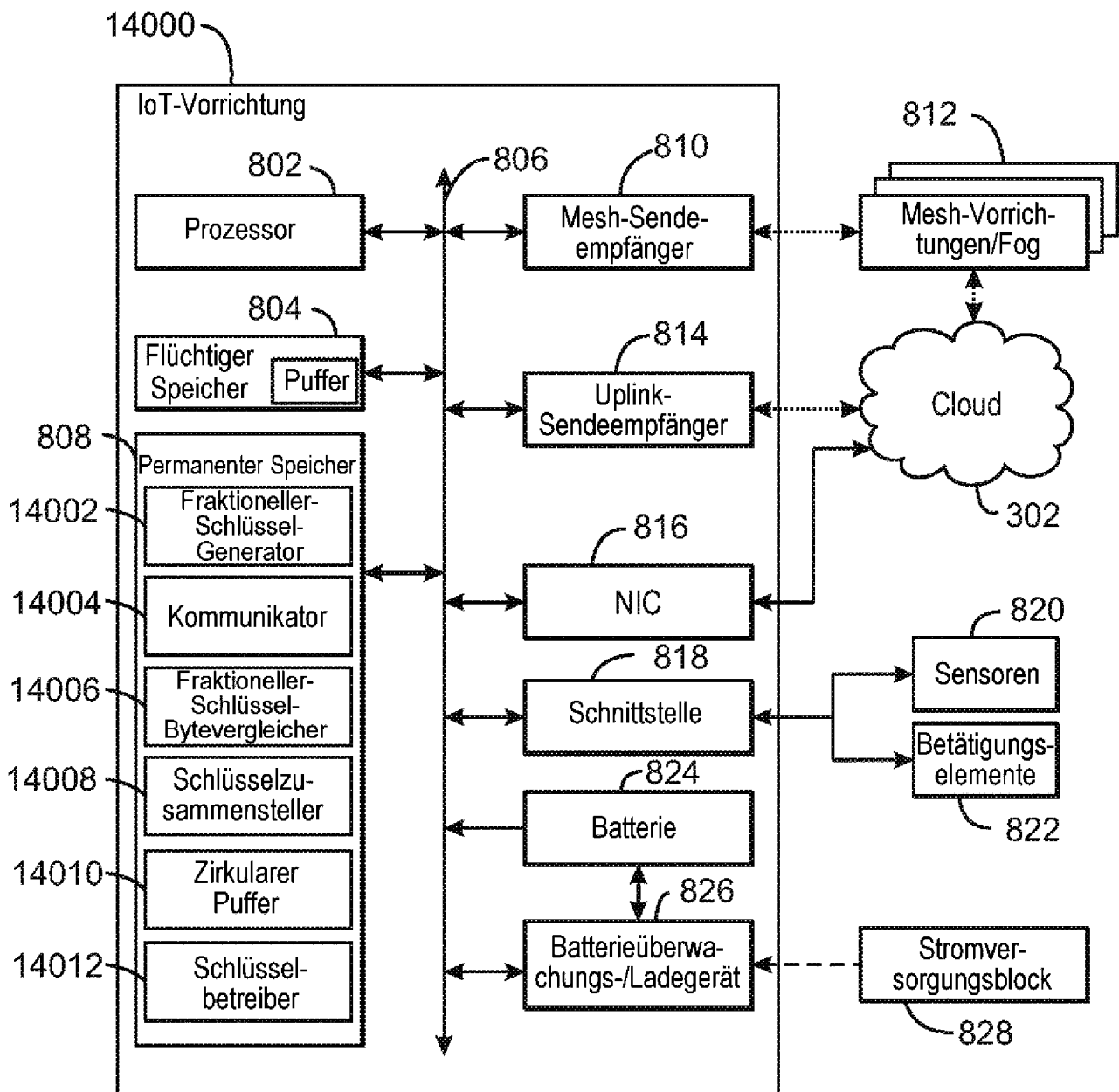


FIG. 140

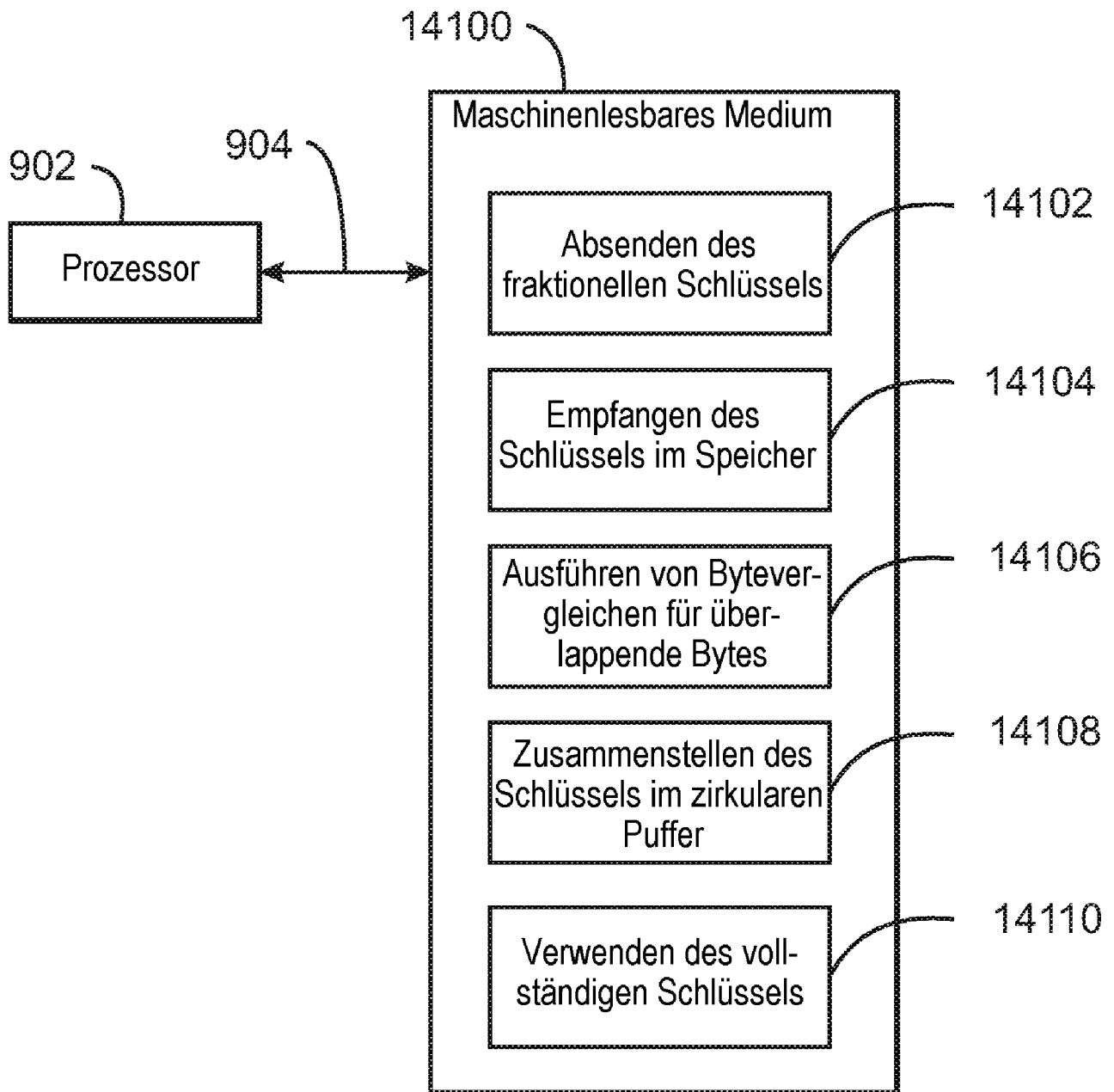
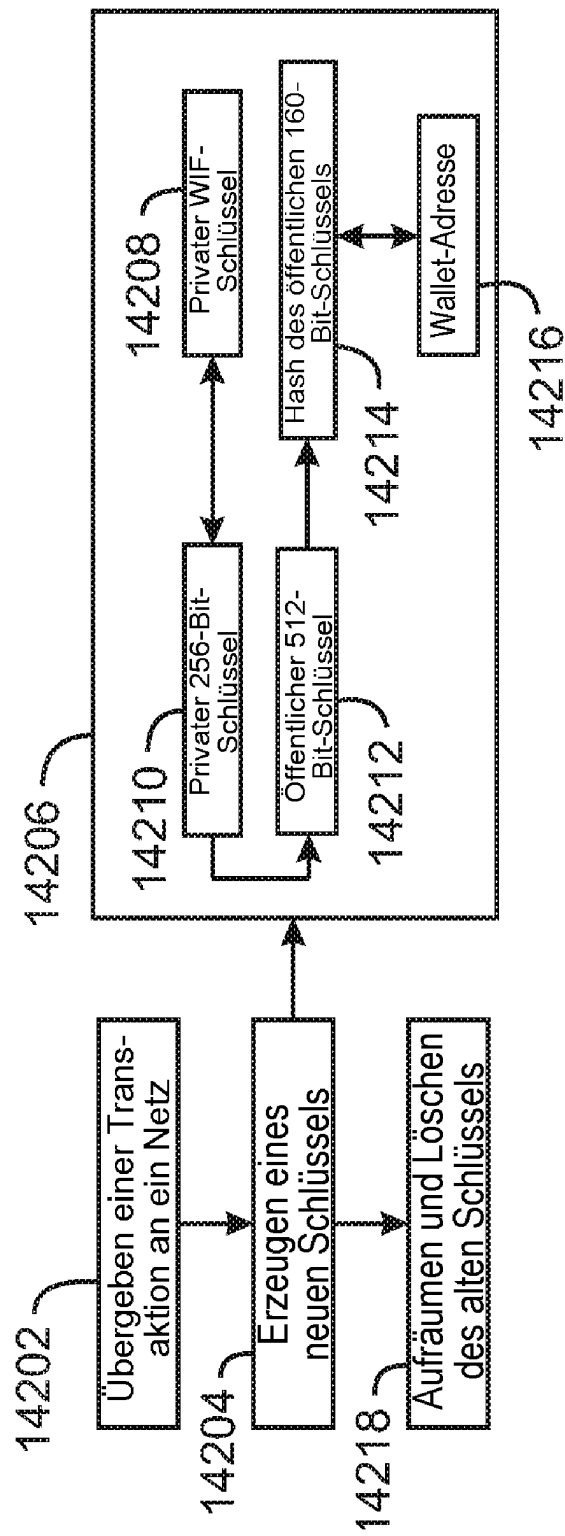
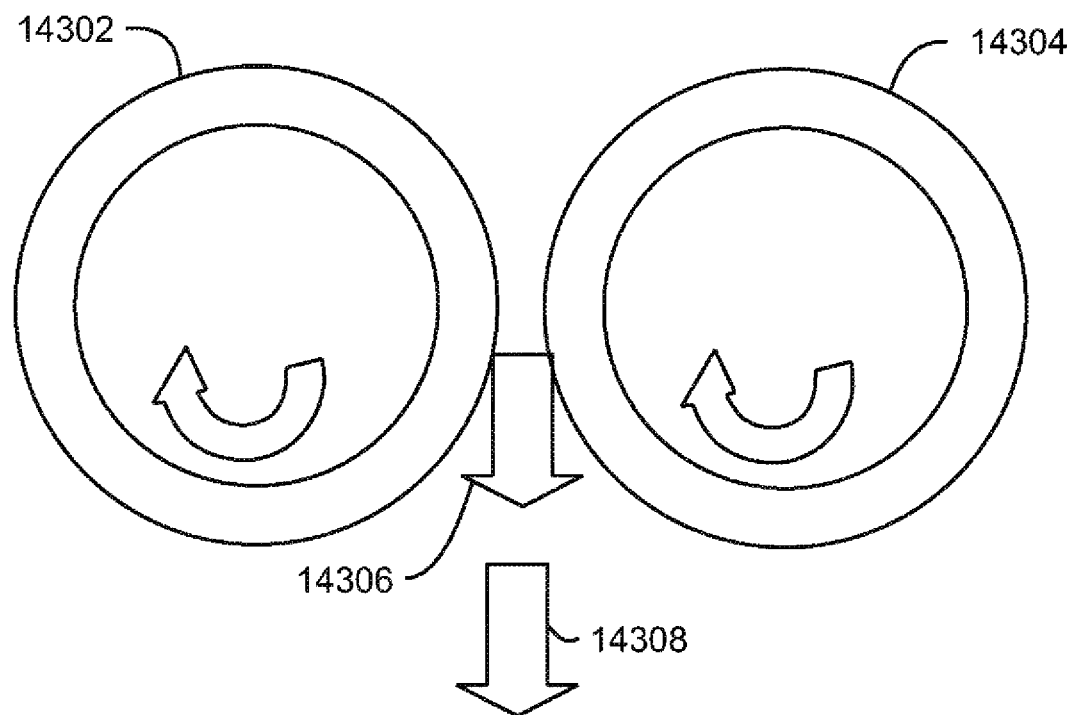


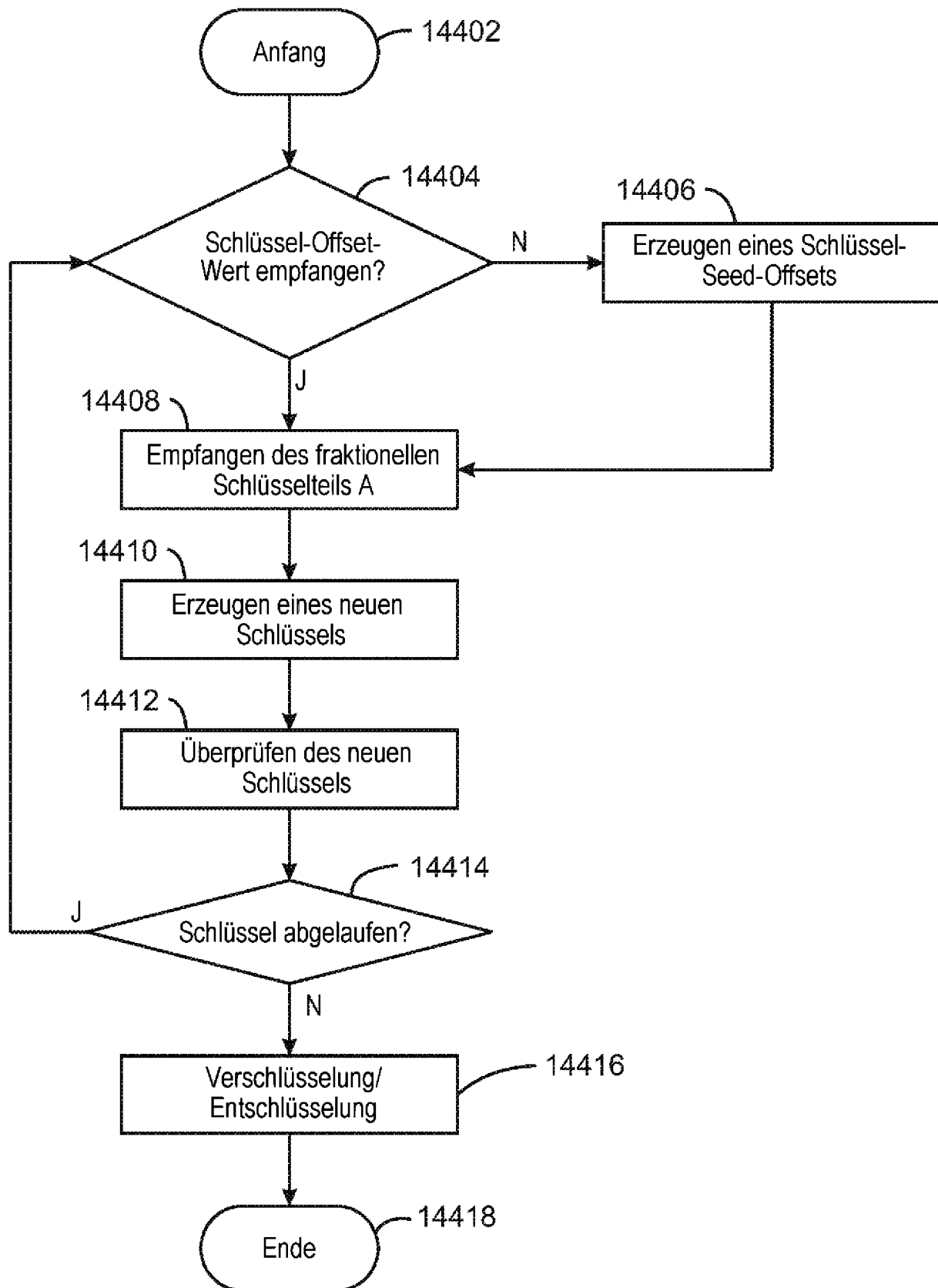
FIG. 141



14200
FIG. 142



14300
FIG. 143



14400
FIG. 144

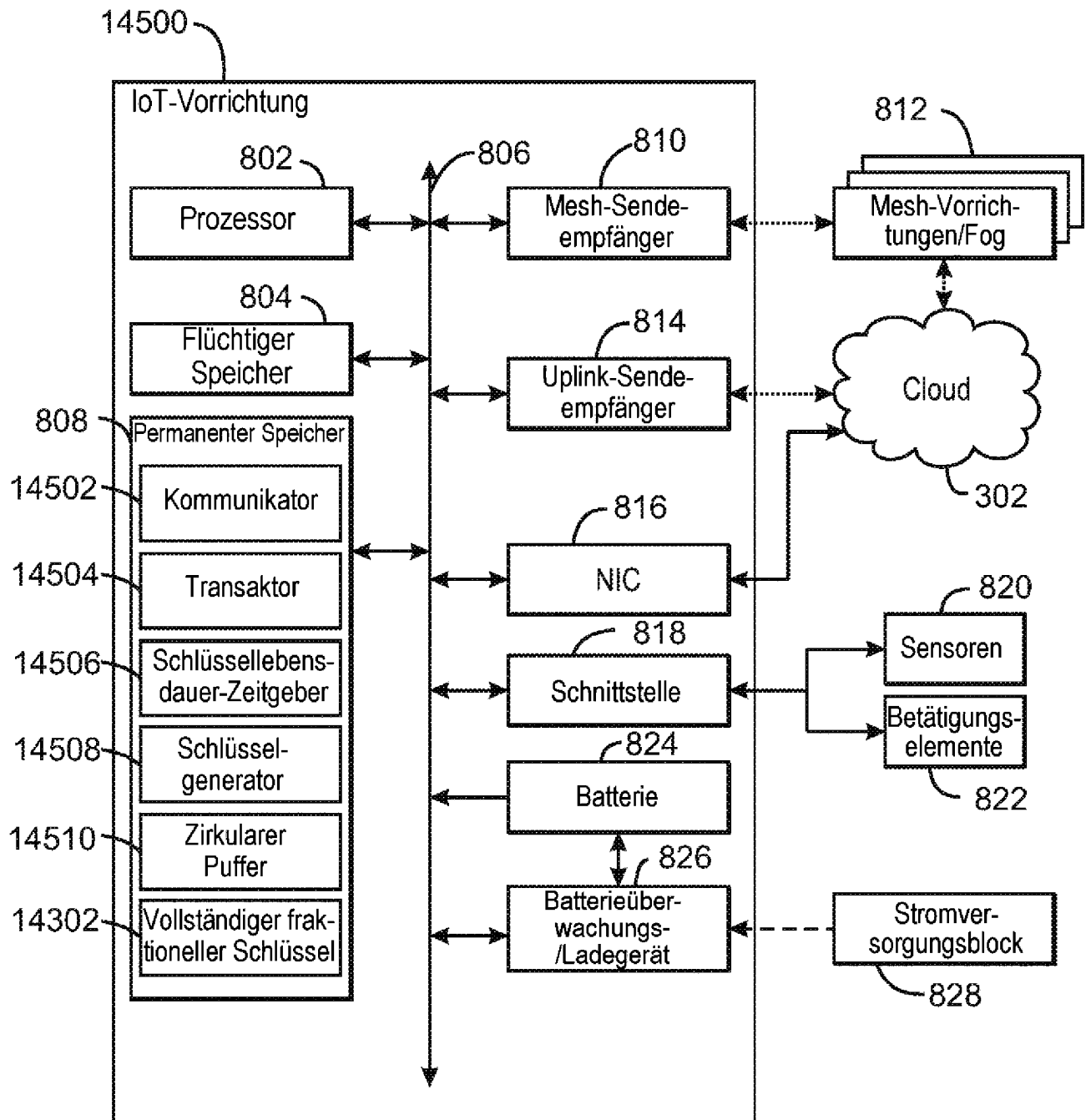


FIG. 145

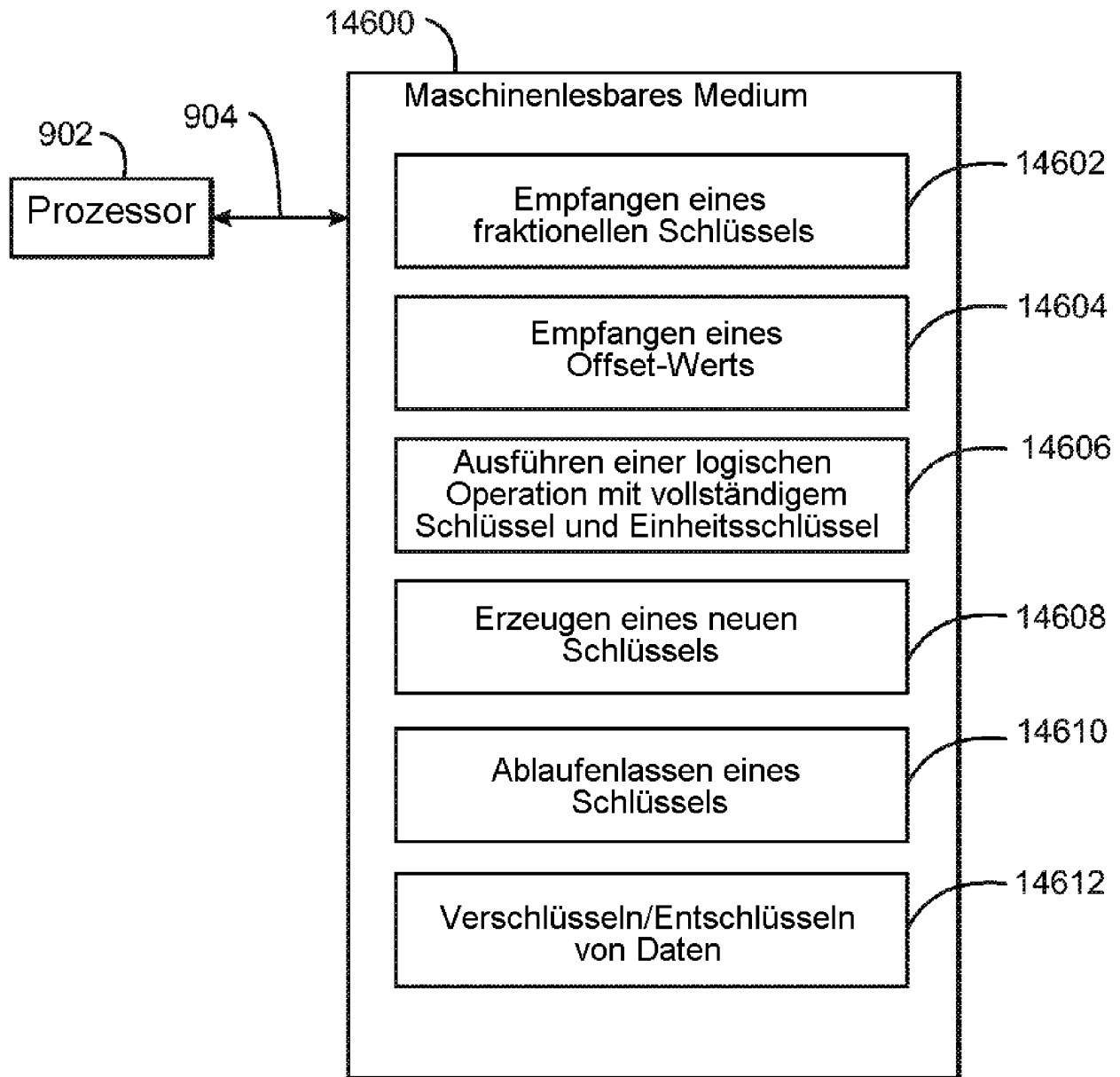
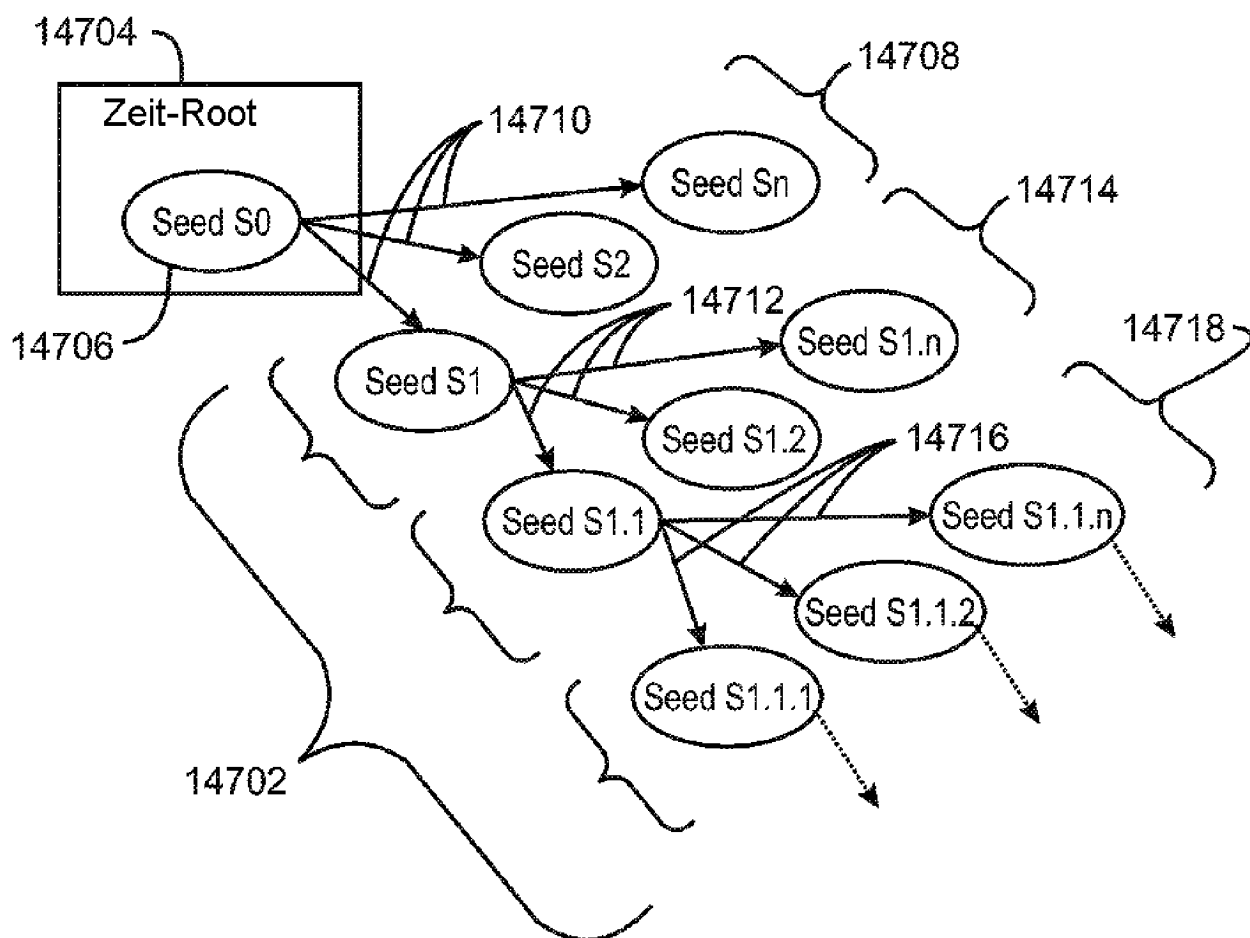
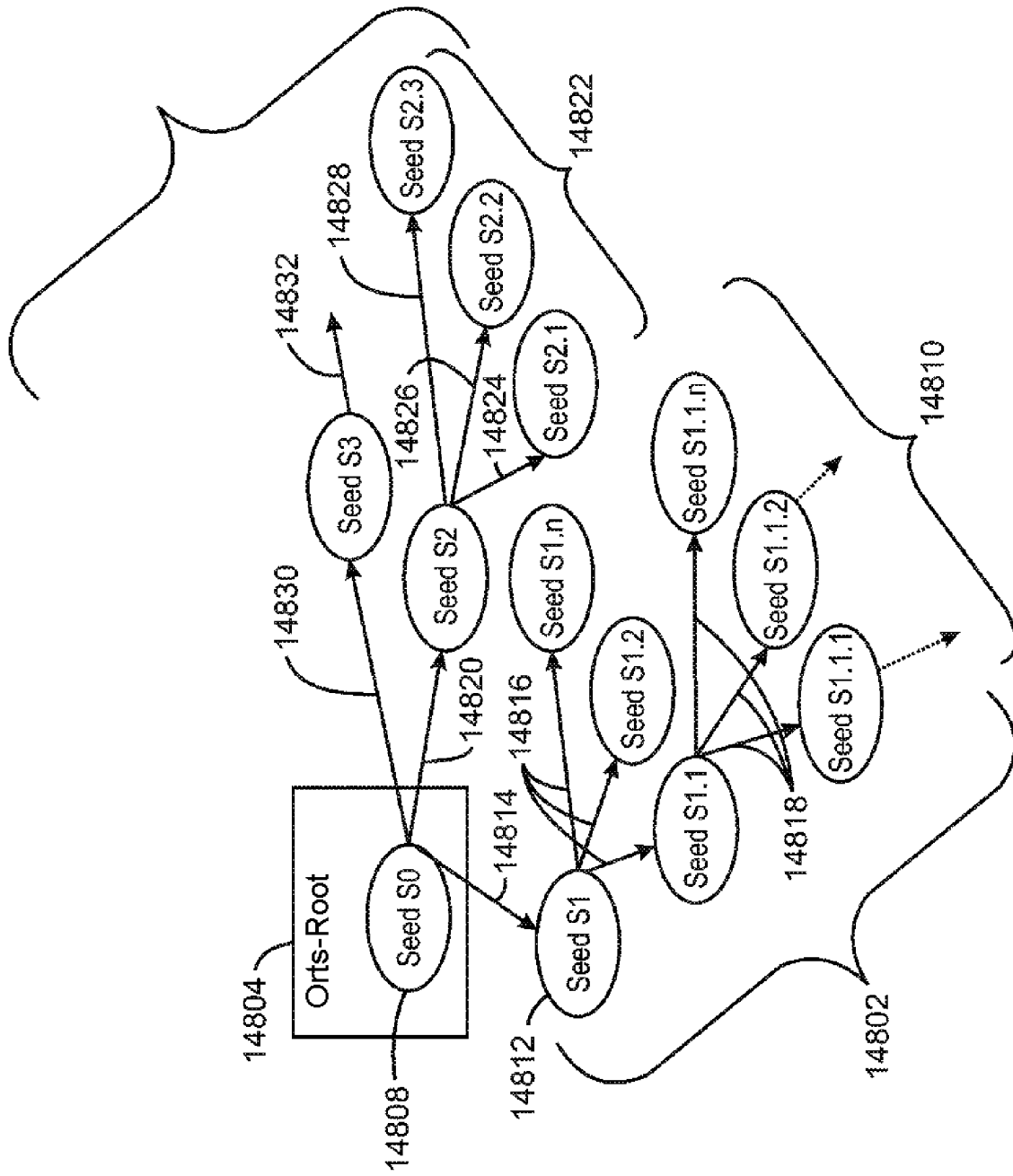


FIG. 146

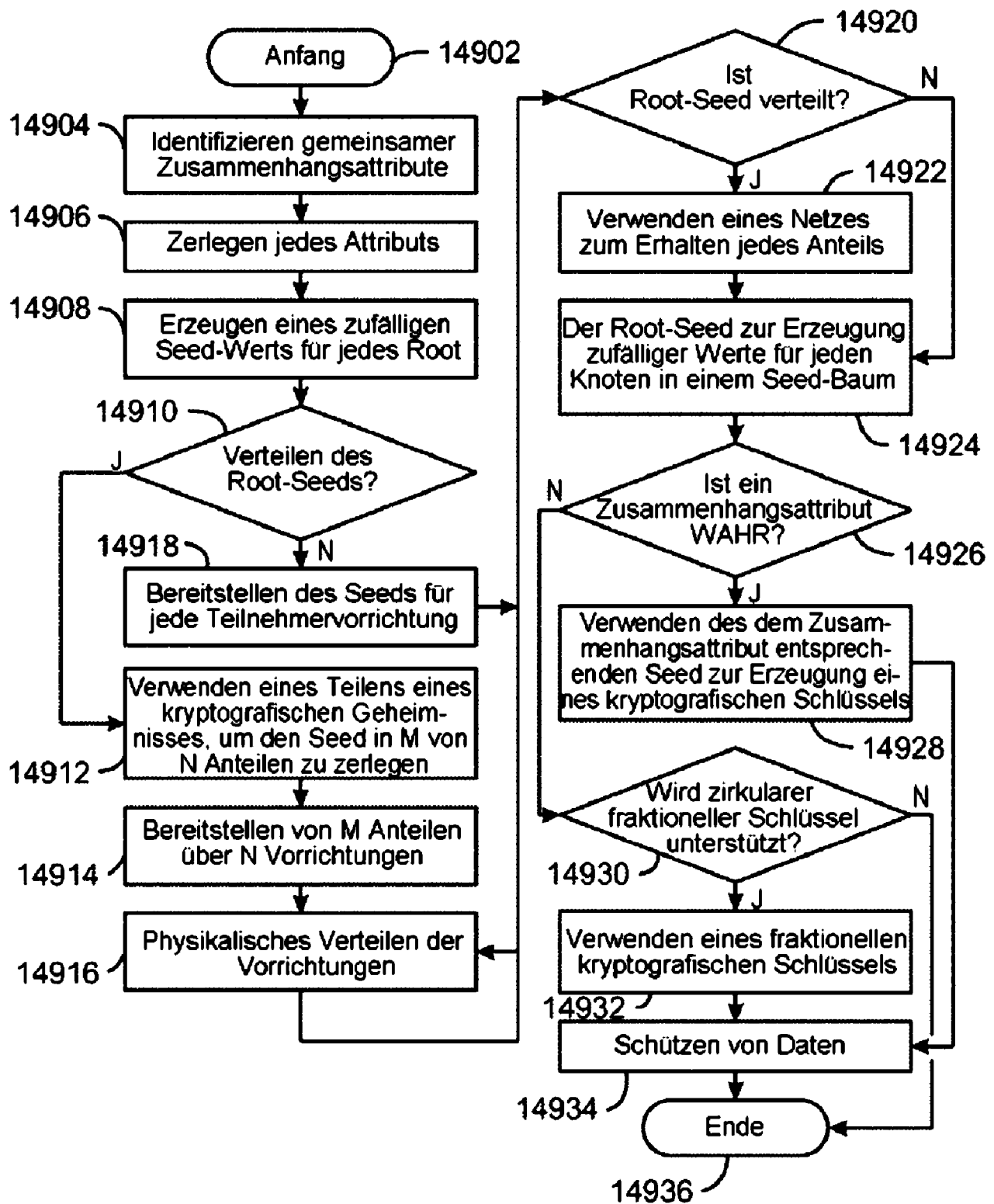


14700
FIG. 147



14800

FIG. 148



14900
FIG. 149

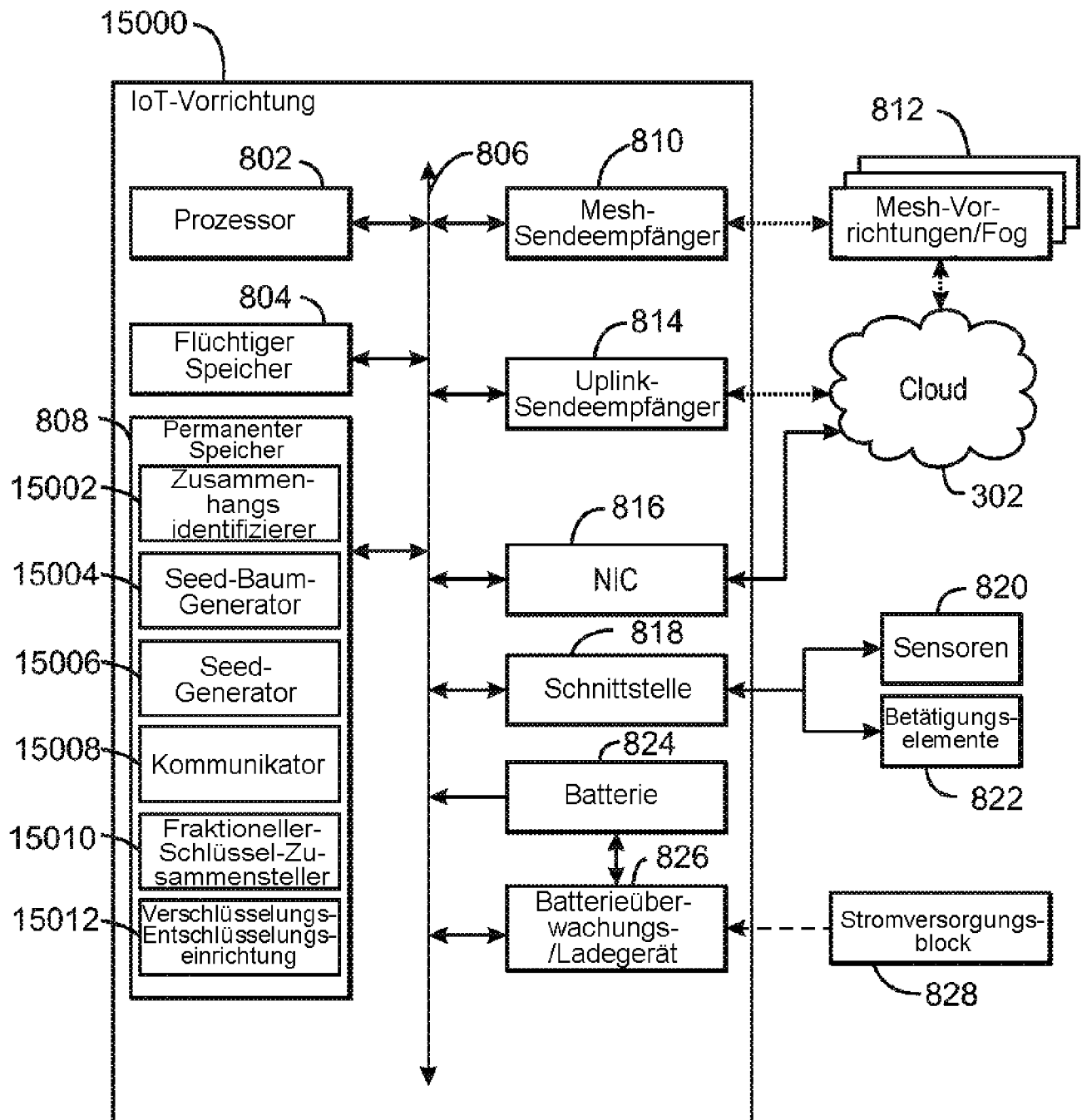


FIG. 150

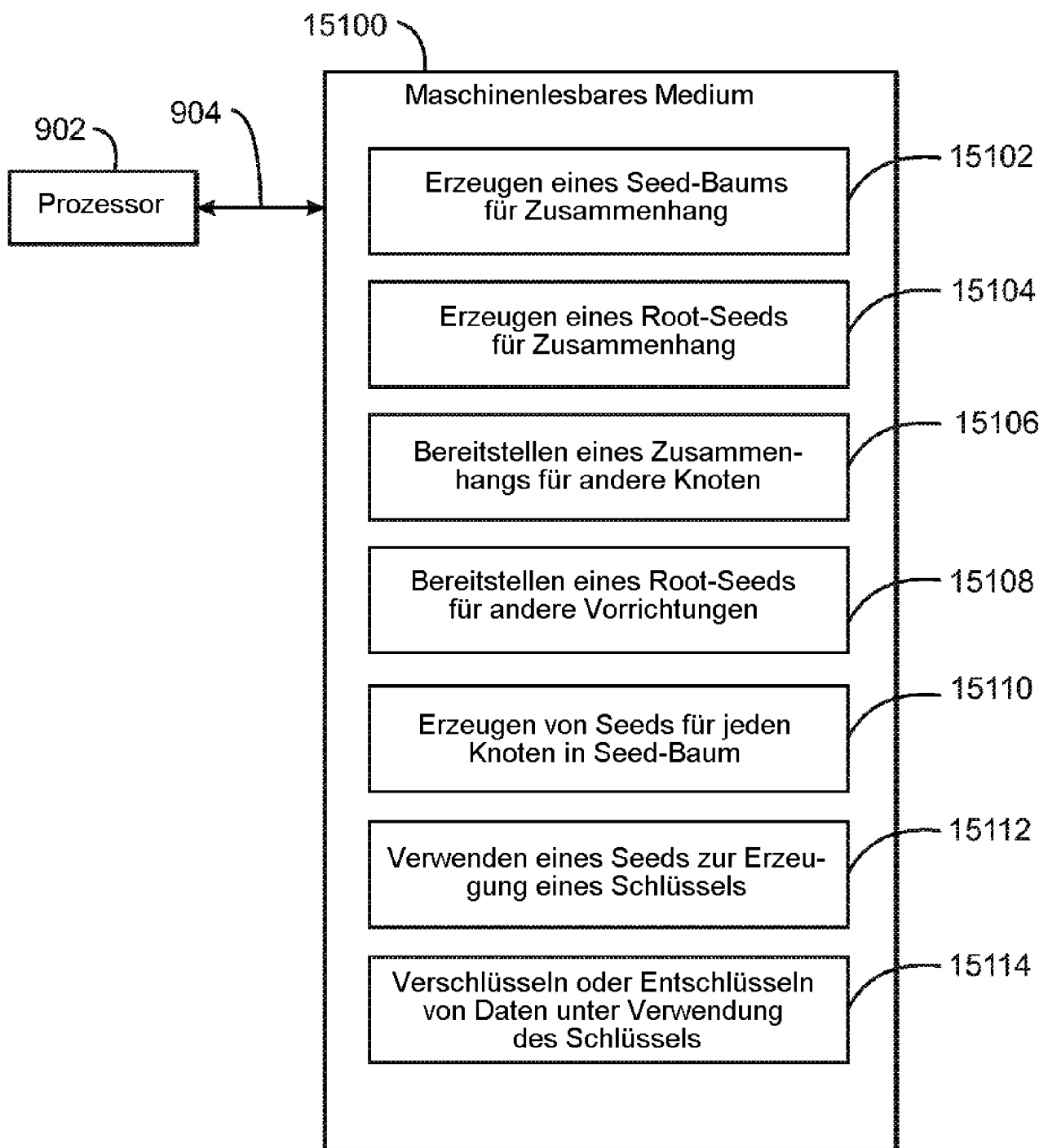
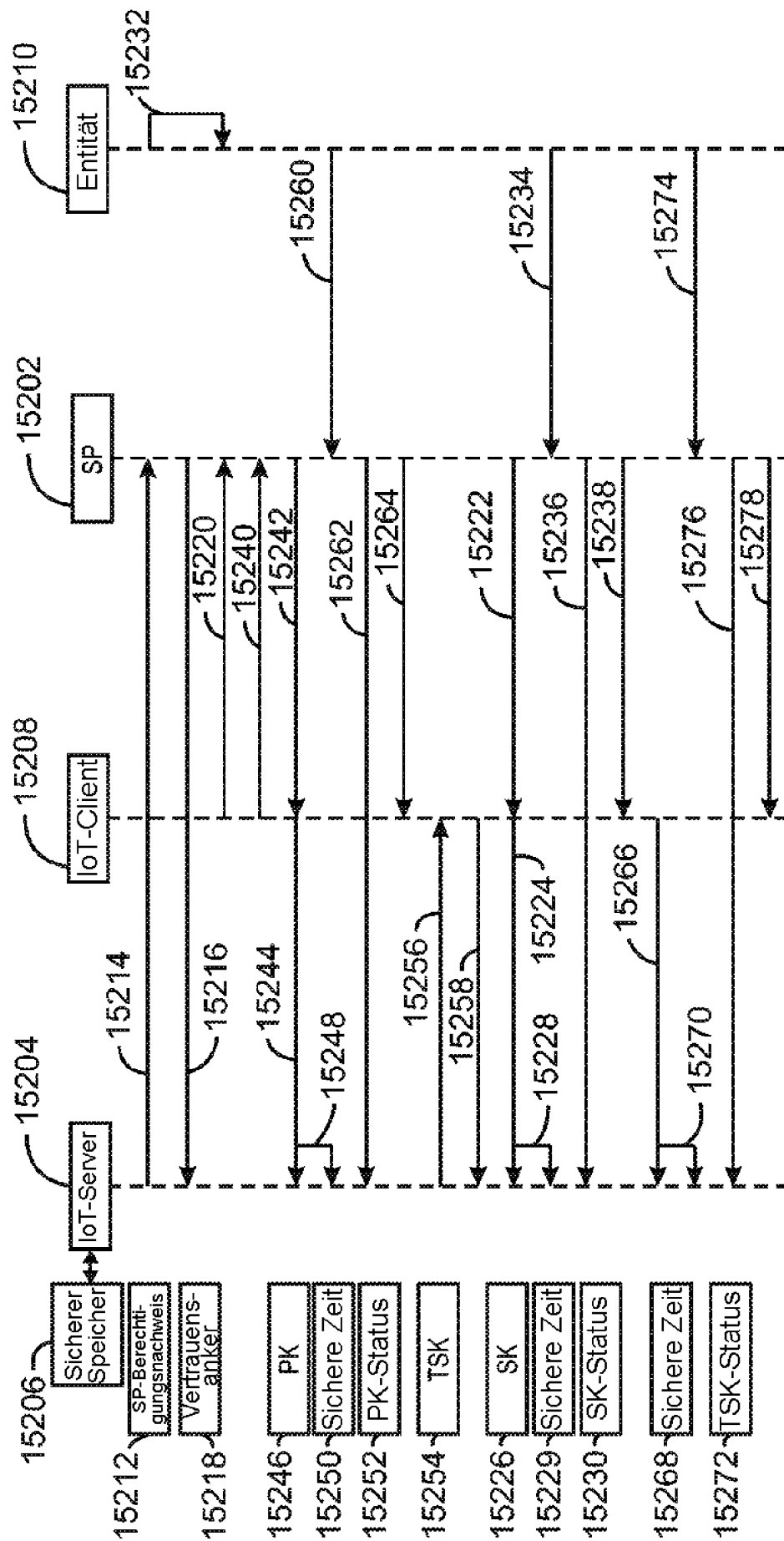


FIG. 151



15200

FIG. 152

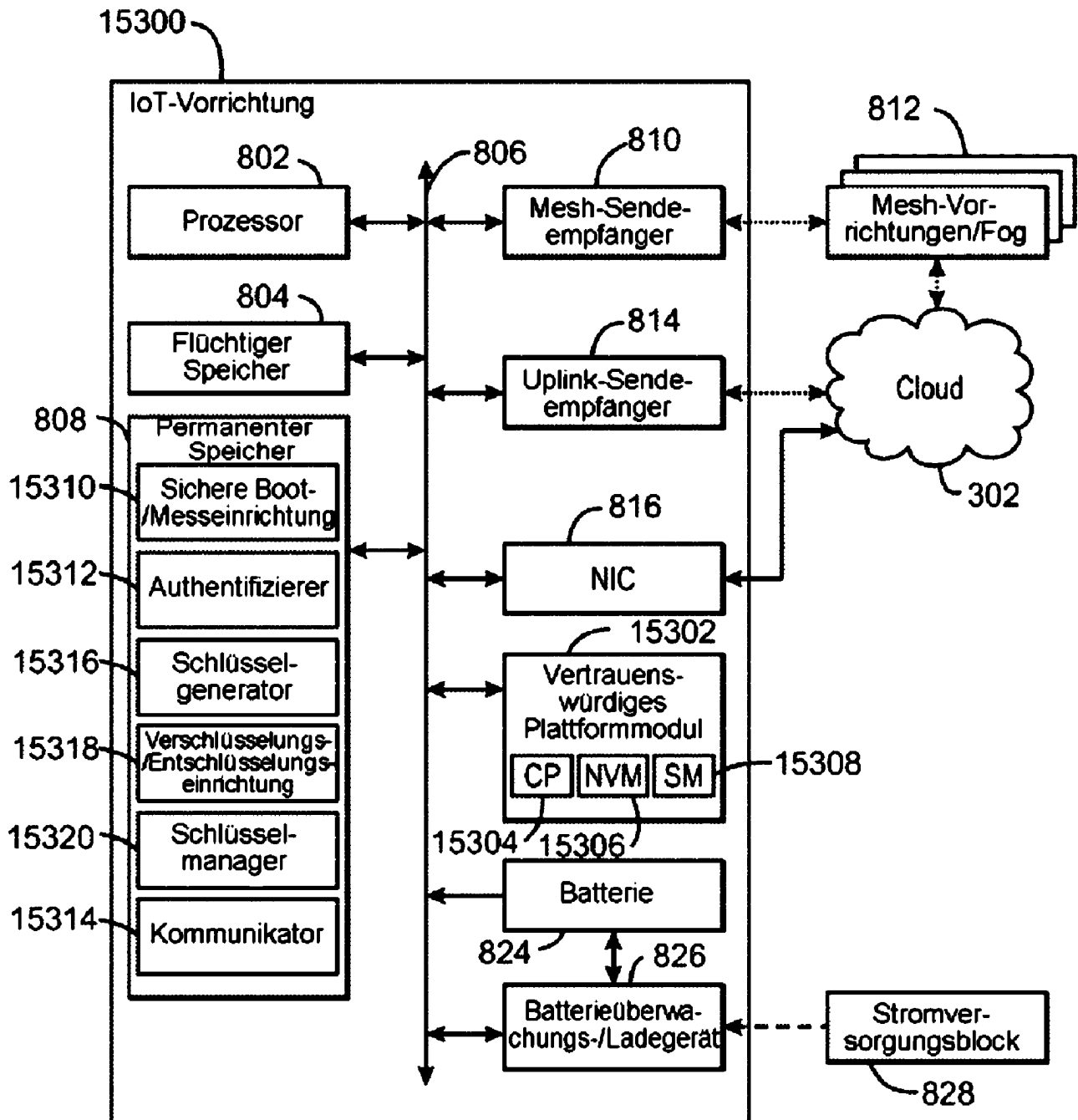


FIG. 153

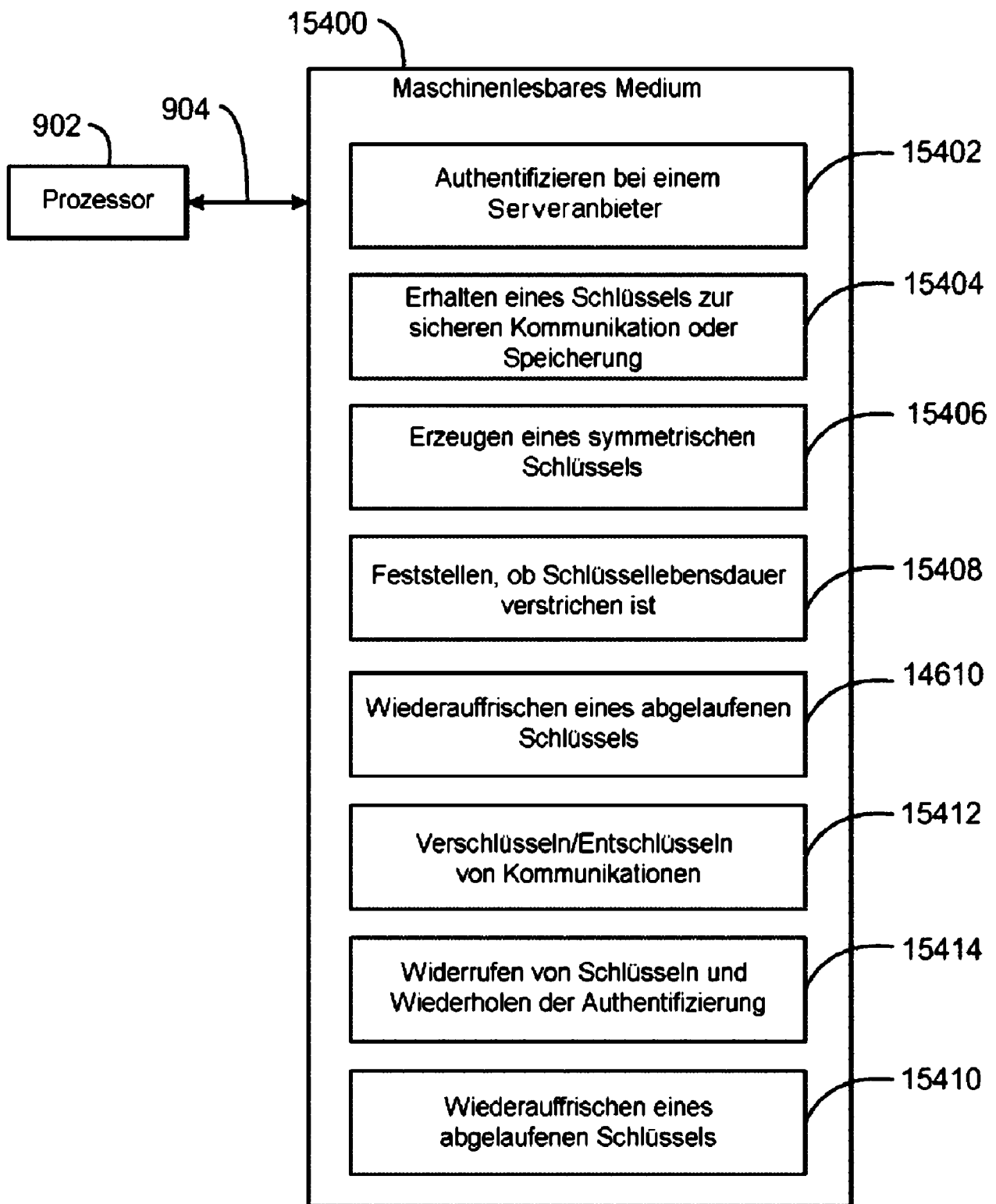
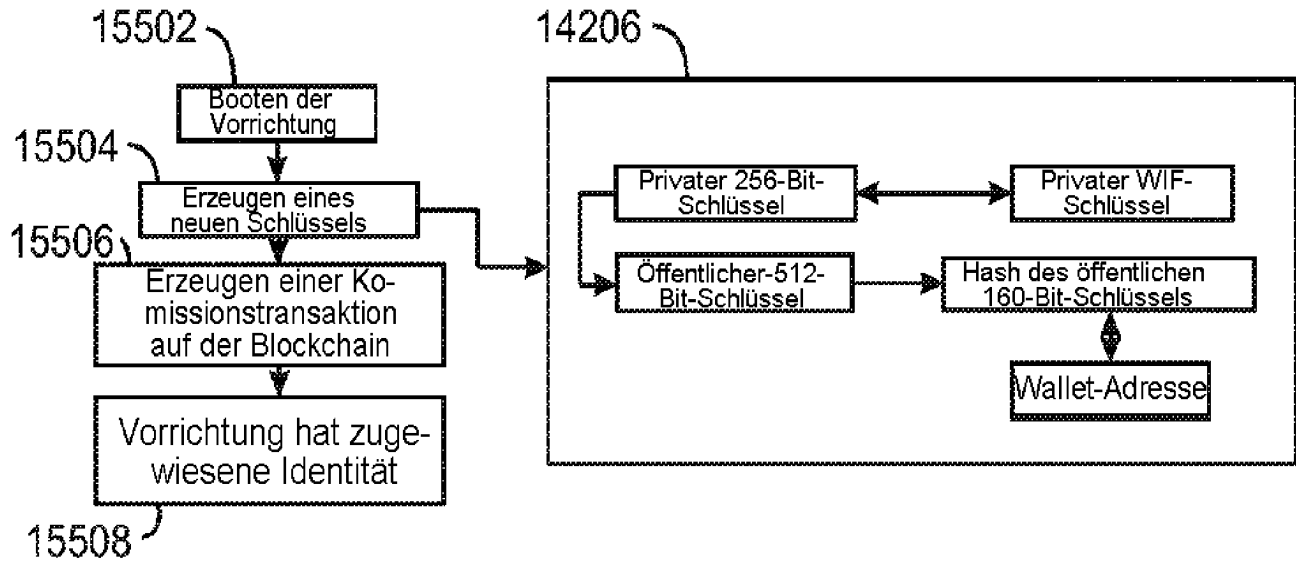
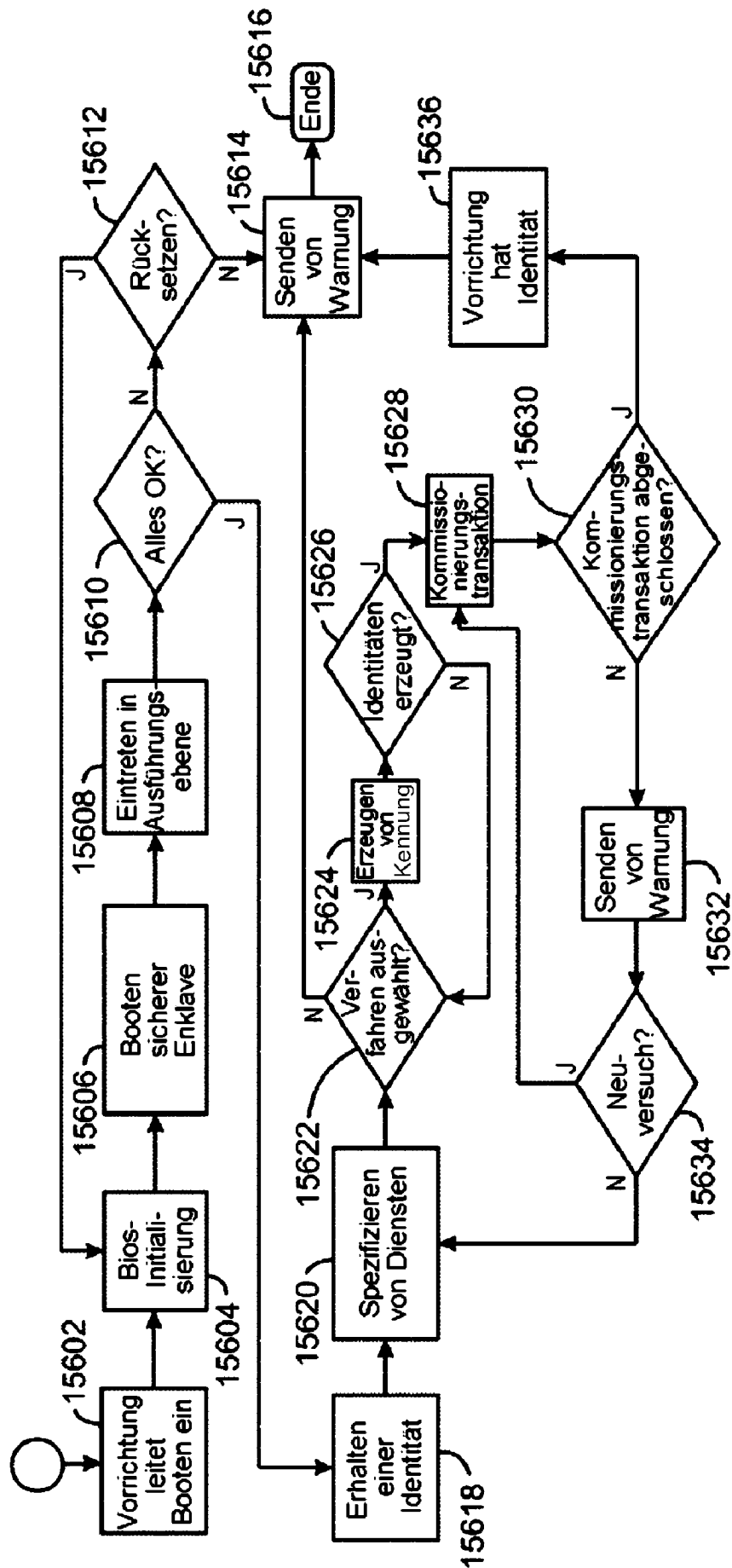


FIG. 154



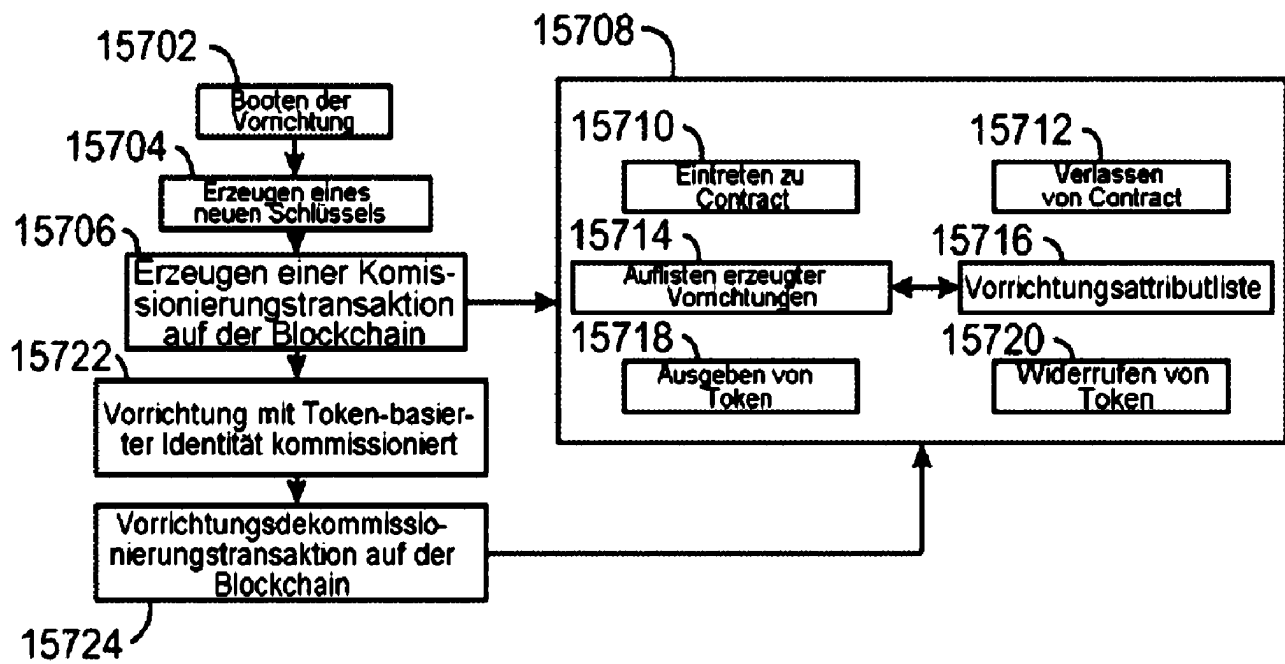
15500

FIG. 155

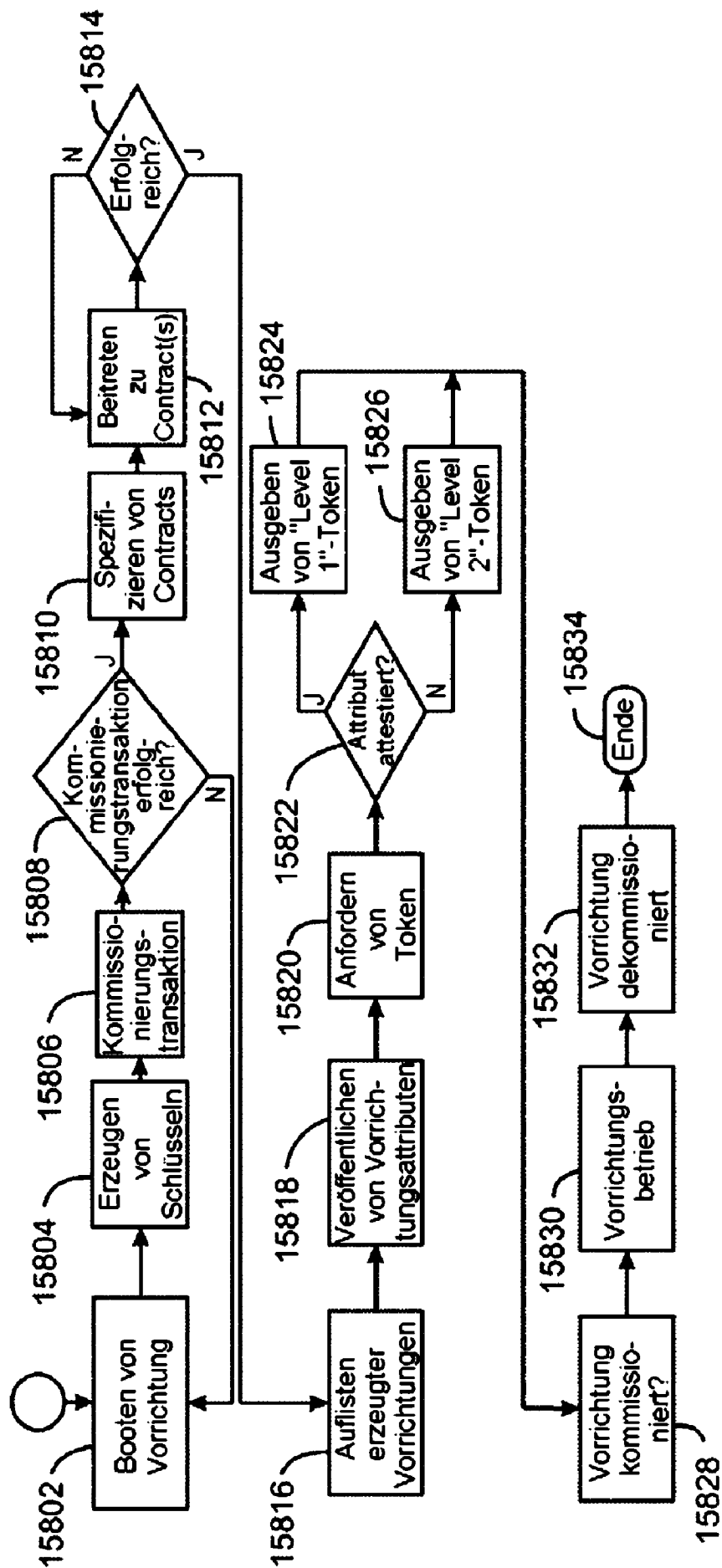


15600

FIG. 156



15700
FIG. 157



15800

FIG. 158

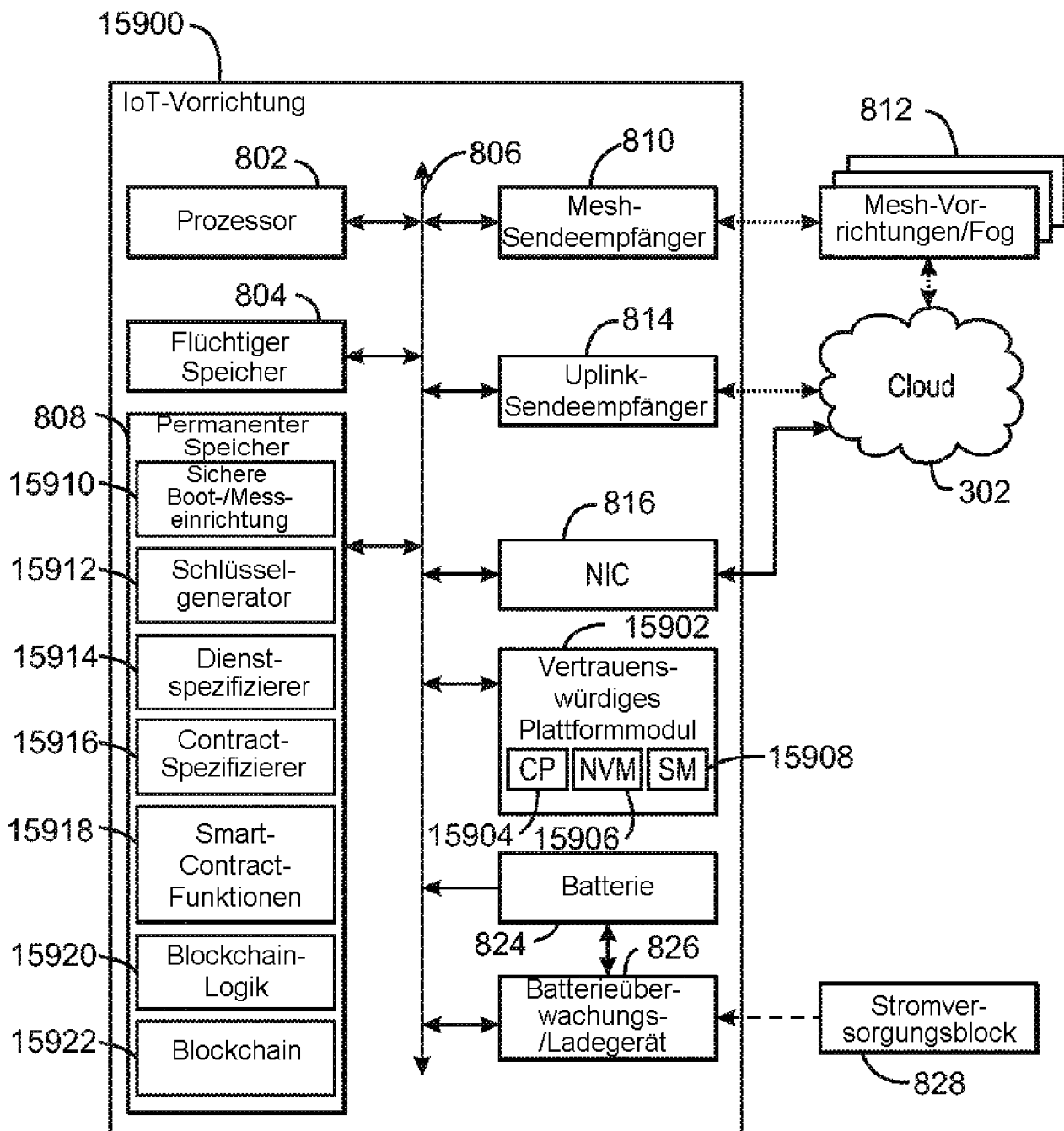


FIG. 159

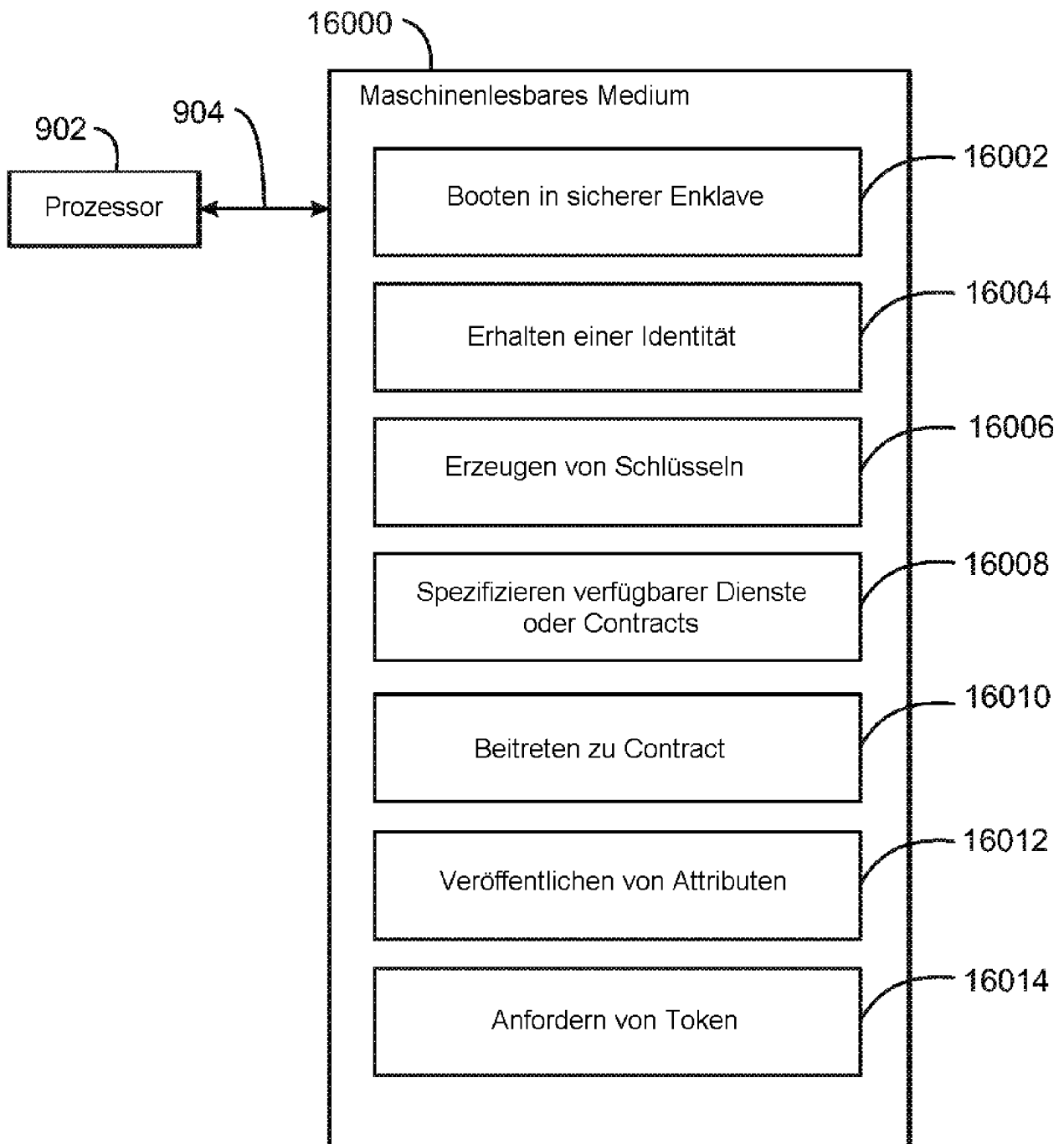
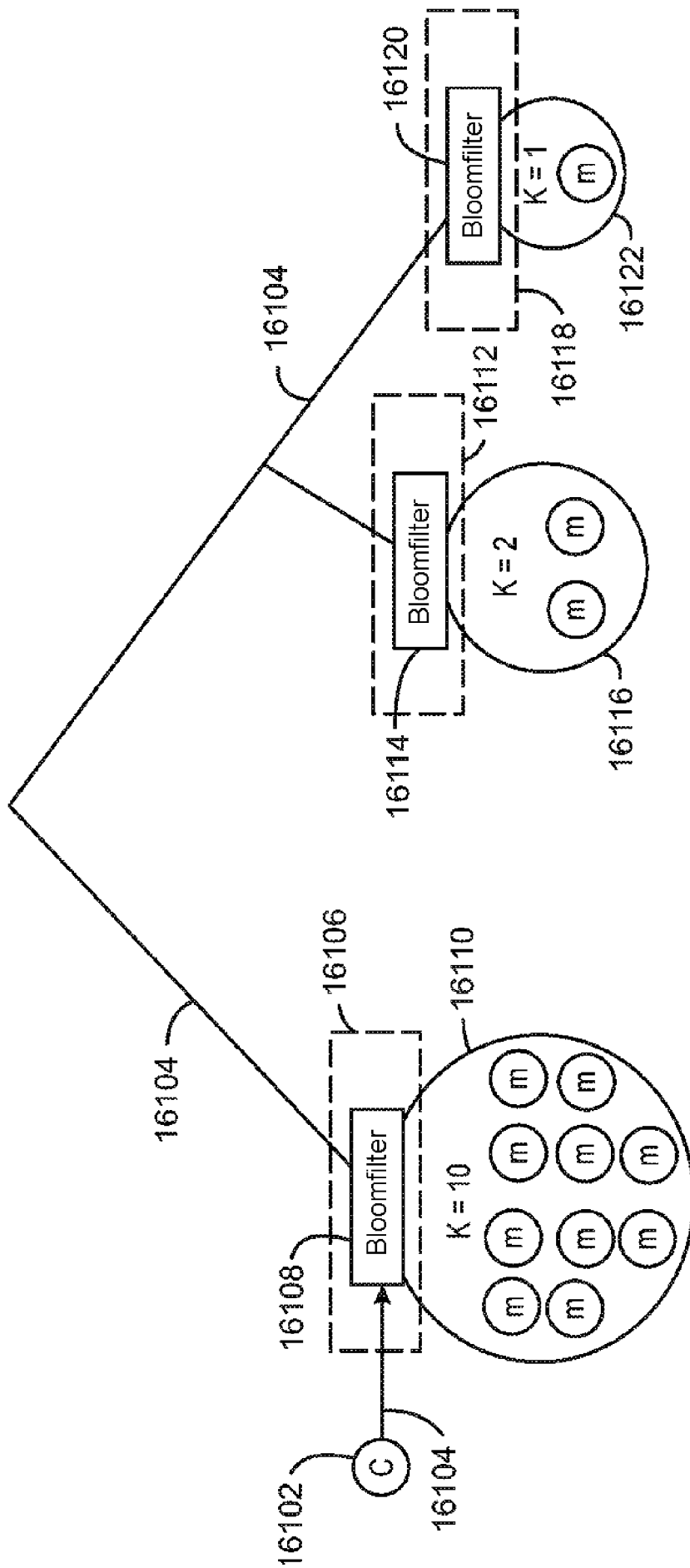
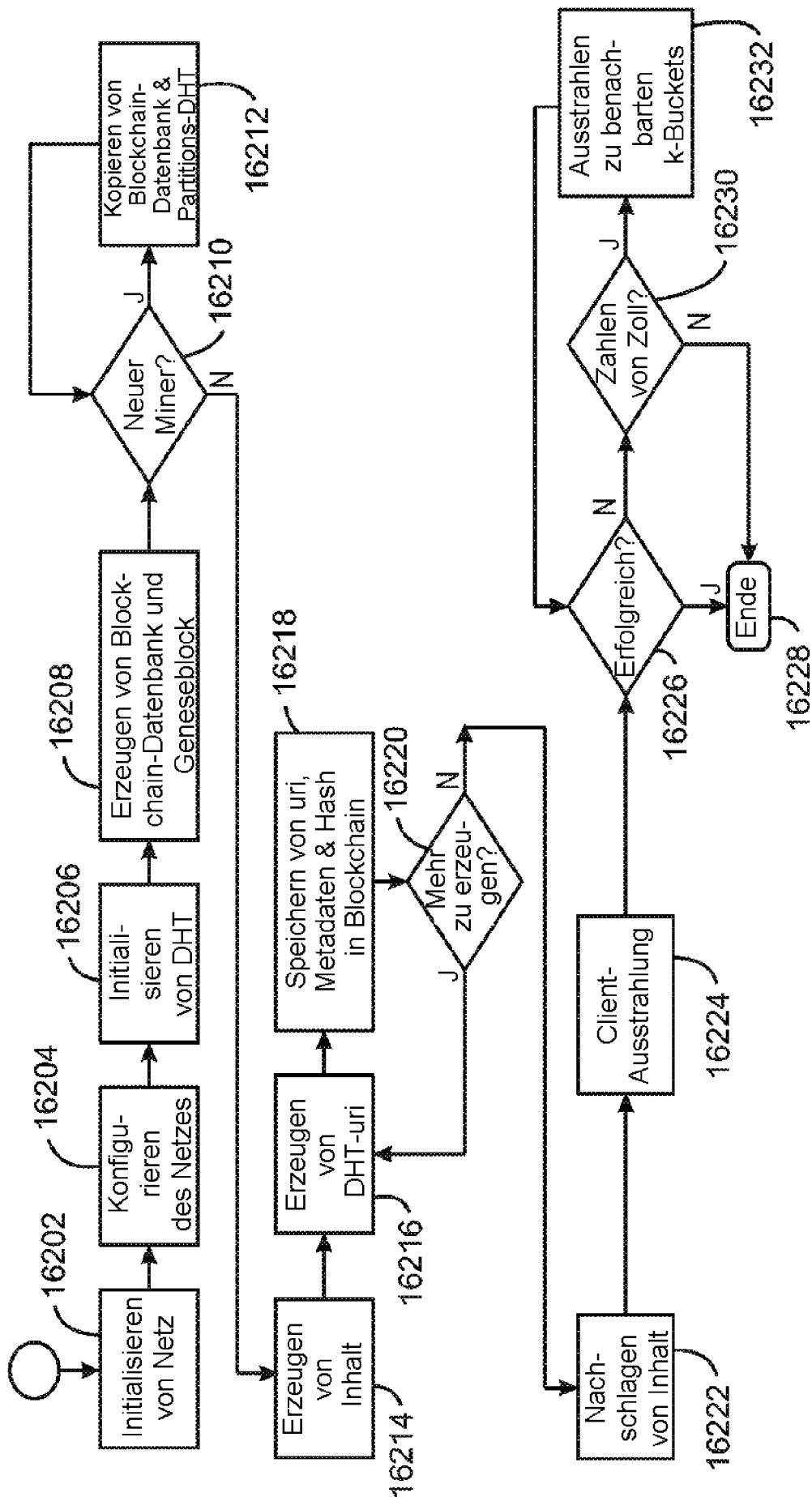


FIG. 160



16100

FIG. 161



16200

FIG. 162

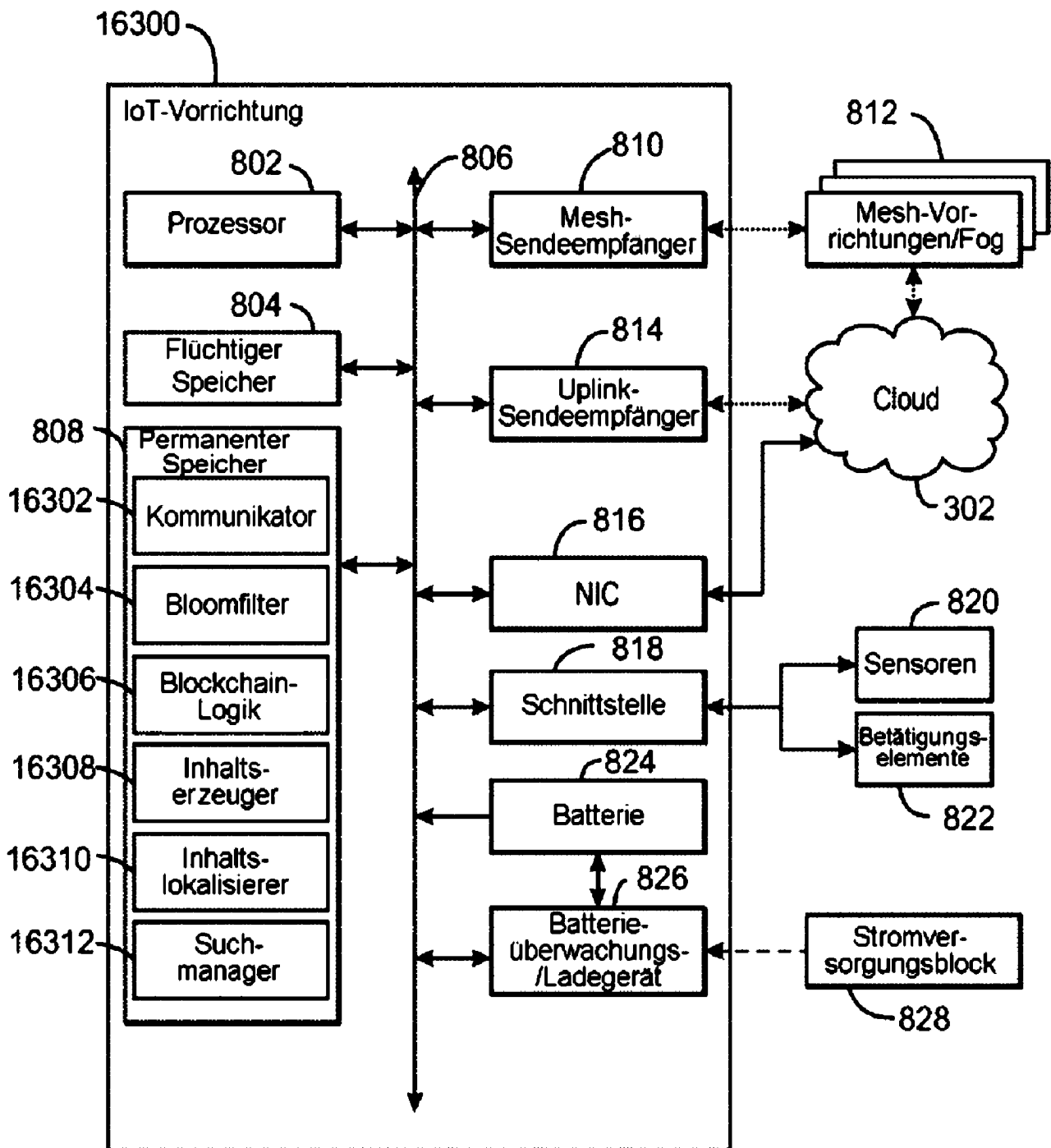


FIG. 163

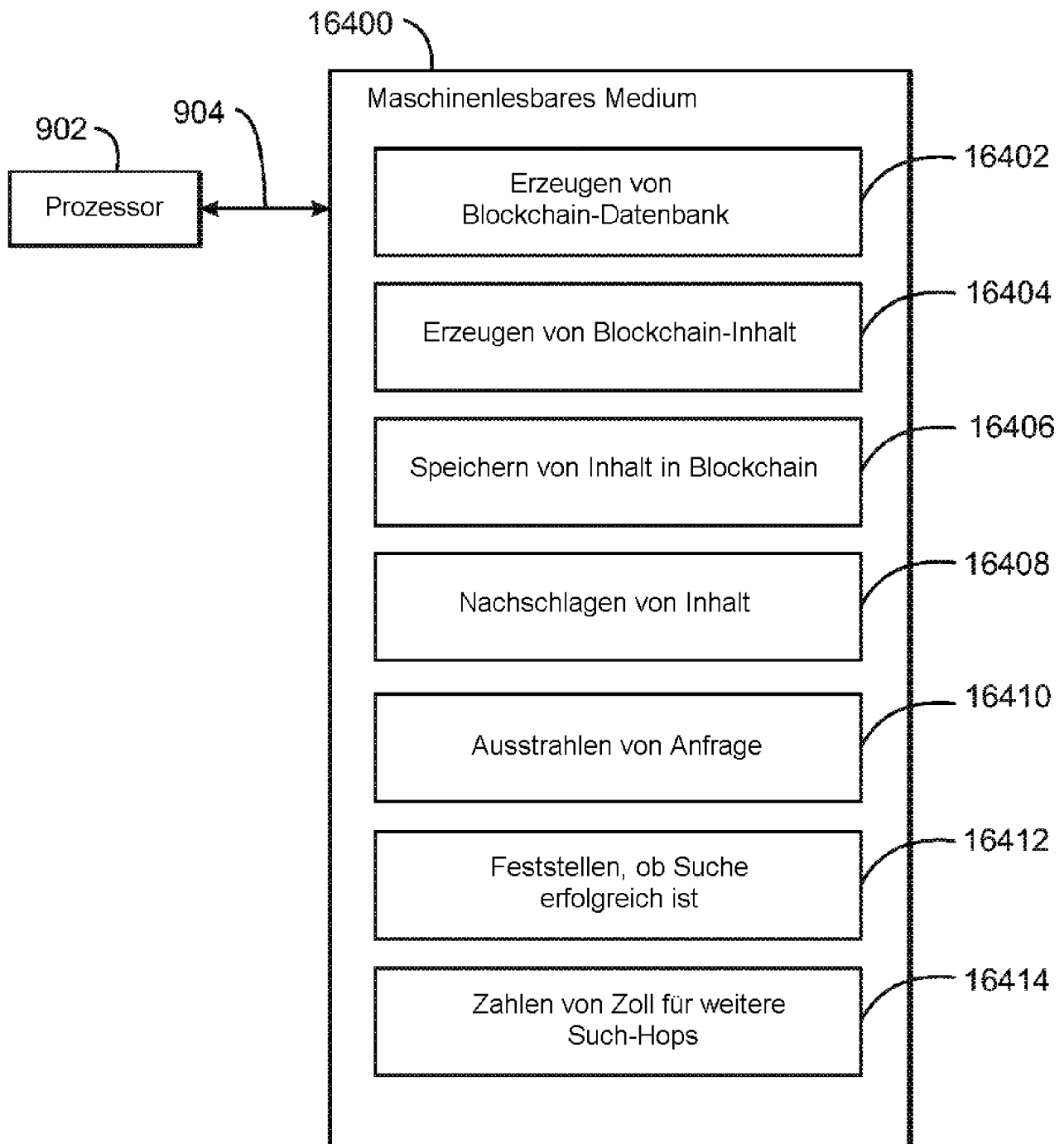
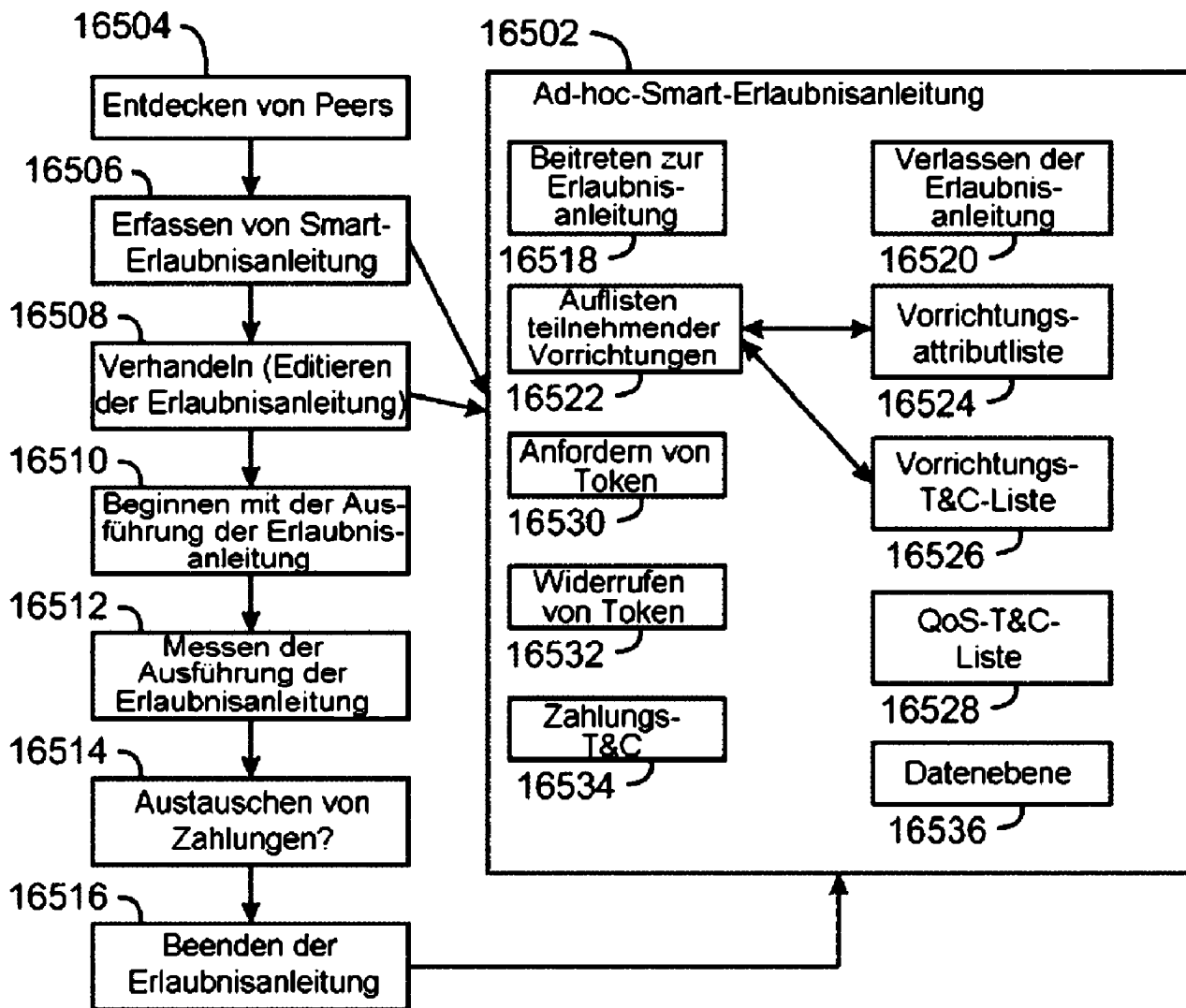
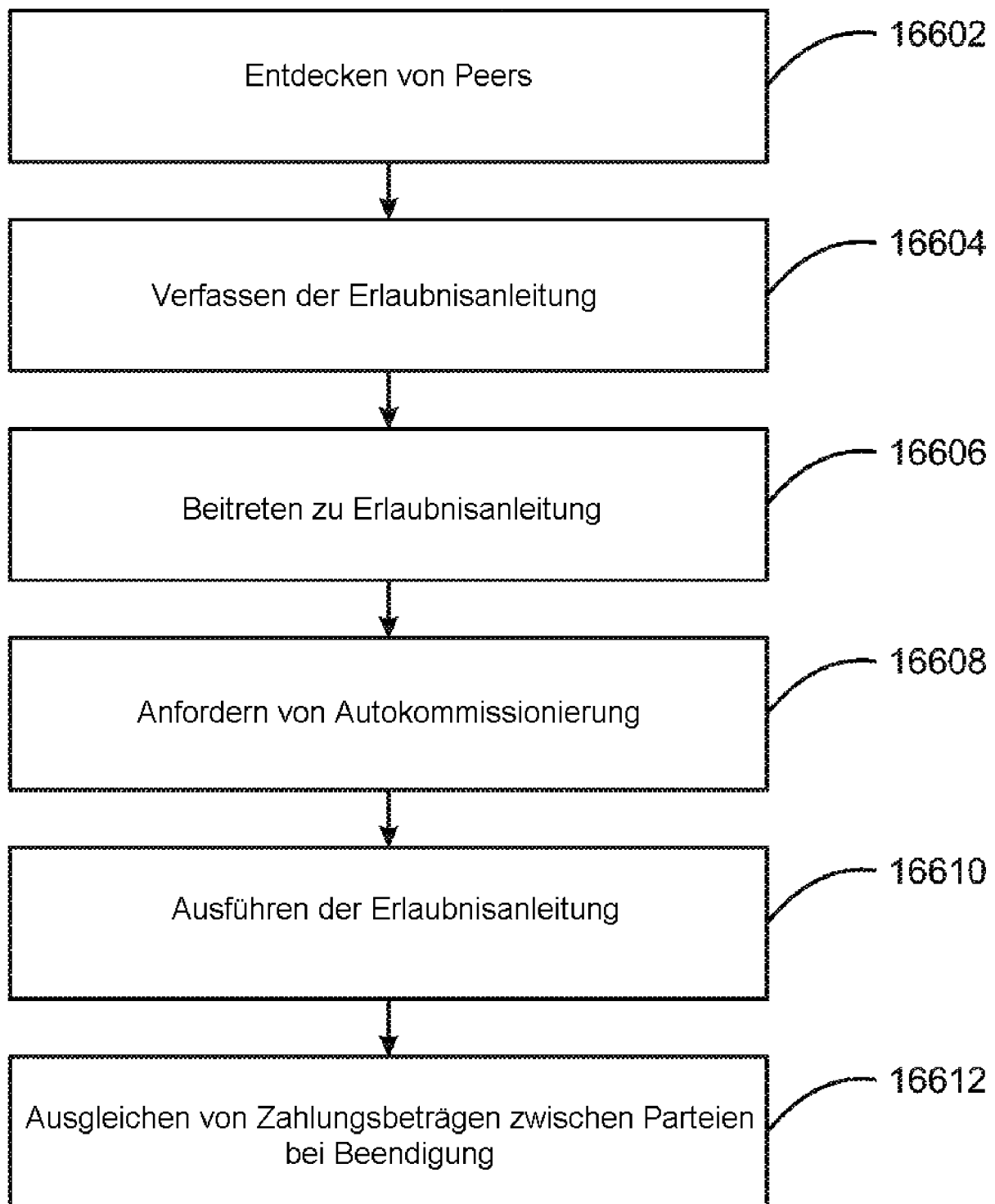


FIG. 164



16500
FIG. 165



16600
FIG. 166

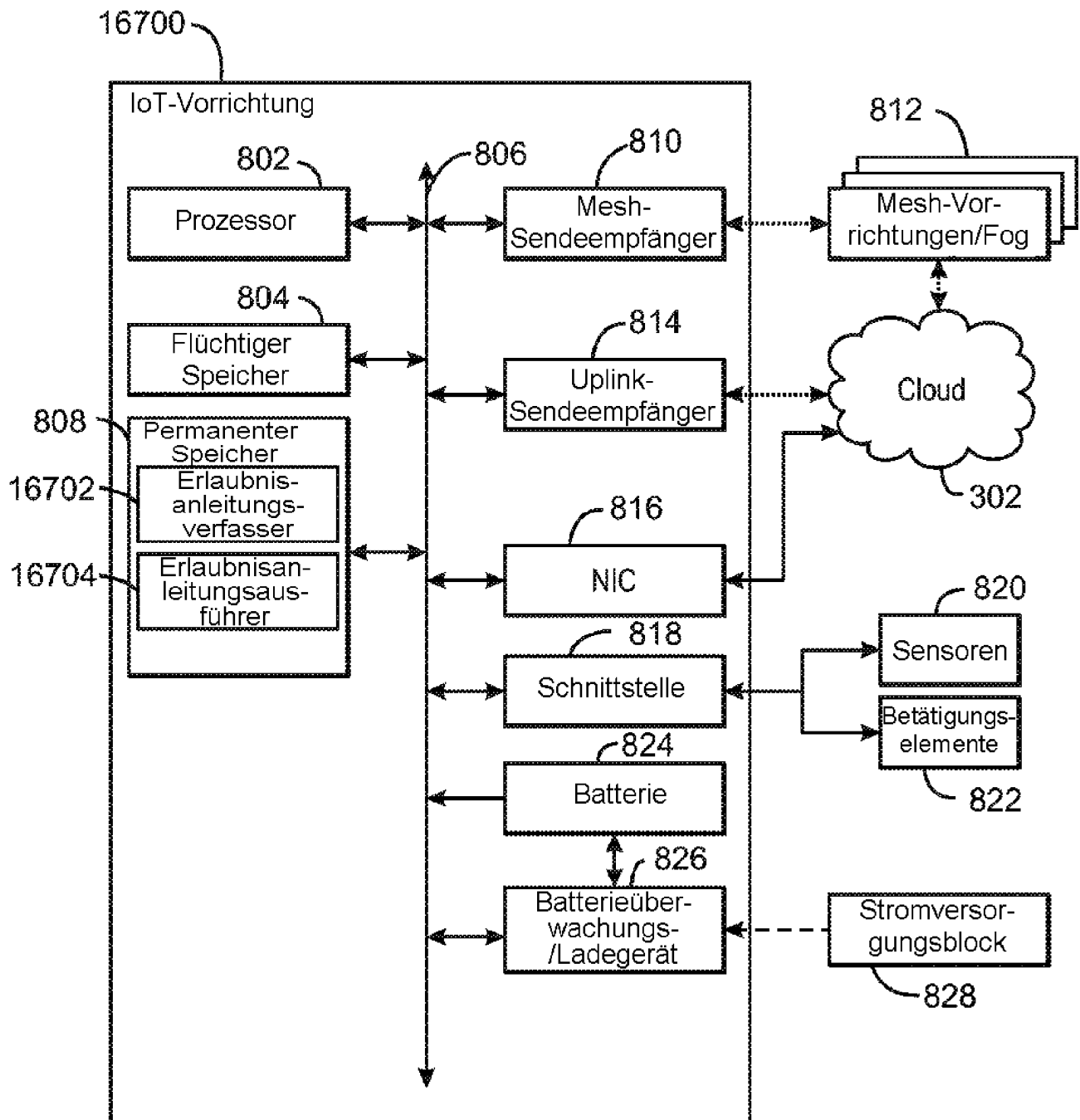


FIG. 167

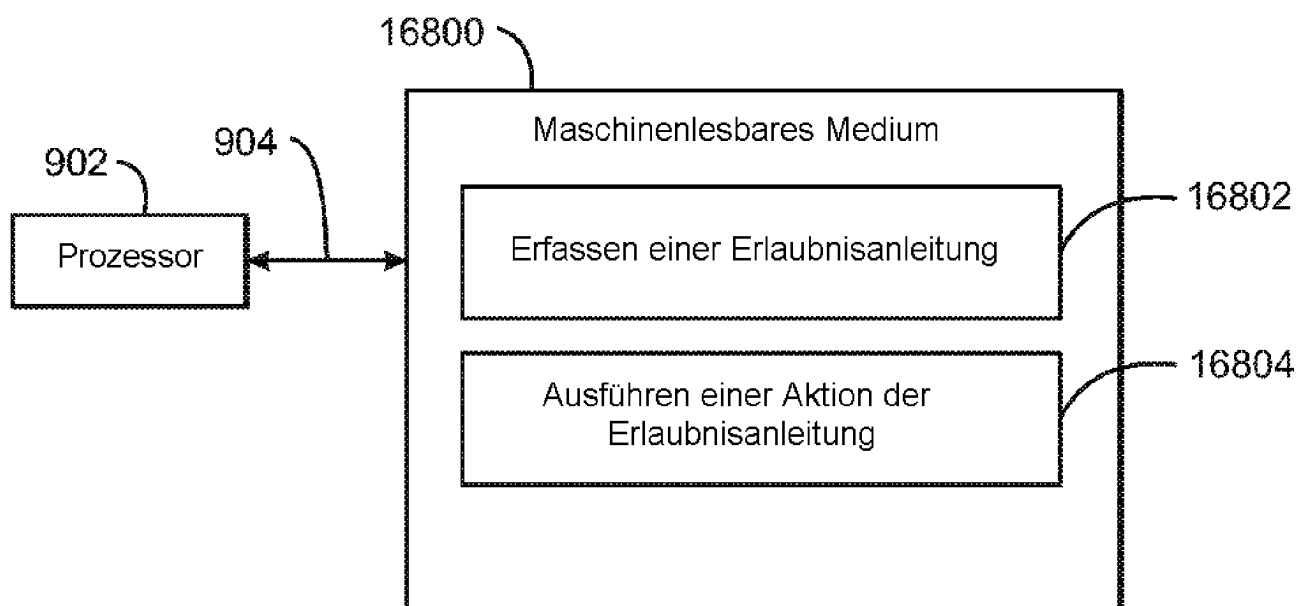
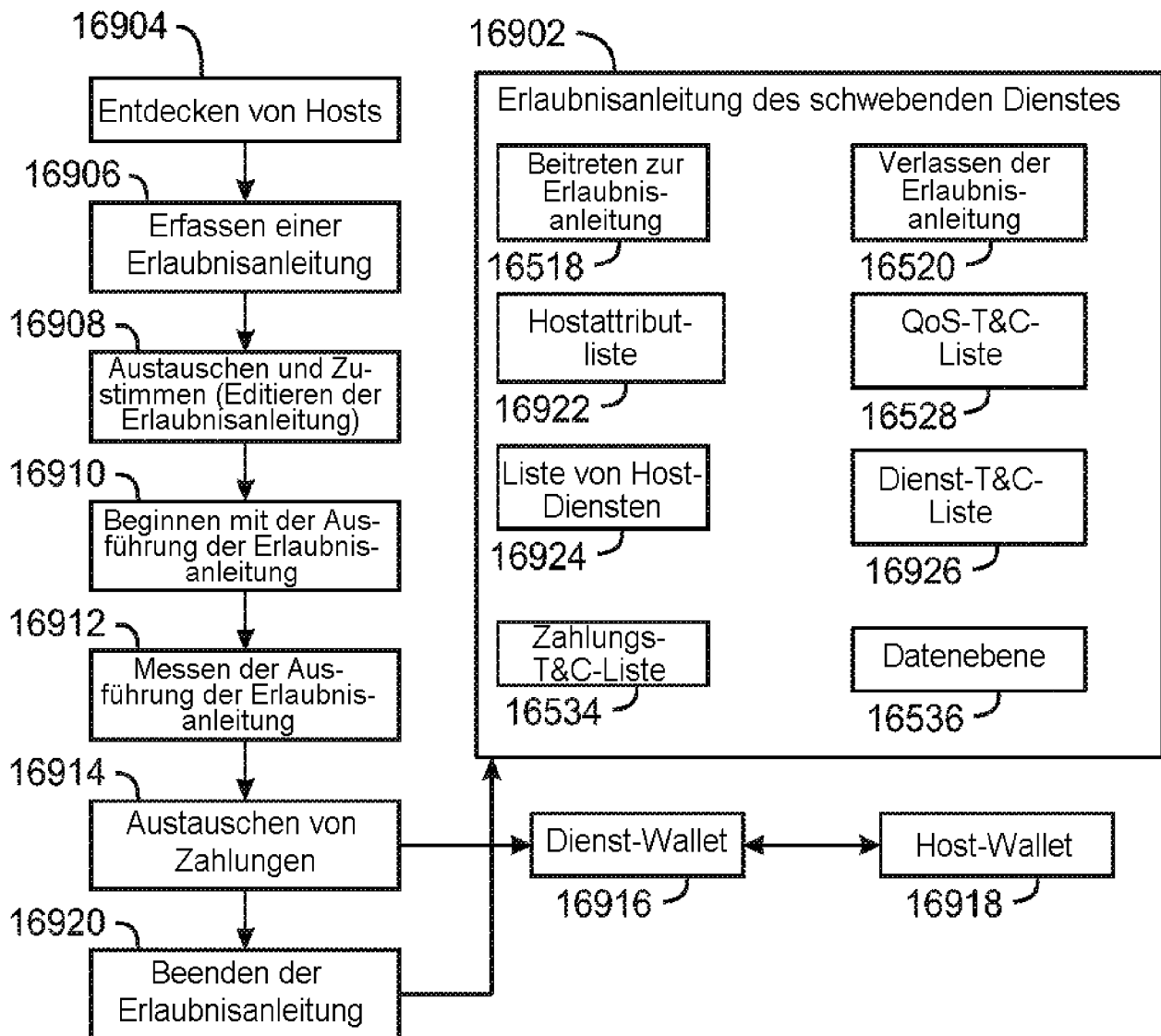


FIG. 168



16900

FIG. 169

17002

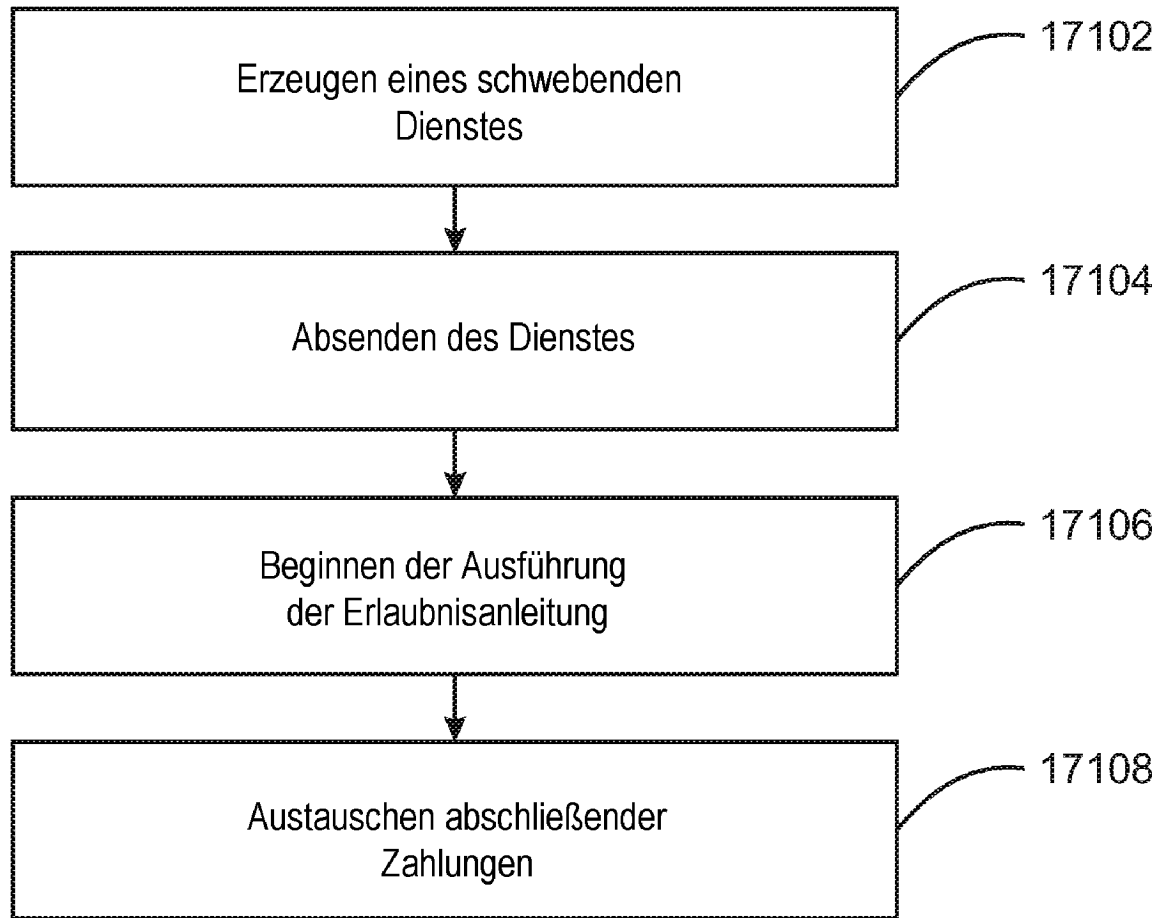
Schwebender Dienst

17000

Merkmal	Wie wichtig?	Host #1	Host #2	Host #3
Verfügbarkeit > 98%	Erforderlich (10)	10	5	10
AES vorhanden	Bevorzugt (7)	3	7	7
SGX vorhanden	Nicht erforderlich (0)	0	0	0
TXT vorhanden	Bevorzugt (5)	0	0	5
Plattenplatz > 10 GB	Erforderlich (10)	4	10	10
flüchtiger Speicher > 16 GB	Erforderlich (10)	5	10	10
Ansprechzeit < 3 s	Wünschenswert (8)	0	3	0
mehrere Knoten = ja	Bevorzugt (5)	0	5	5
Verpachtung = geteilt	Egal (2)	2	1	2
...				
...				
Kosten (pro Stunde)		0.004	0.012	0.014
Möglich?		Abgelehnt	Möglich	Akzeptabel
Entscheidungsbewertung (Wert)		0.00016667	0.00029268	0.000285714

17000

FIG. 170



17100

FIG. 171

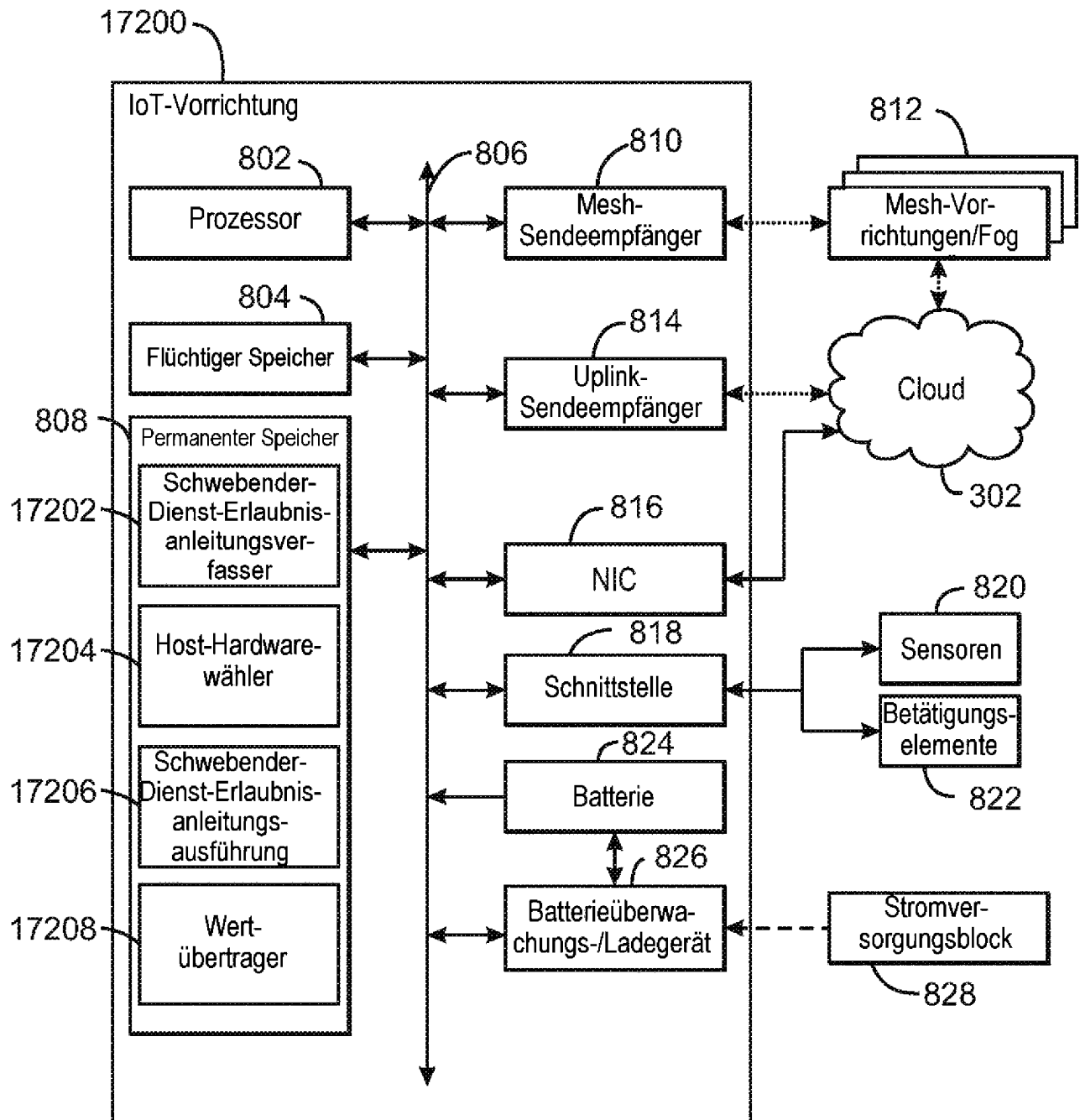


FIG. 172

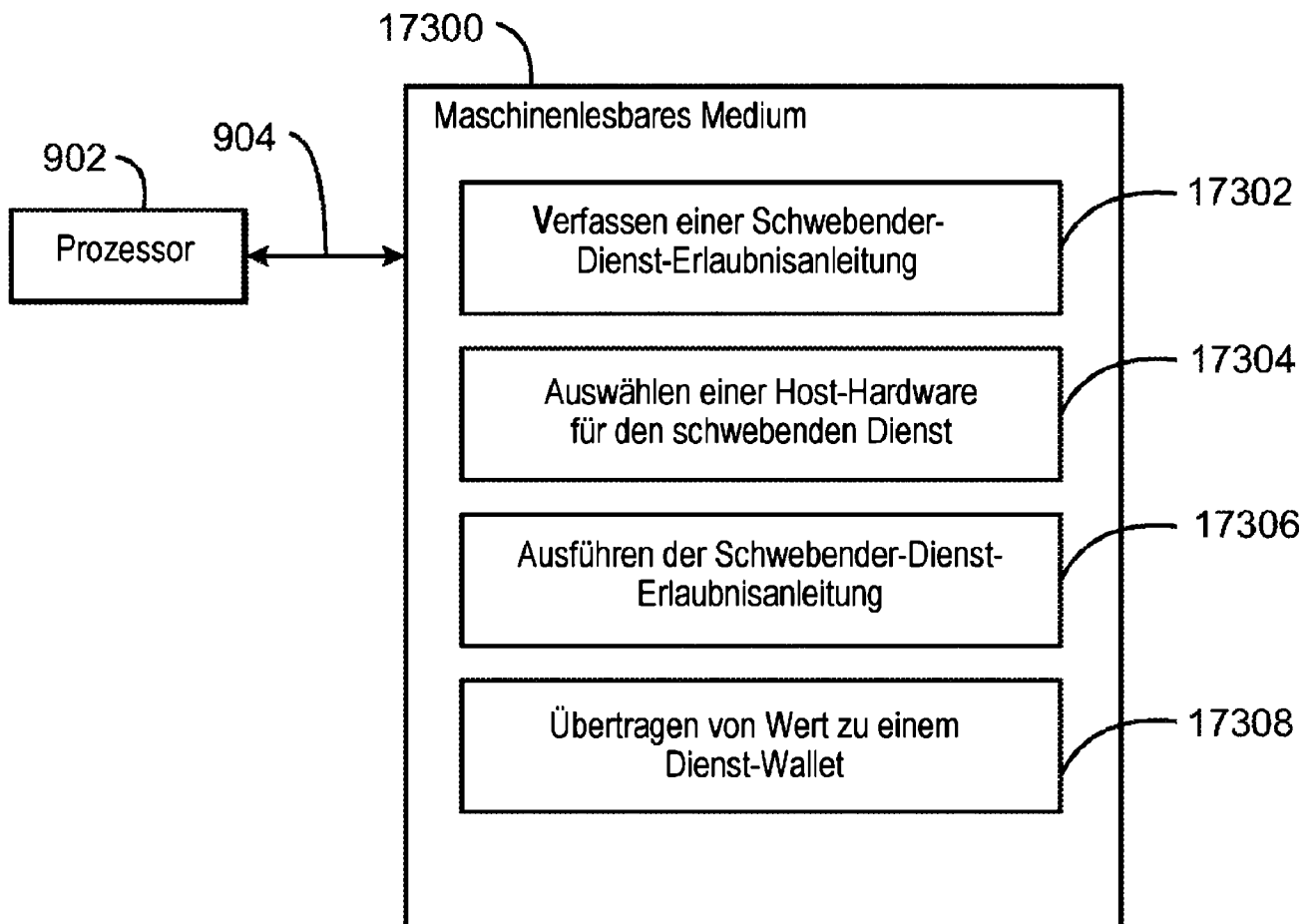


FIG. 173

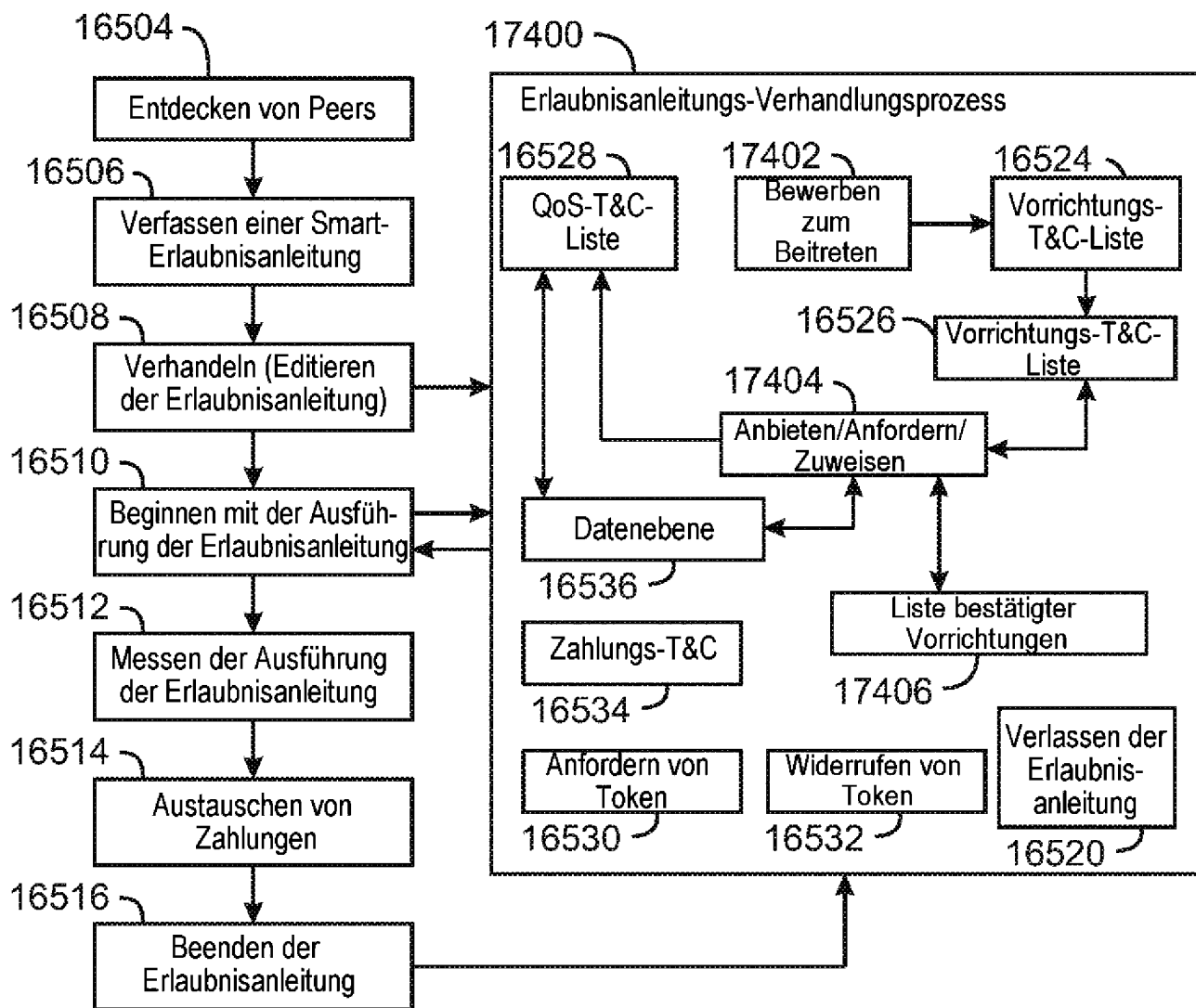
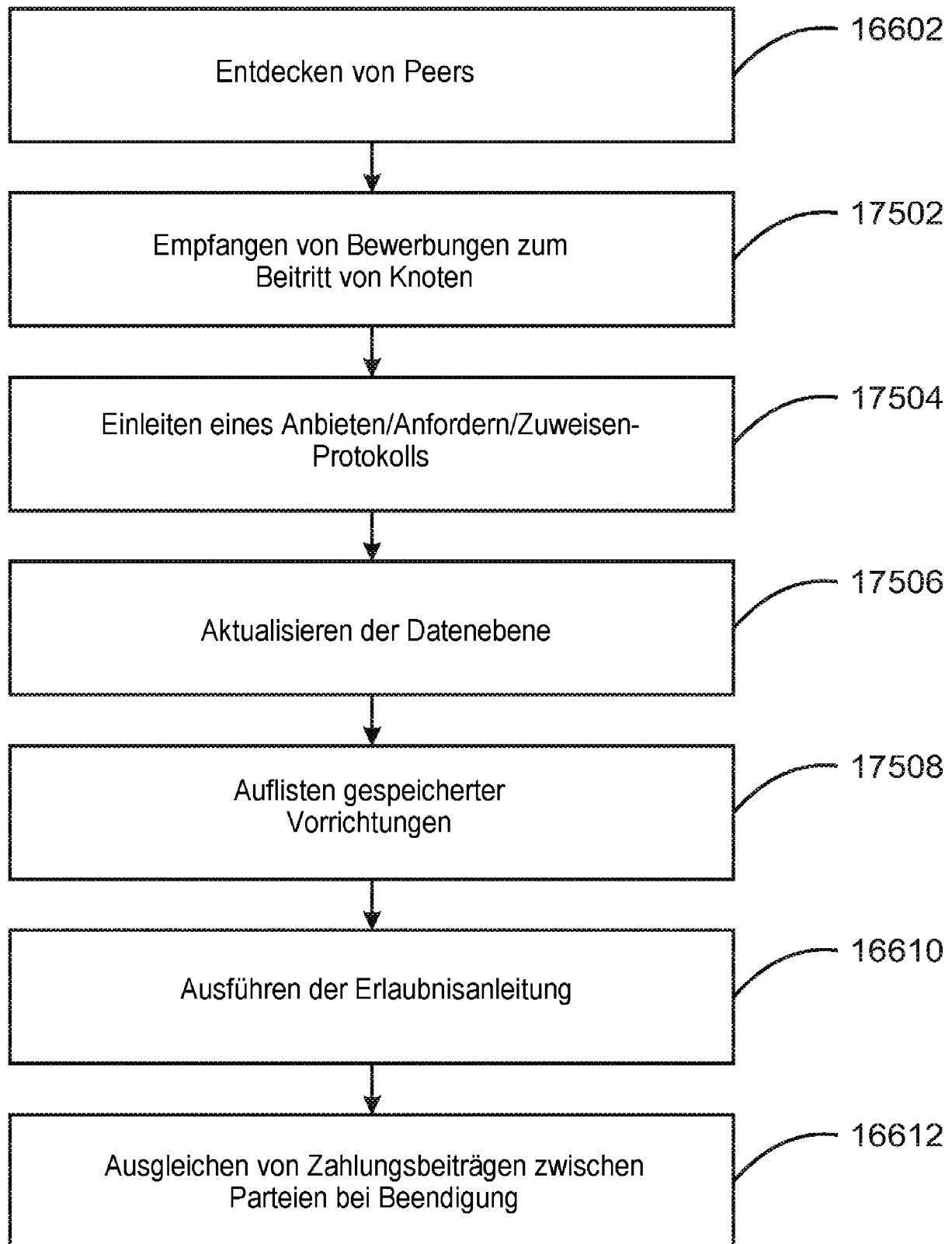


FIG. 174



17500
FIG. 175

17600

Knoten/ Ort	Typ	Bin	% Nicht- Ereig- nisse	% der Ereig- nisse	WoE $\log_n$ (%-Nicht- Ereignisse/% Ereignisse)	Div	Cacq (\$IC)
001/FS-URI	RAW-CO2	1	5%	6%	-0.0952	0.0005	0.5
001/FS-URI	RAW-CO2	2	12%	10%	0.2002	0.0045	0.5
001/FS-URI	RAW-CO2	3	13%	12%	0.1522	0.0029	0.5
001/FS-URI	RAW-CO2	4	16%	15%	0.0774	0.0009	0.5
001/FS-URI	RAW-CO2	5	17%	16%	0.0401	0.0003	0.5
001/FS-URI	RAW-CO2	6	16%	16%	-0.0236	0.0001	0.5
001/FS-URI	RAW-CO2	7	38%	12%	1.1379	0.2926	0.5
001/FS-URI	RAW-CO2	8	5%	7%	-0.4123	0.0095	0.5
001/FS-URI	RAW-CO2	9	5%	6%	-0.2123	0.0025	0.5
	Total:					0.3138	

17602

FIG. 176

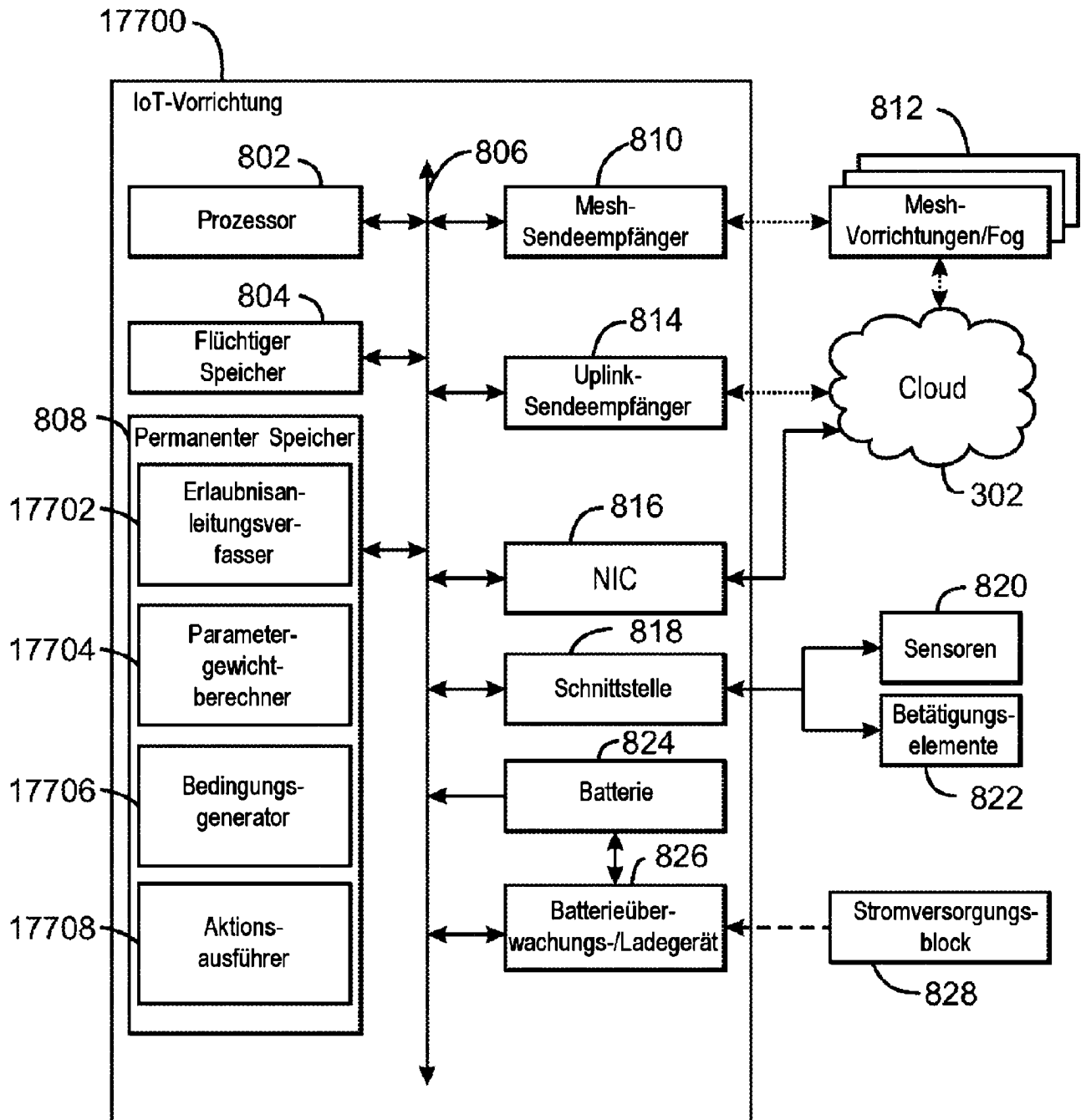


FIG. 177

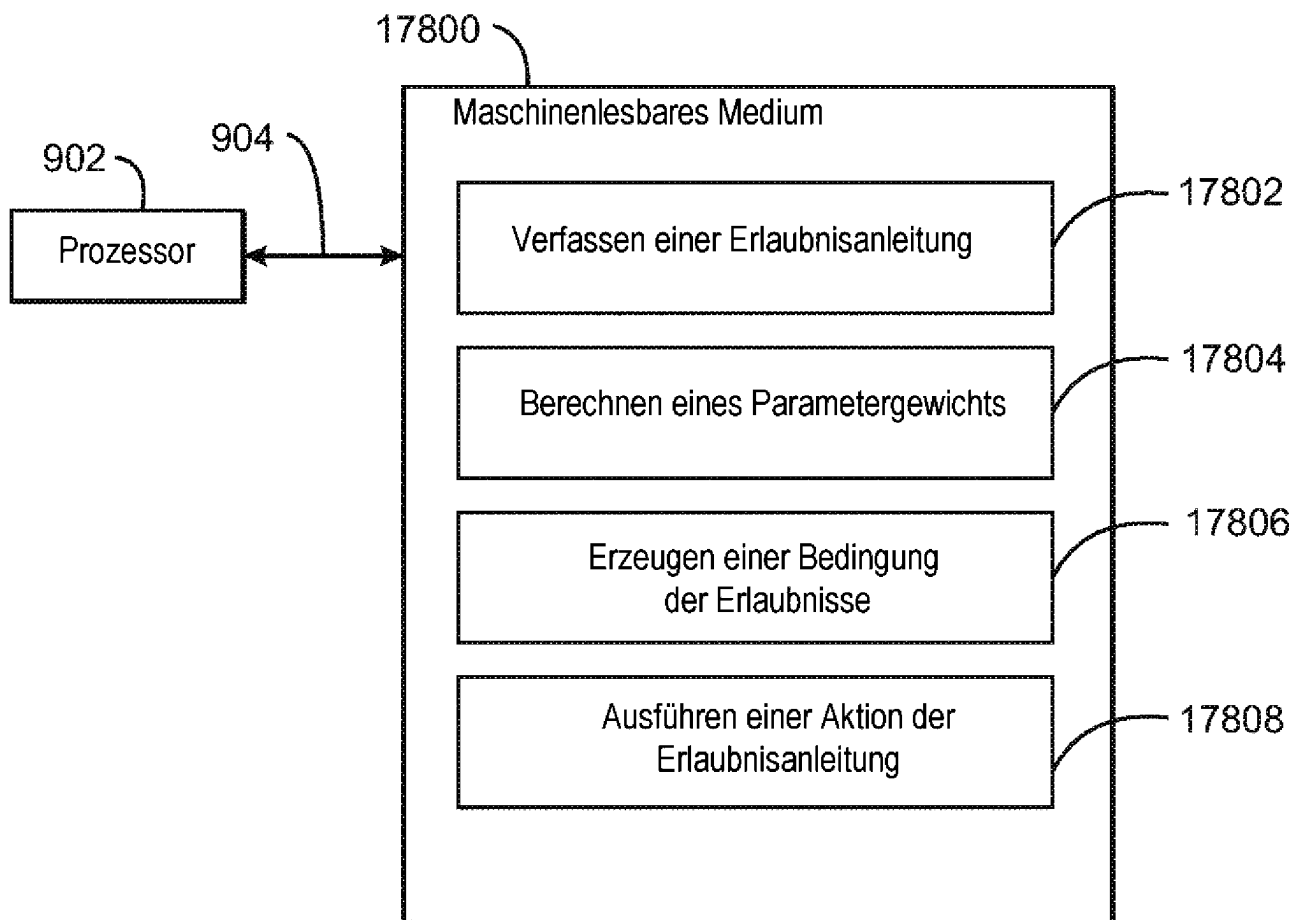
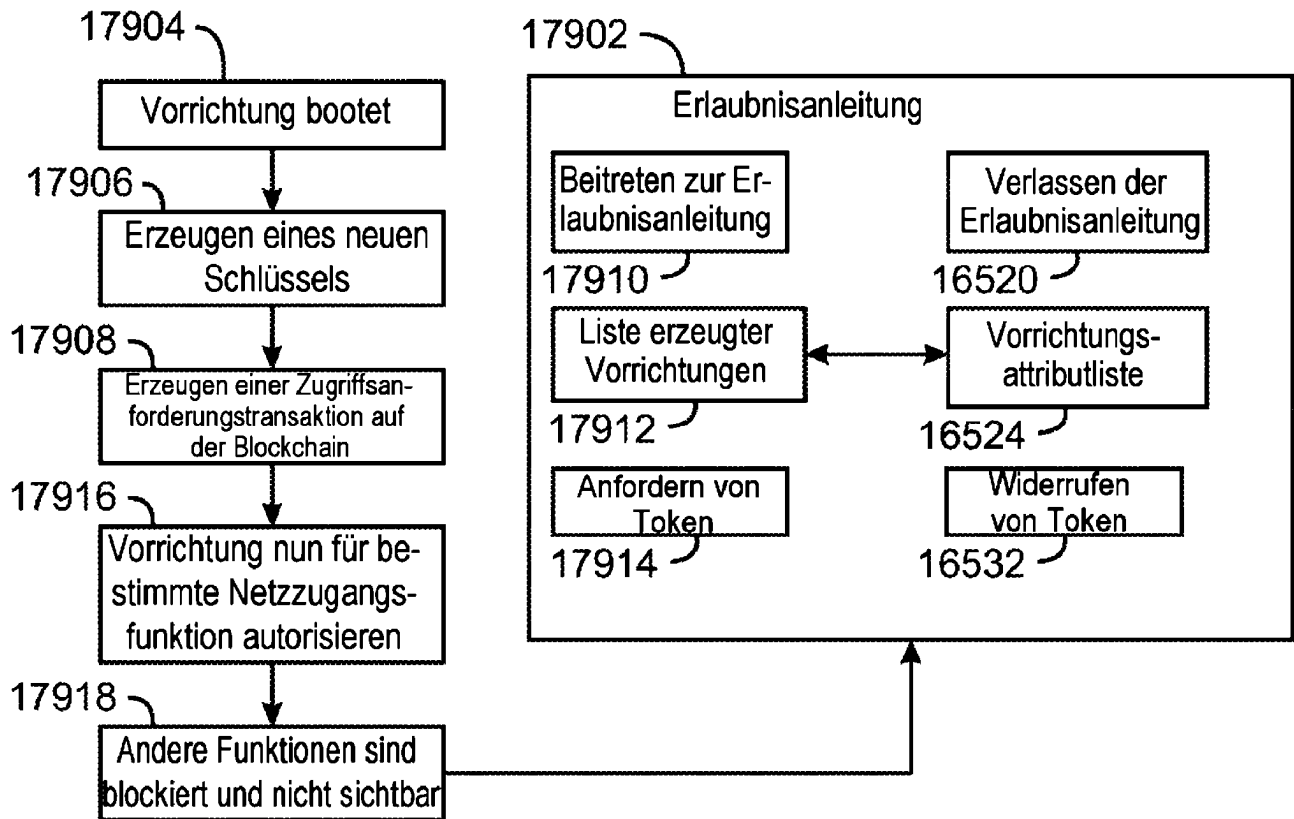
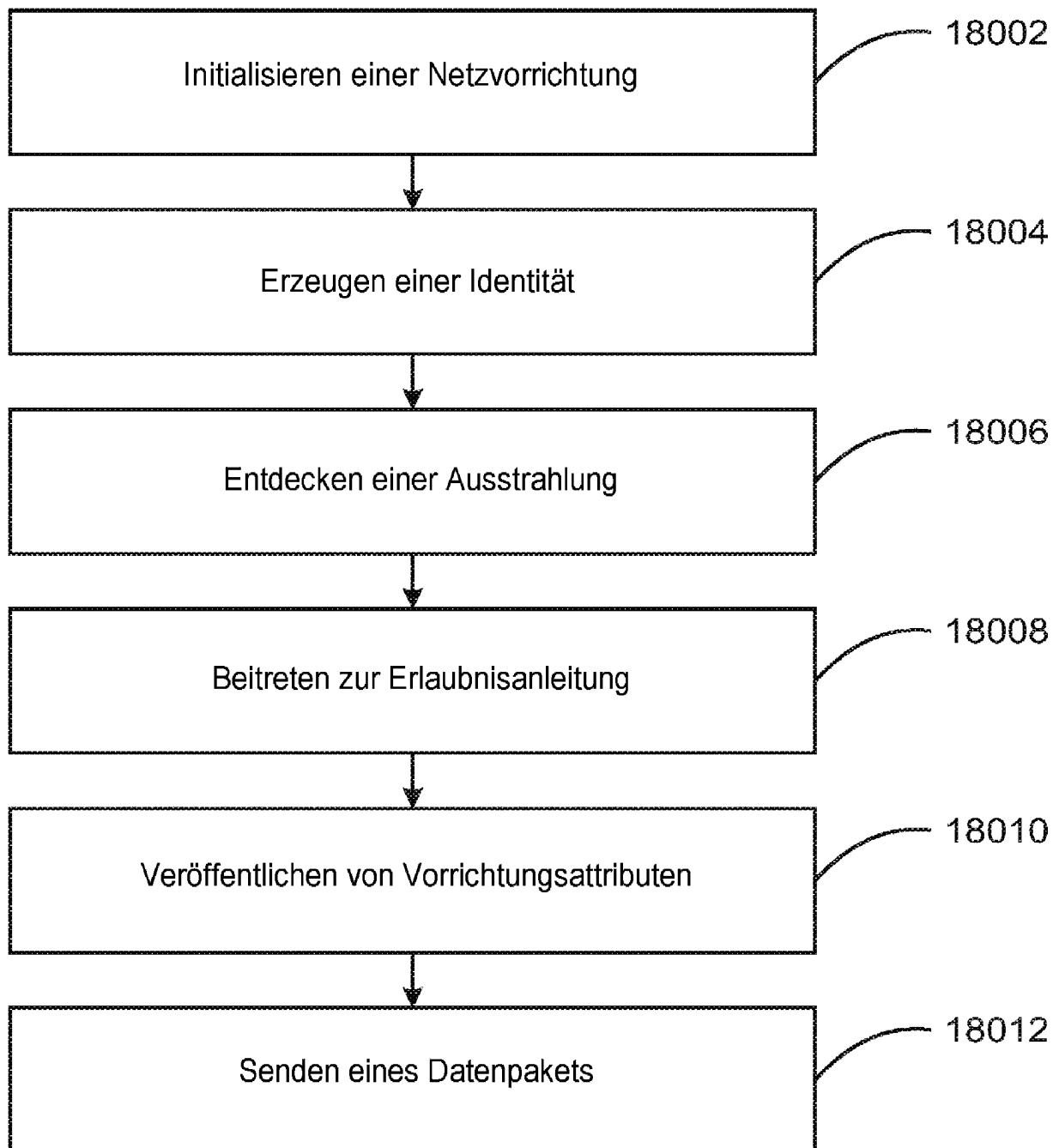


FIG. 178



17900
FIG. 179



18000
FIG. 180

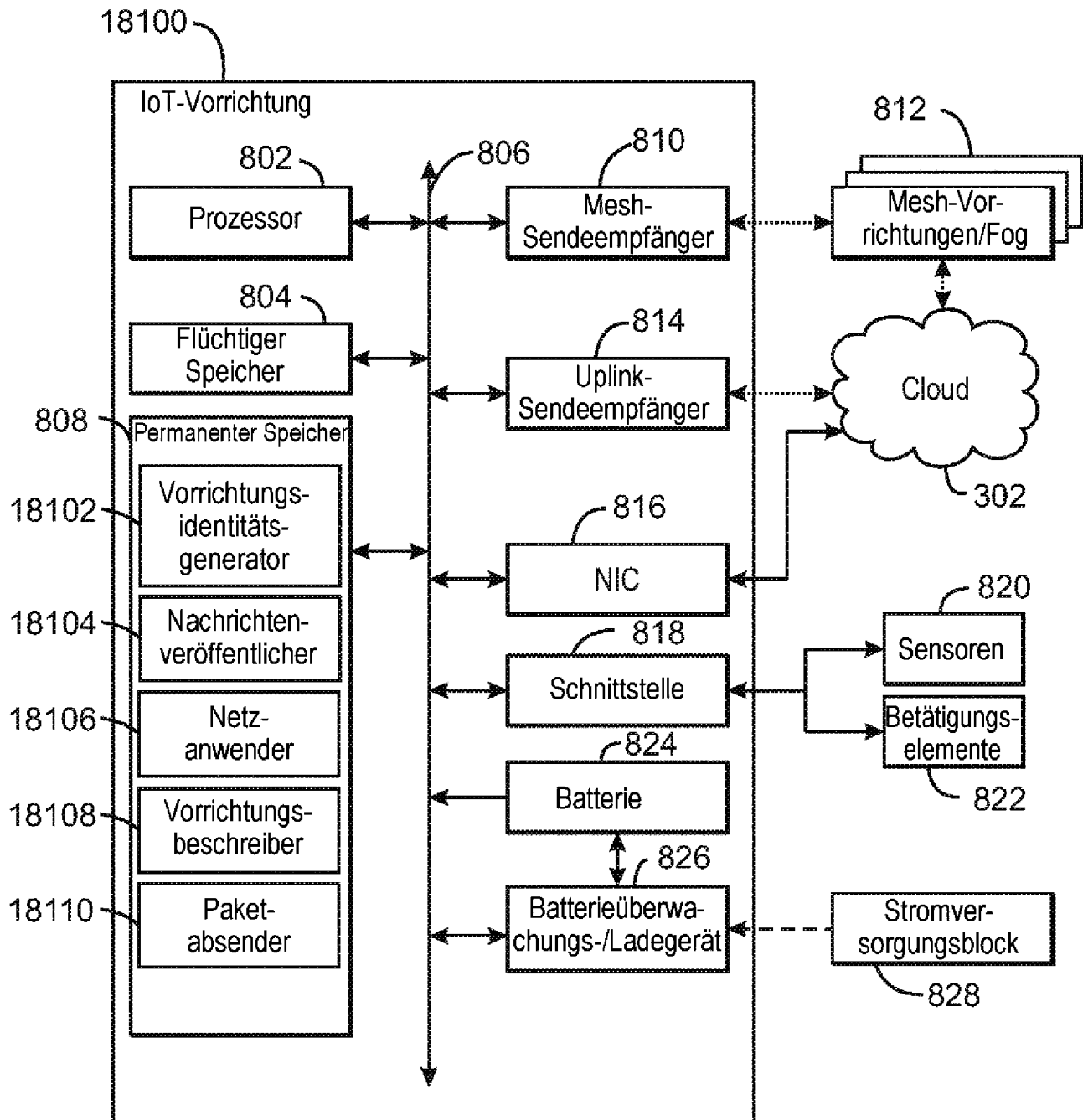


FIG. 181

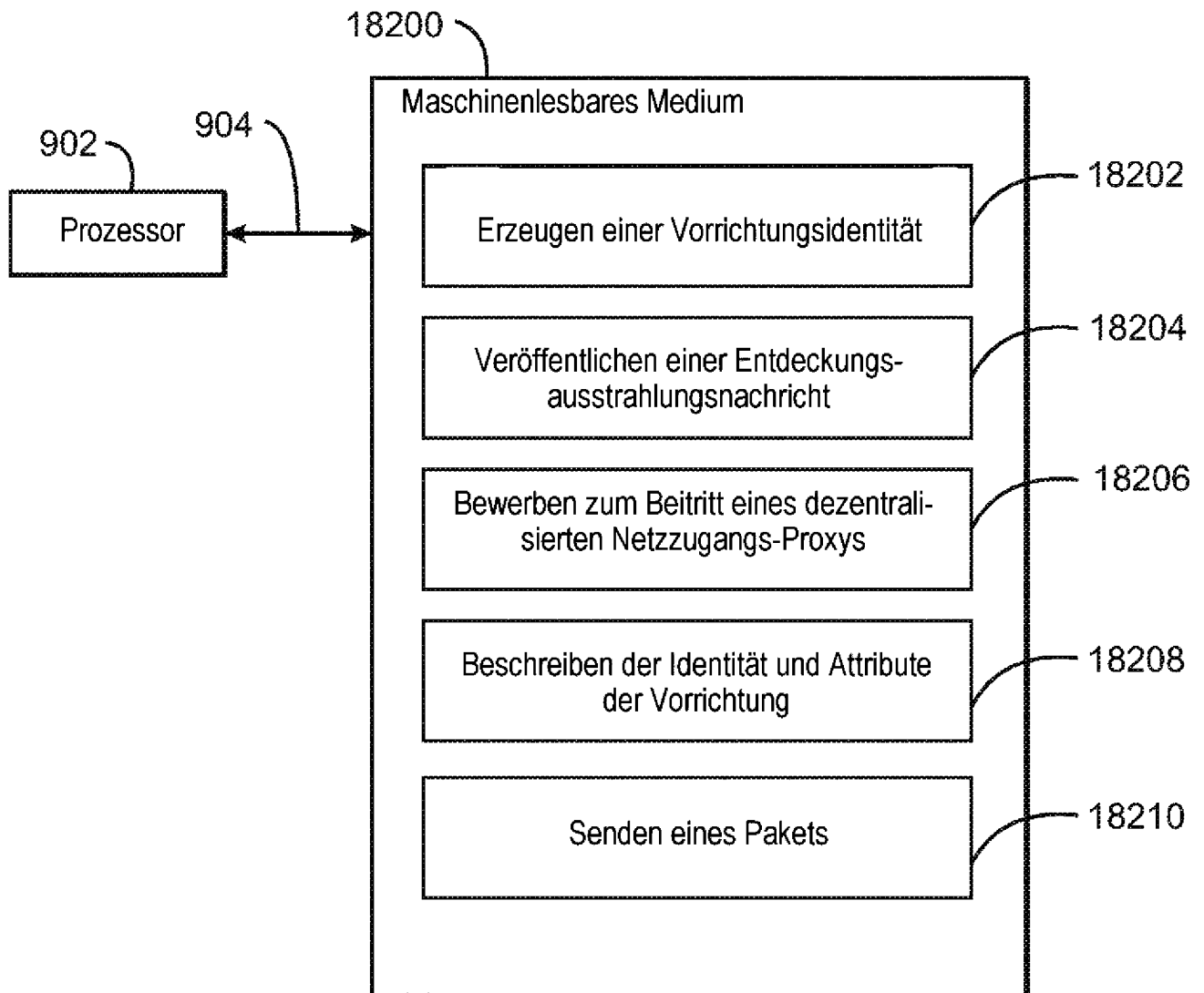
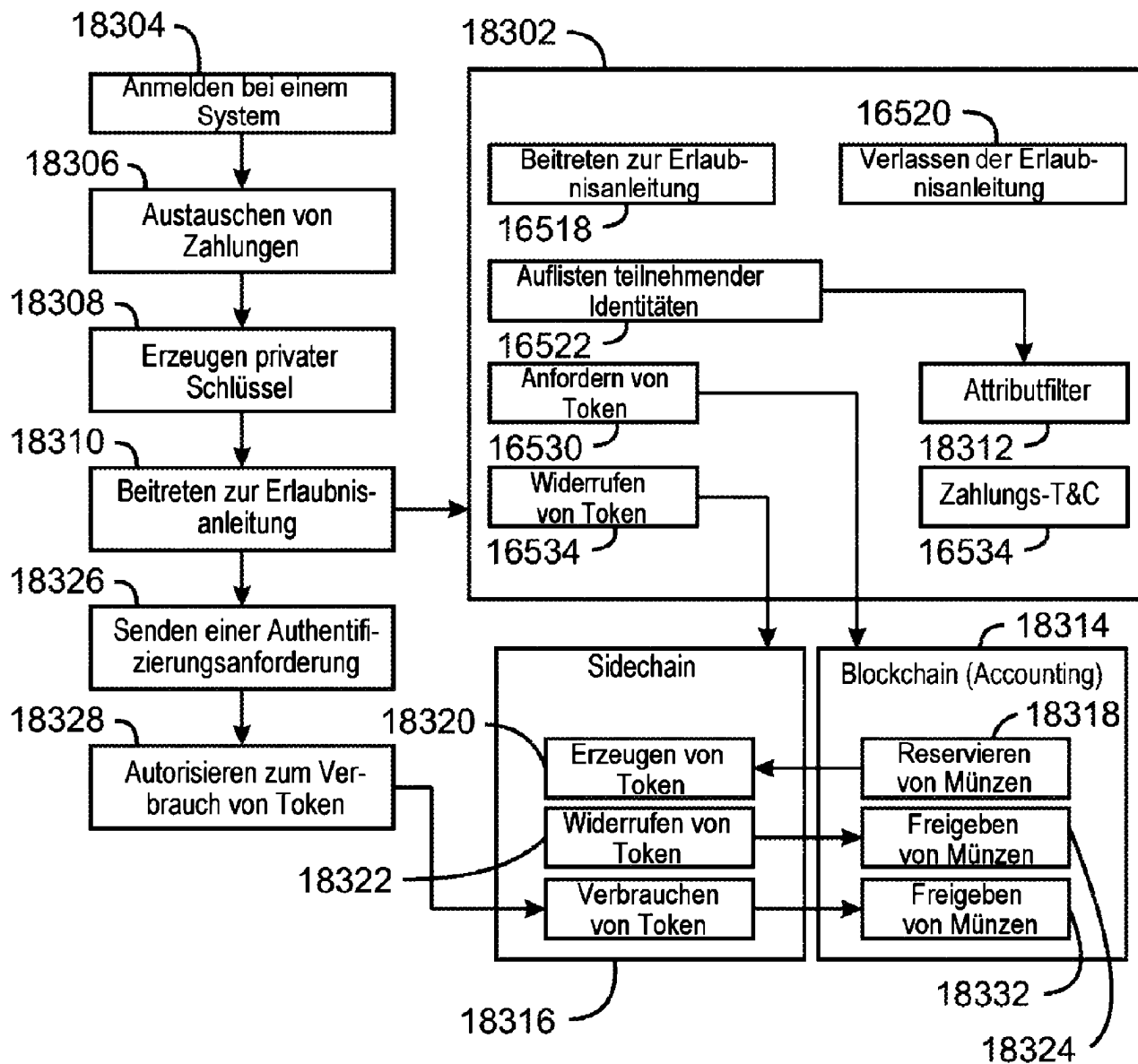
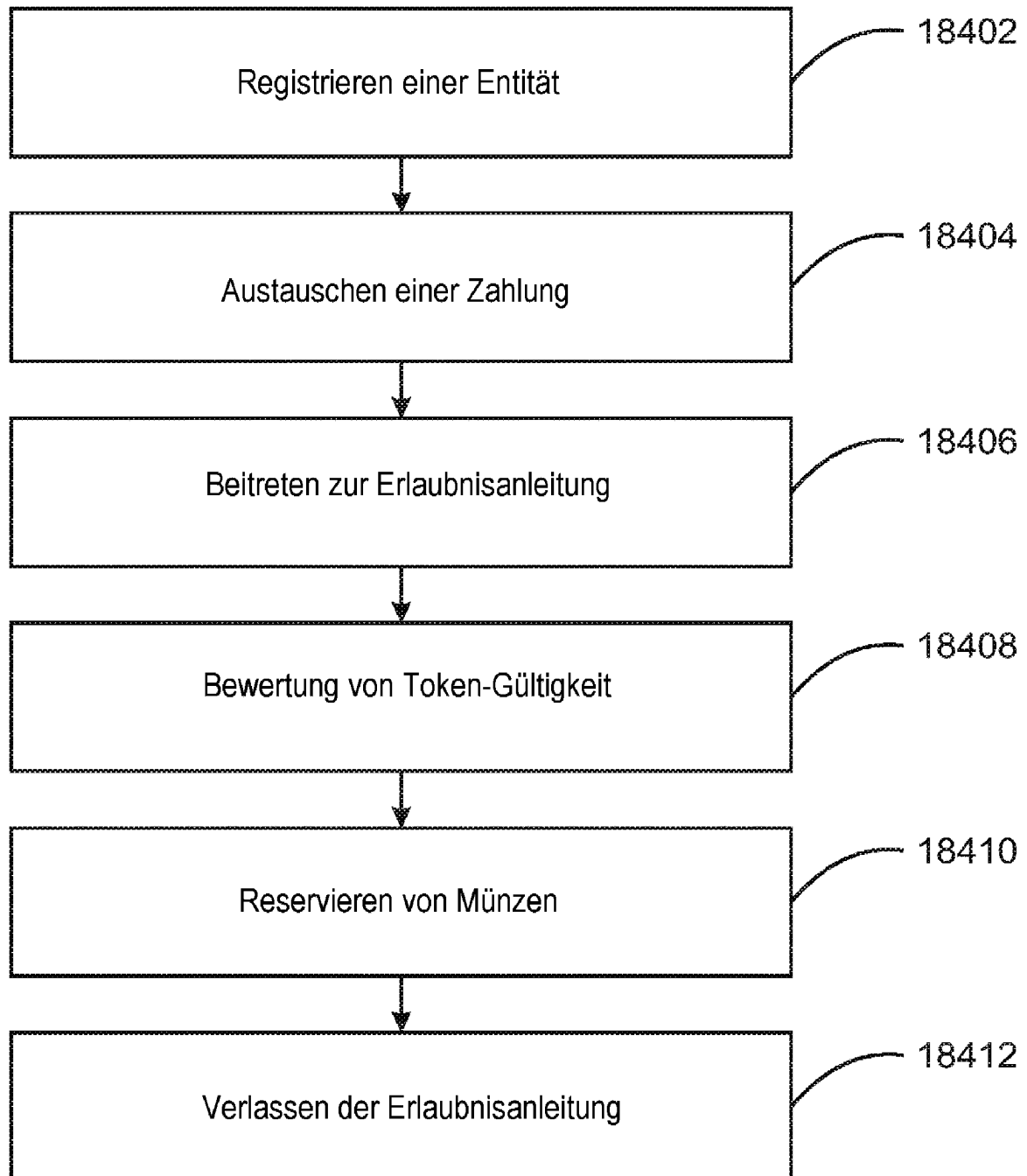


FIG. 182



18300
FIG. 183



18400
FIG. 184

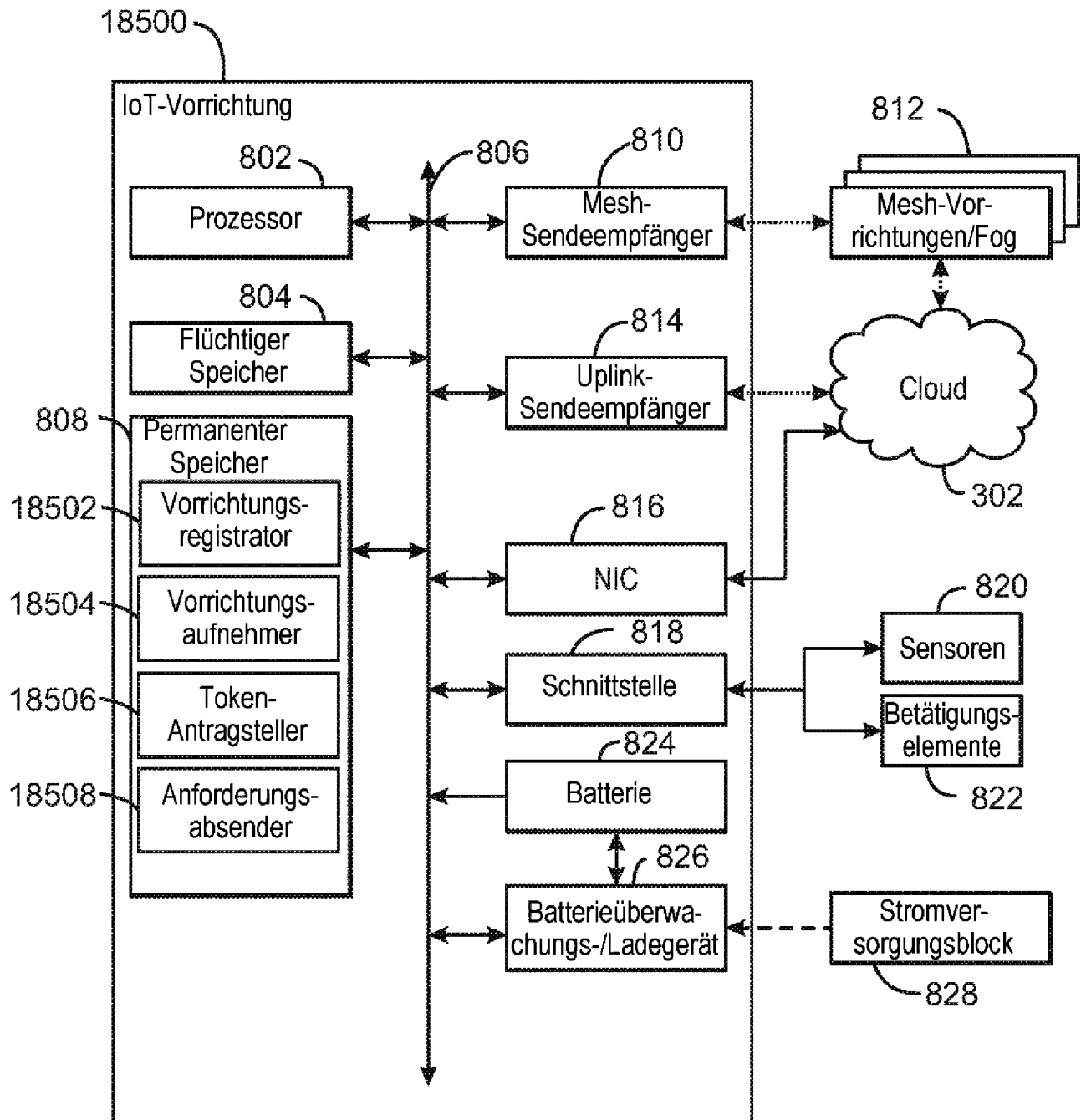


FIG. 185

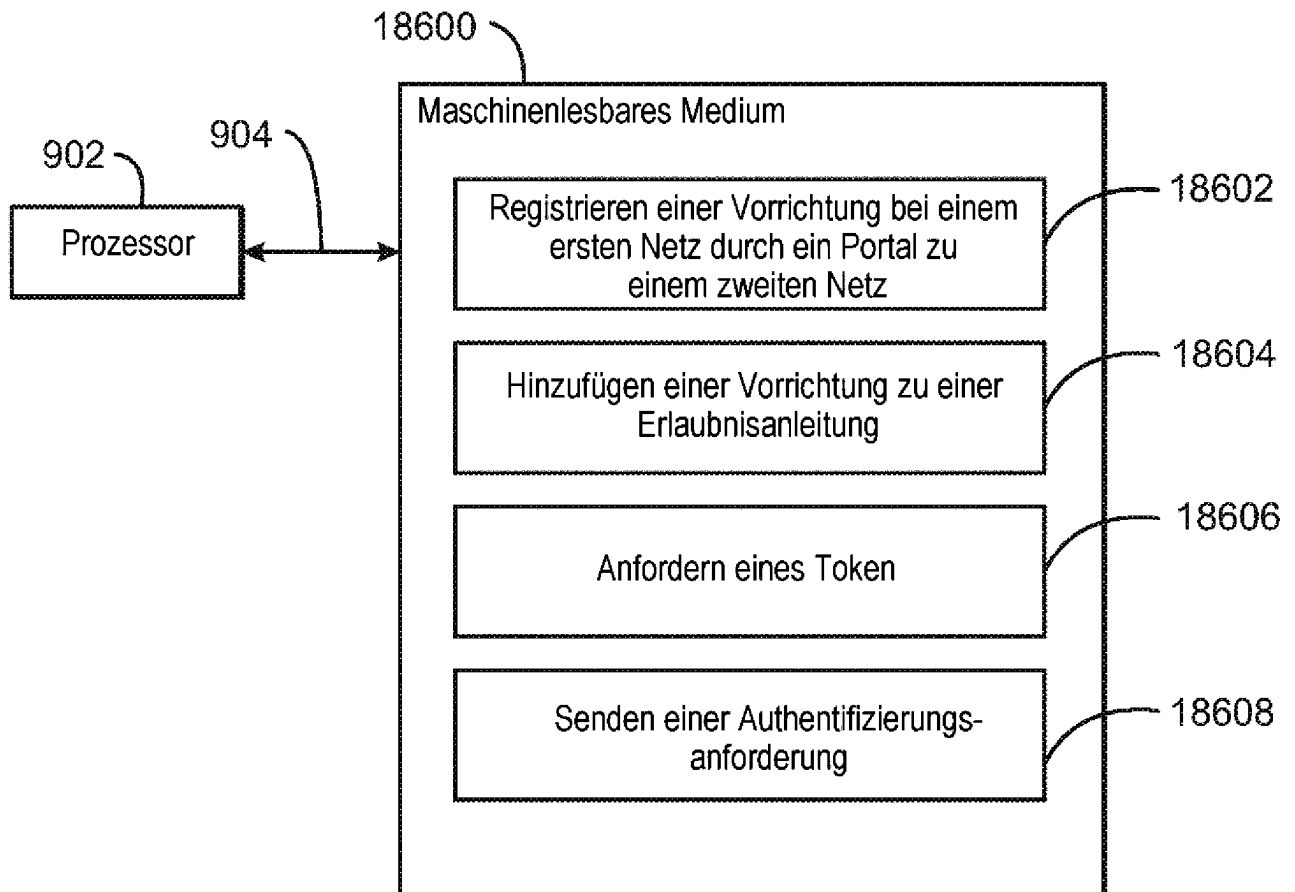
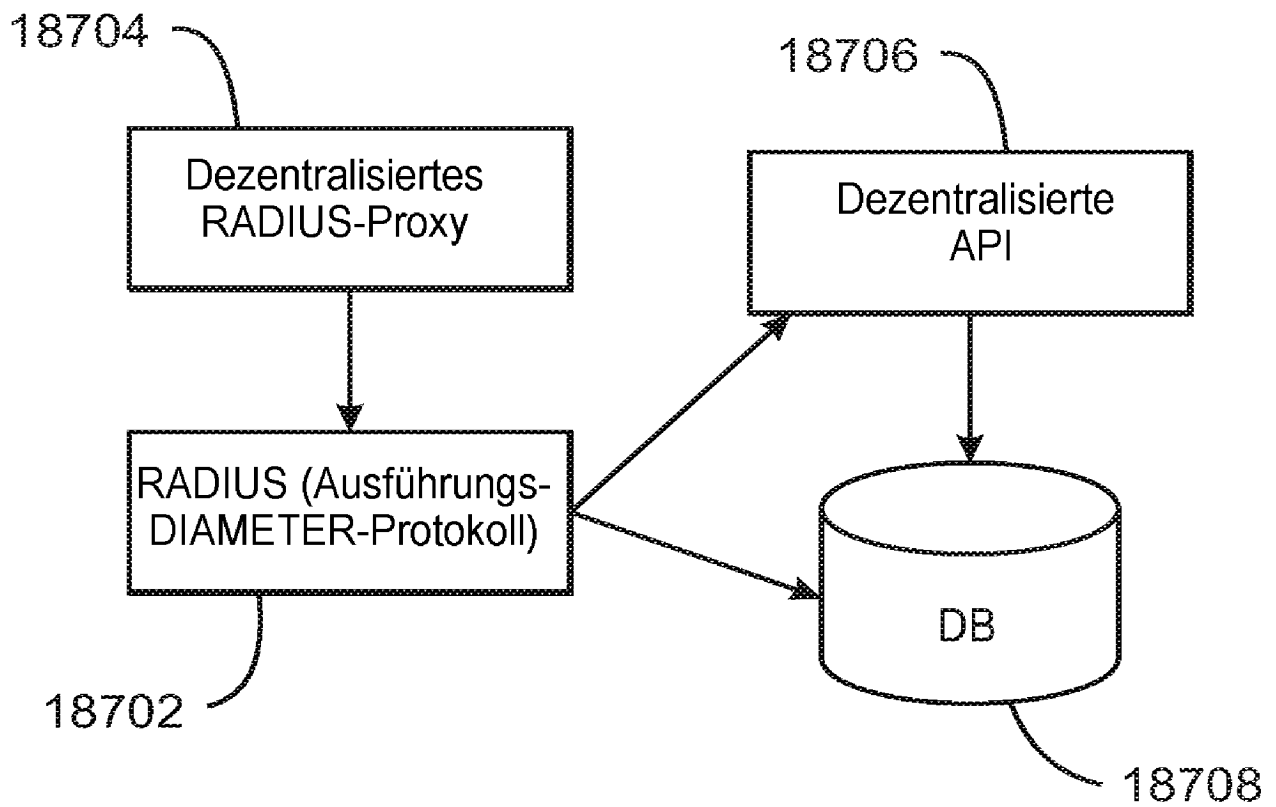
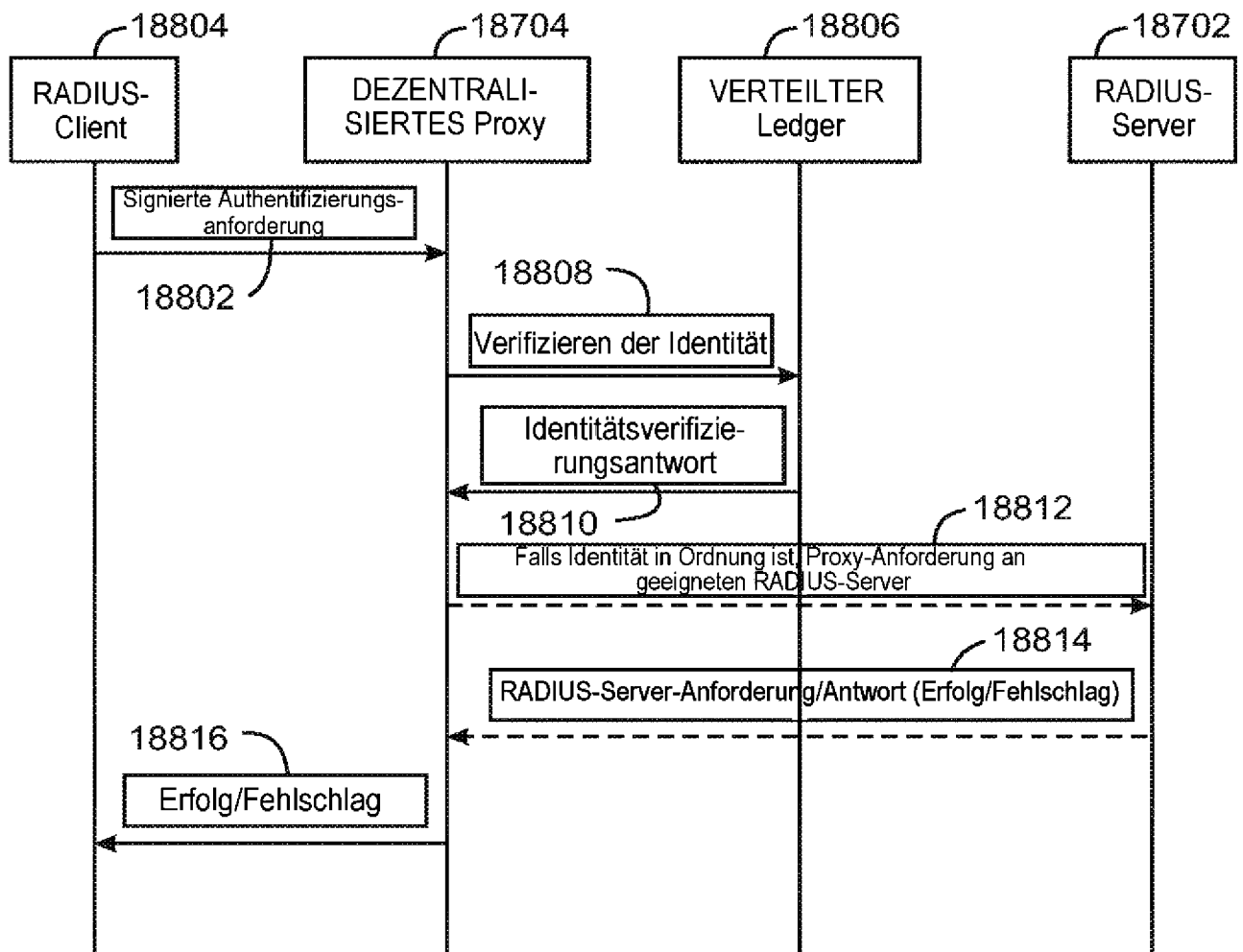


FIG. 186



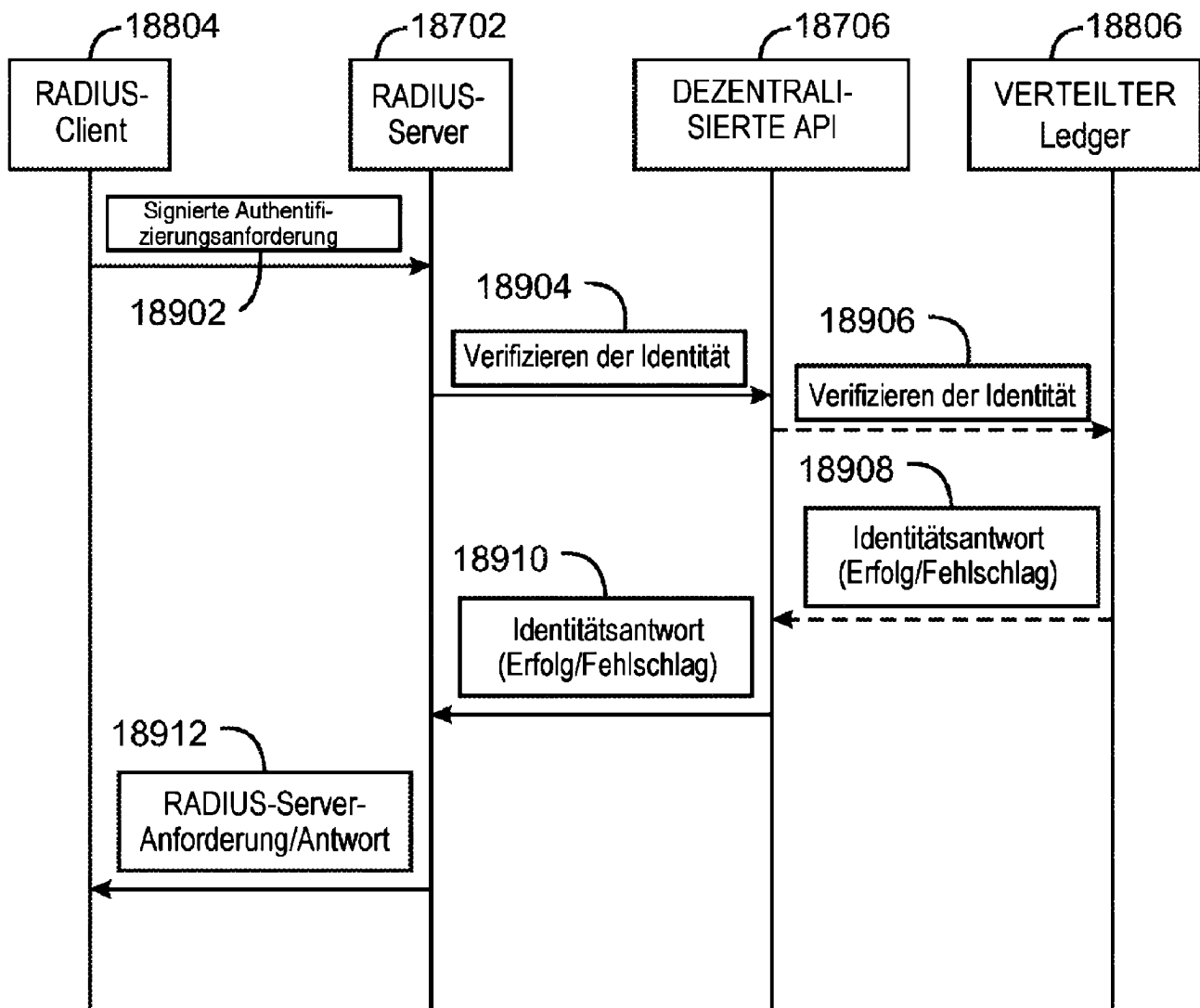
18700

FIG. 187

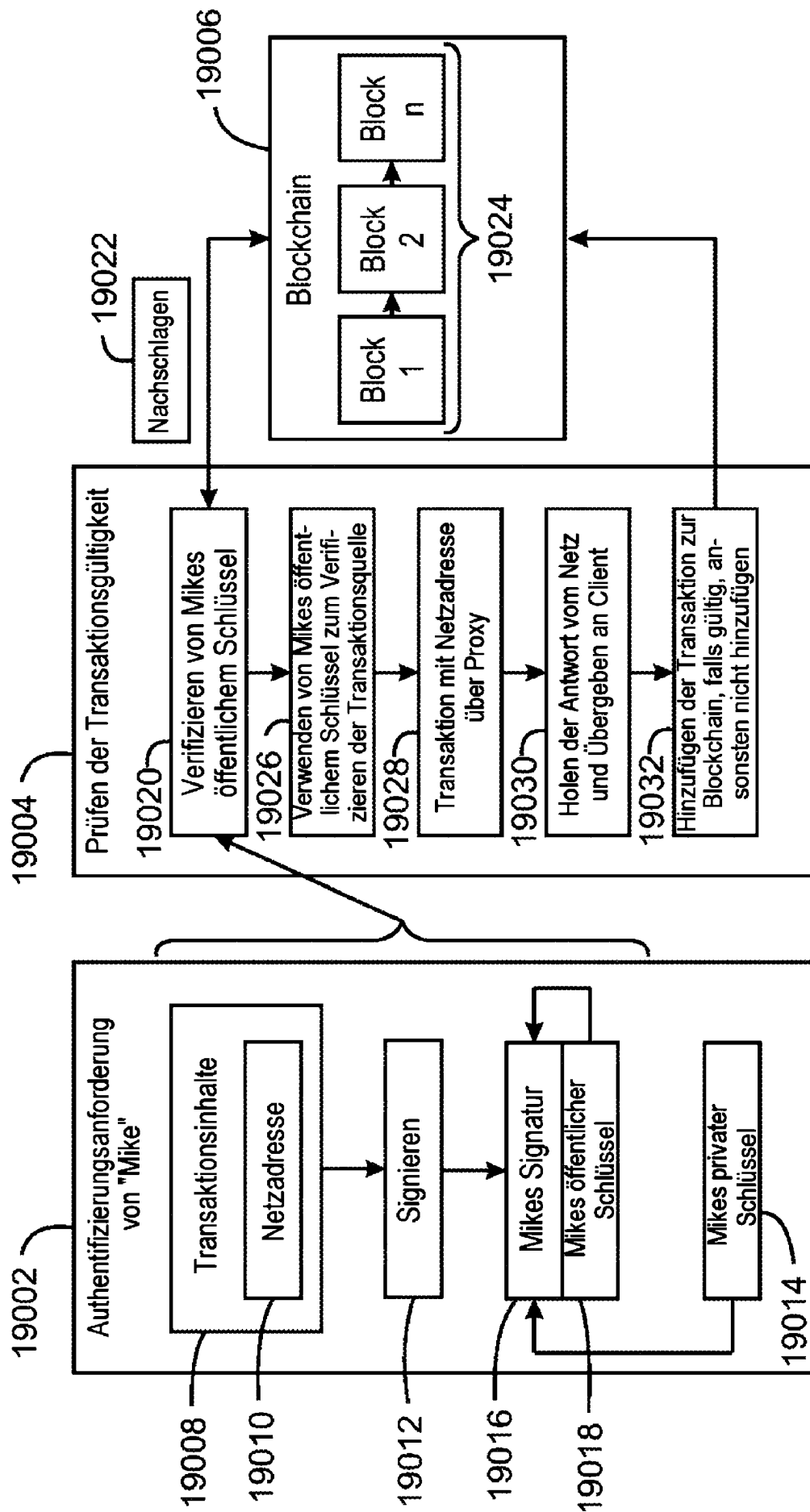


18800

FIG. 188



18900
FIG. 189



19000

FIG. 190

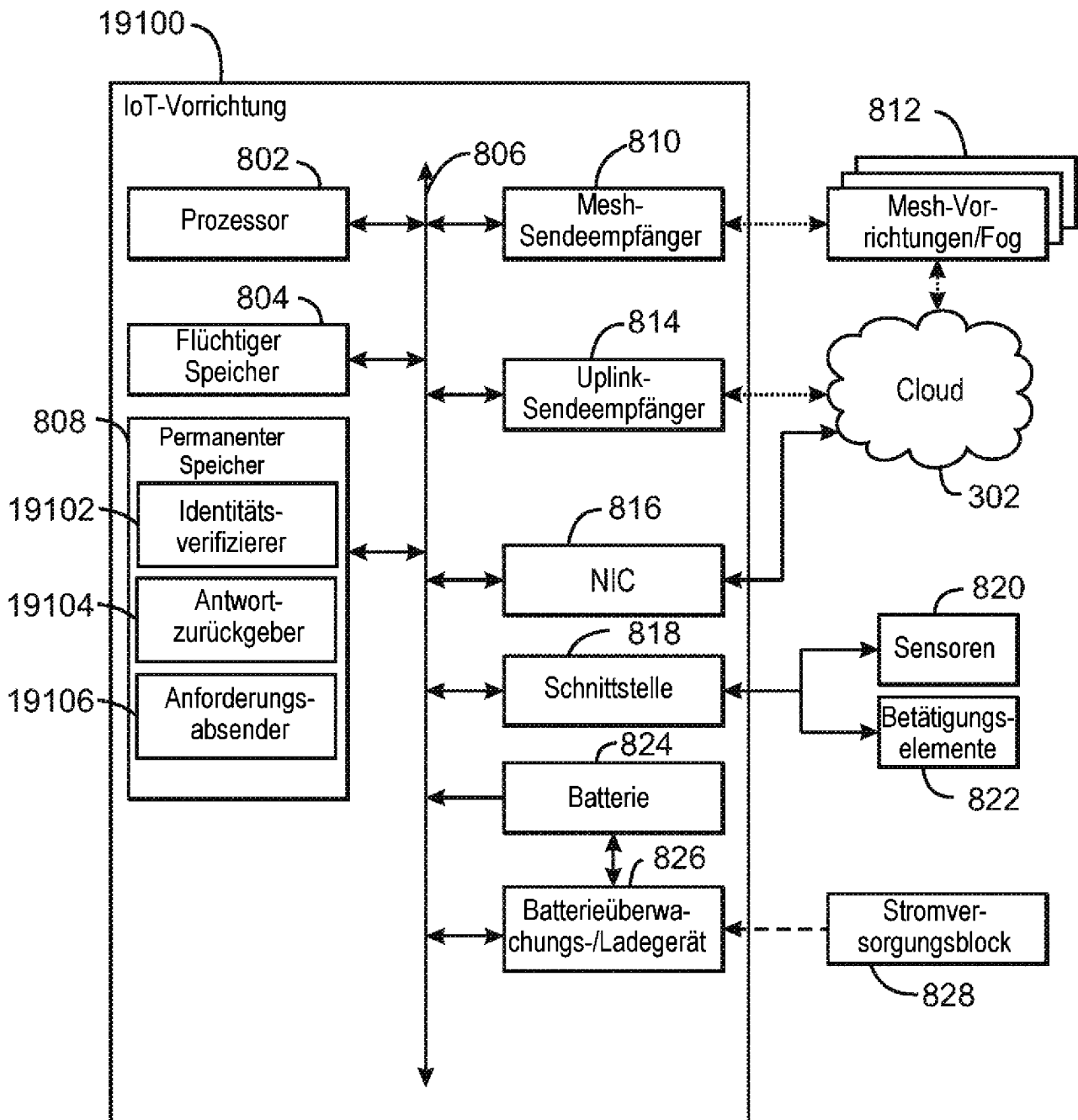


FIG. 191

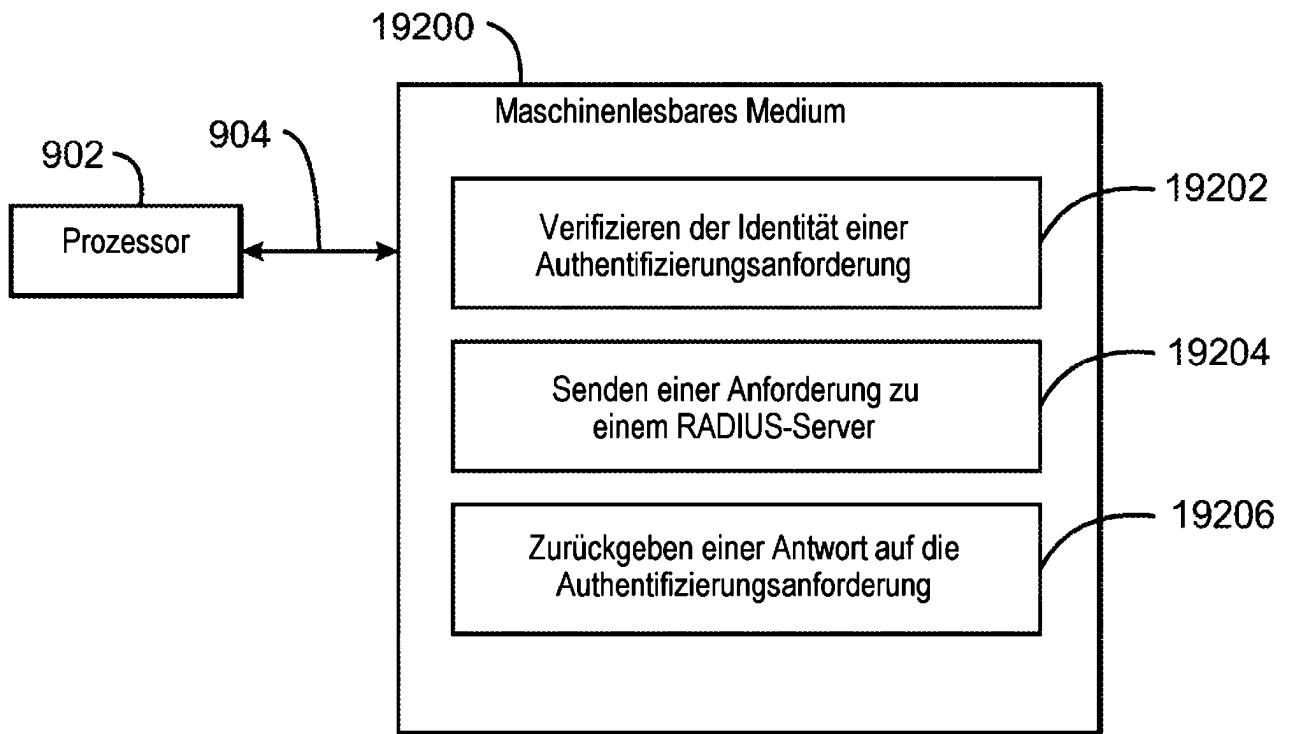
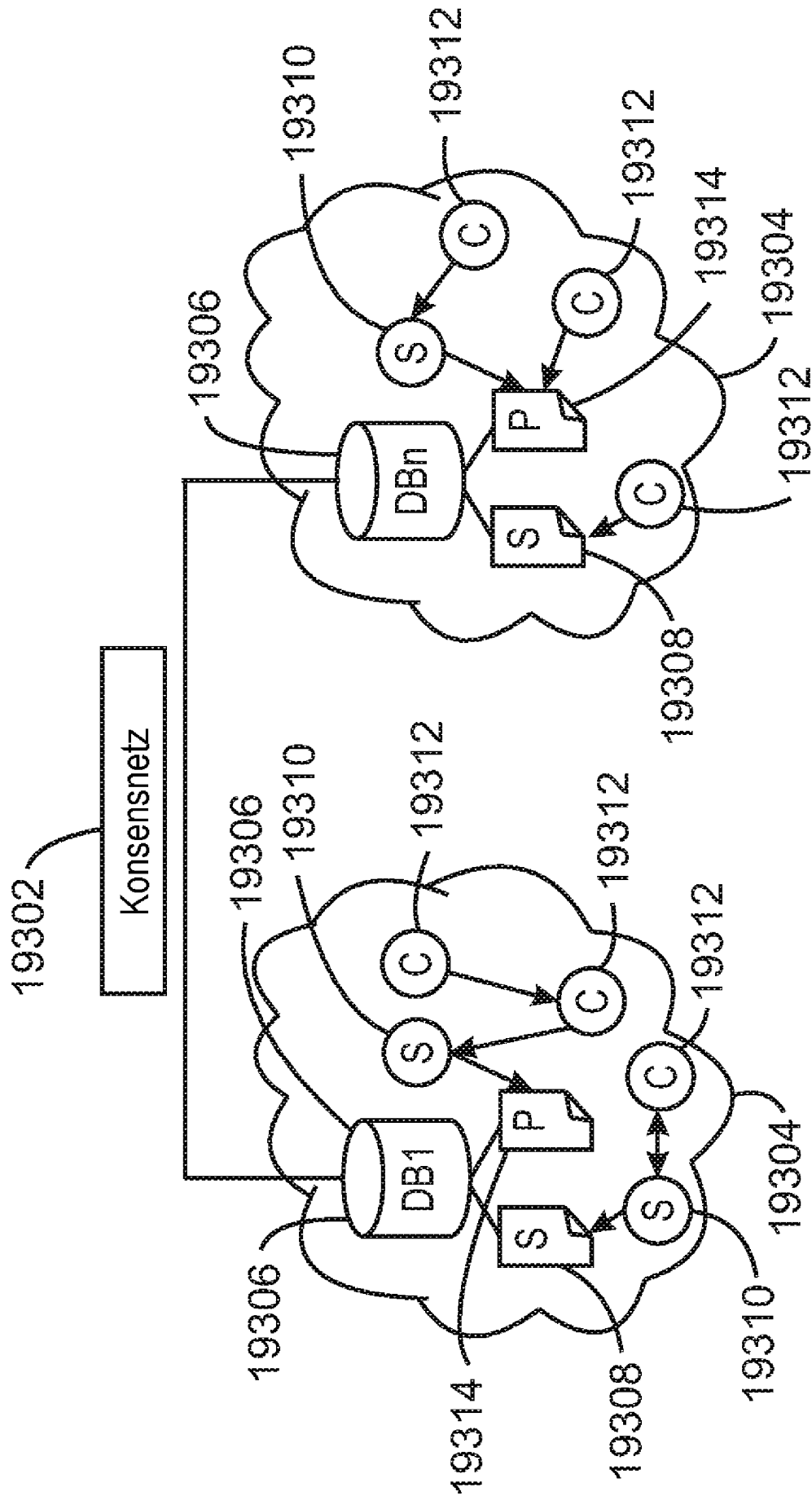
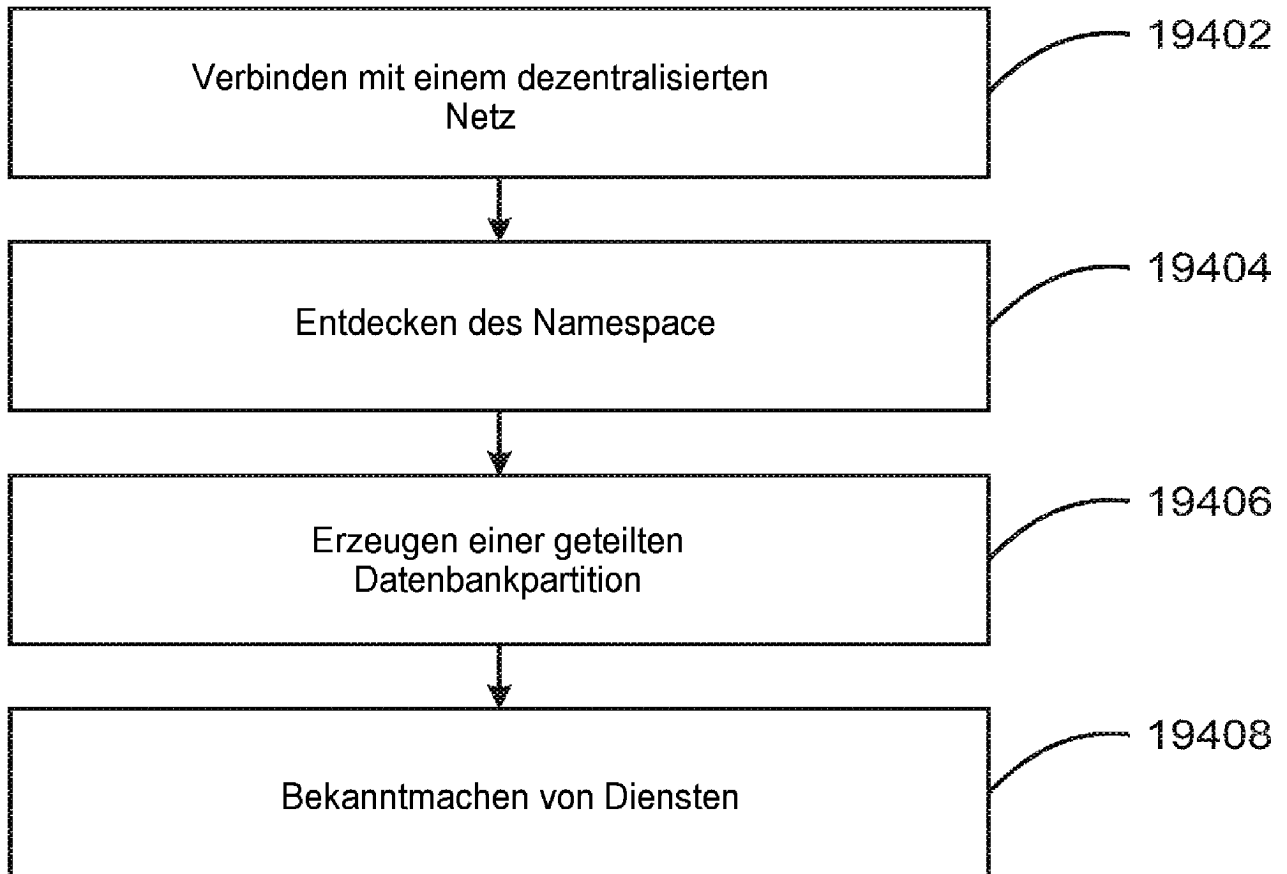


FIG. 192



19300

FIG. 193



19400
FIG. 194

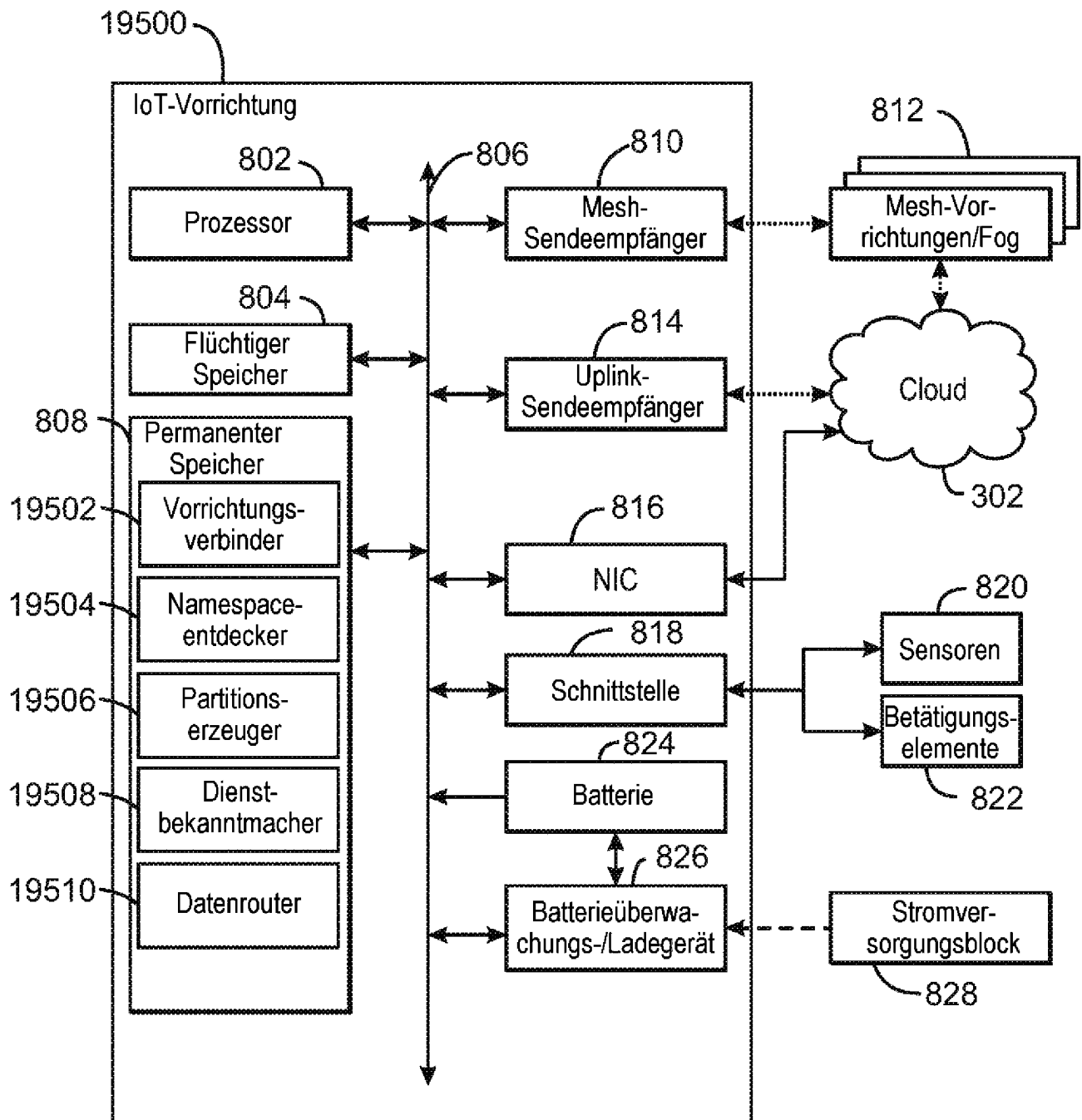


FIG. 195

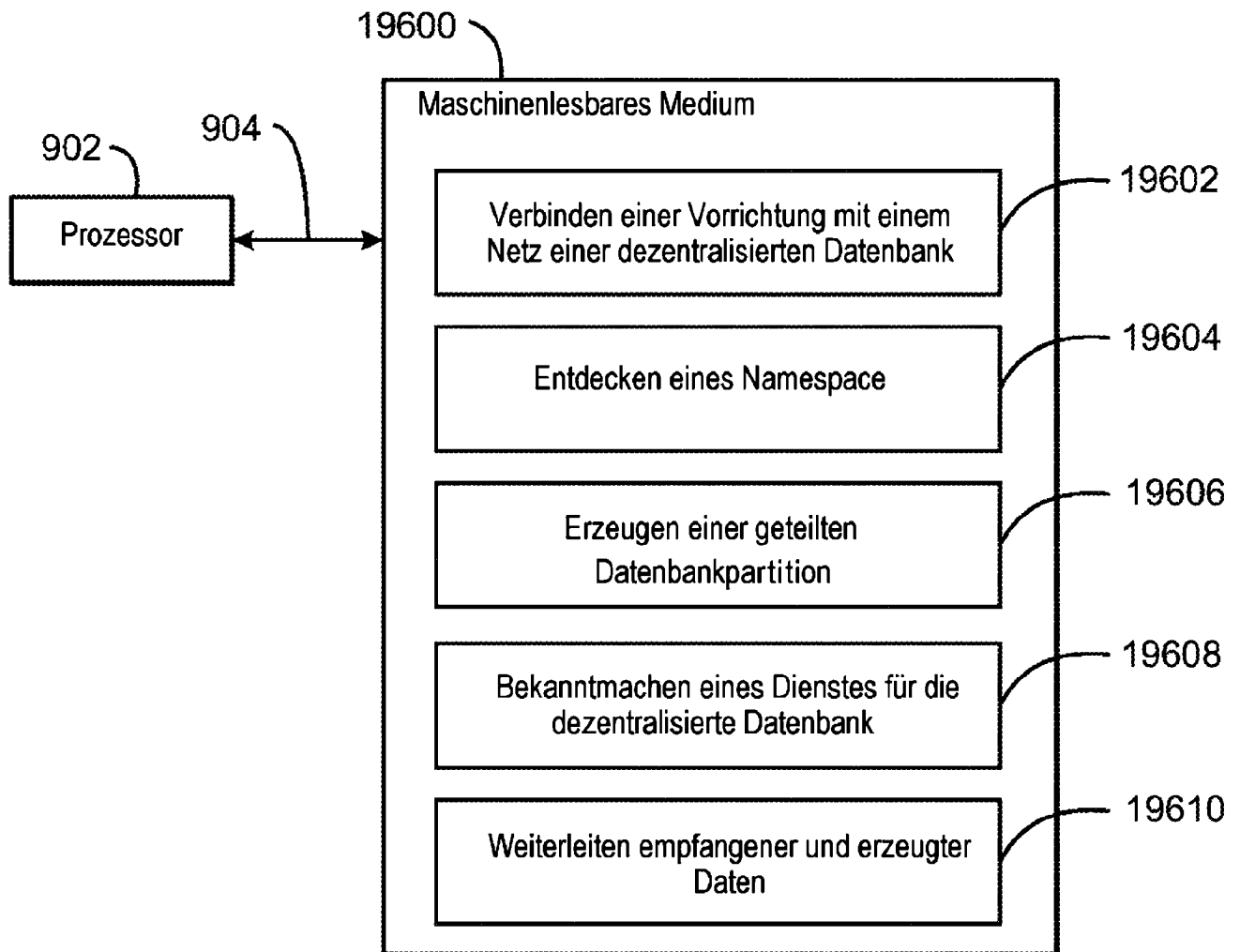
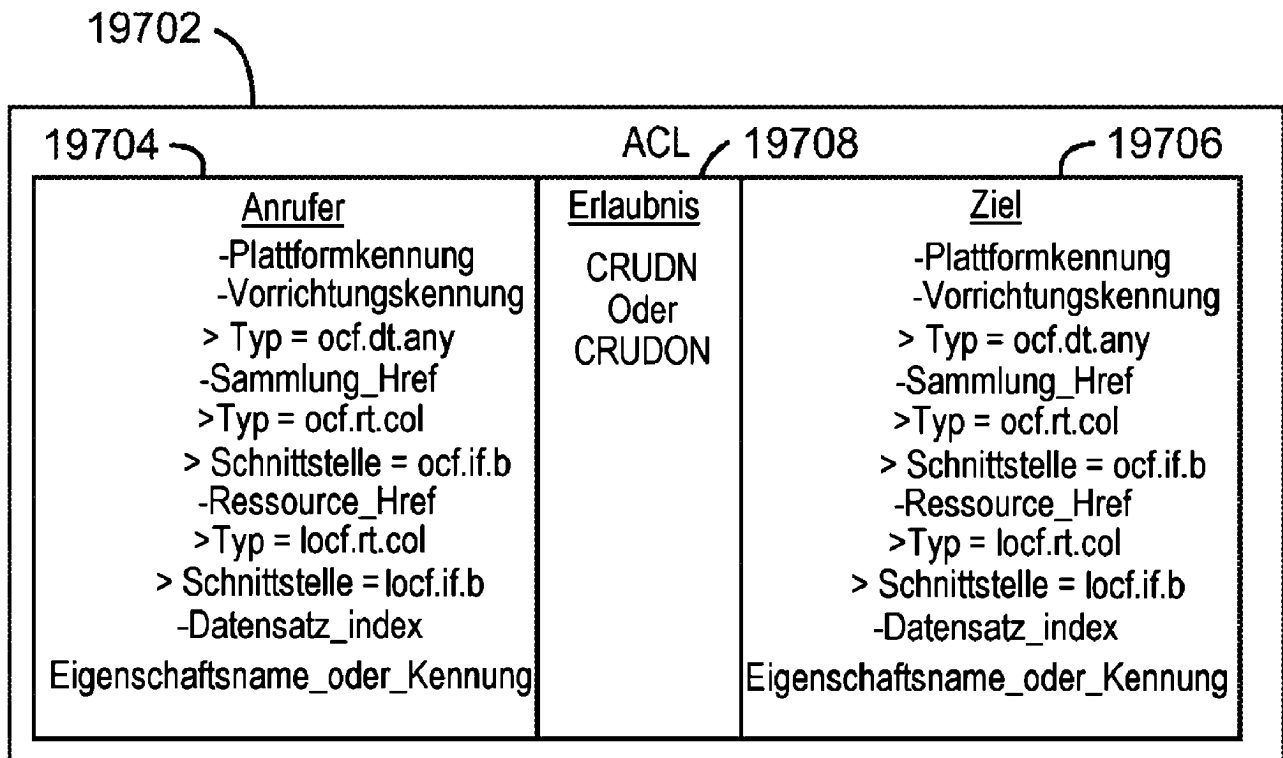


FIG. 196



19700

FIG. 197

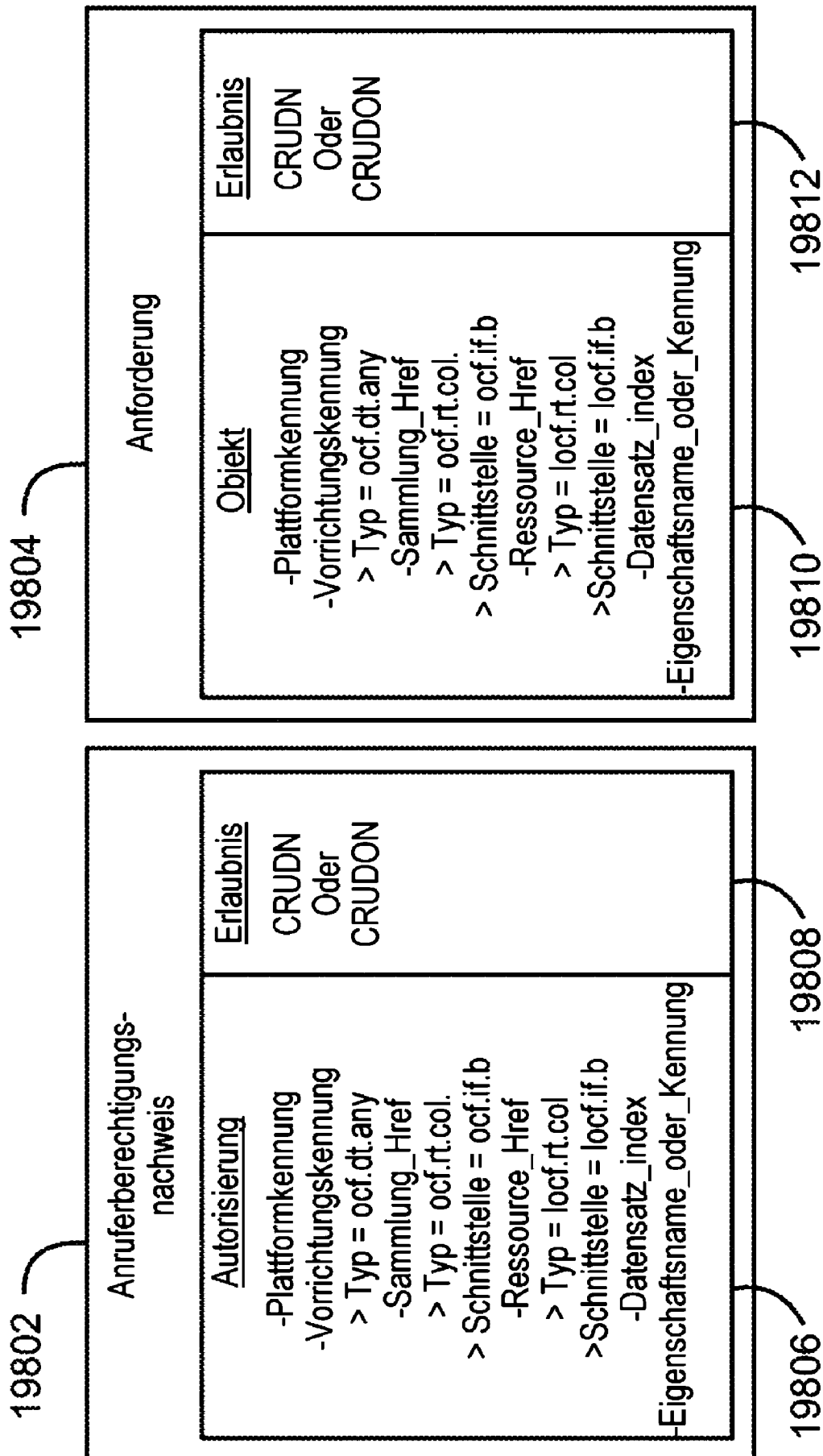
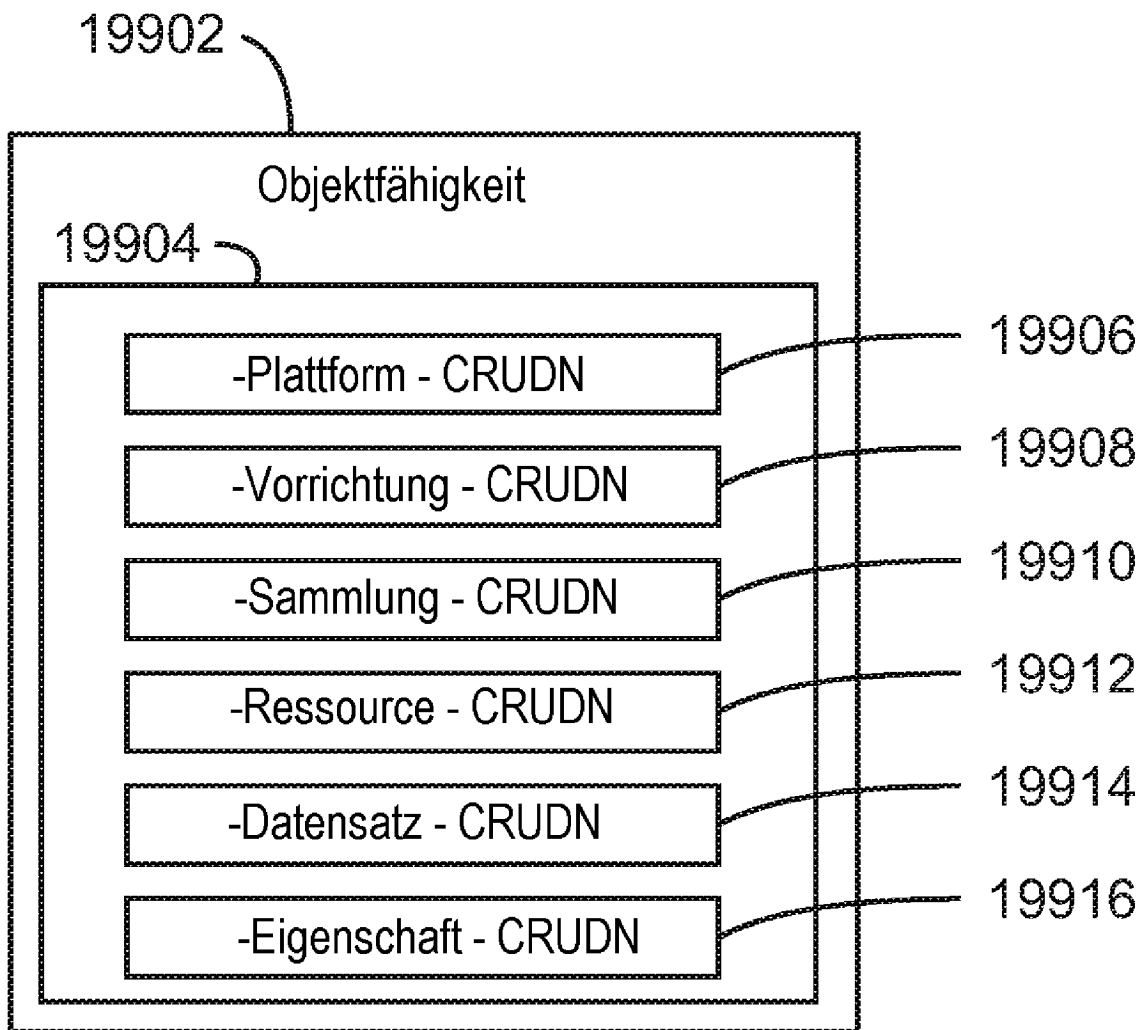
19800

FIG. 198



19900

FIG. 199

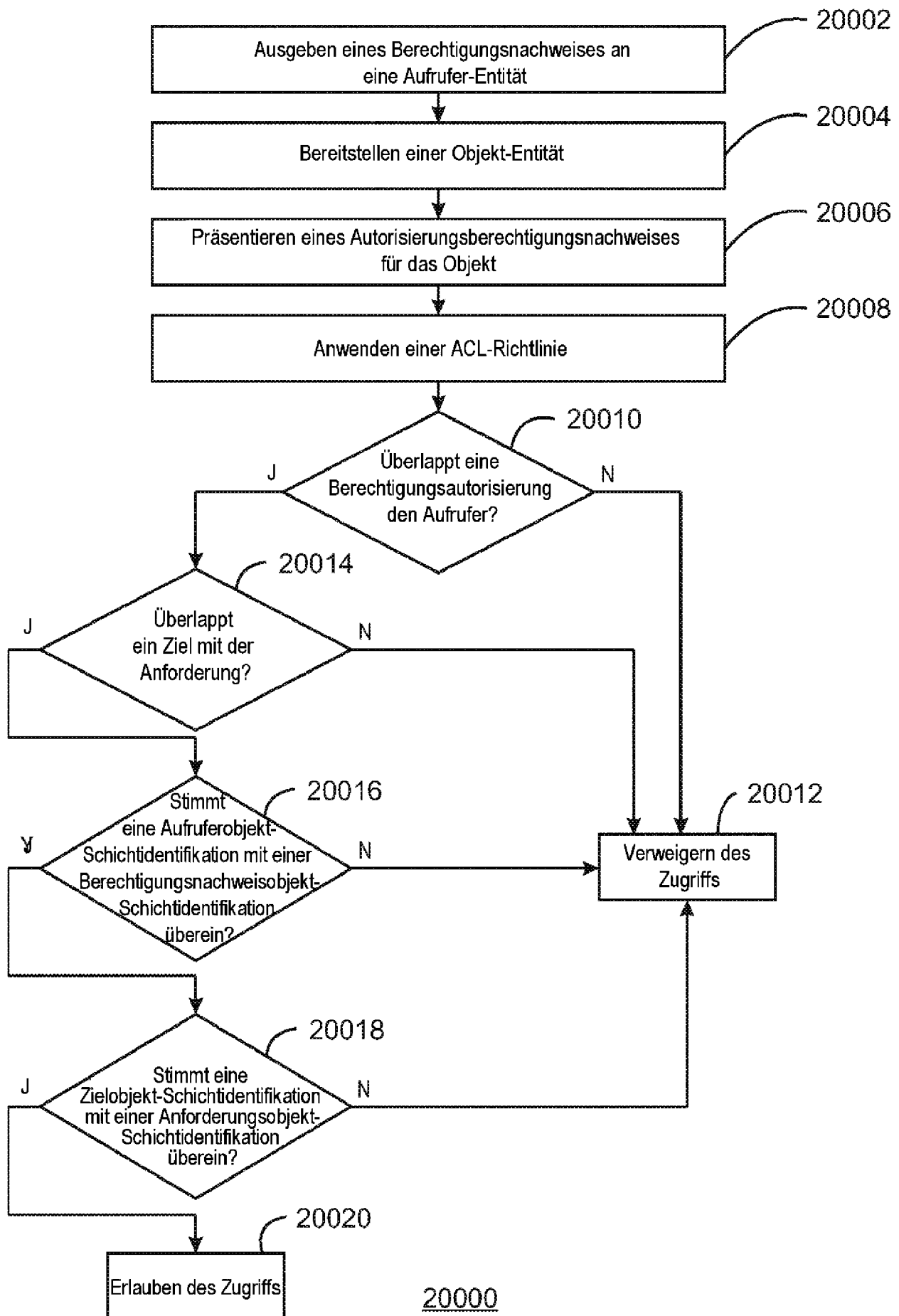


FIG. 200

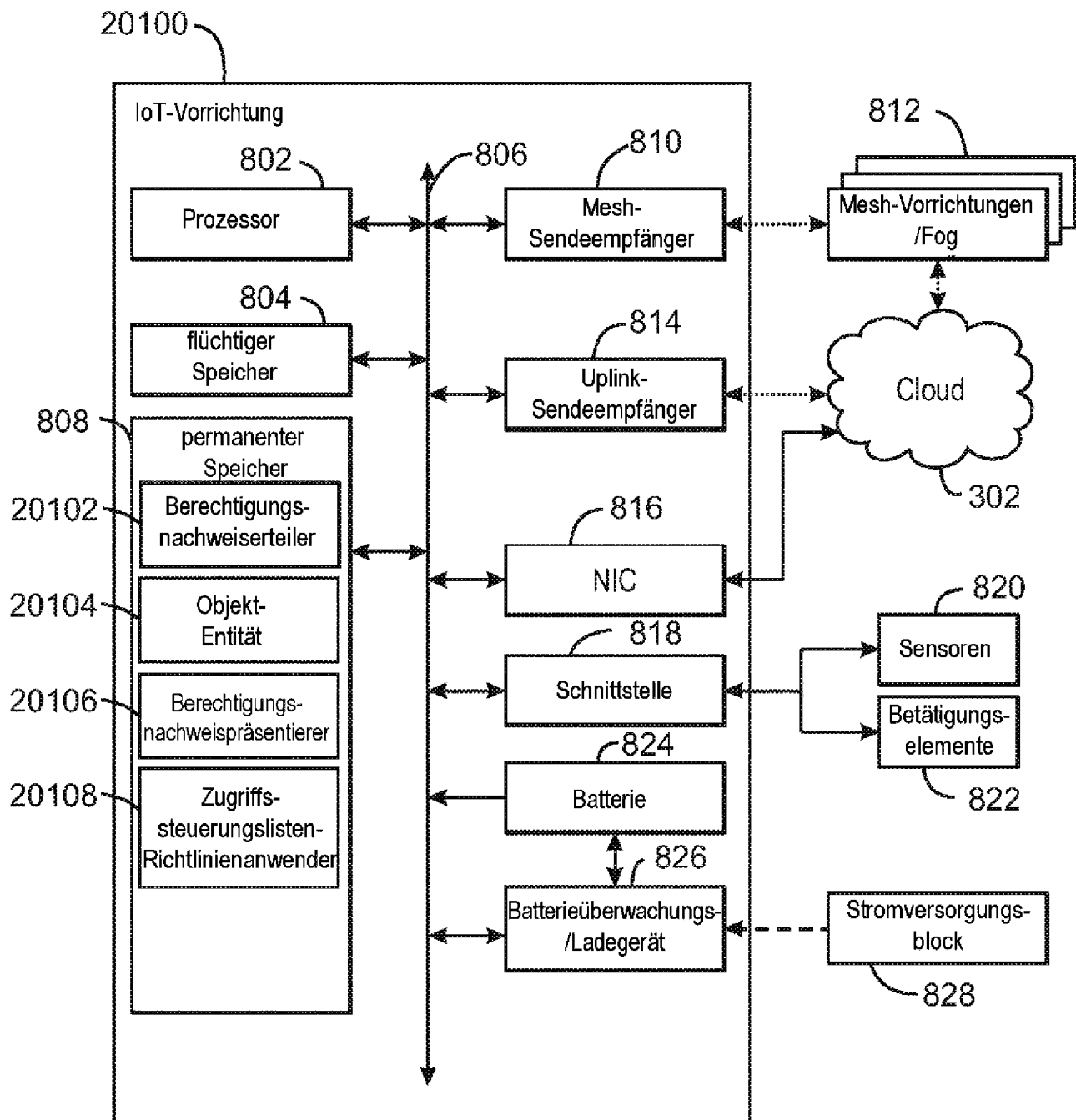


FIG. 201

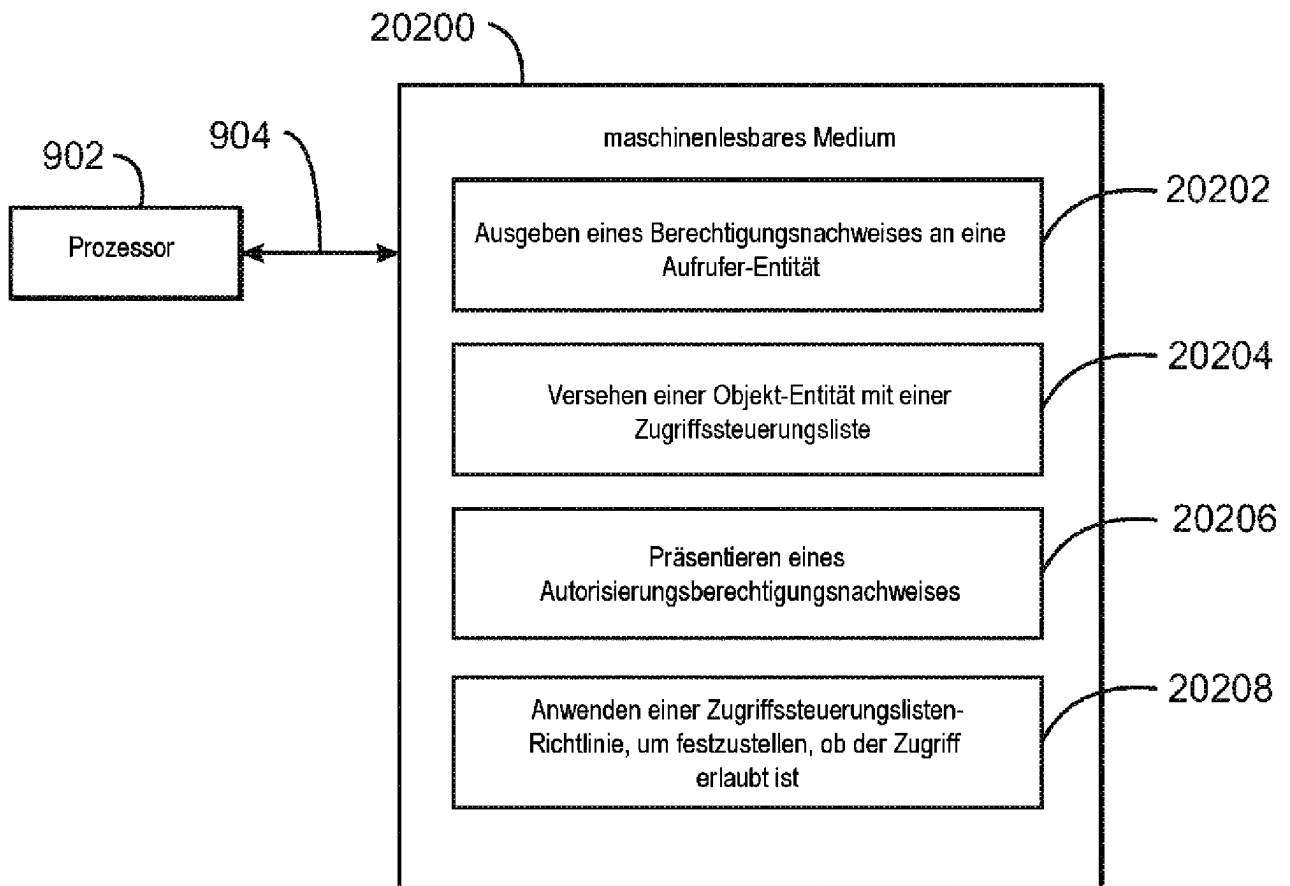
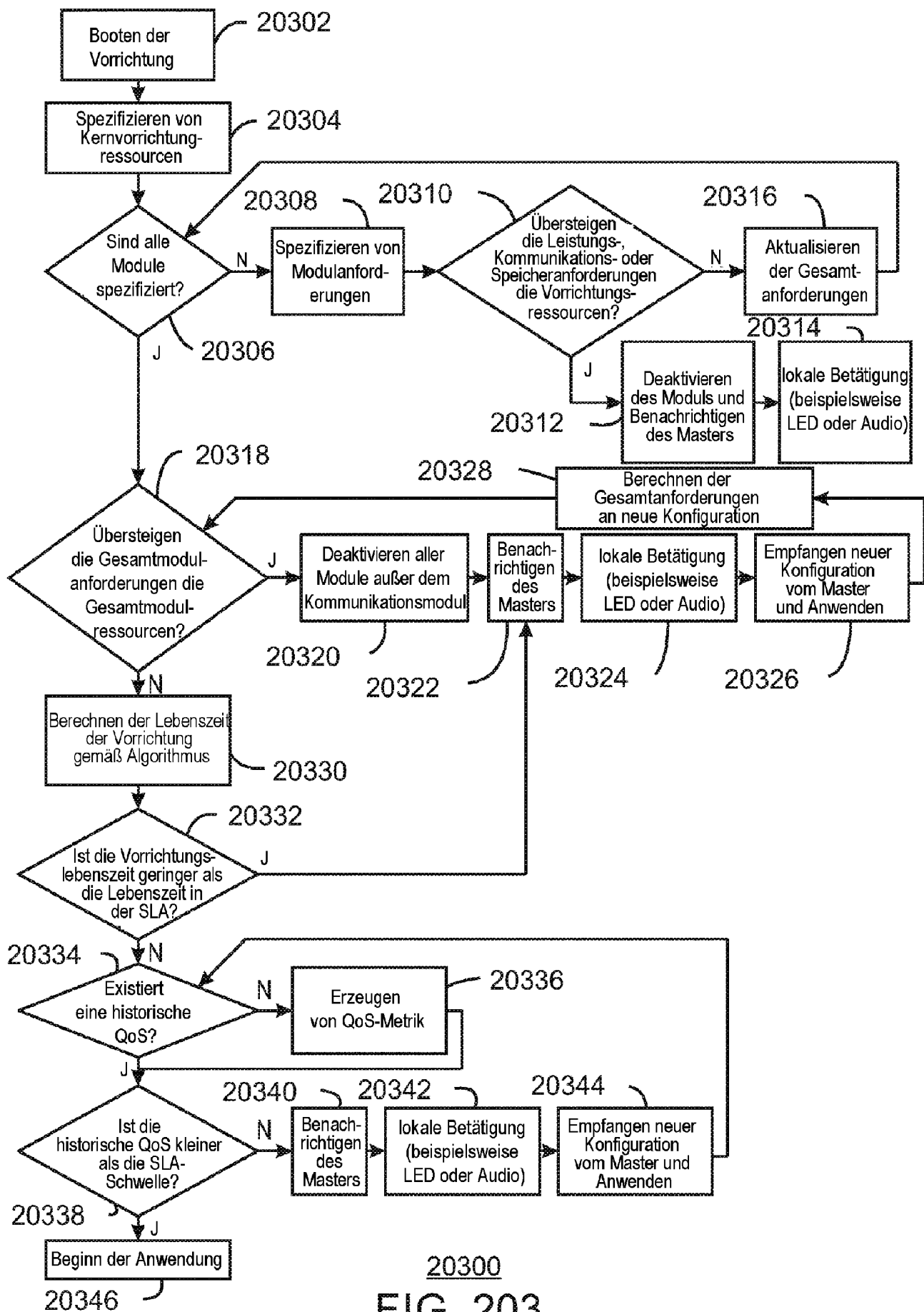


FIG. 202



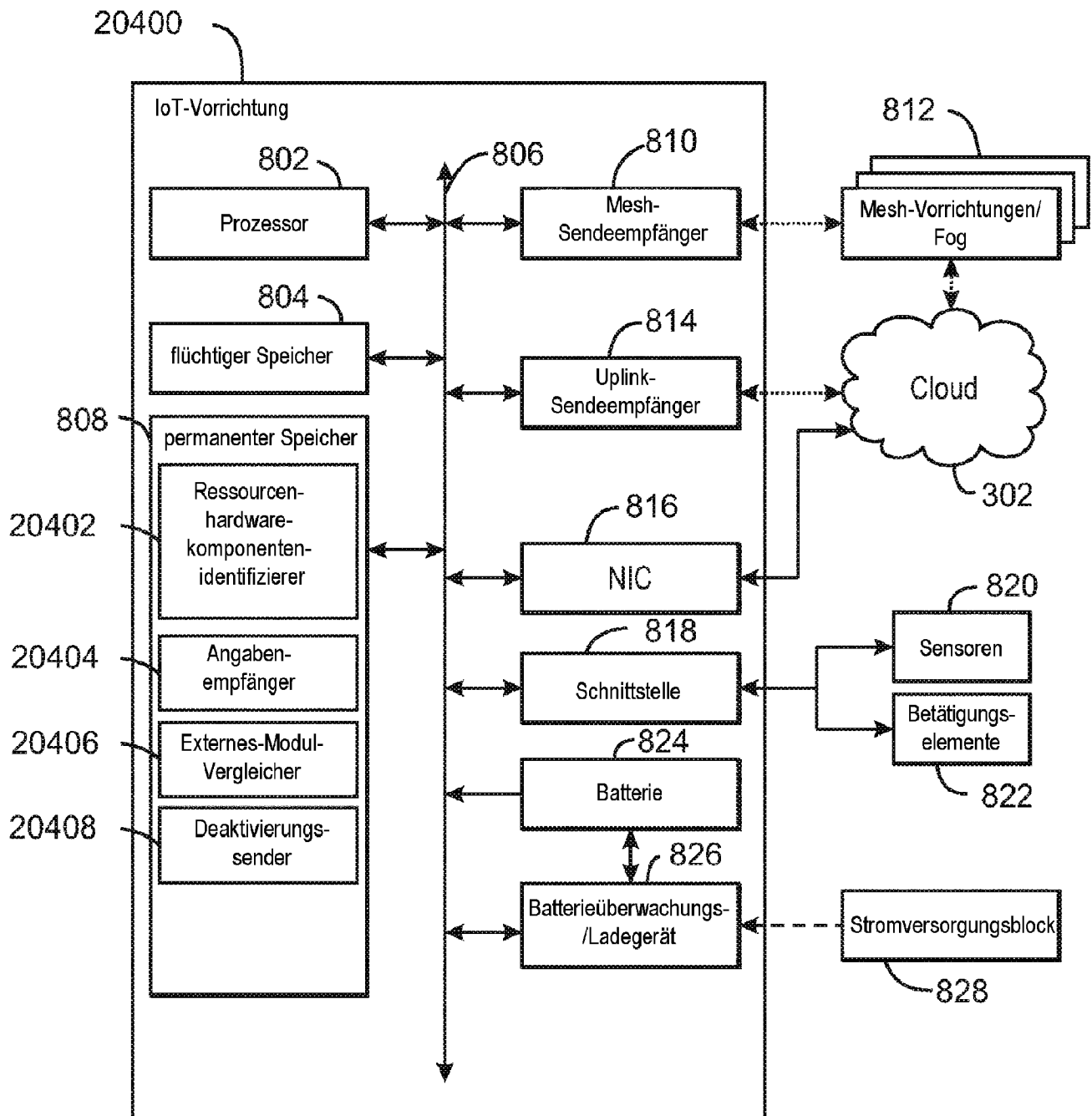


FIG. 204

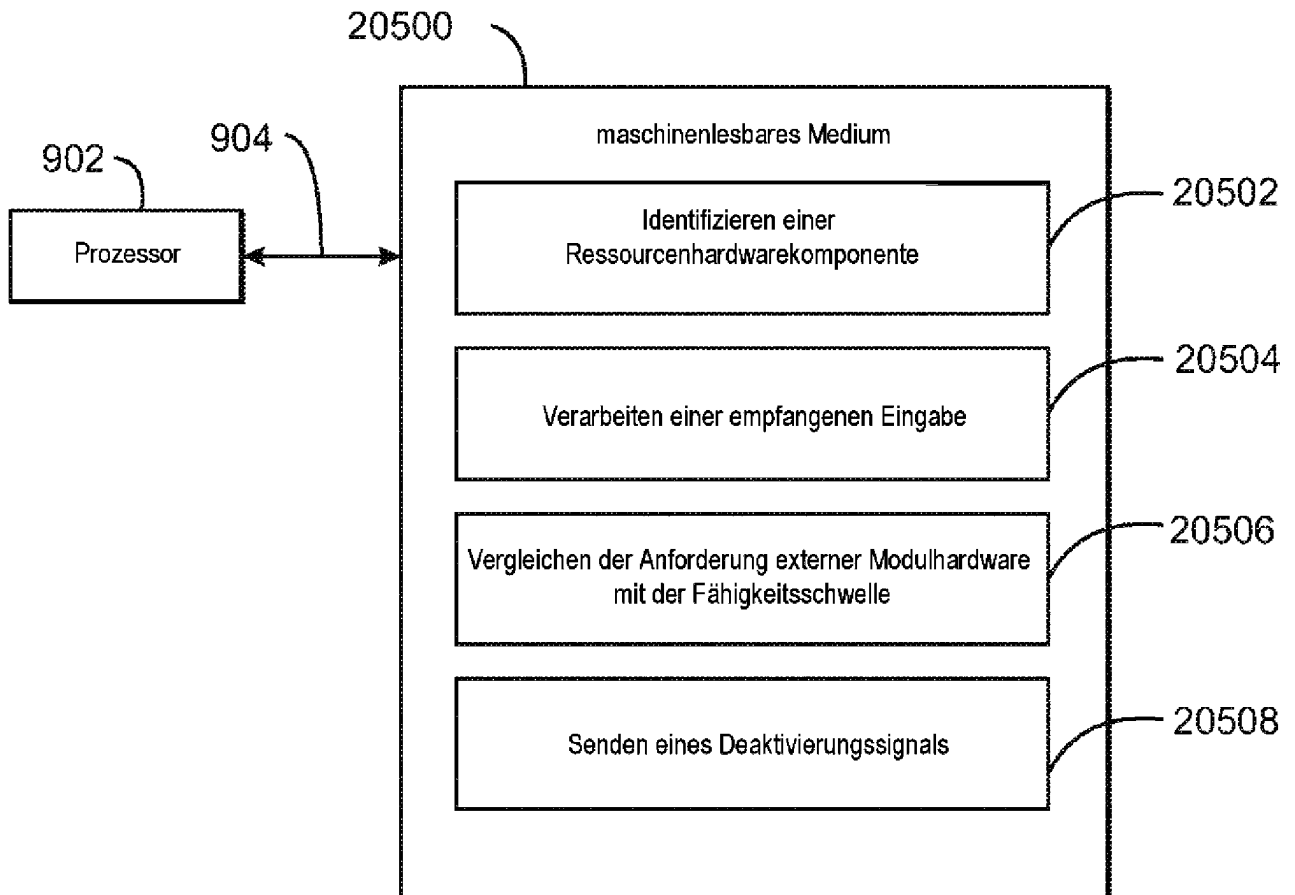
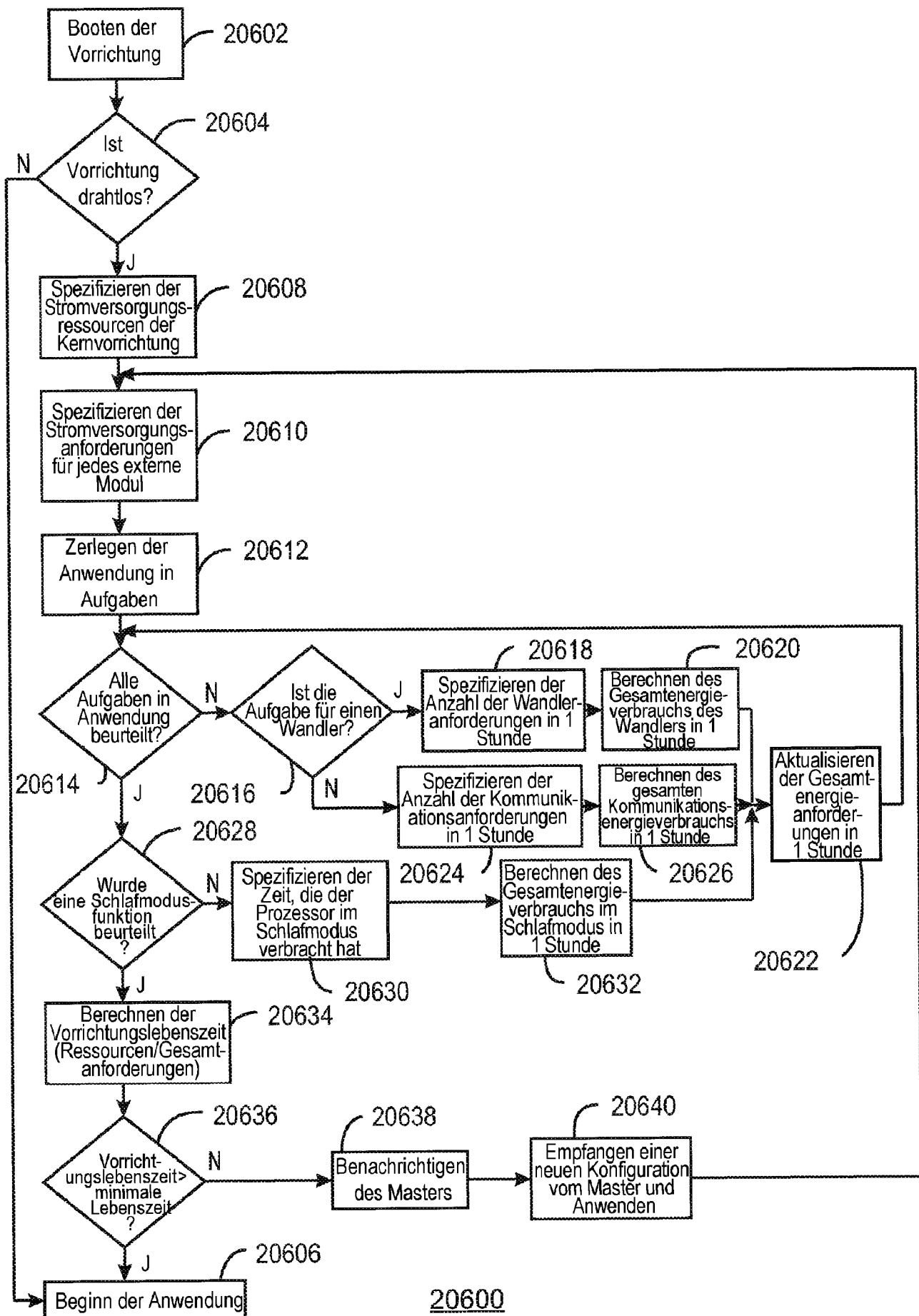


FIG. 205



20600
FIG. 206

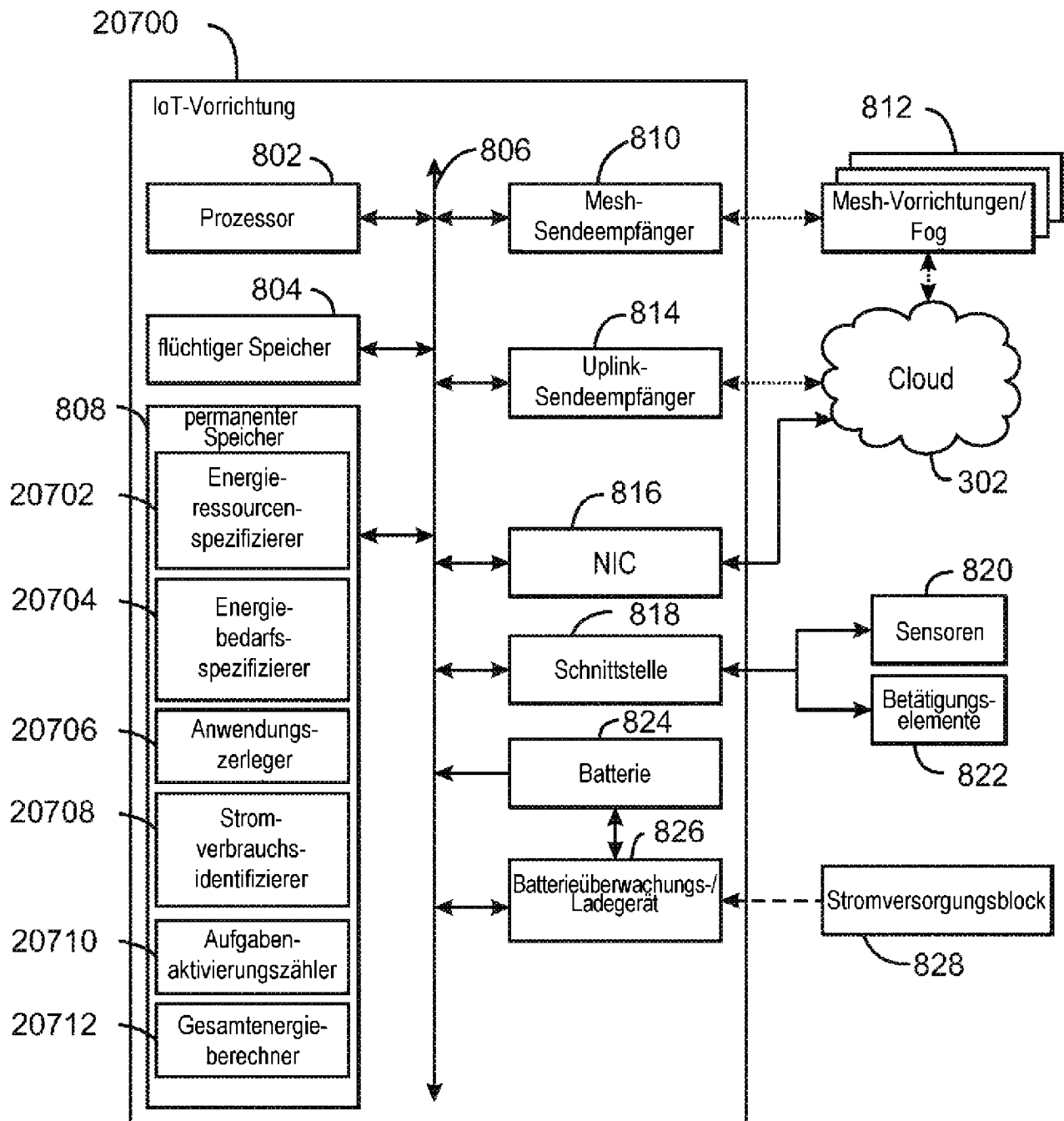


FIG. 207

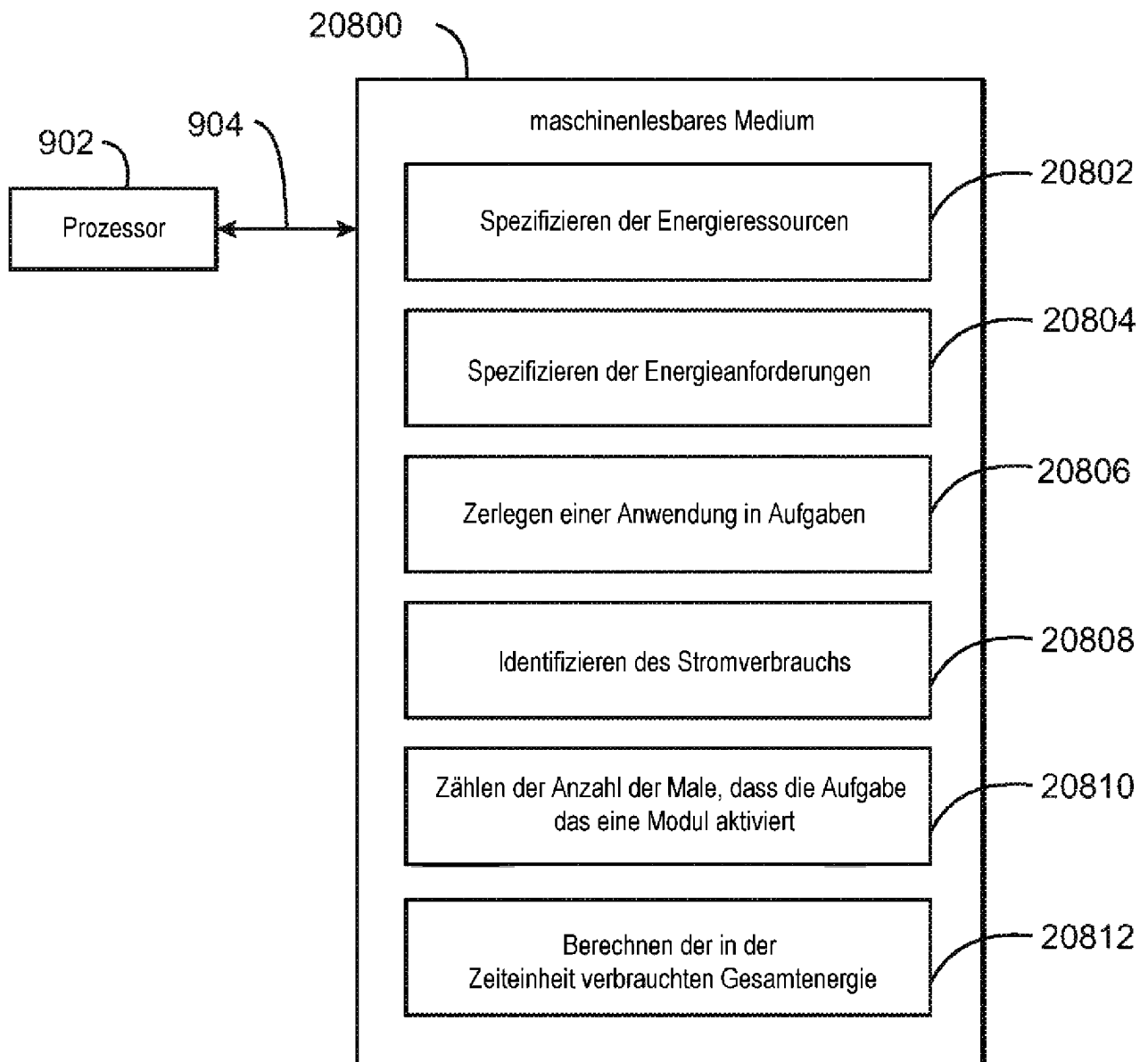
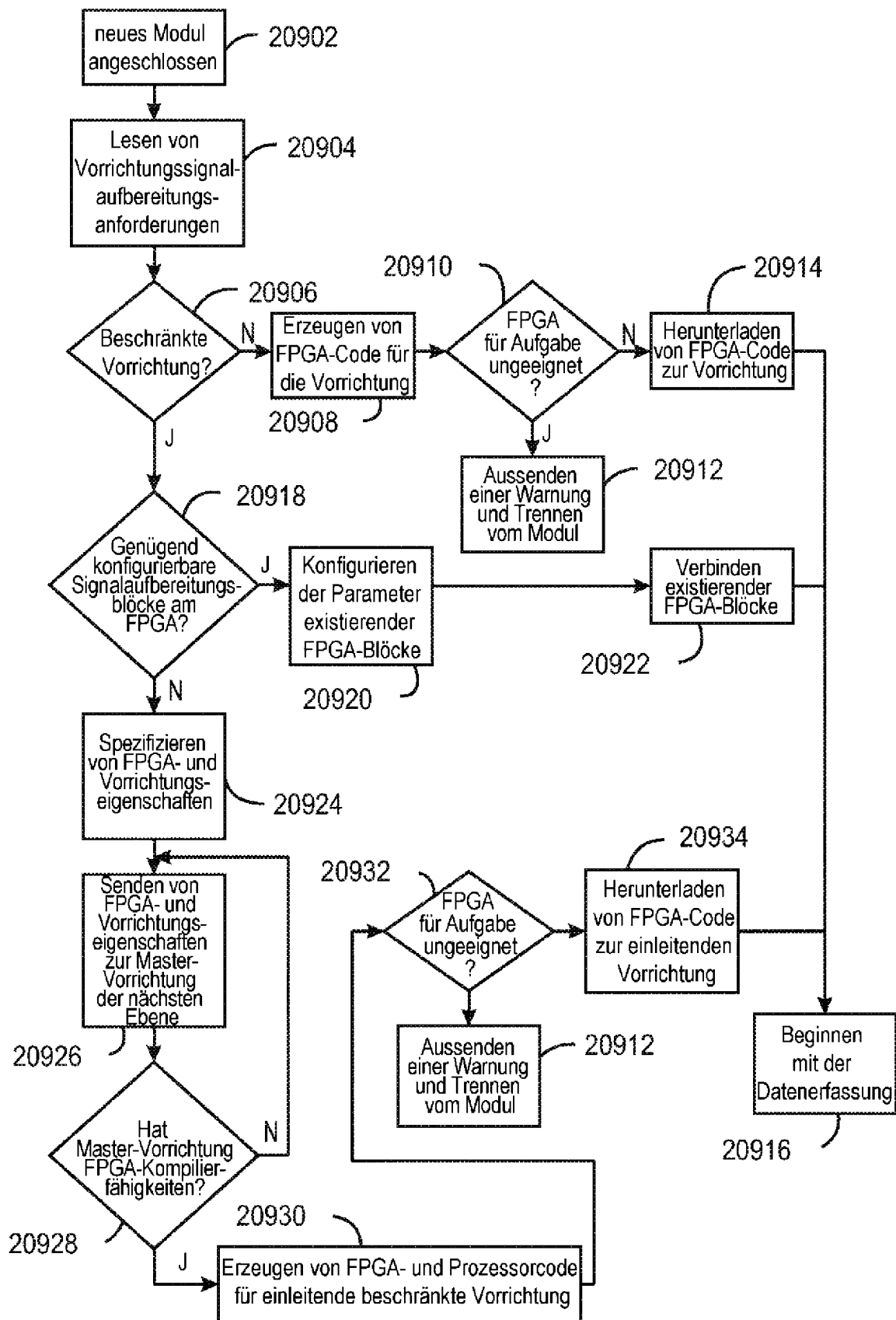


FIG. 208



20900
FIG. 209

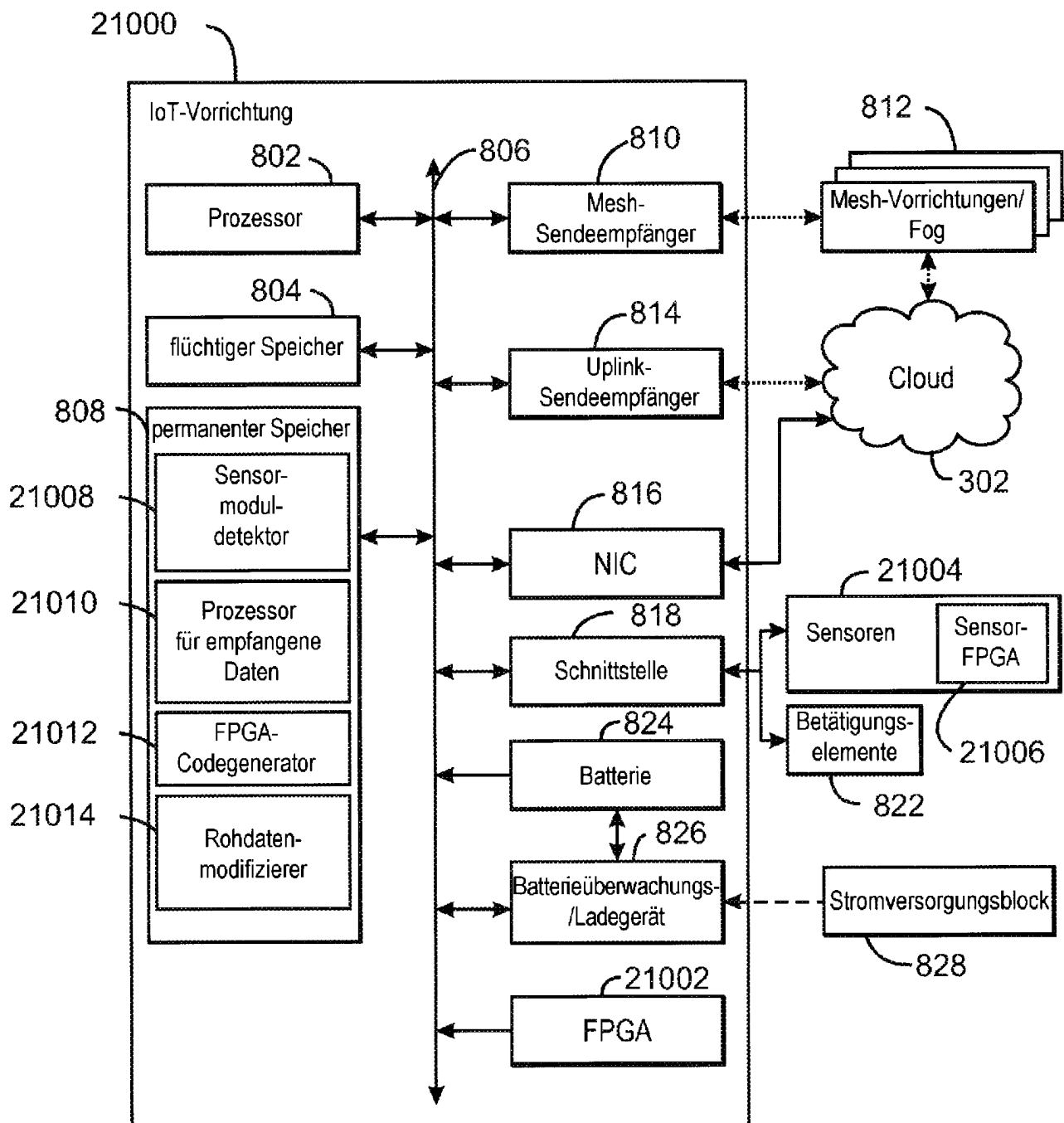


FIG. 210

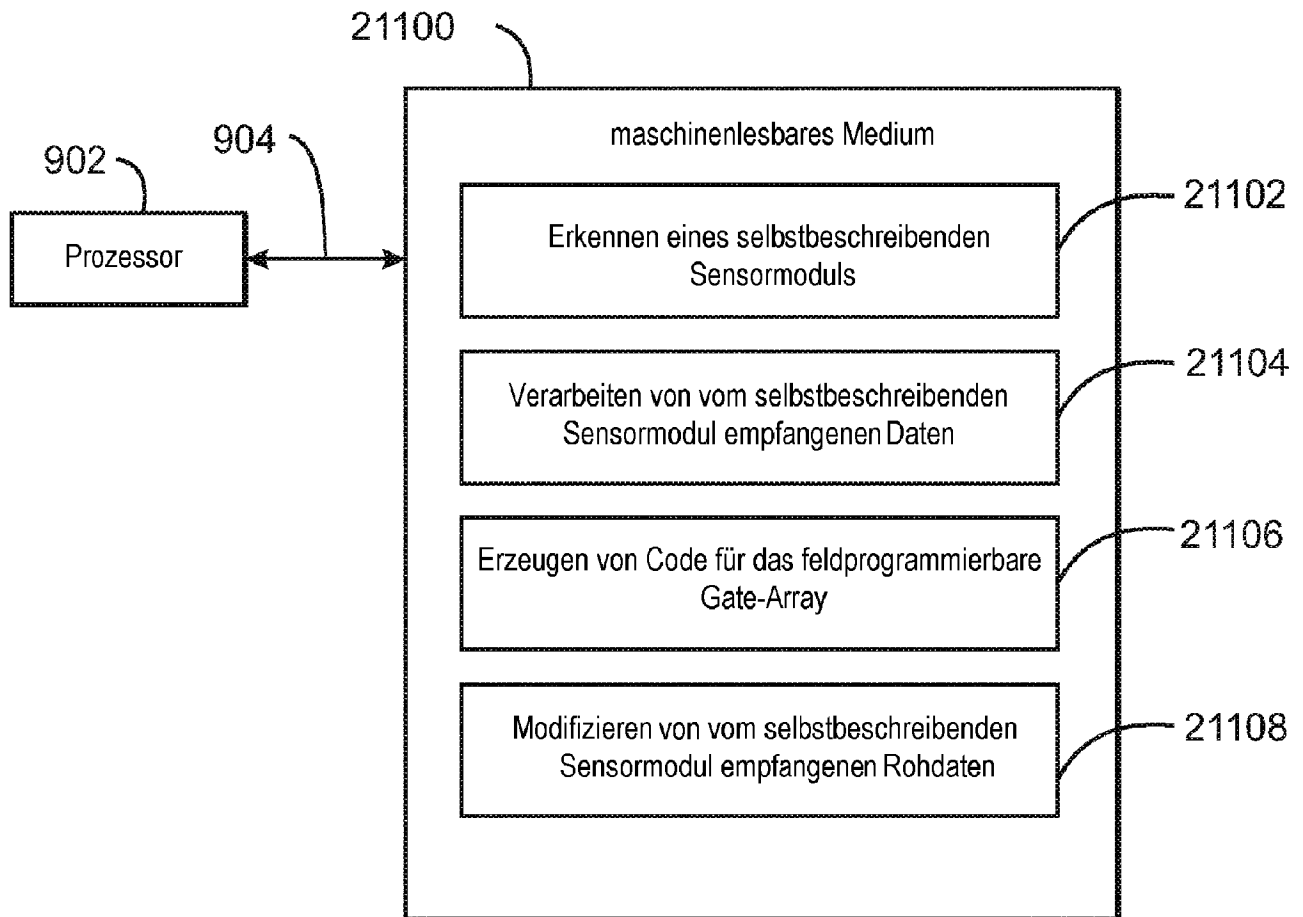
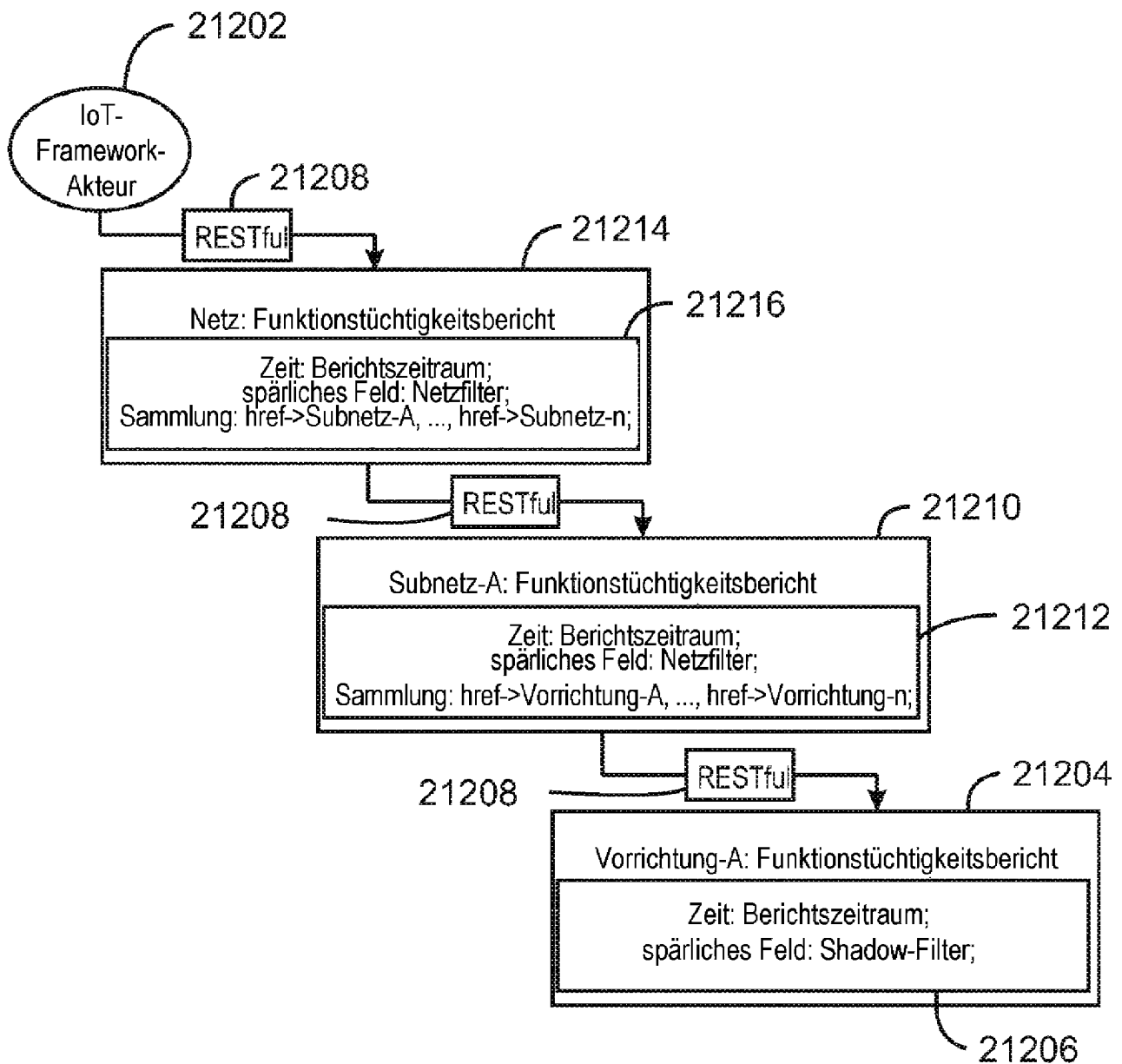
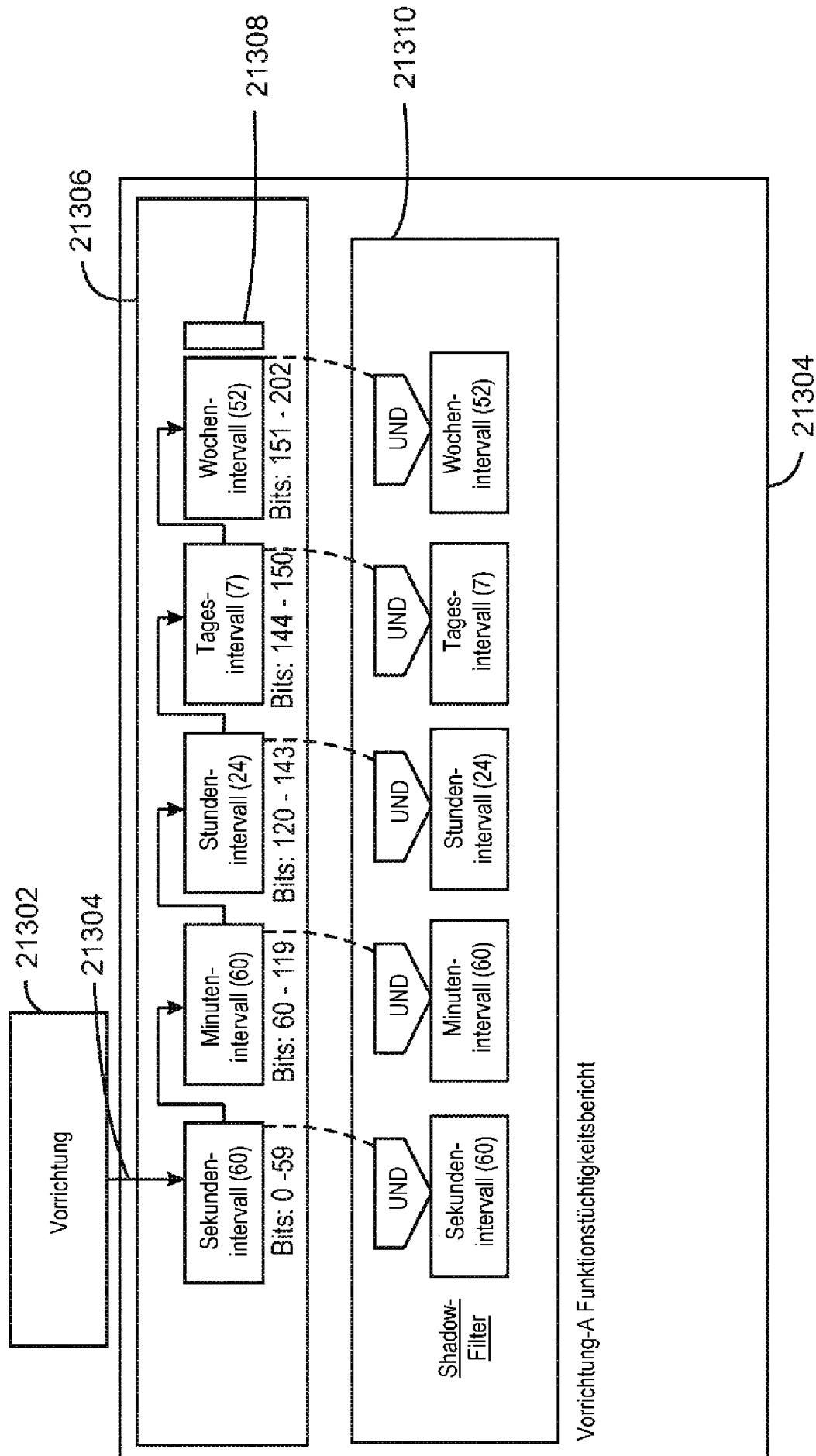


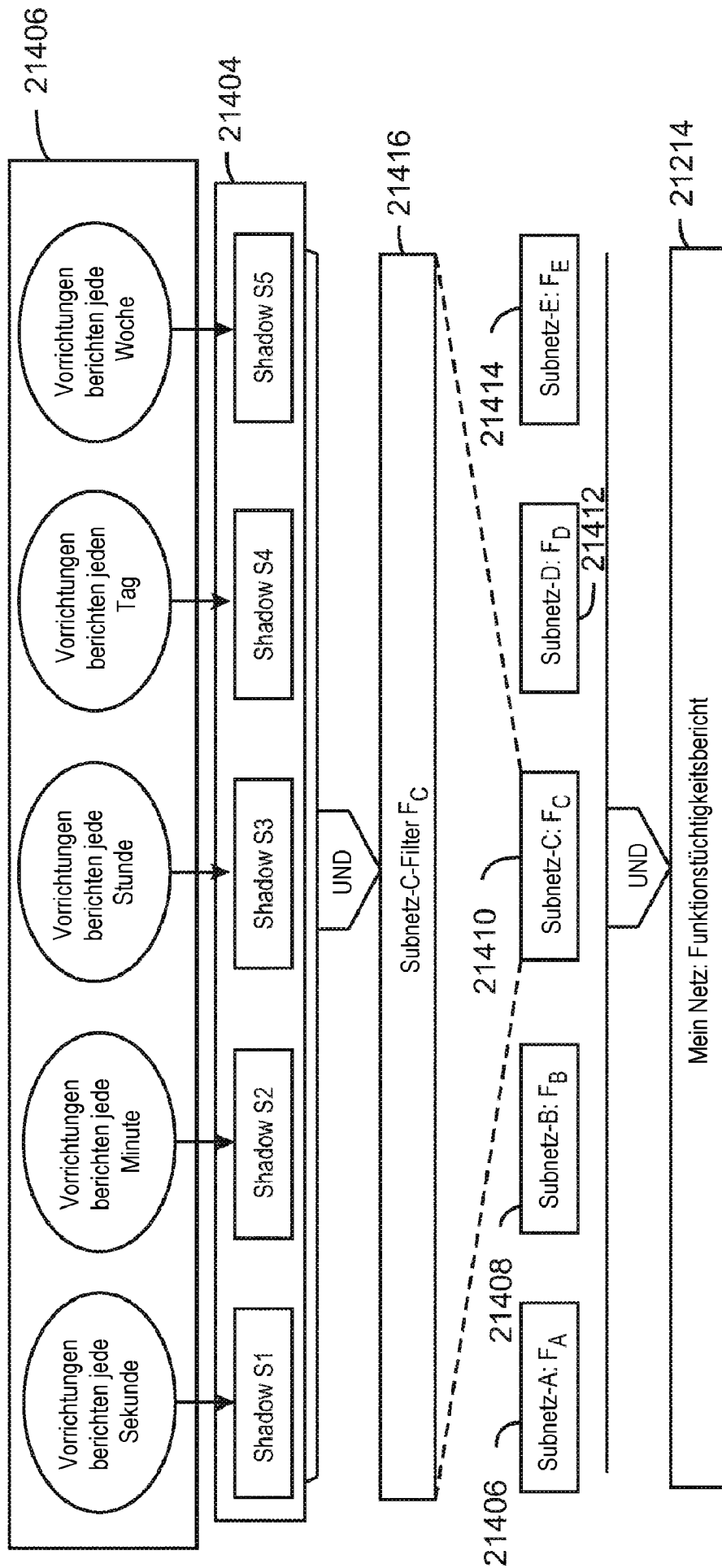
FIG. 211



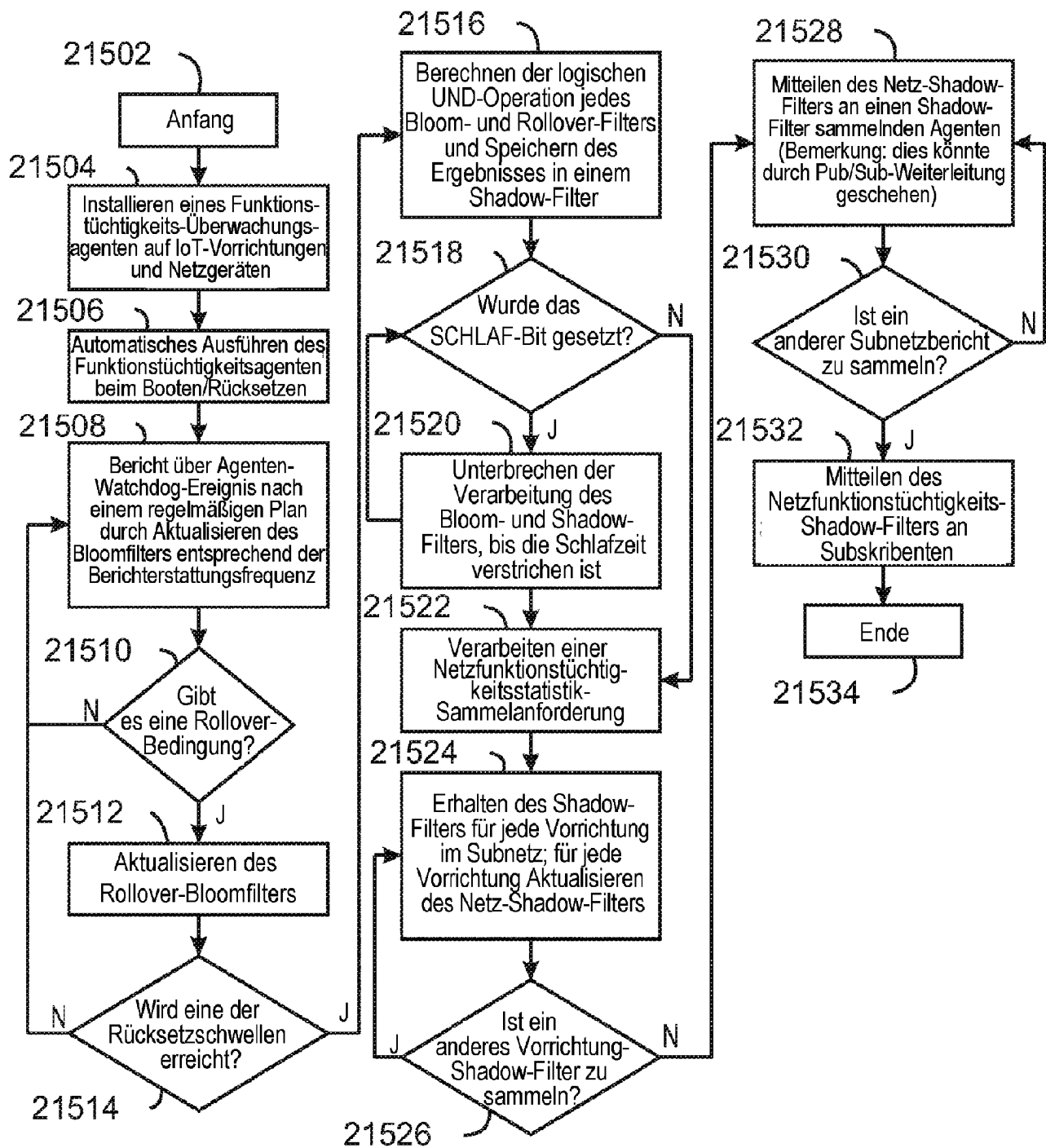
21200
FIG. 212



21300
FIG. 213



21400
FIG. 214



21500
FIG. 215

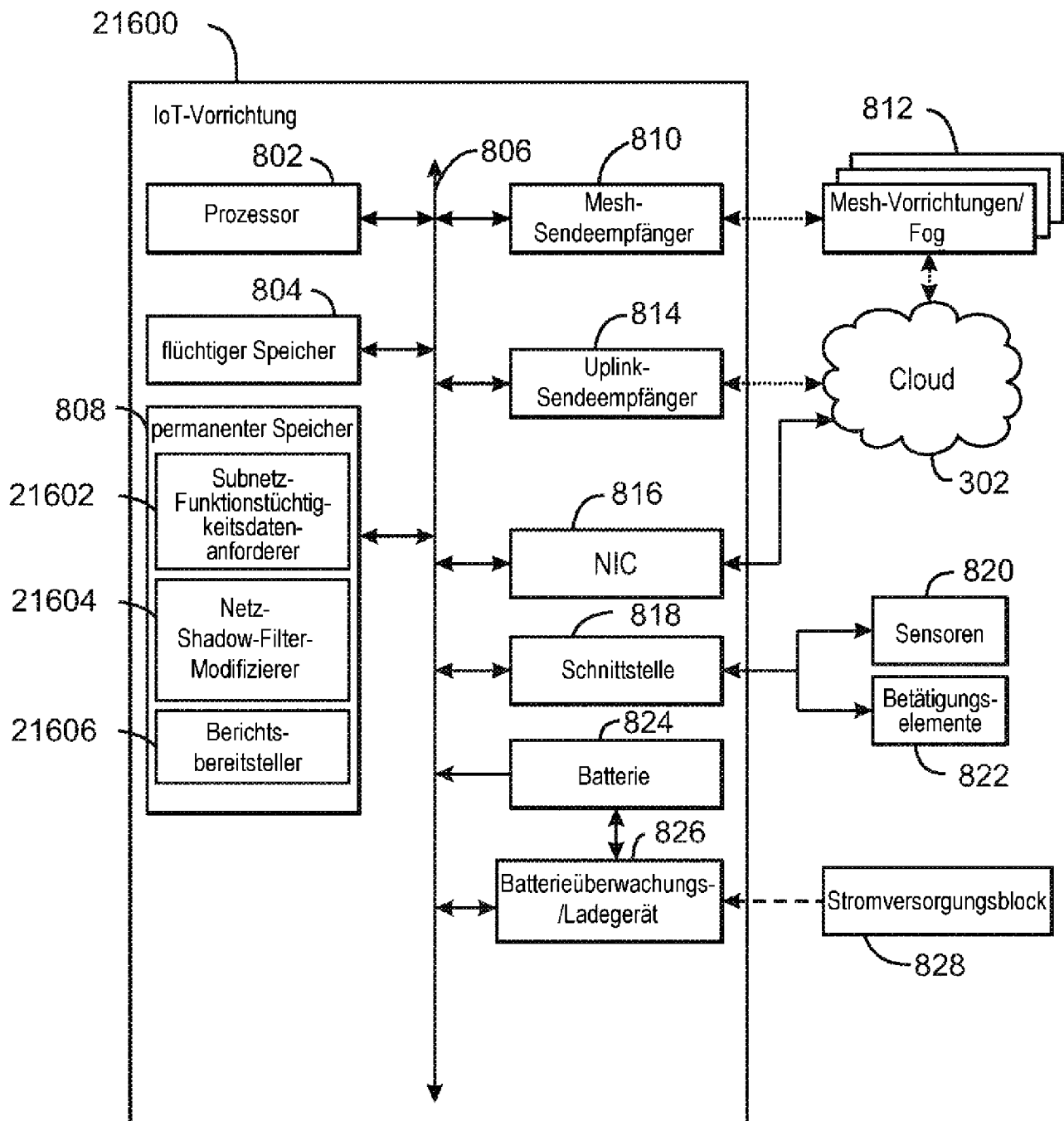


FIG. 216

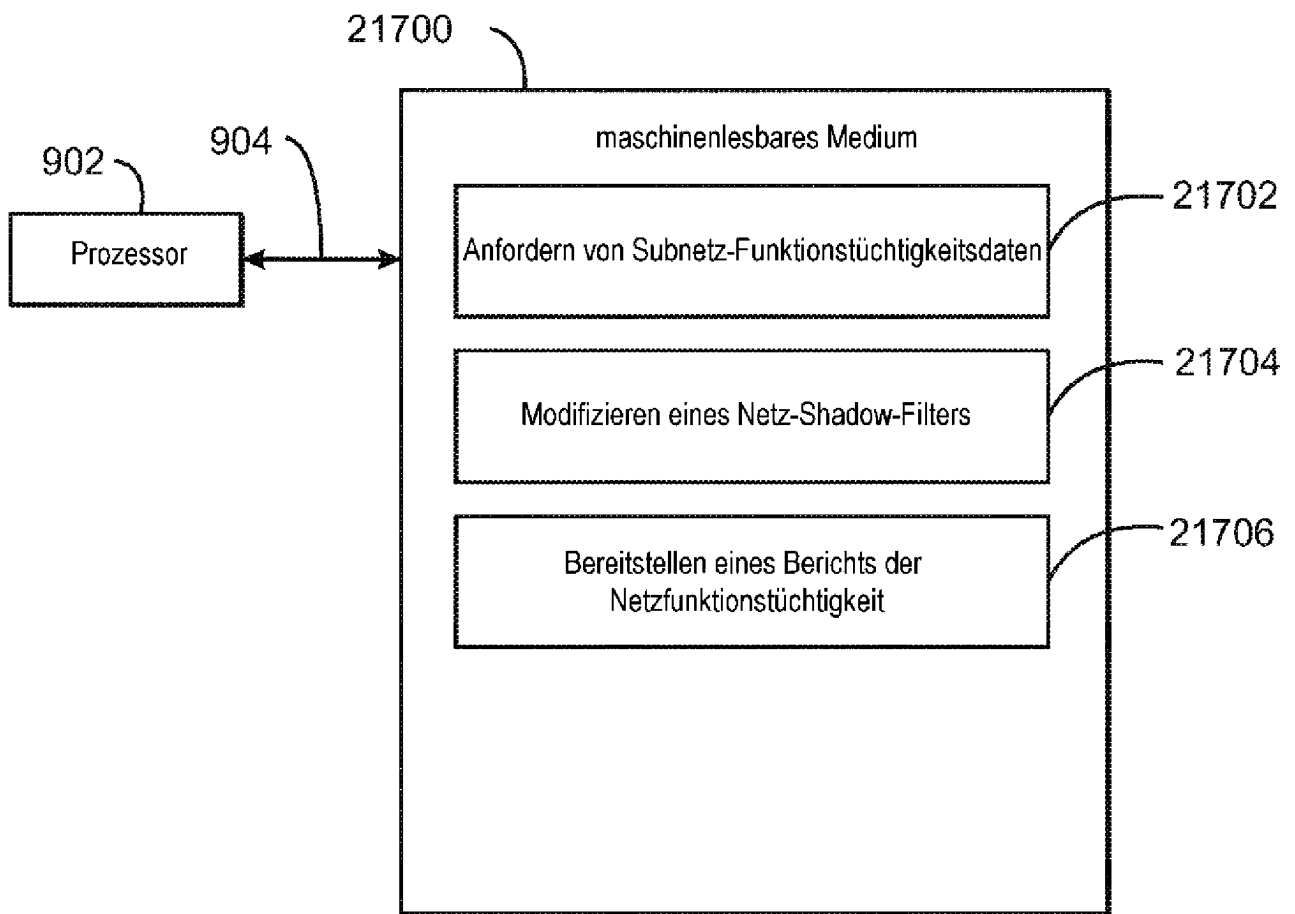
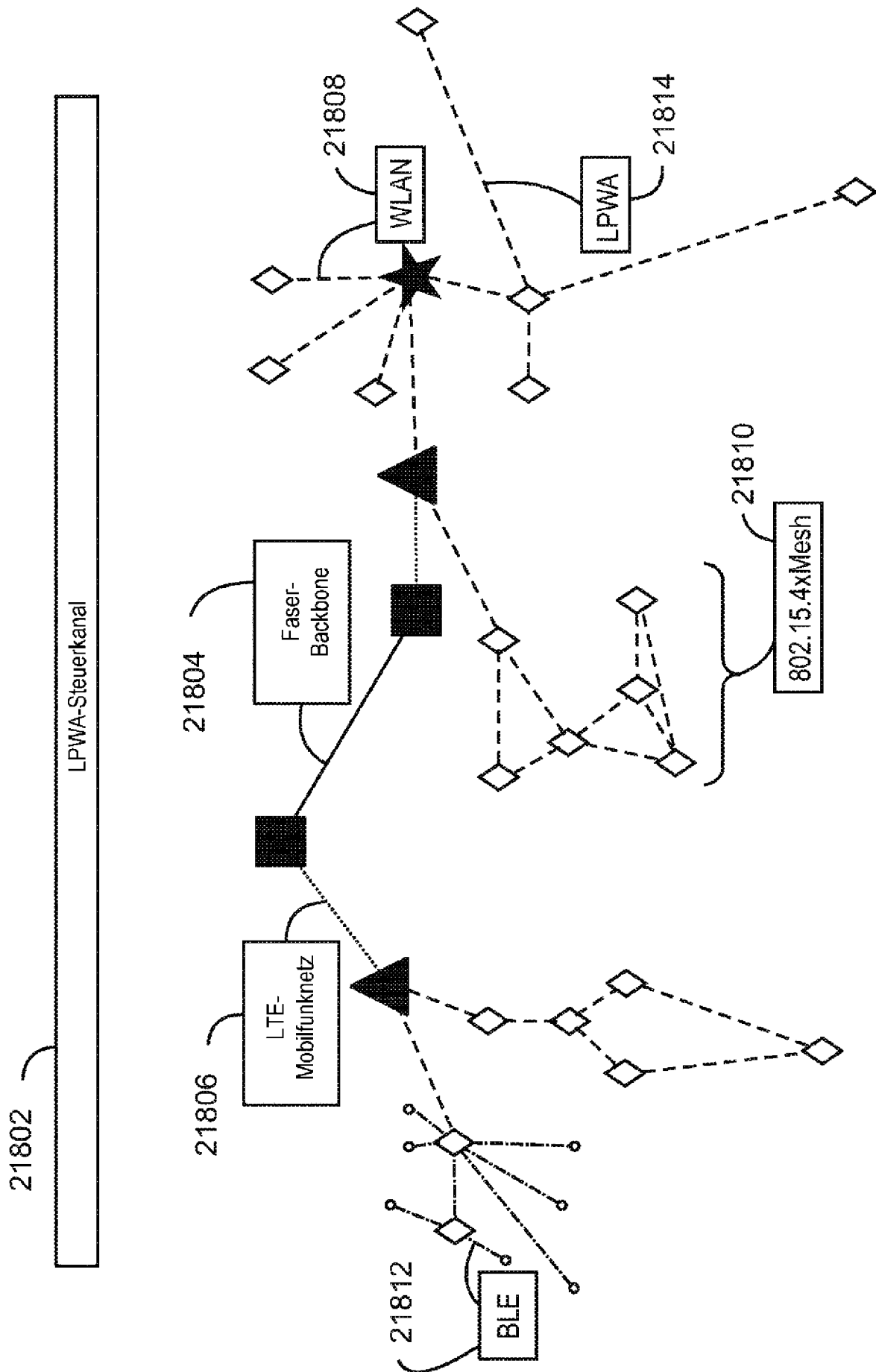
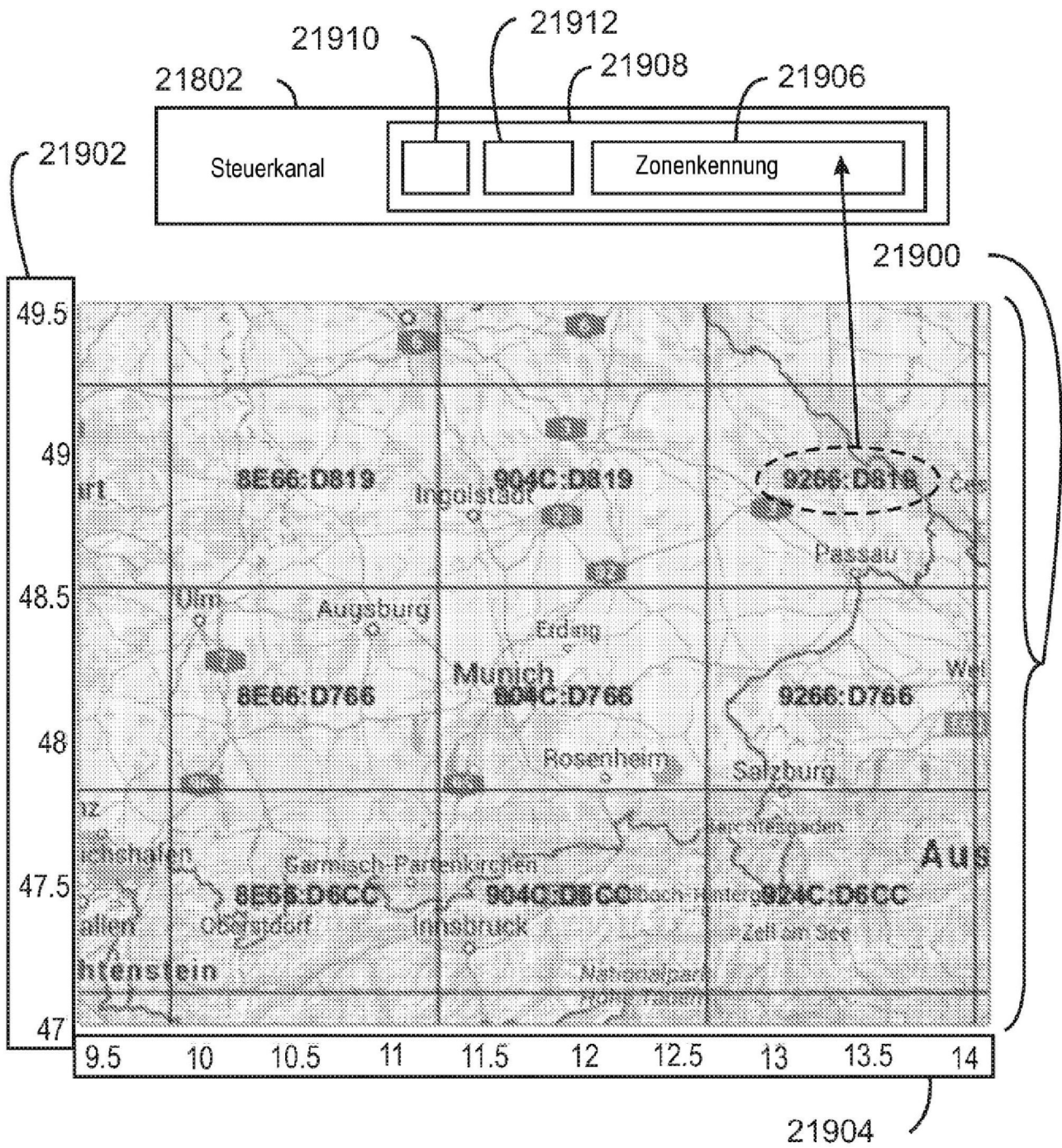


FIG. 217

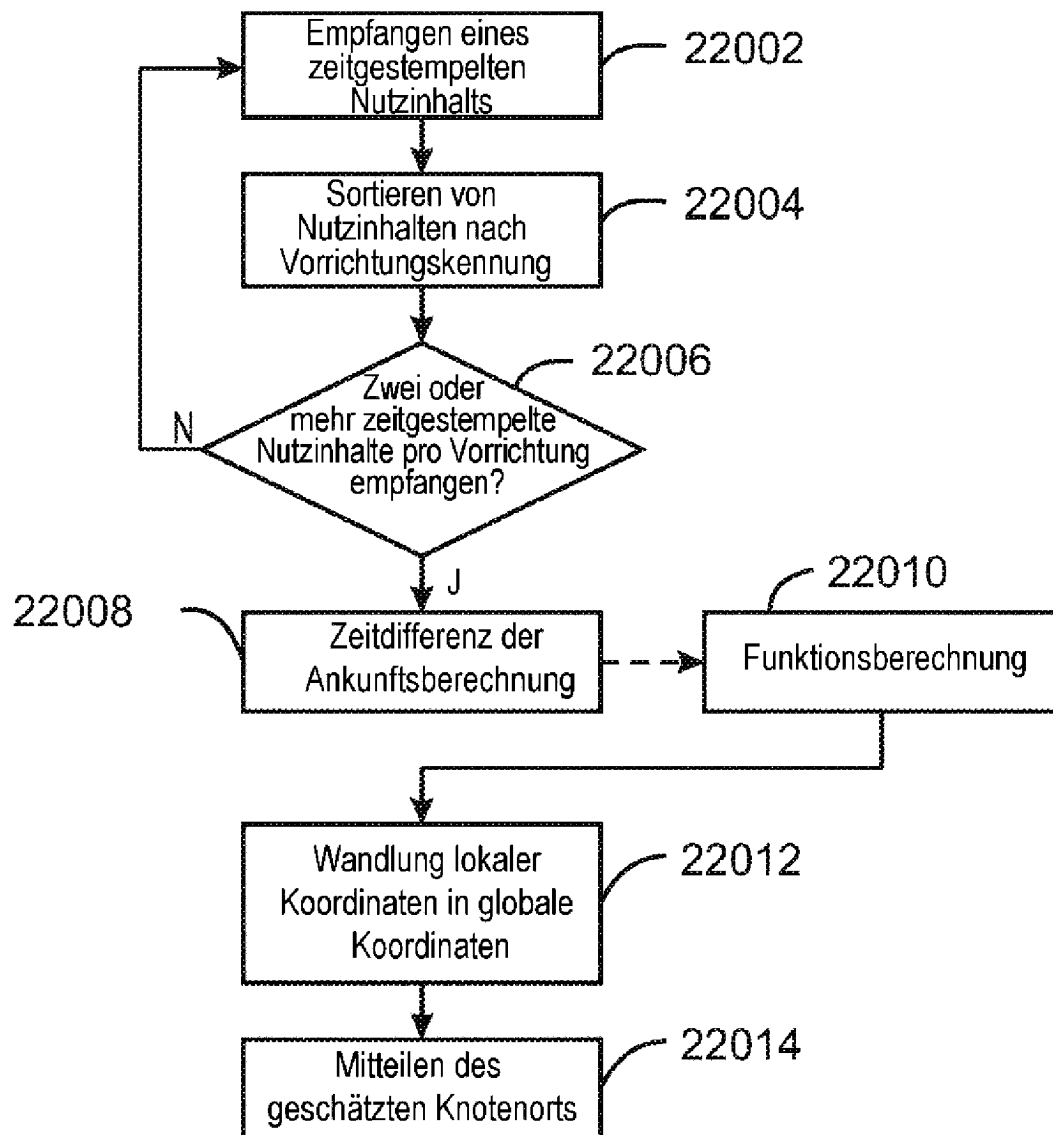


21800

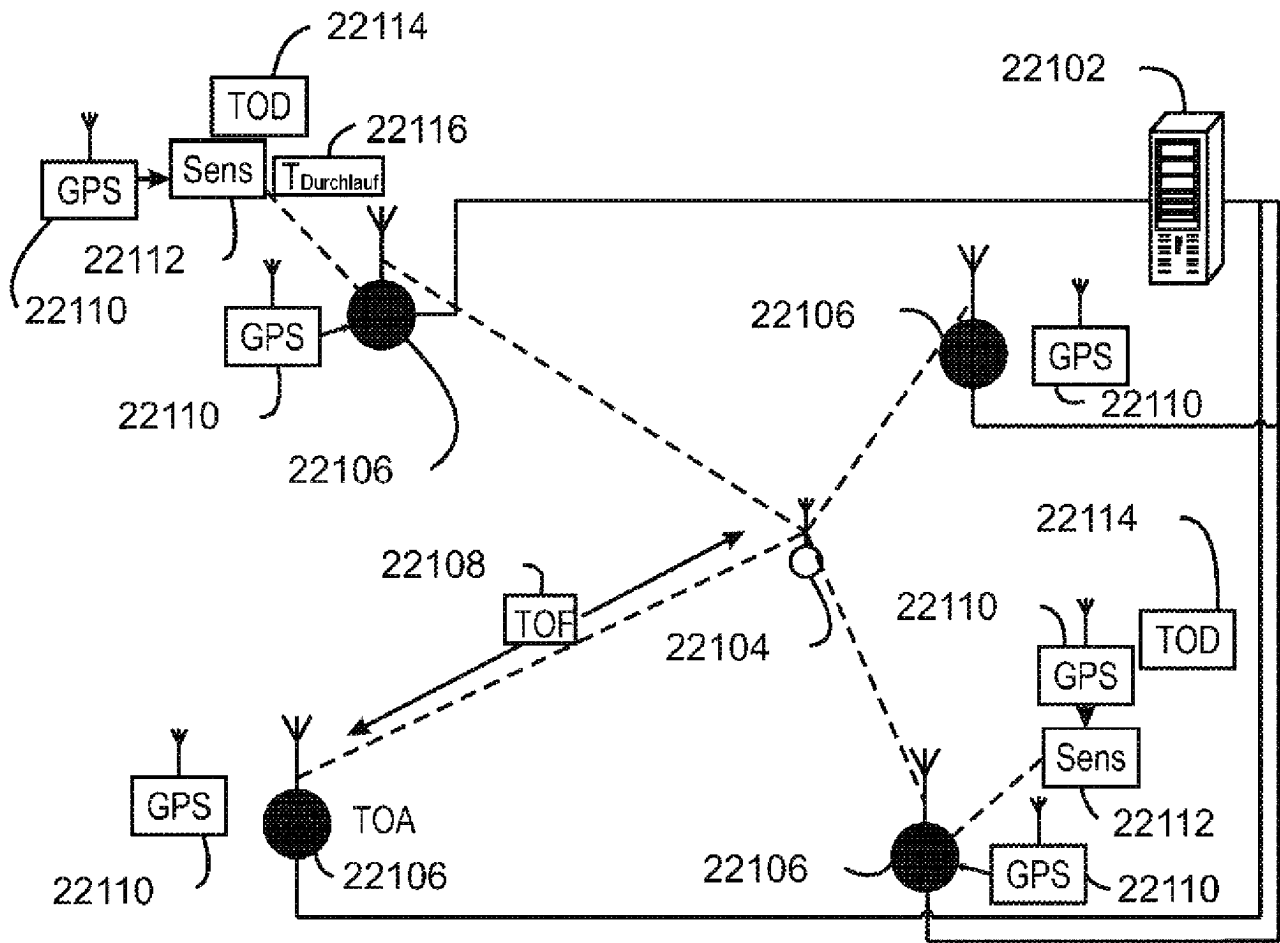
FIG. 218



21900
FIG. 219



22000
FIG. 220



22100
FIG. 221

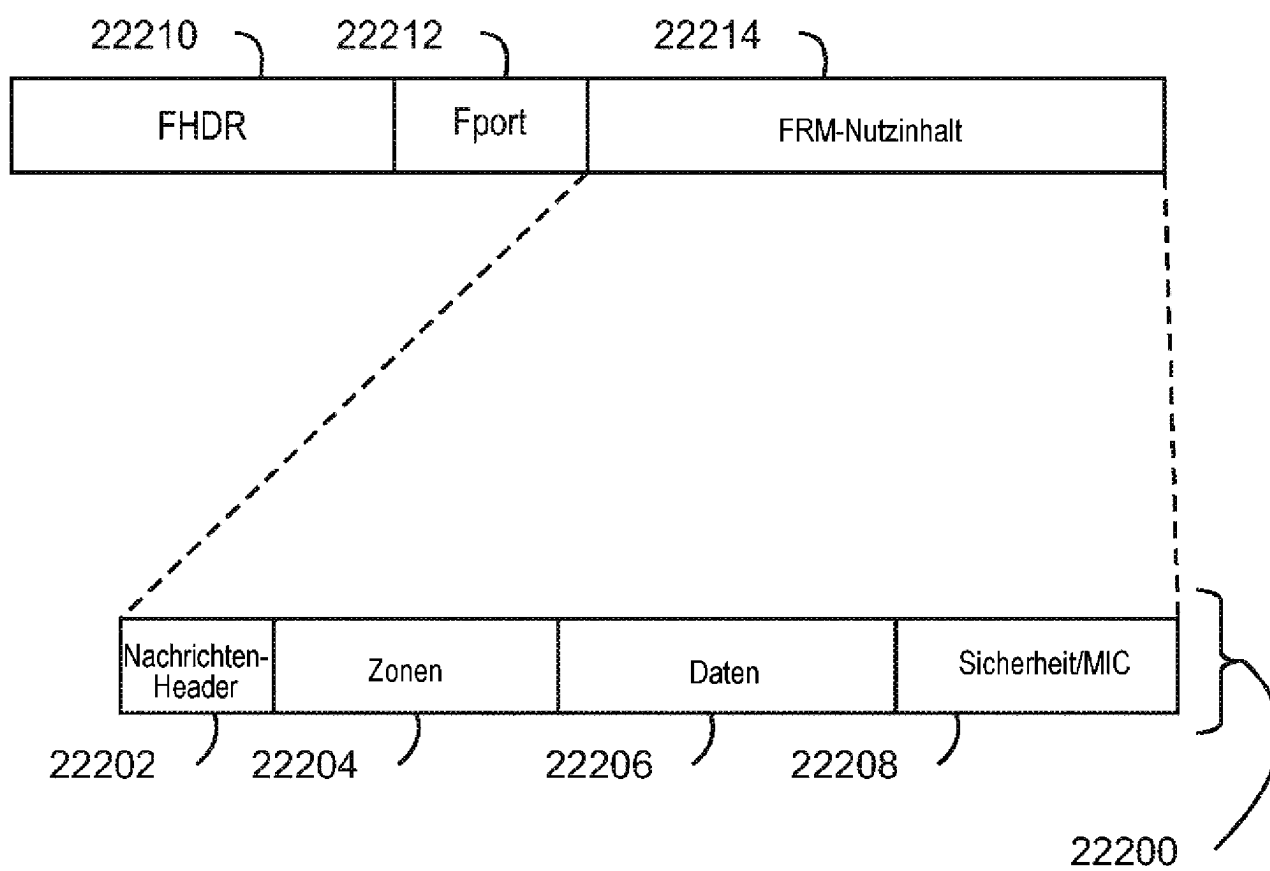


FIG. 222

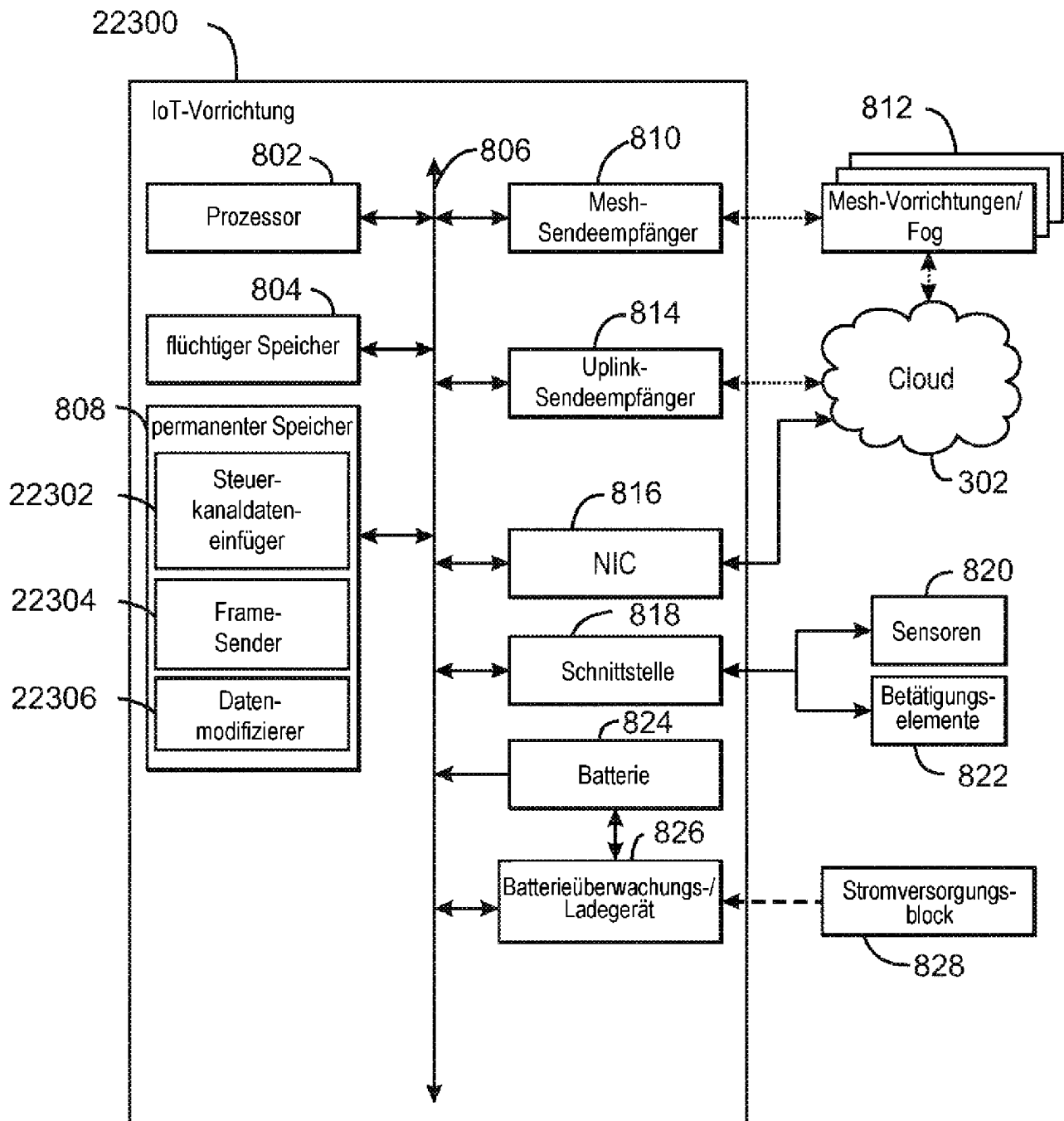


FIG. 223

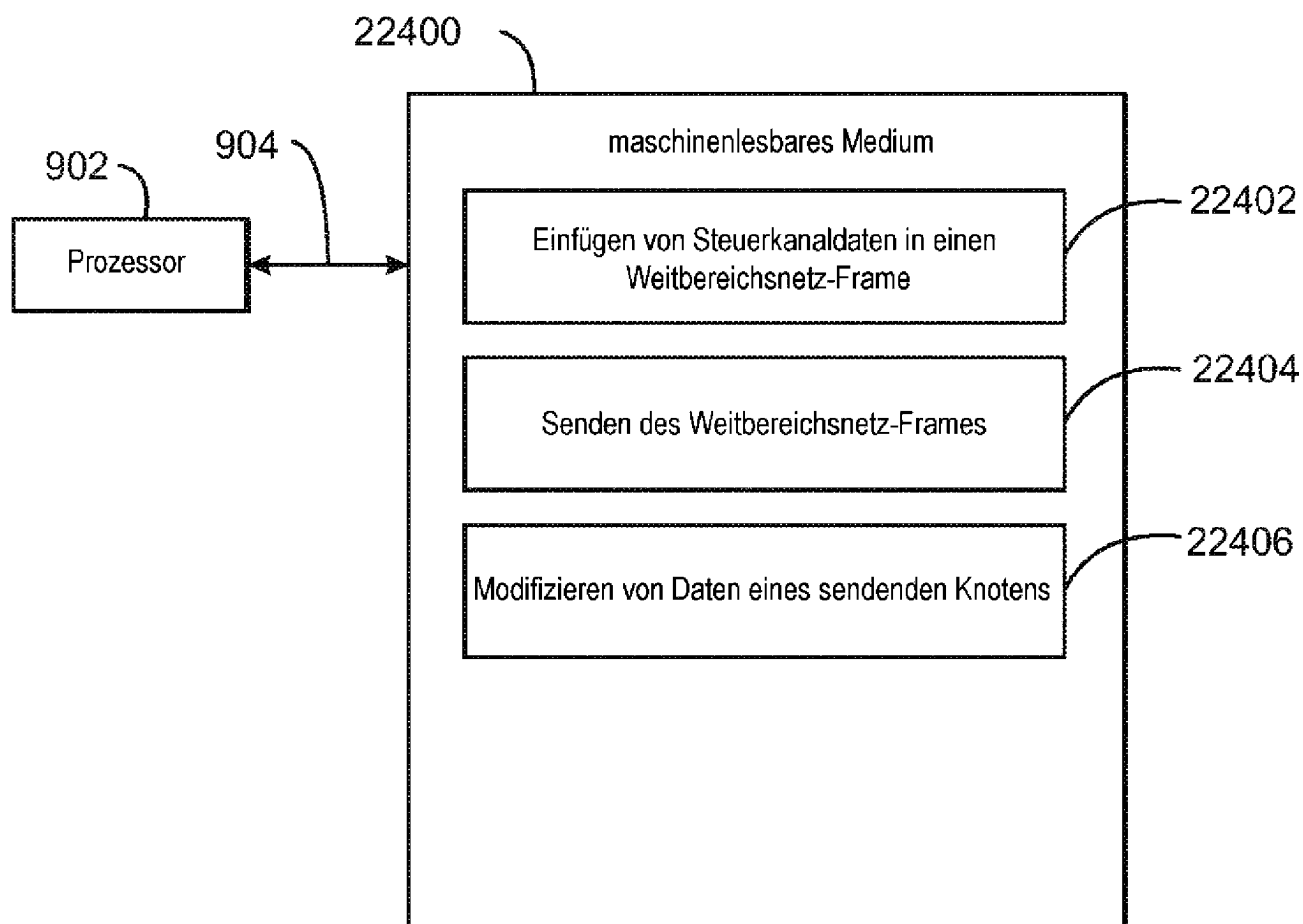
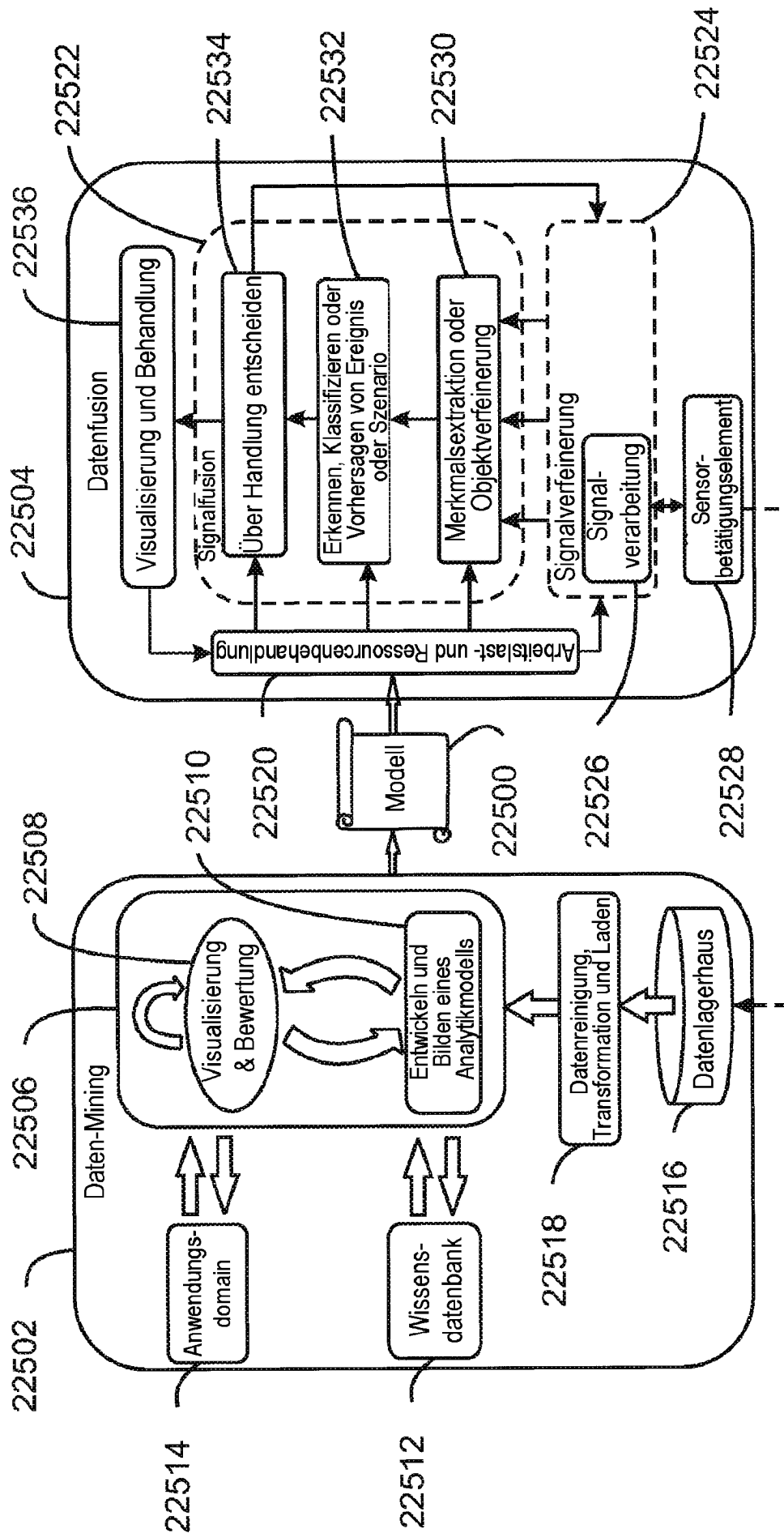
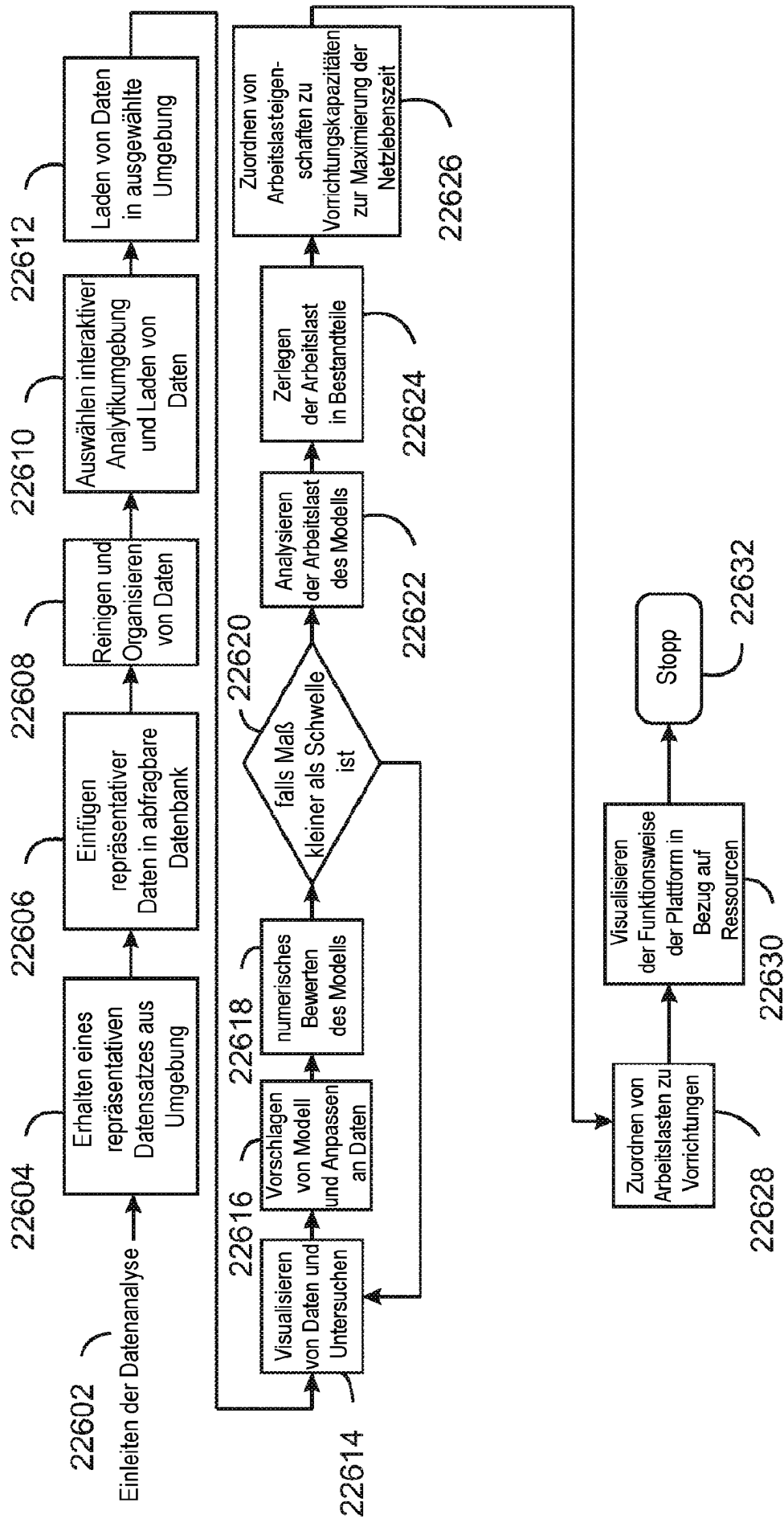


FIG. 224



22500

FIG. 225



22600

FIG. 226

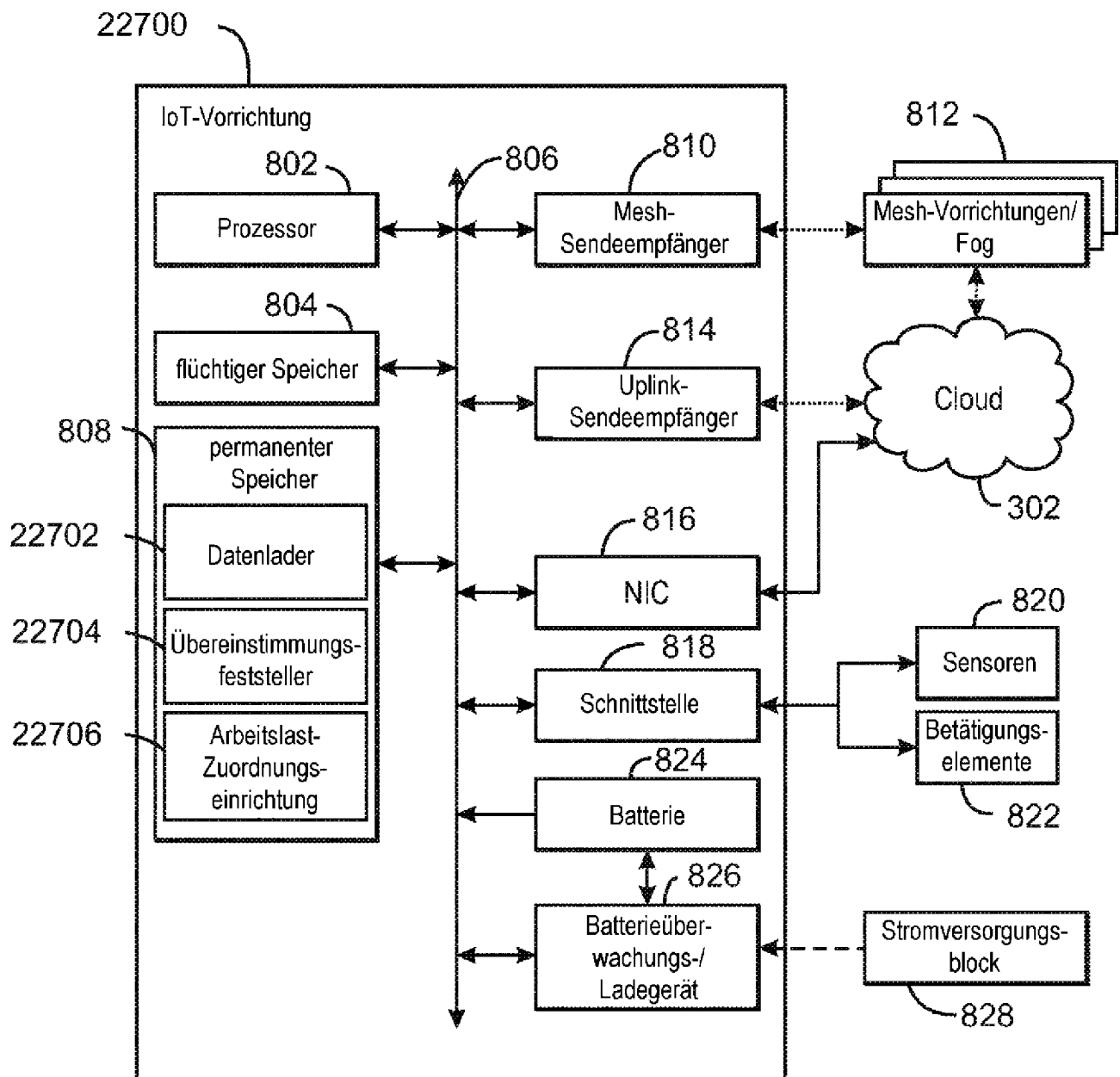


FIG. 227

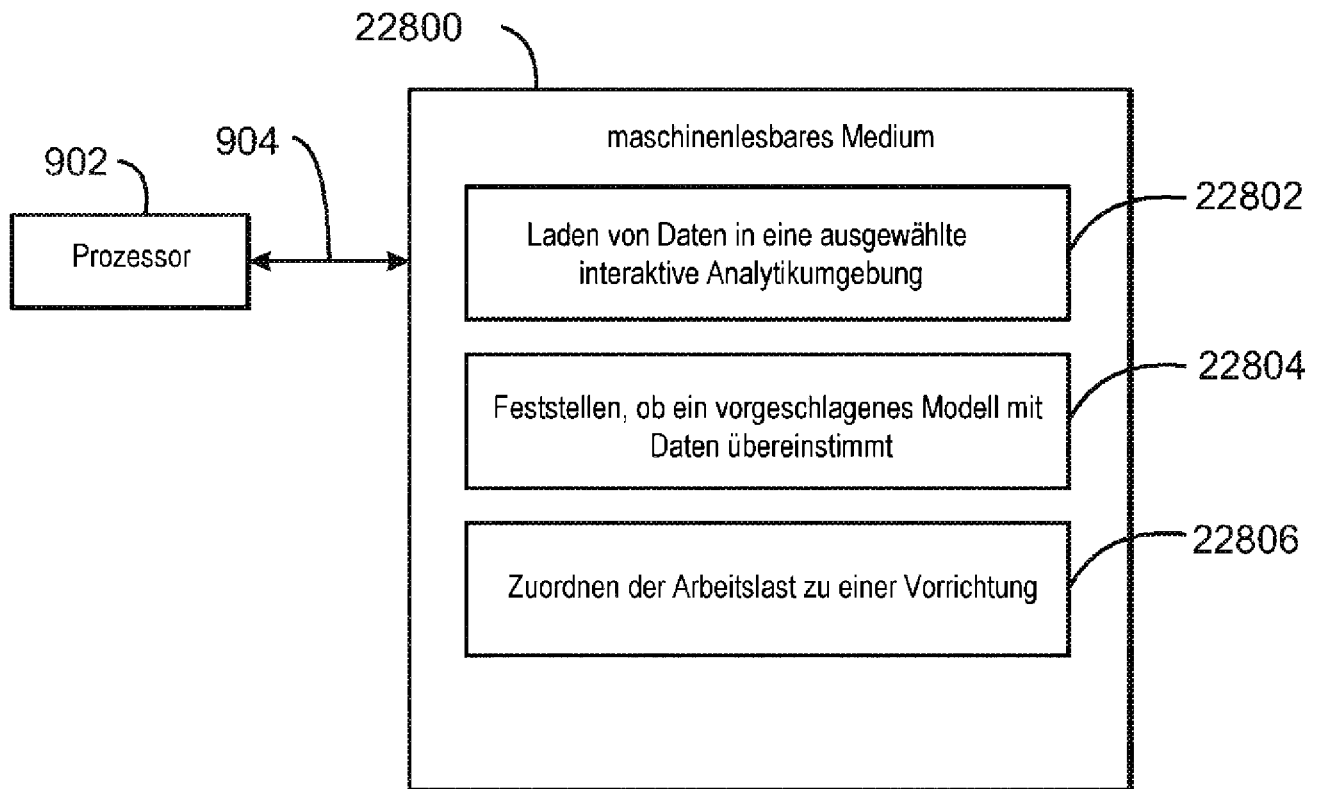
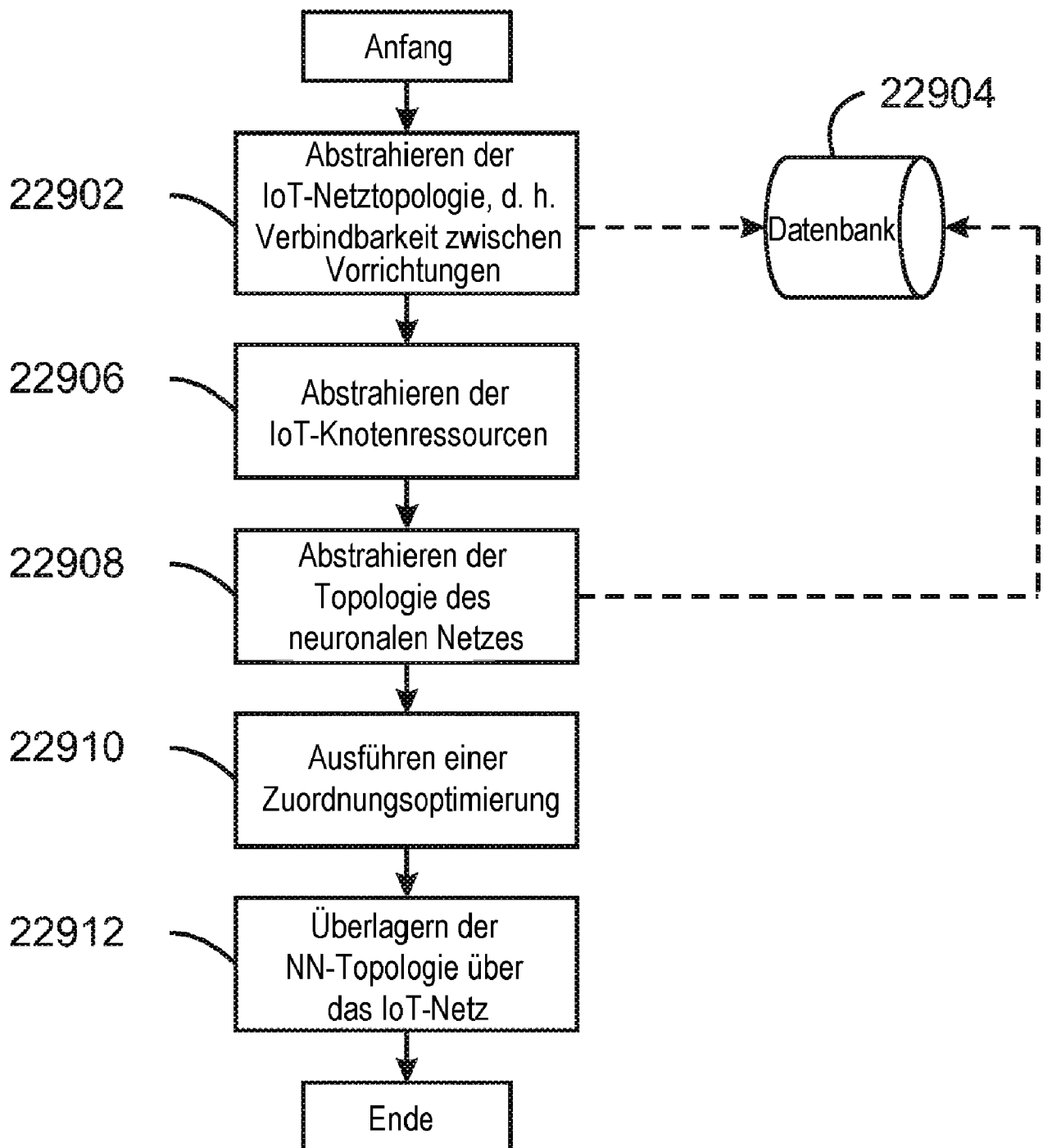
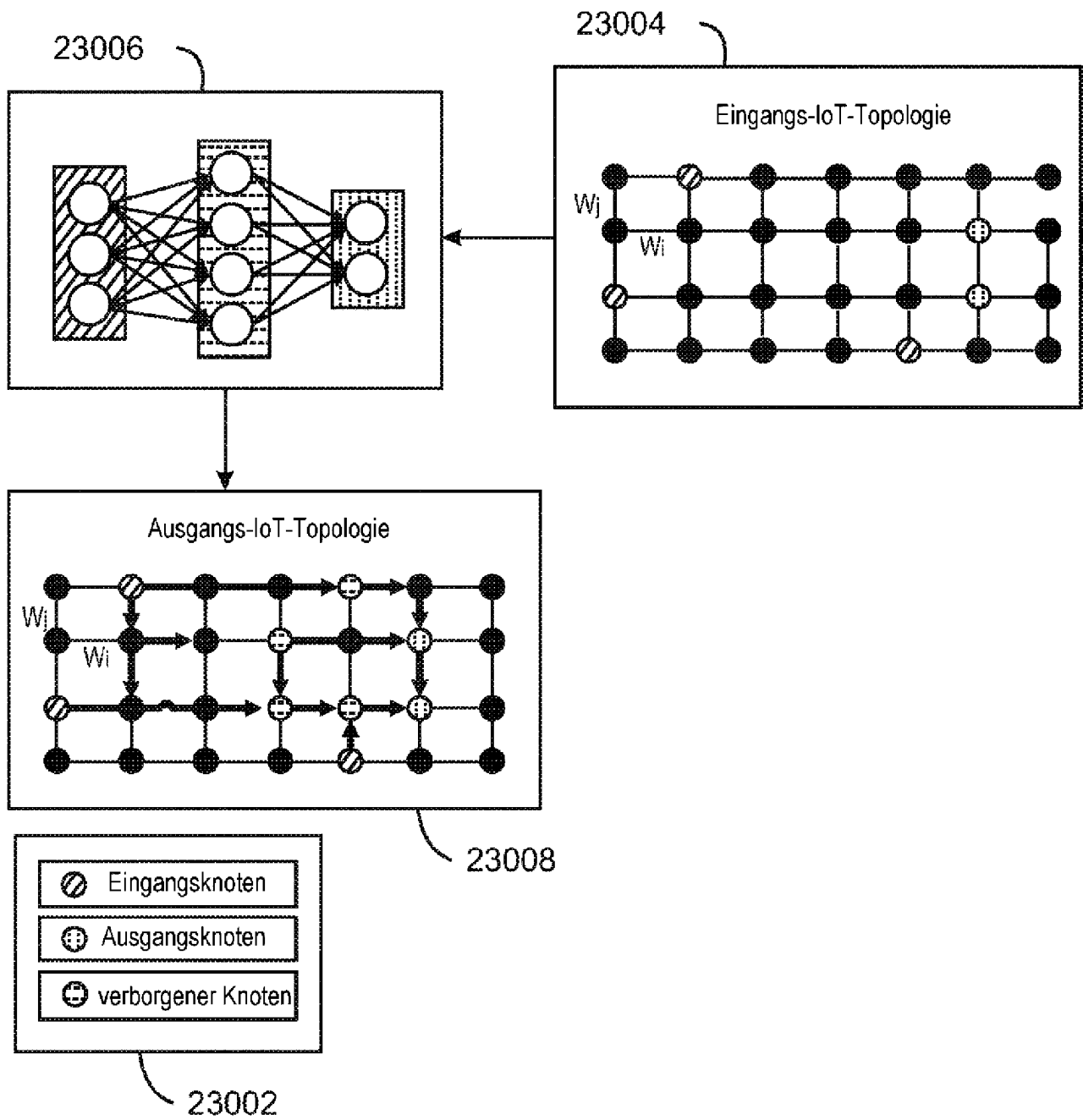


FIG. 228



22900
FIG. 229



23000
FIG. 230

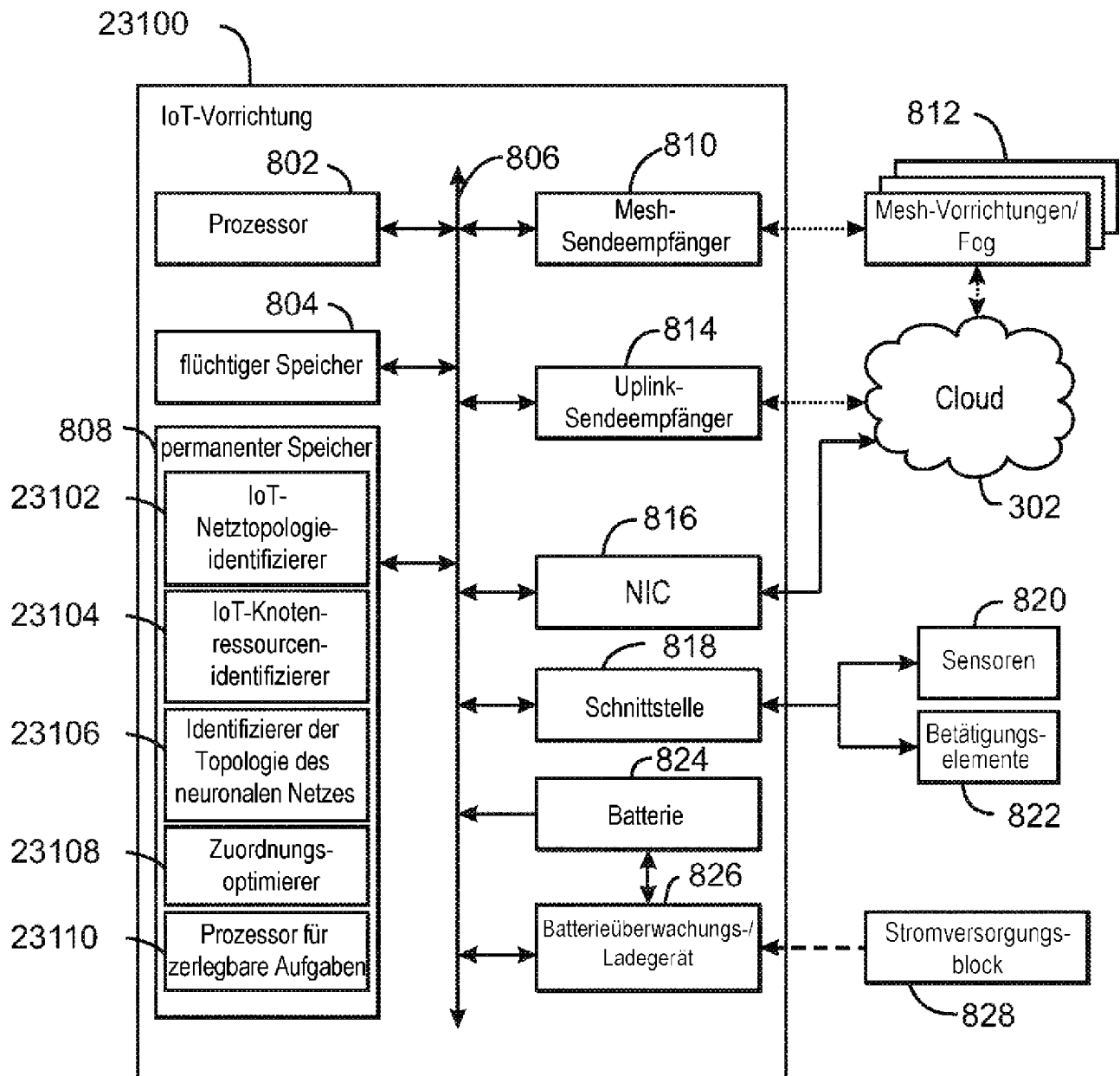


FIG. 231

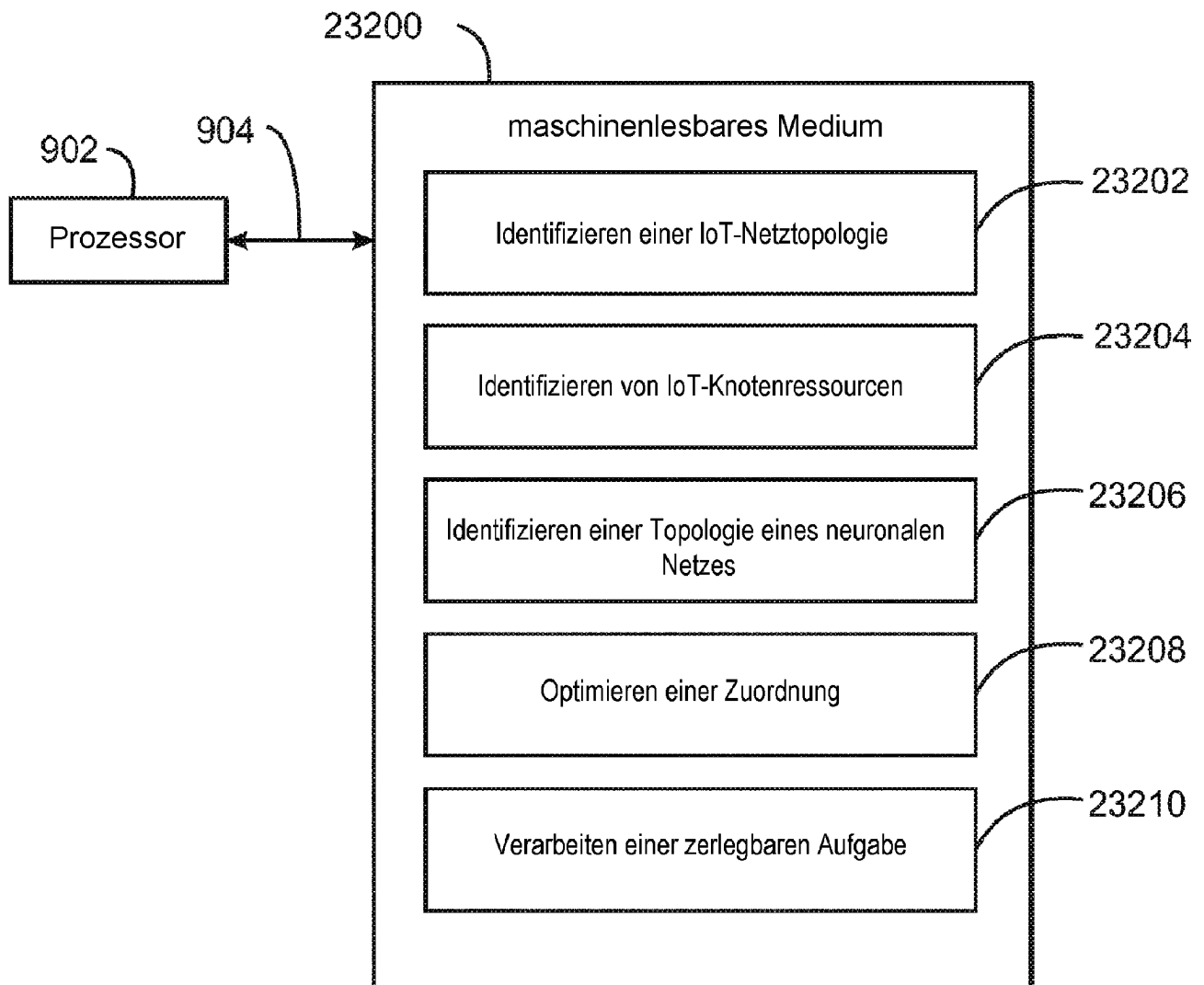
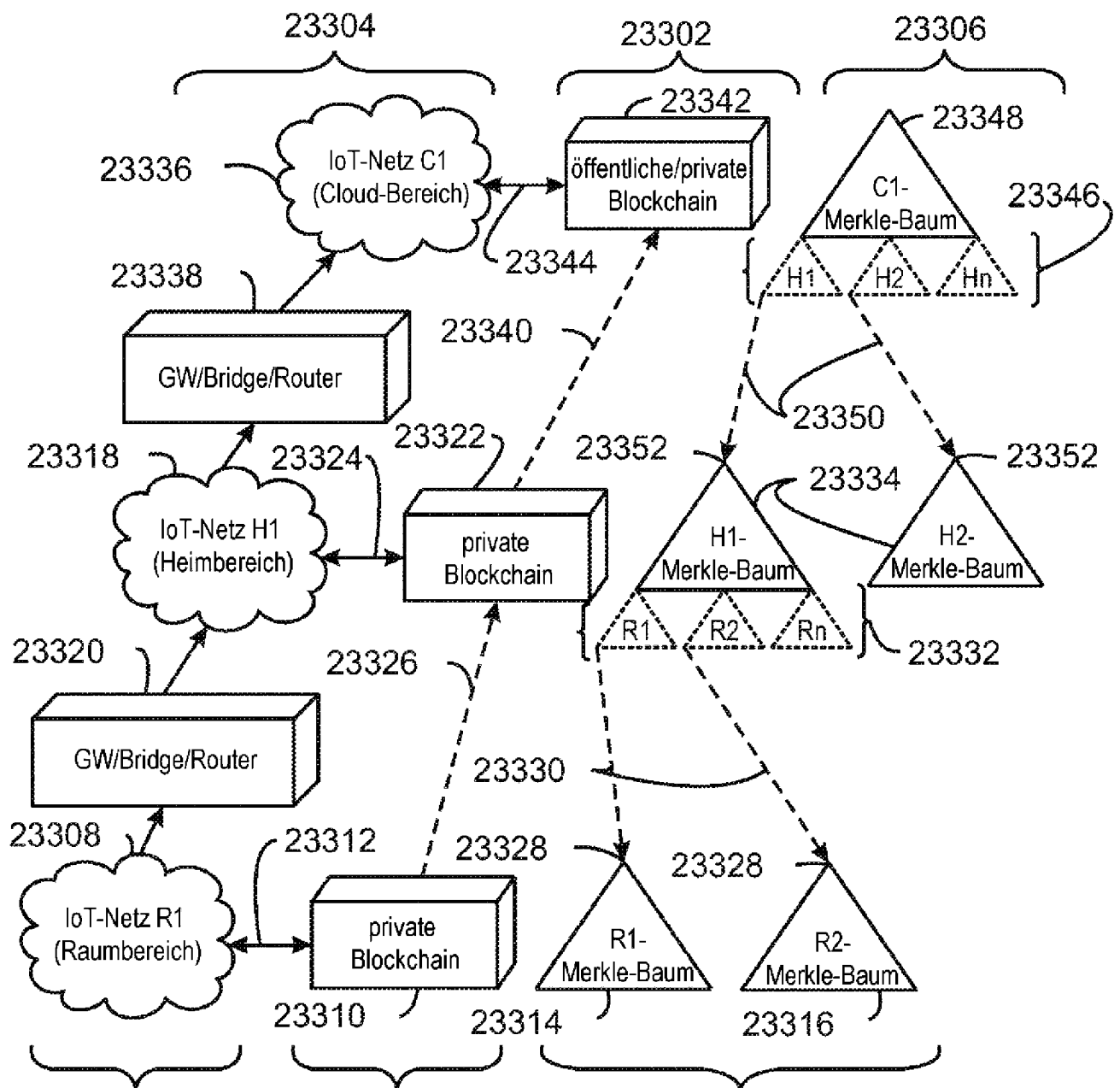


FIG. 232



23300
FIG. 233

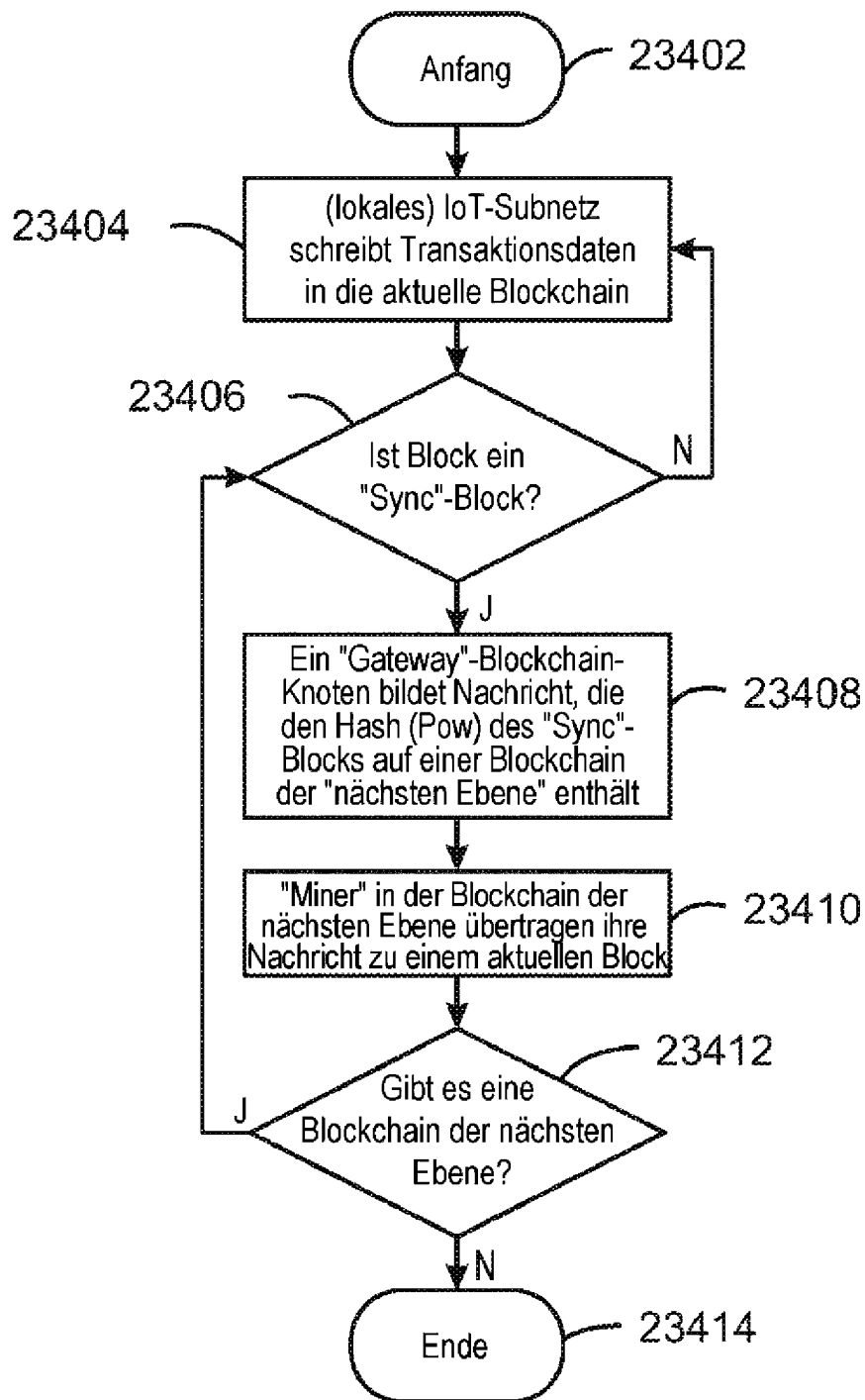
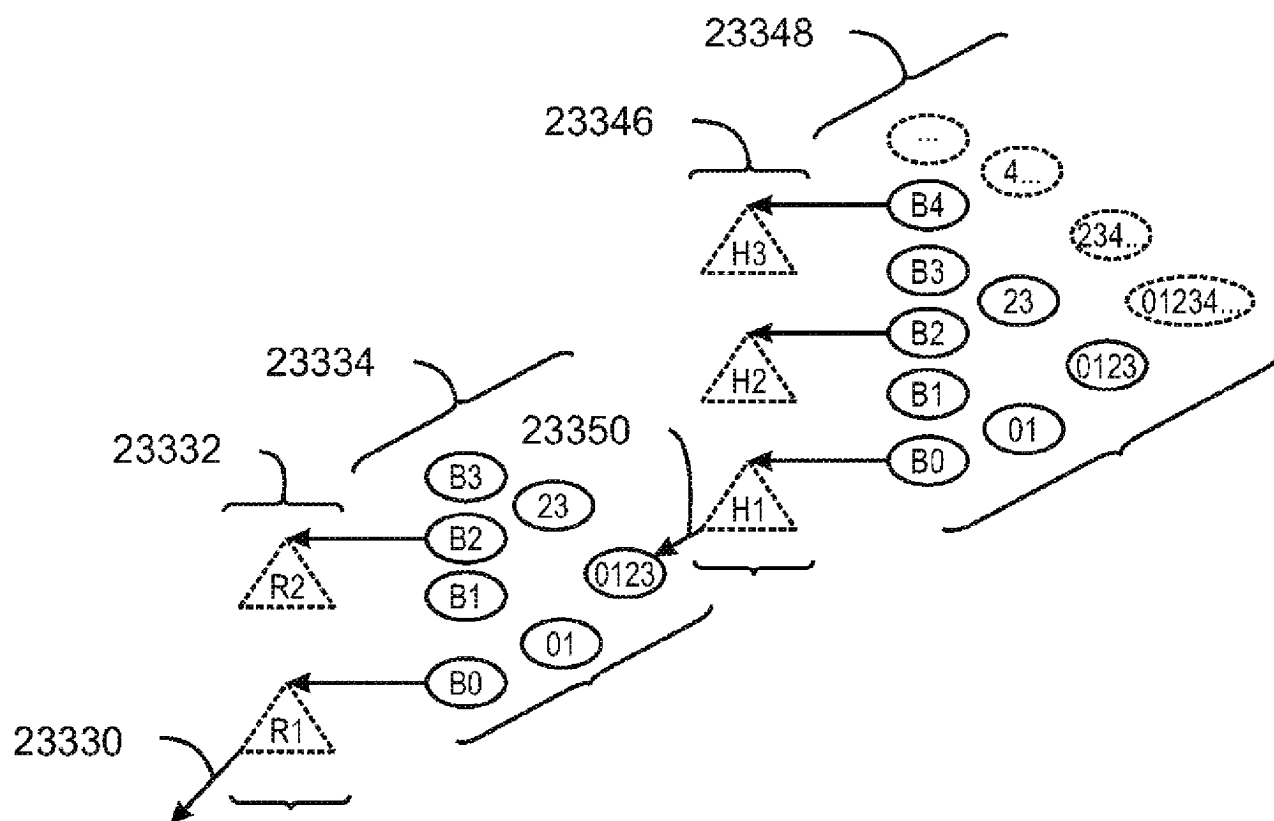
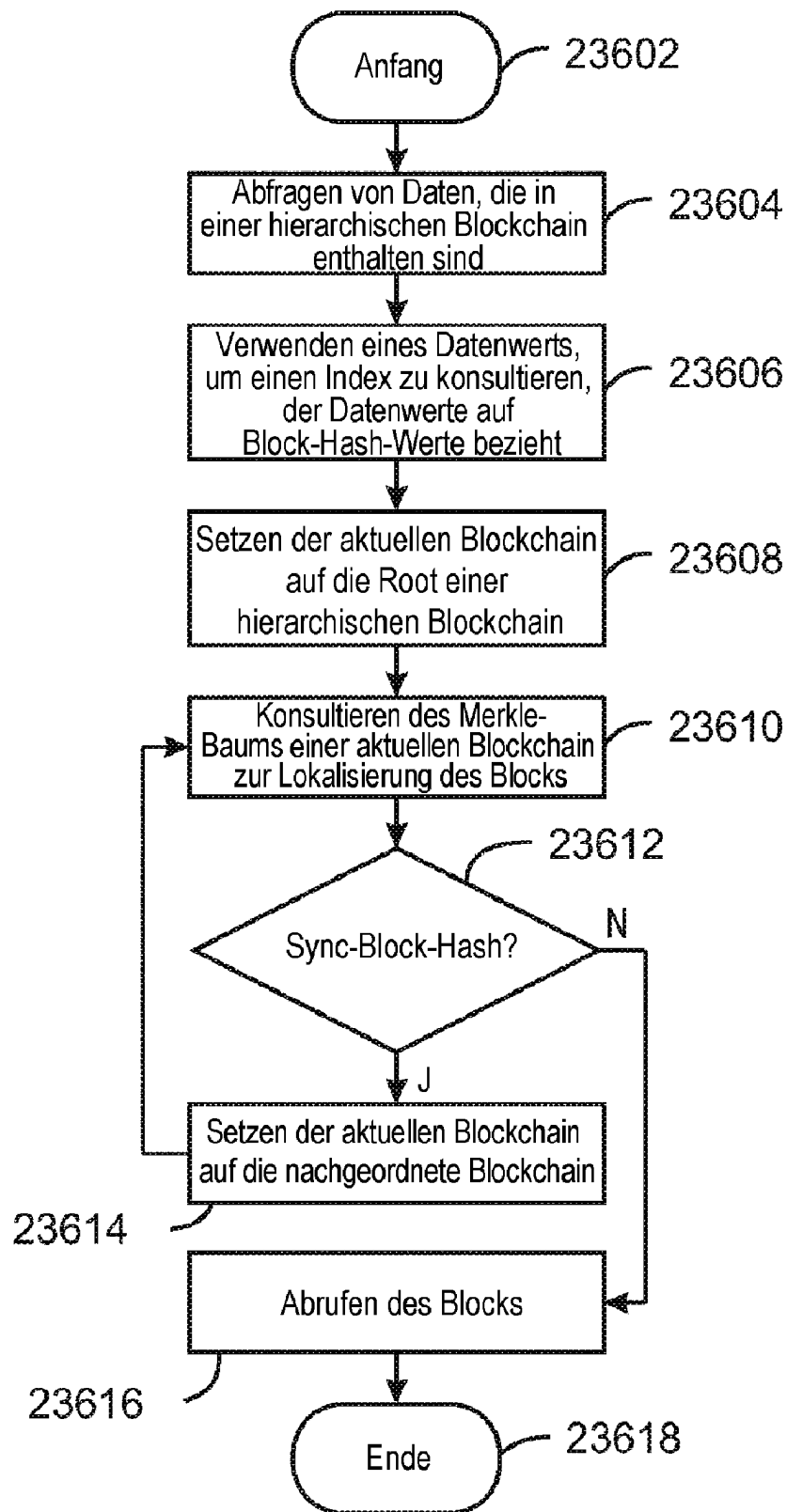
23400

FIG. 234

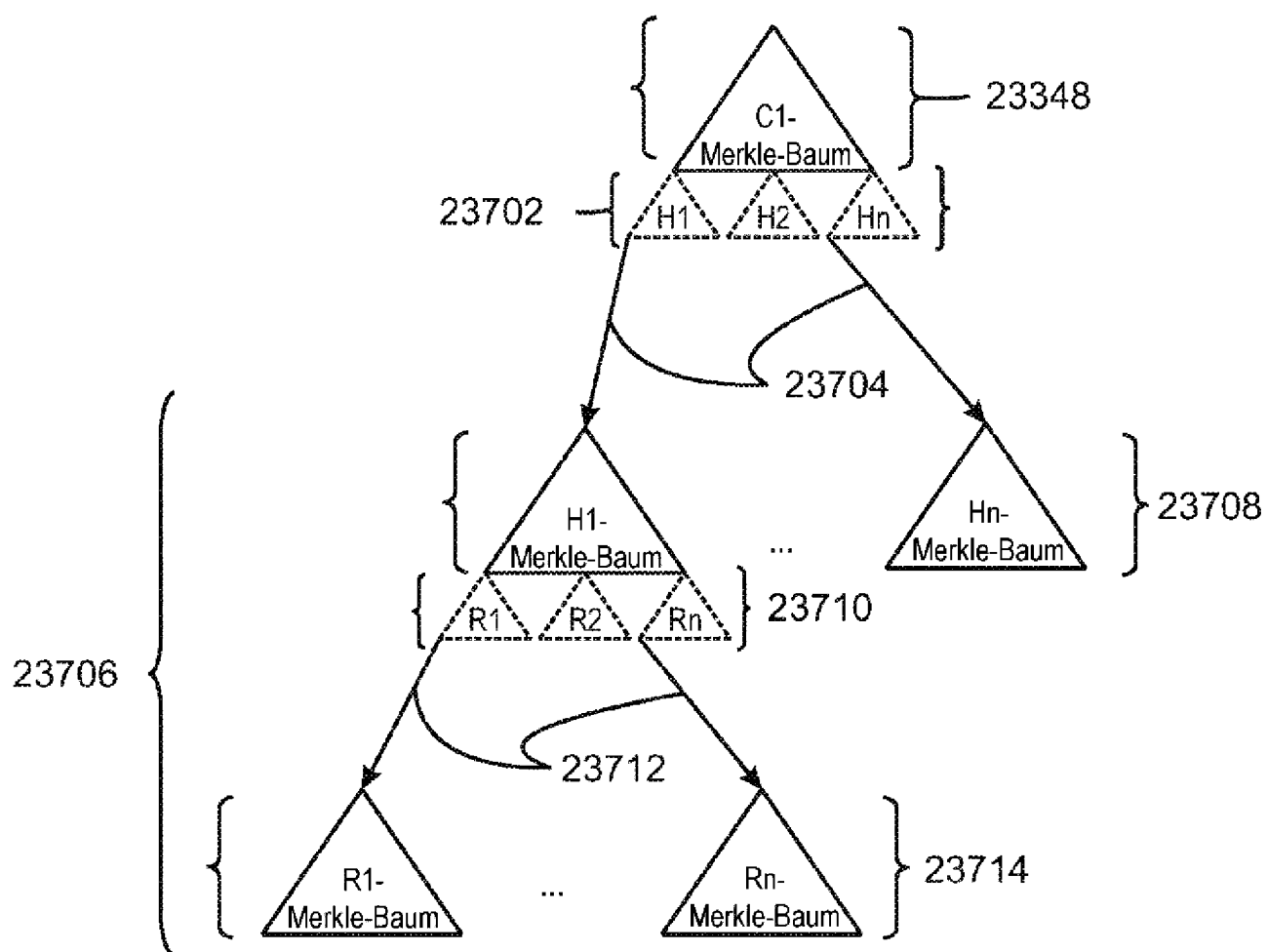


23500

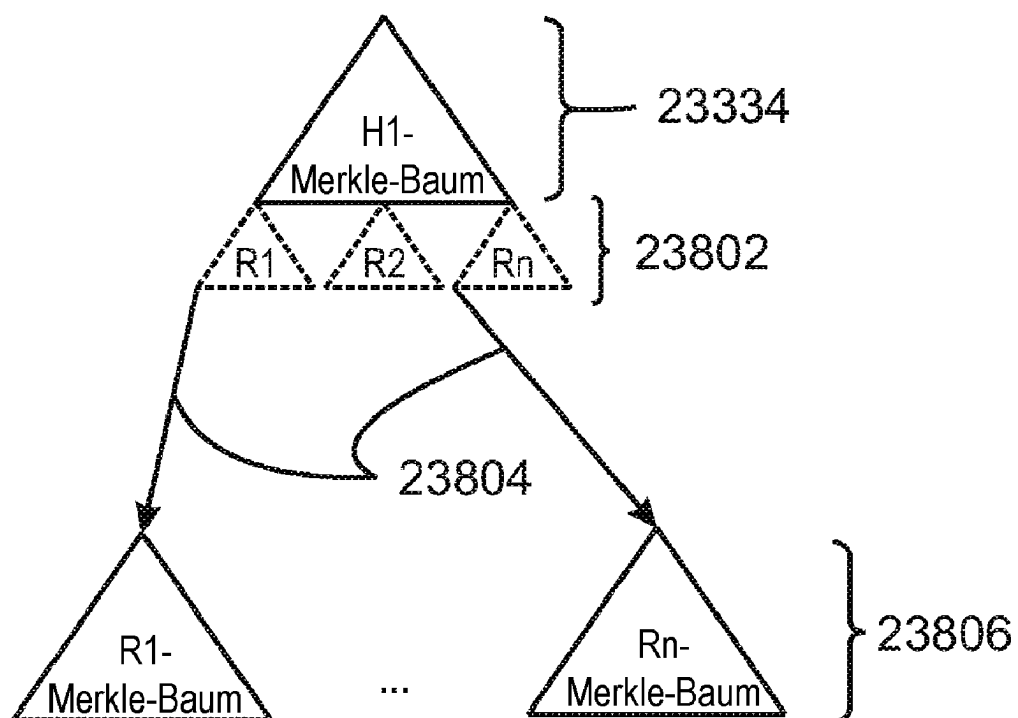
FIG. 235



23600
FIG. 236

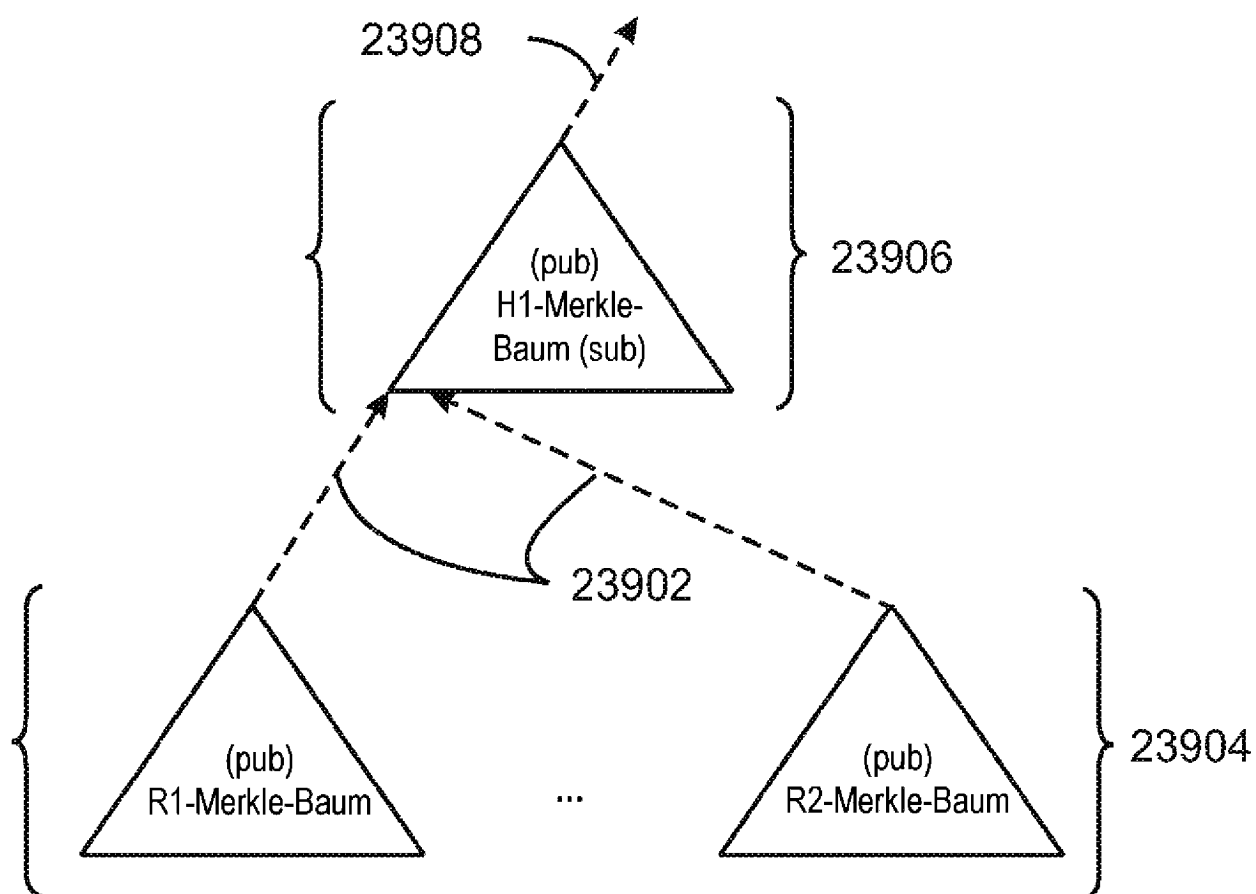


23700
FIG. 237

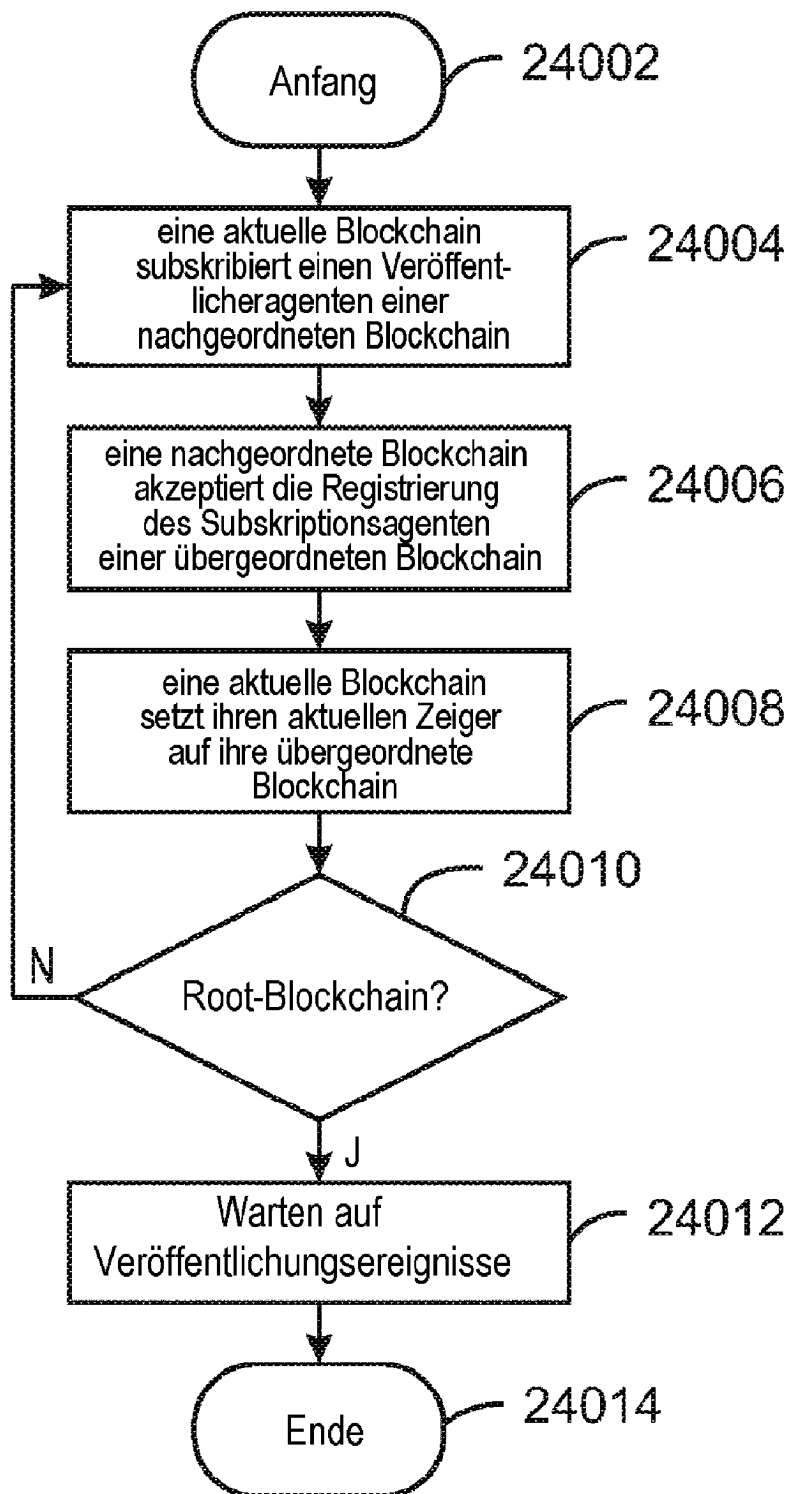


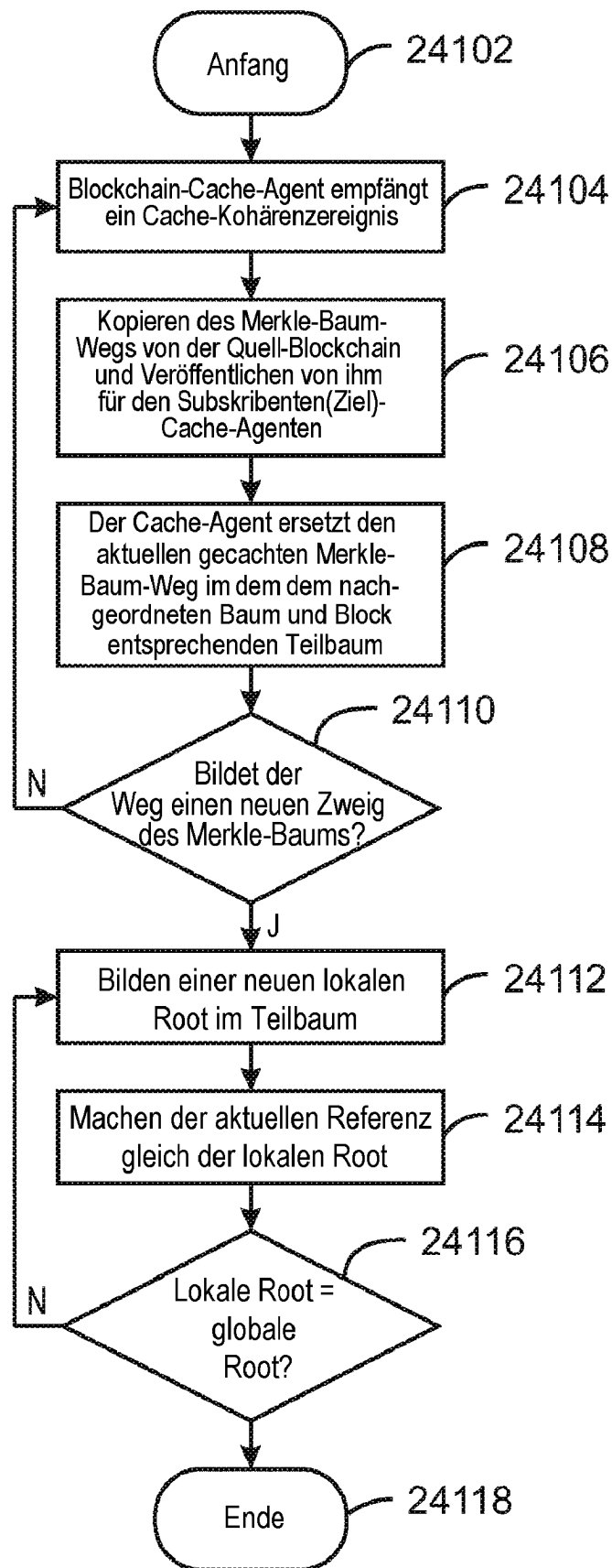
23800

FIG. 238



23900
FIG. 239

24000**FIG. 240**



24100

FIG. 241

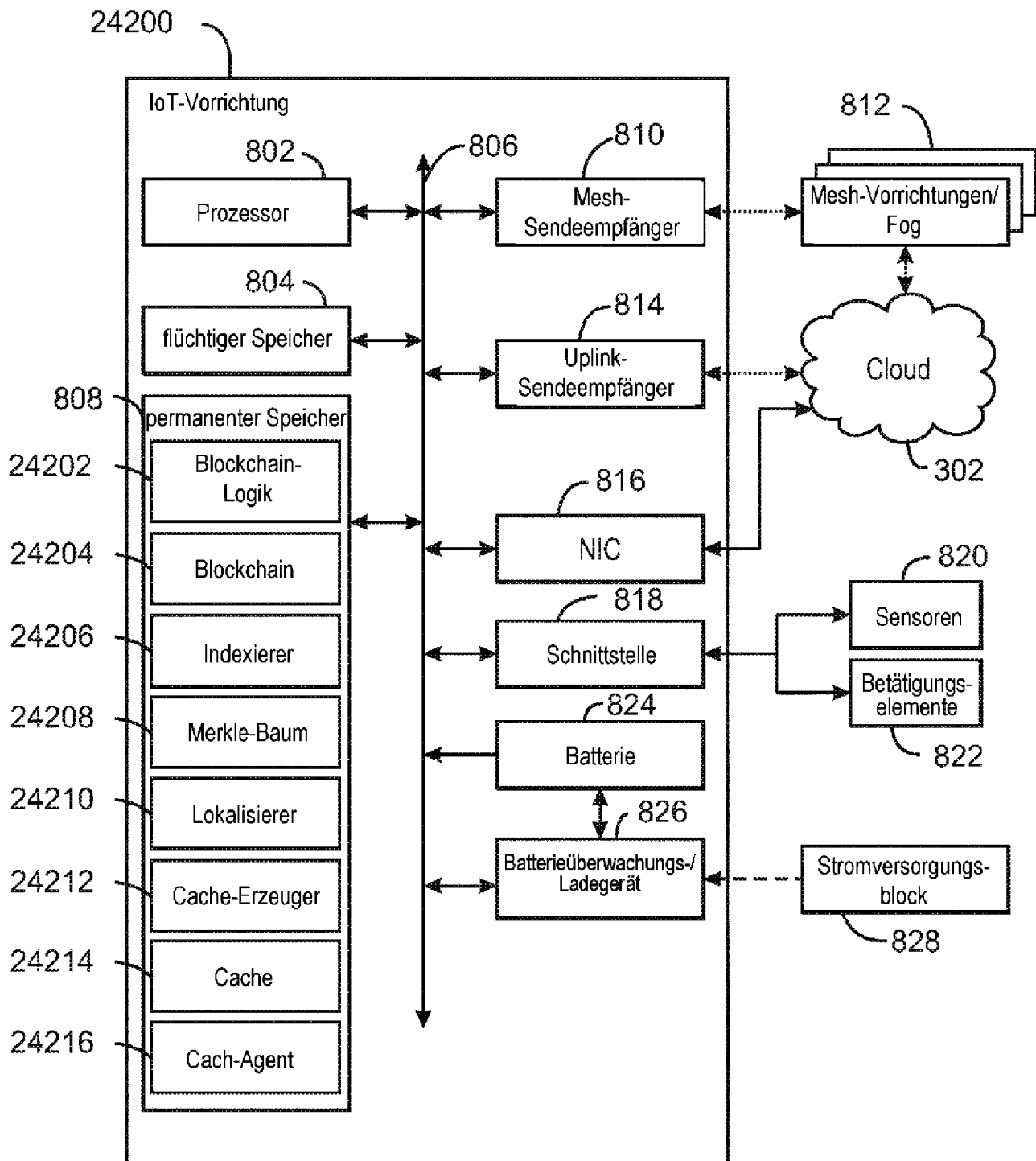


FIG. 242

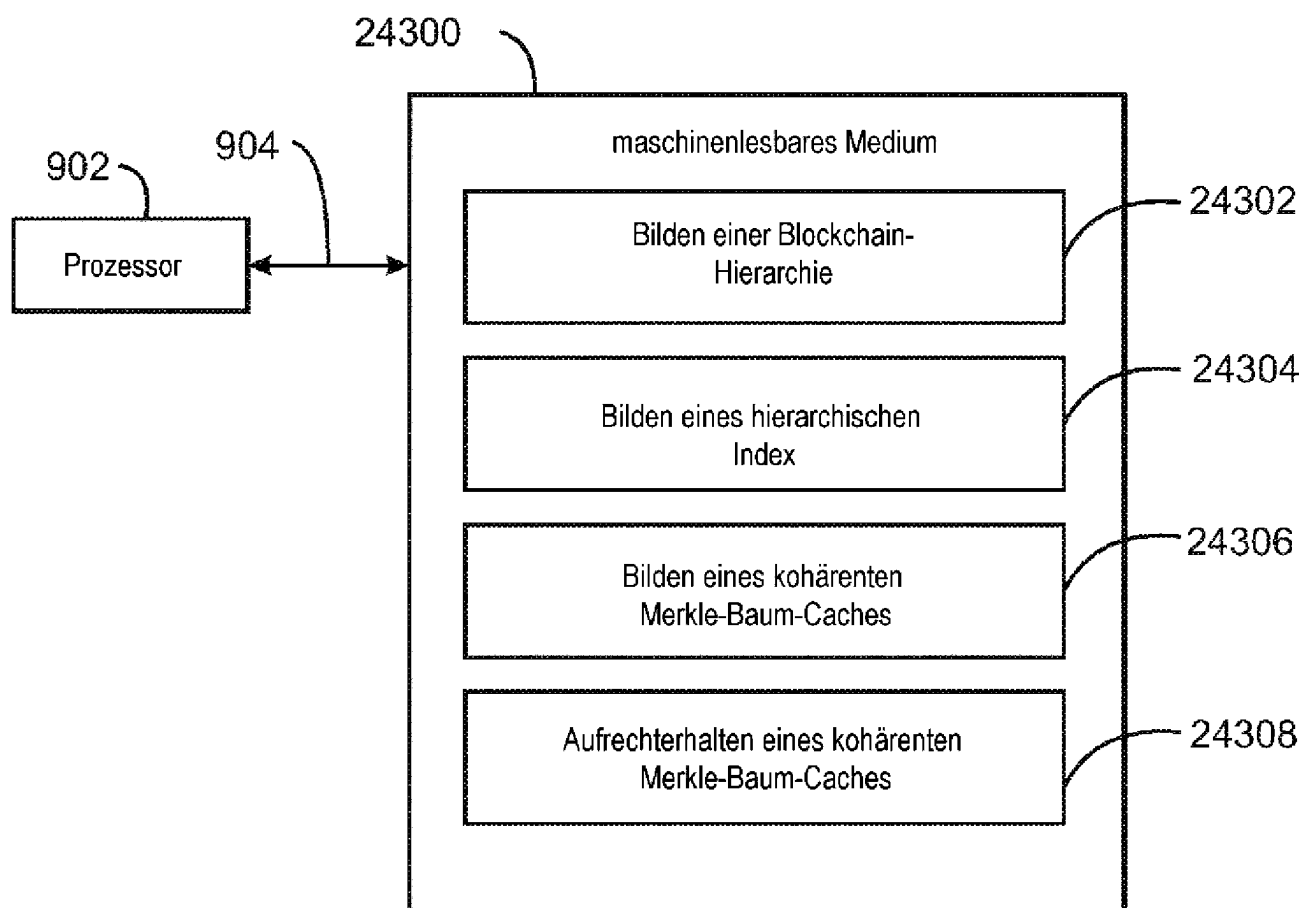
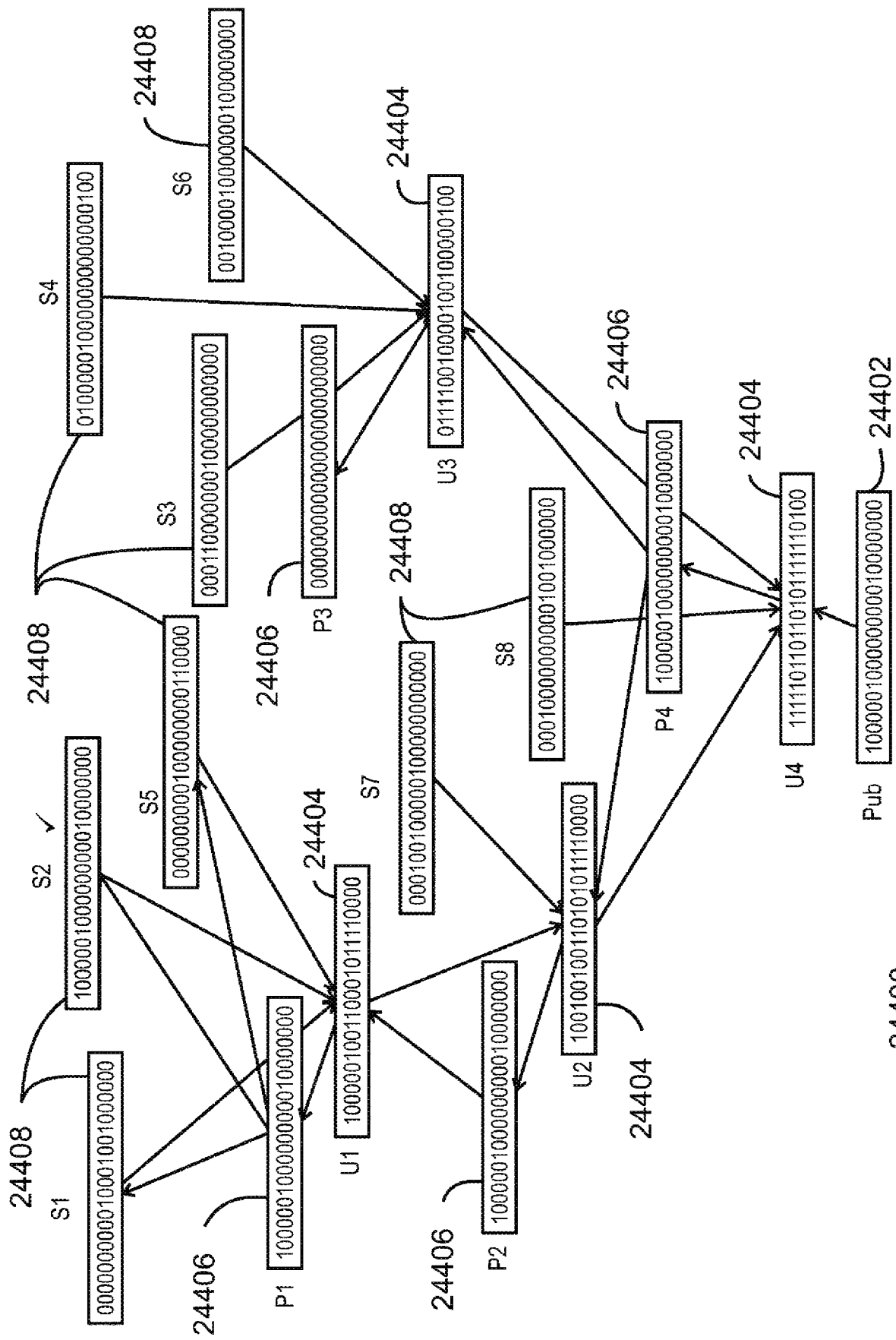
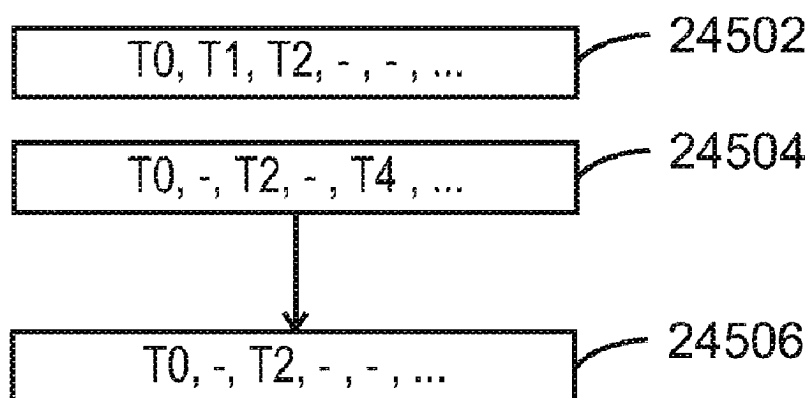


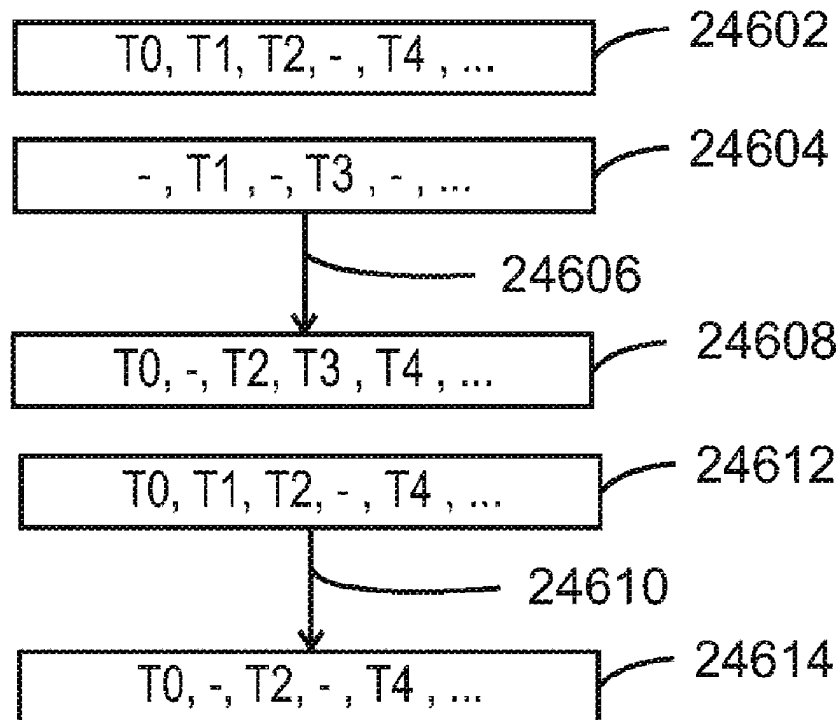
FIG. 243



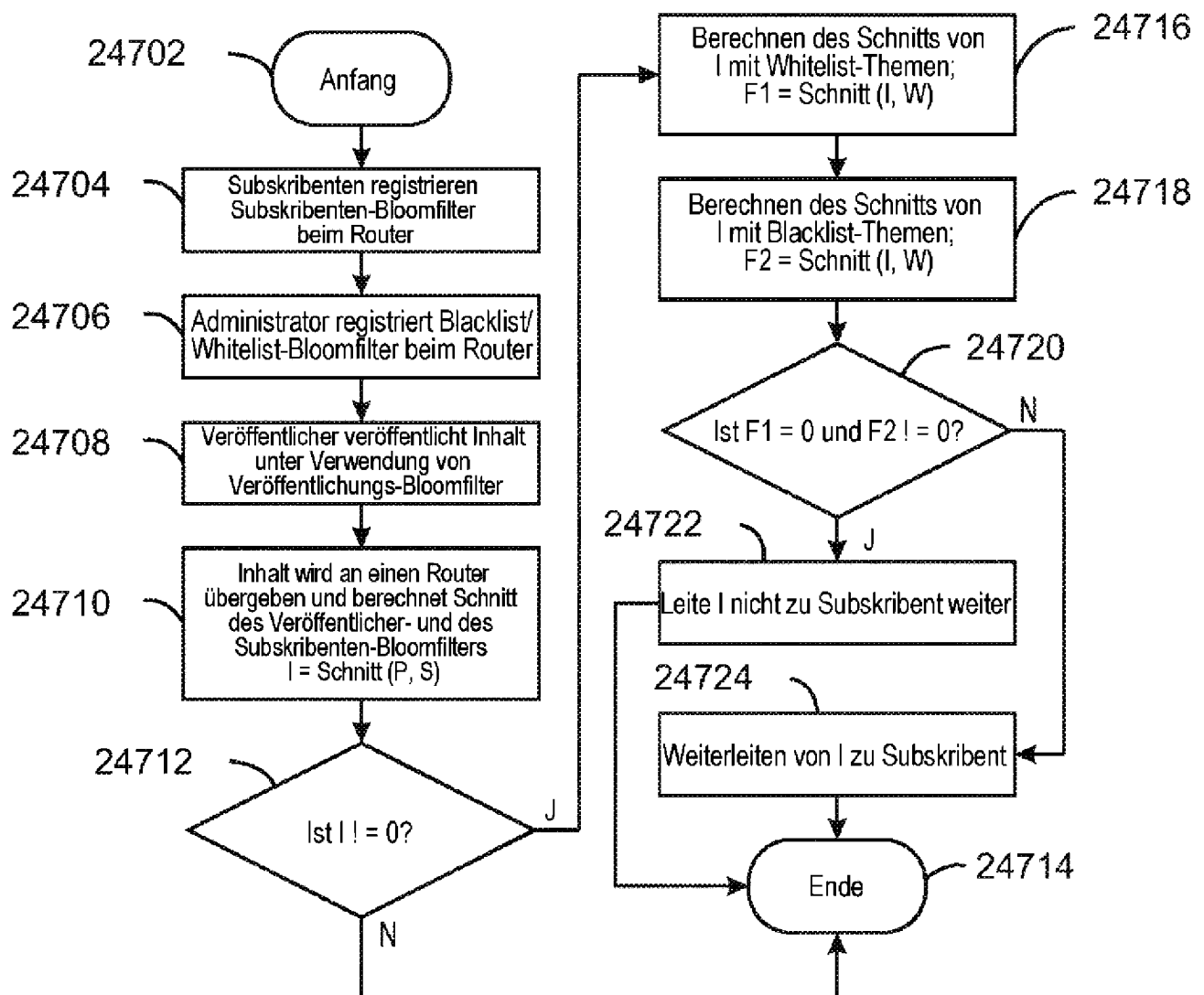
24400
FIG. 244



24500
FIG. 245



24600
FIG. 246



24700

FIG. 247

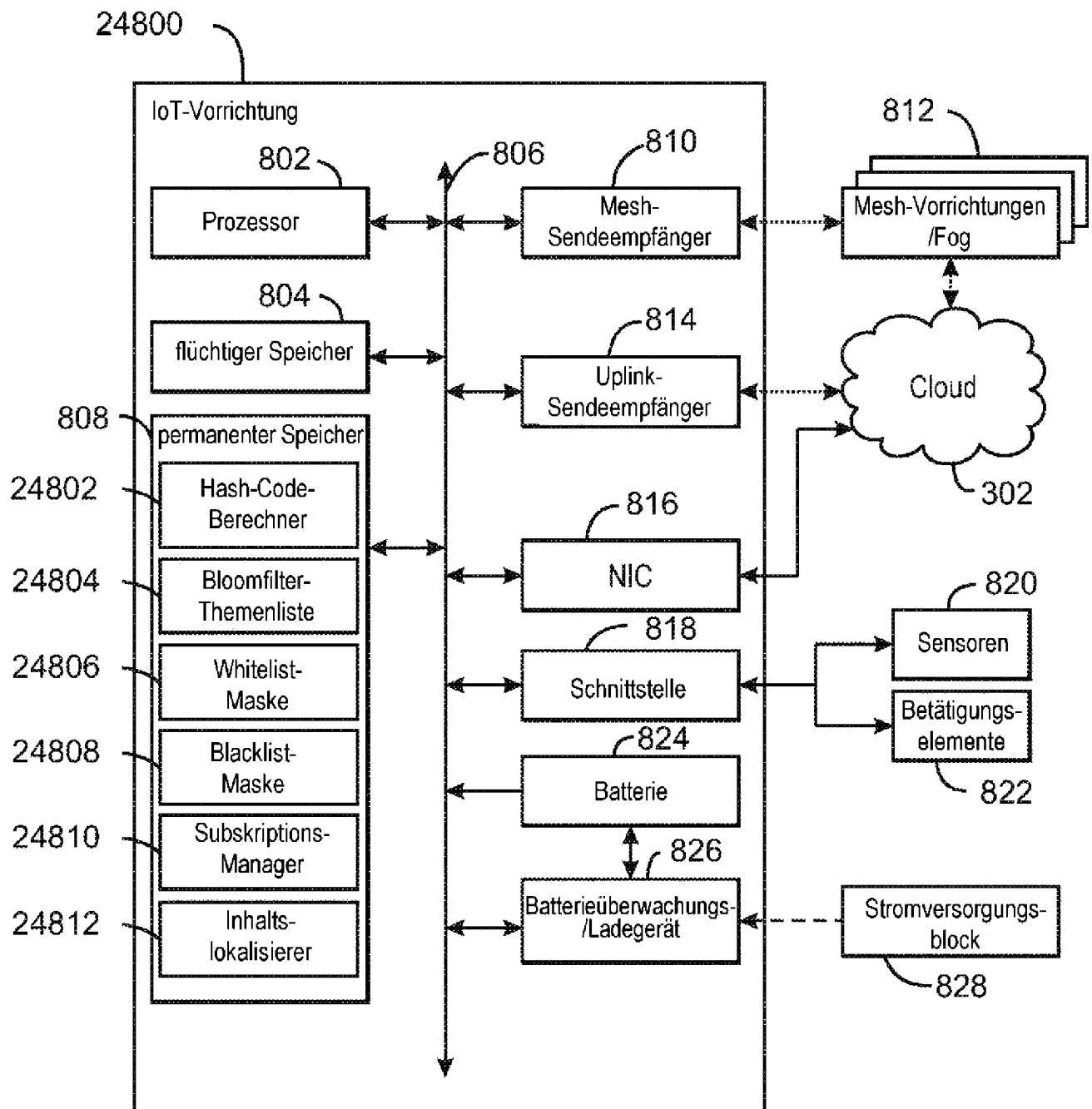


FIG. 248

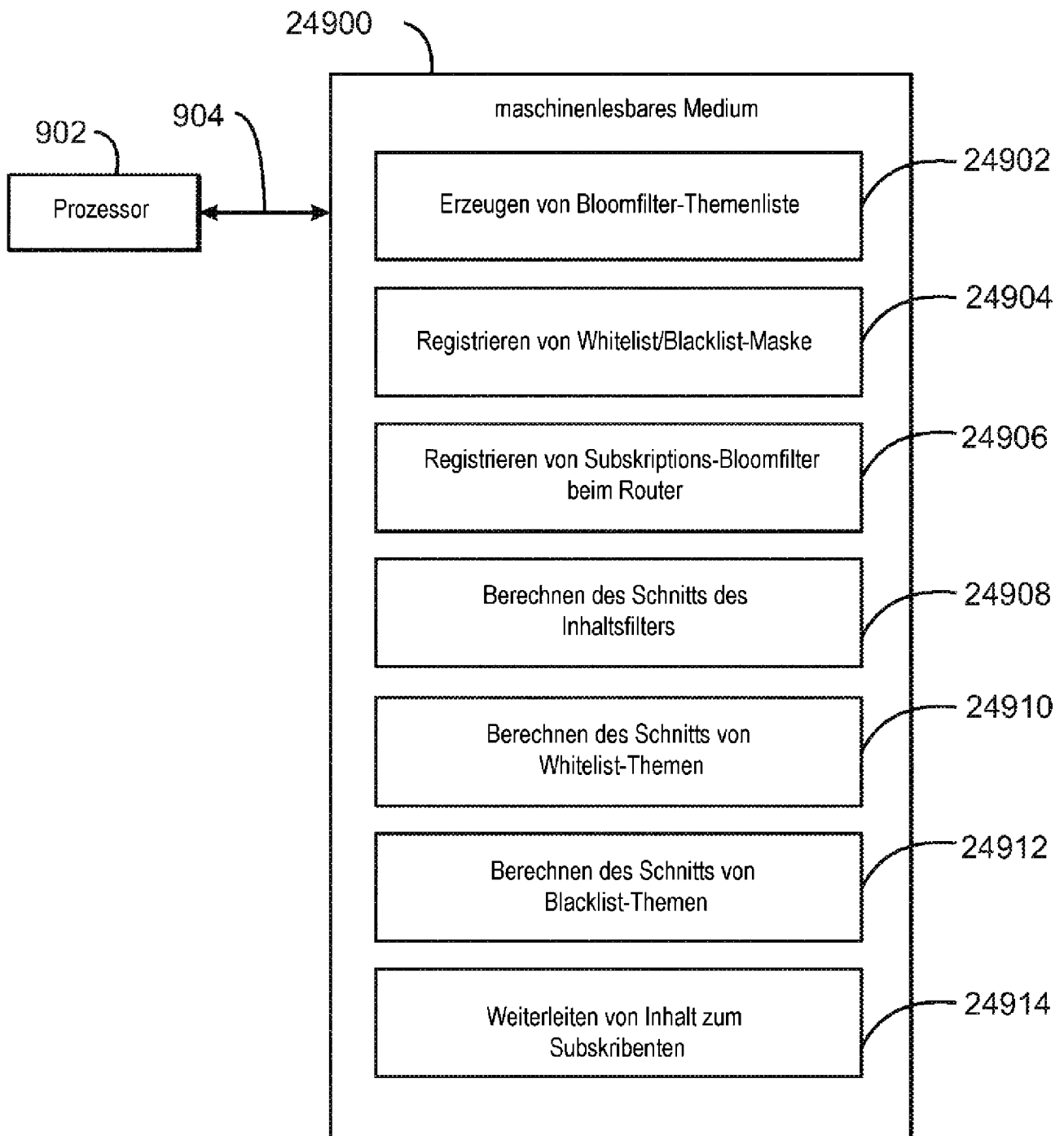
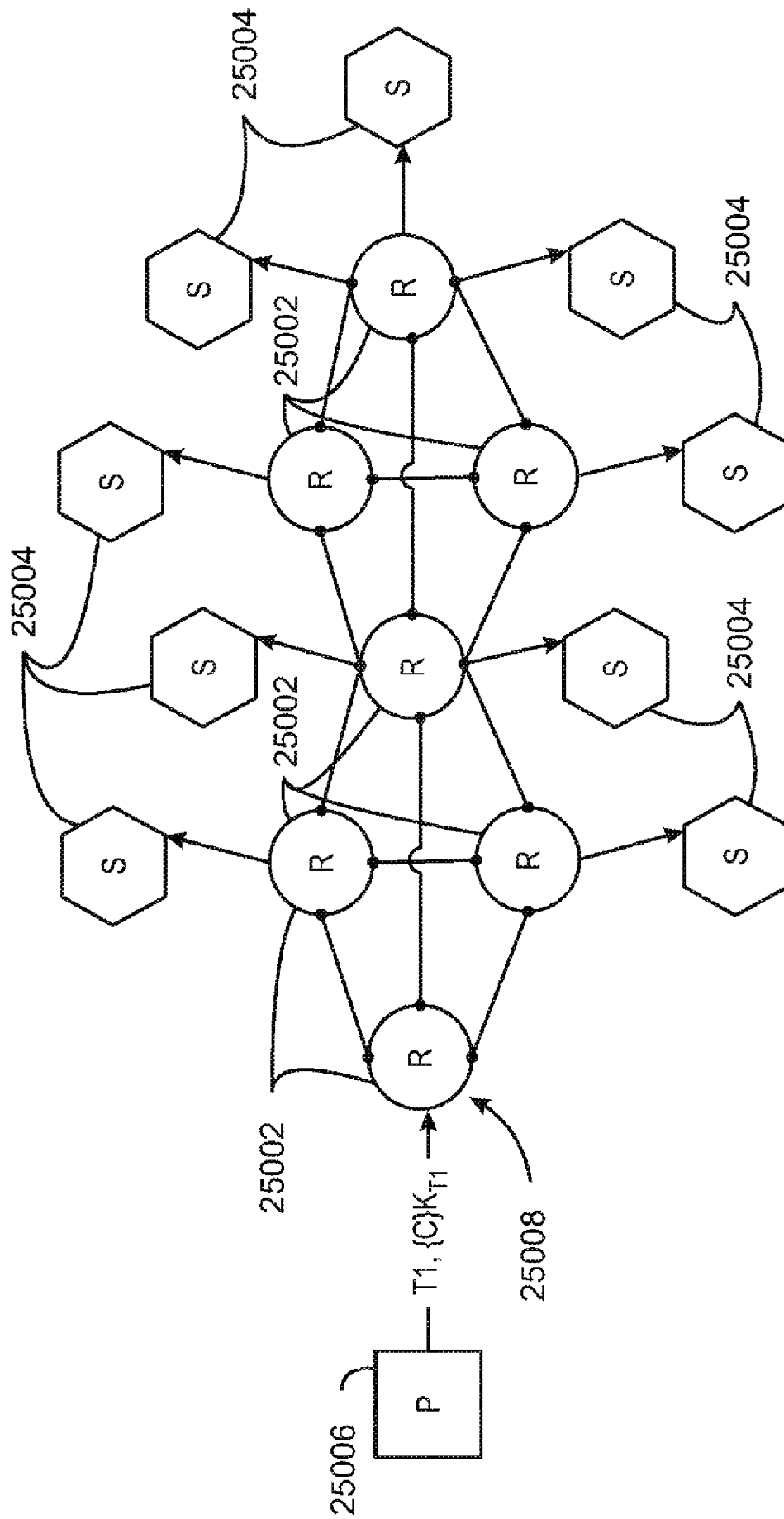
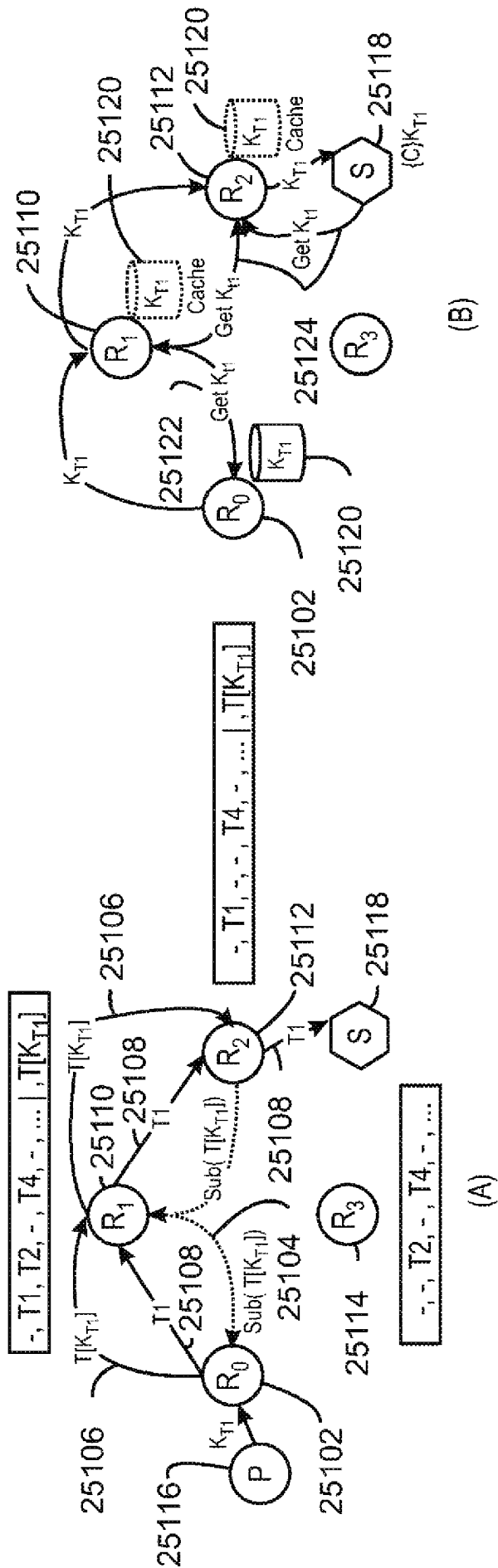


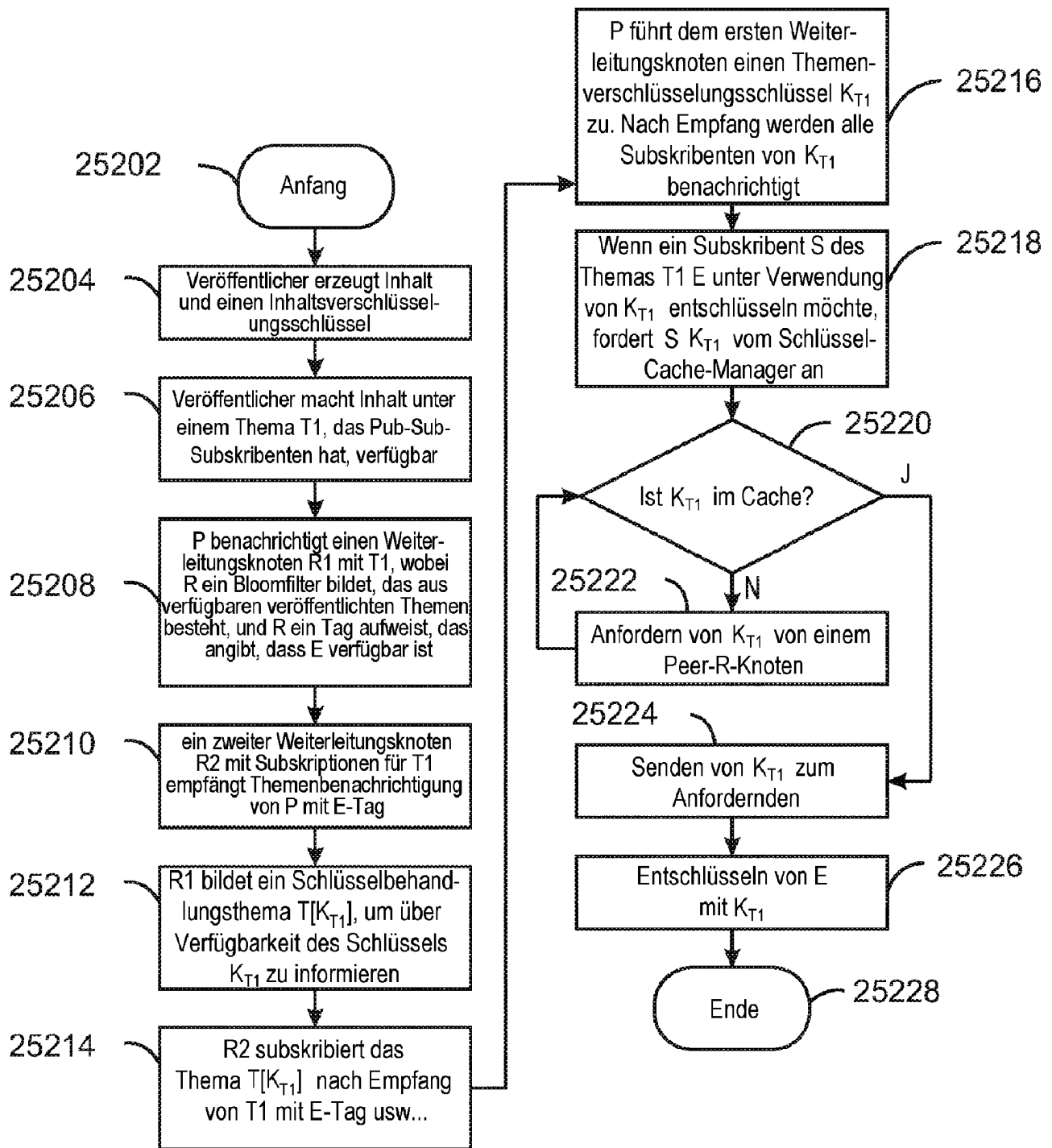
FIG. 249



25000
FIG. 250



25100
FIG. 251



25200

FIG. 252

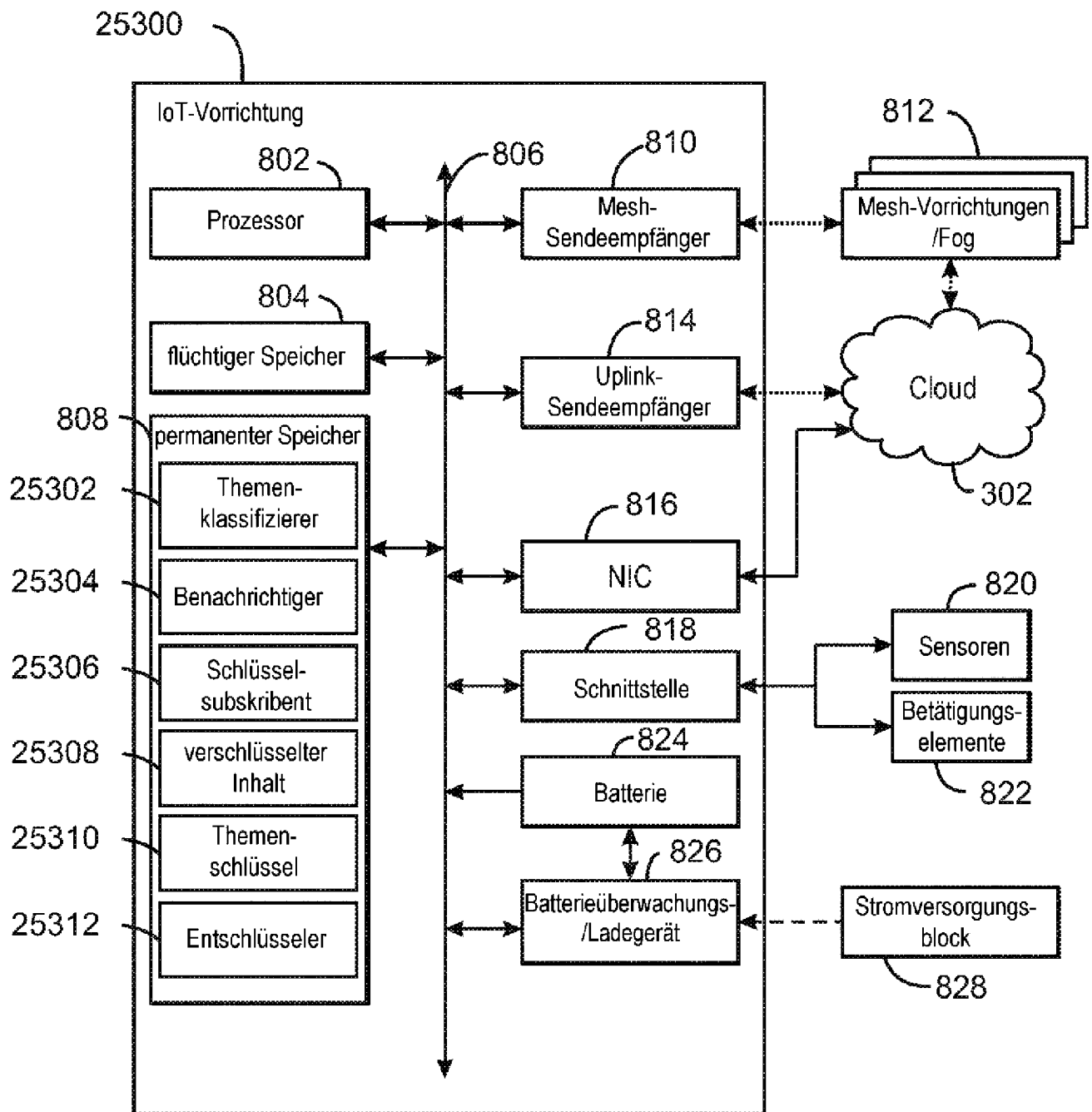


FIG. 253

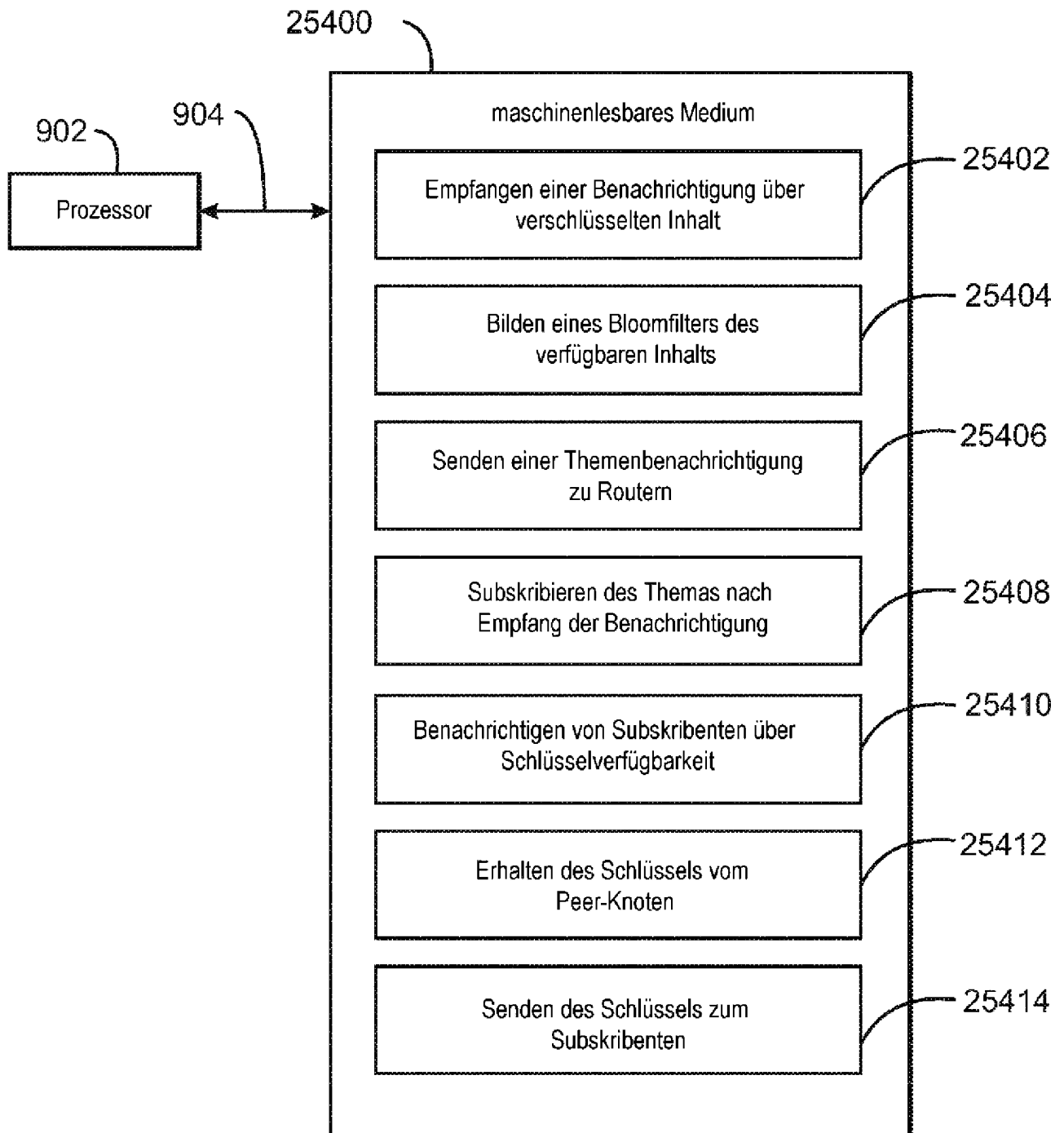
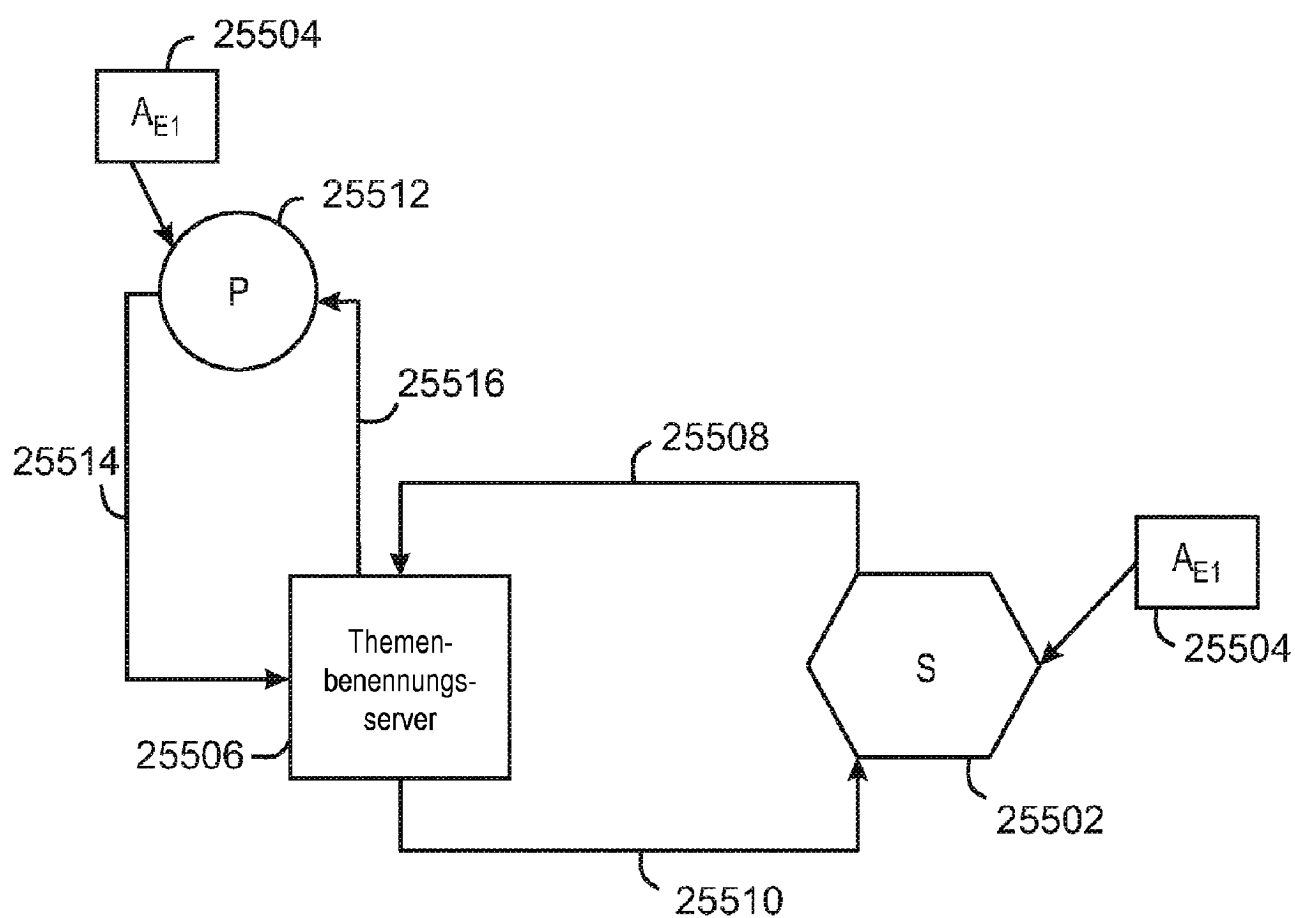
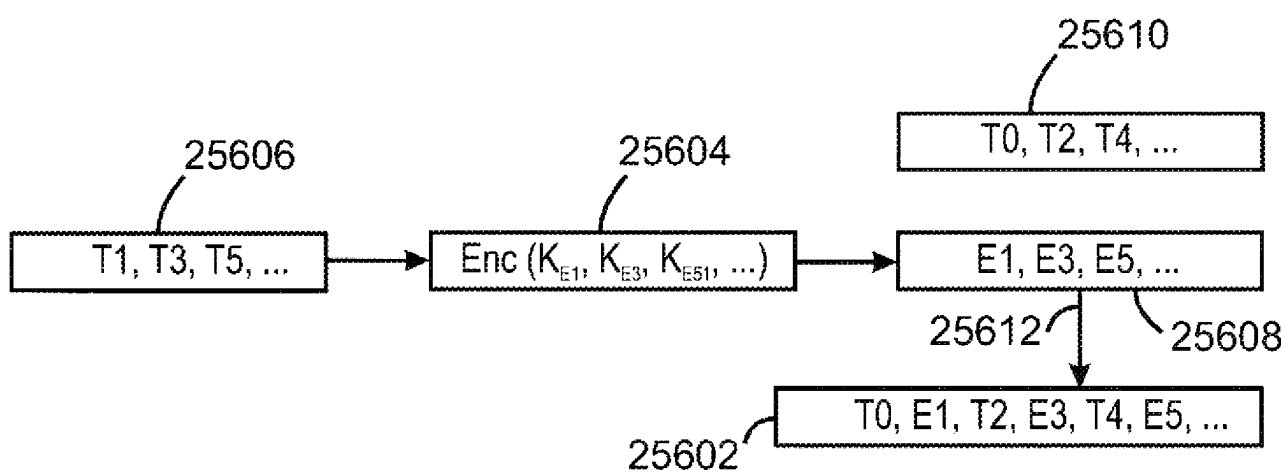


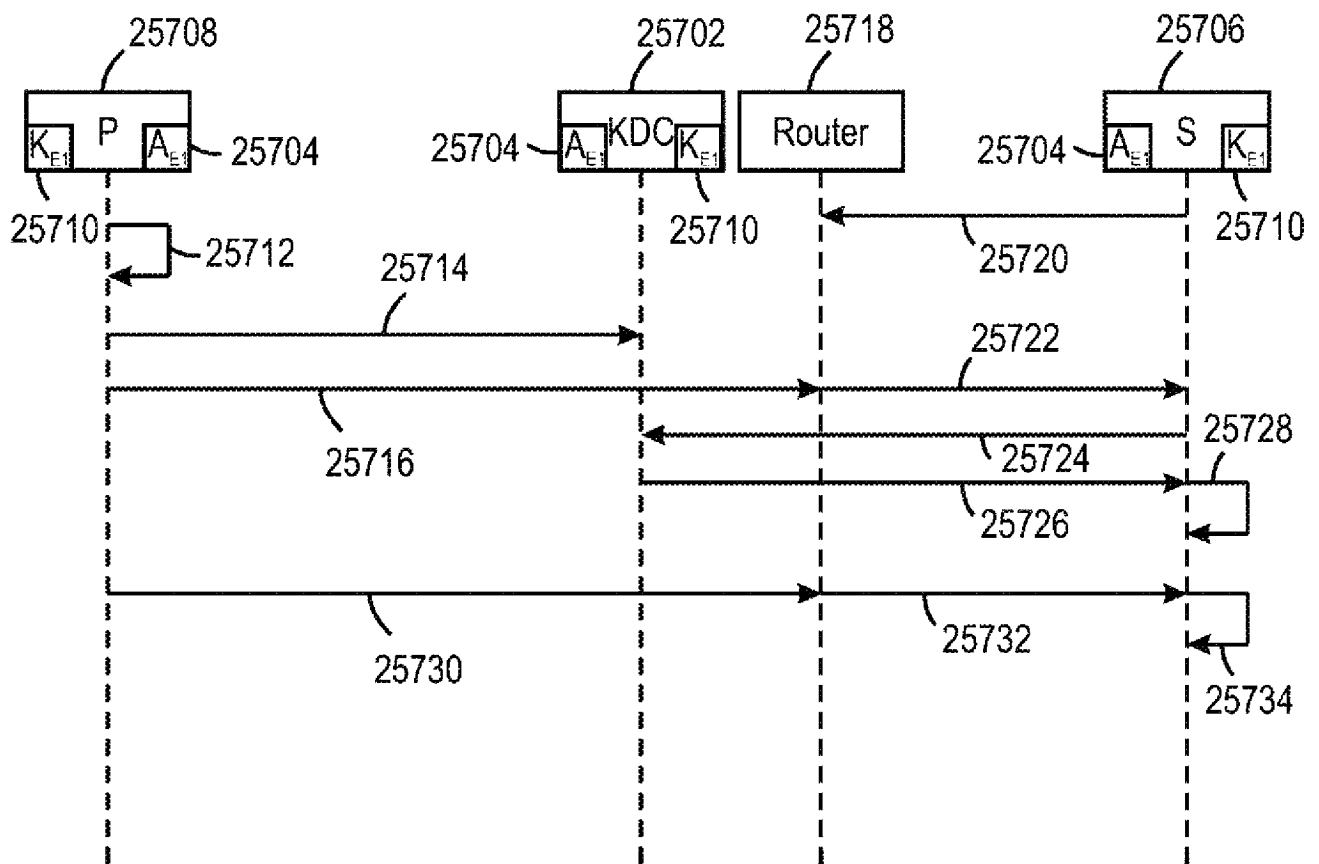
FIG. 254



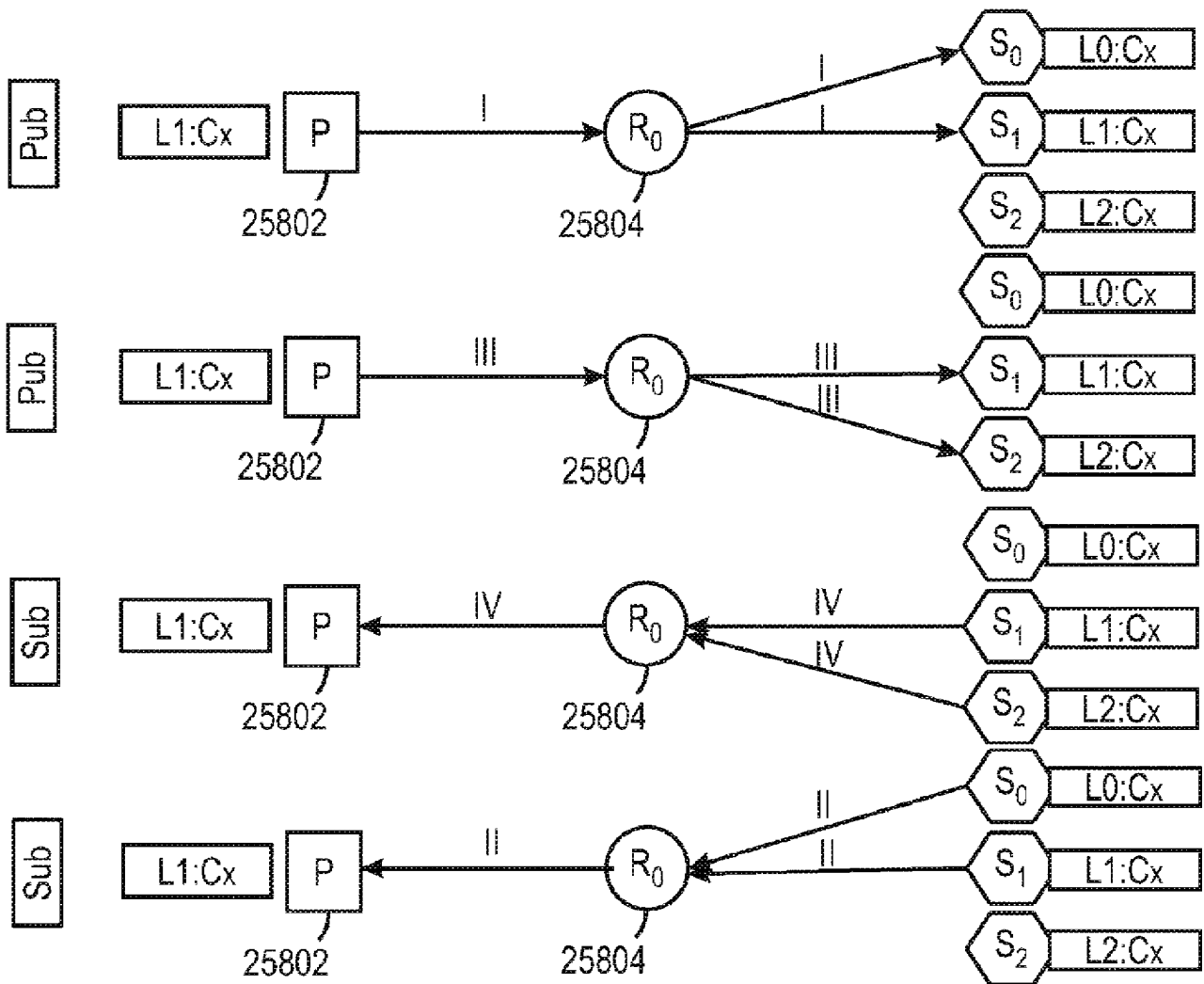
25500
FIG. 255



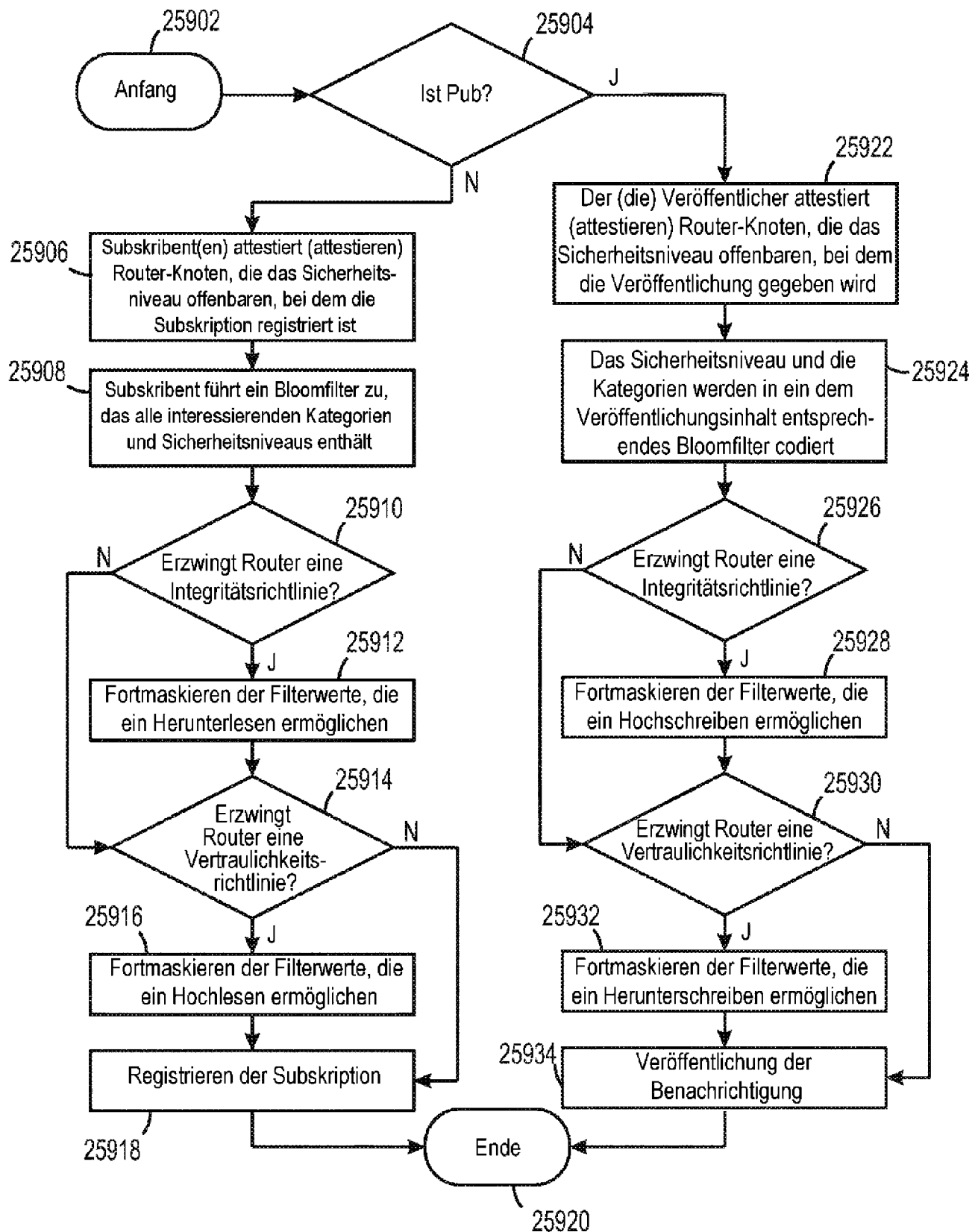
25600
FIG. 256



25700
FIG. 257



25800
FIG. 258



25900

FIG. 259

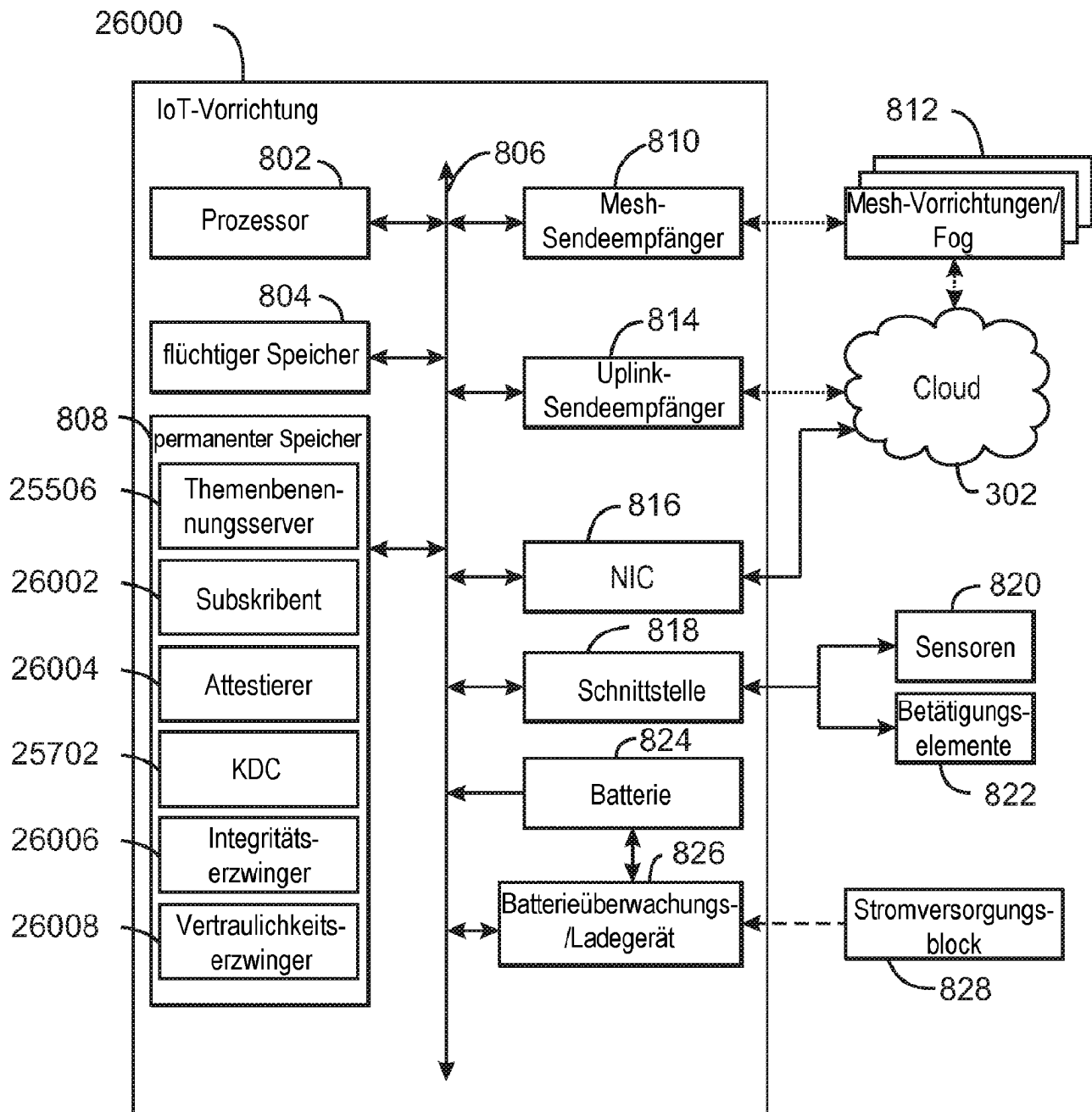


FIG. 260

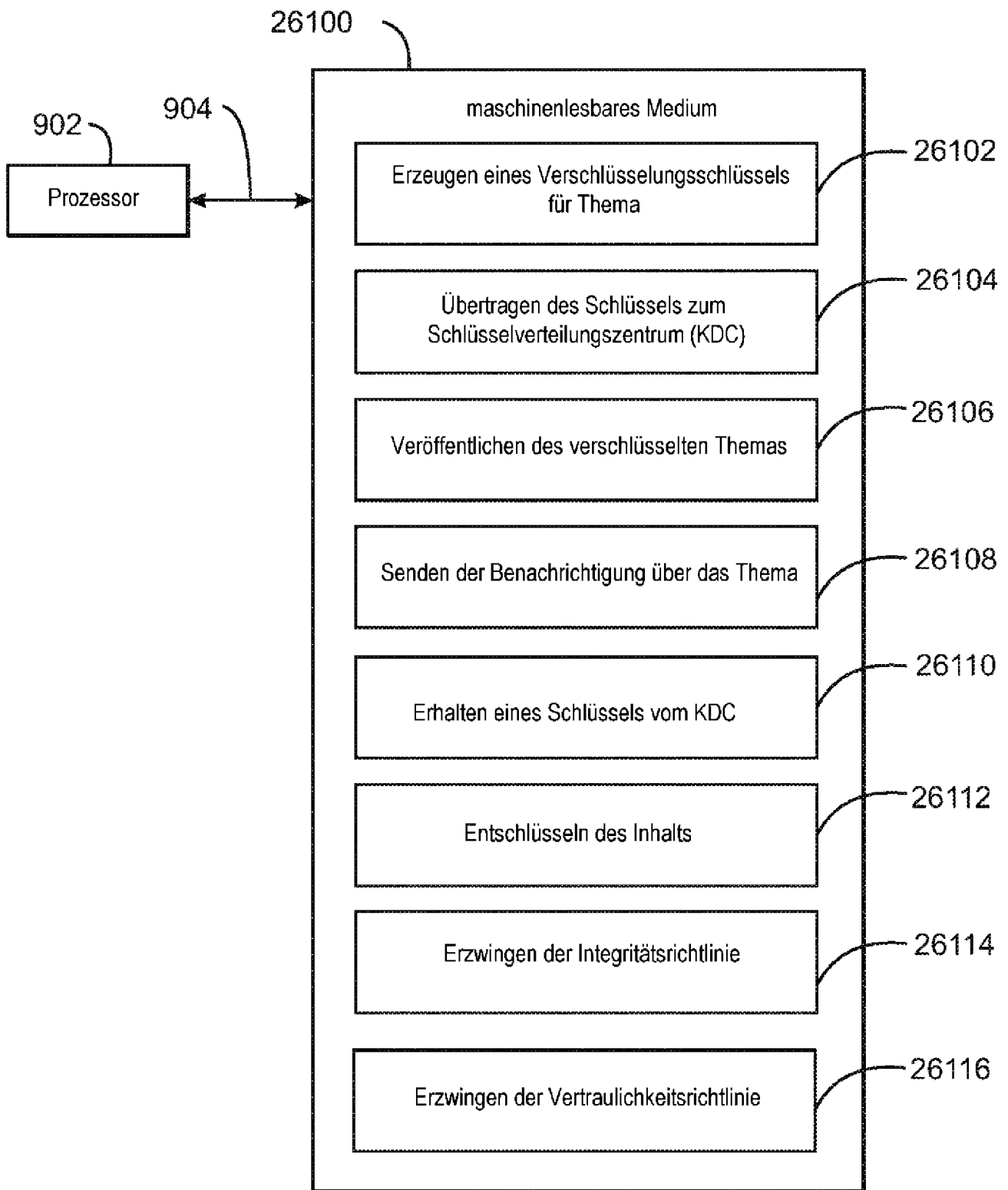
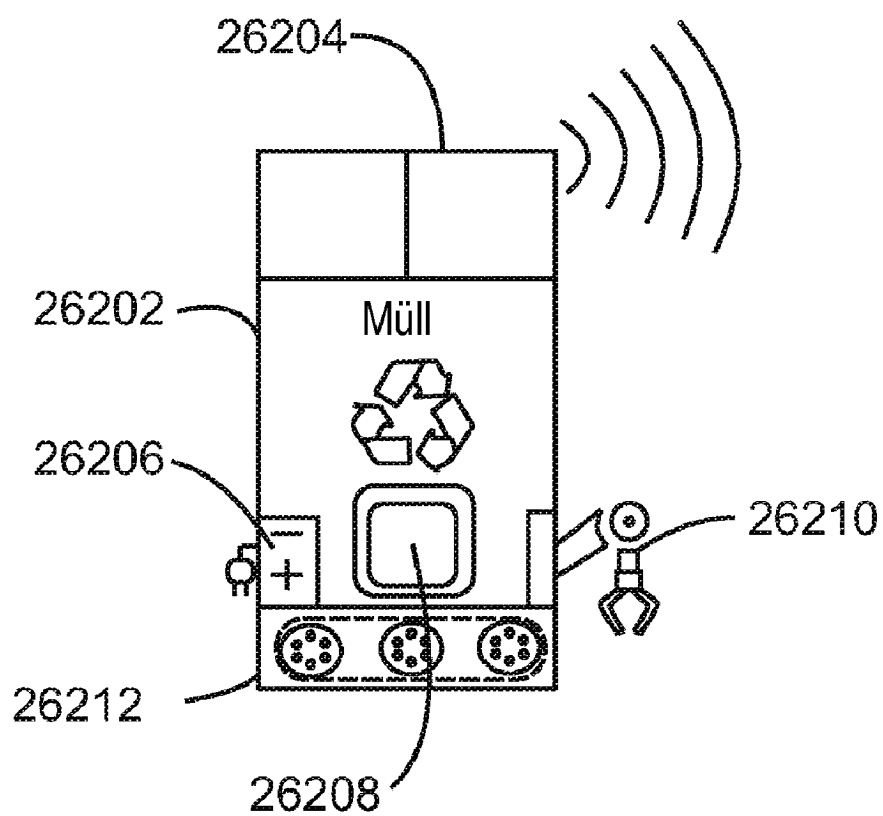
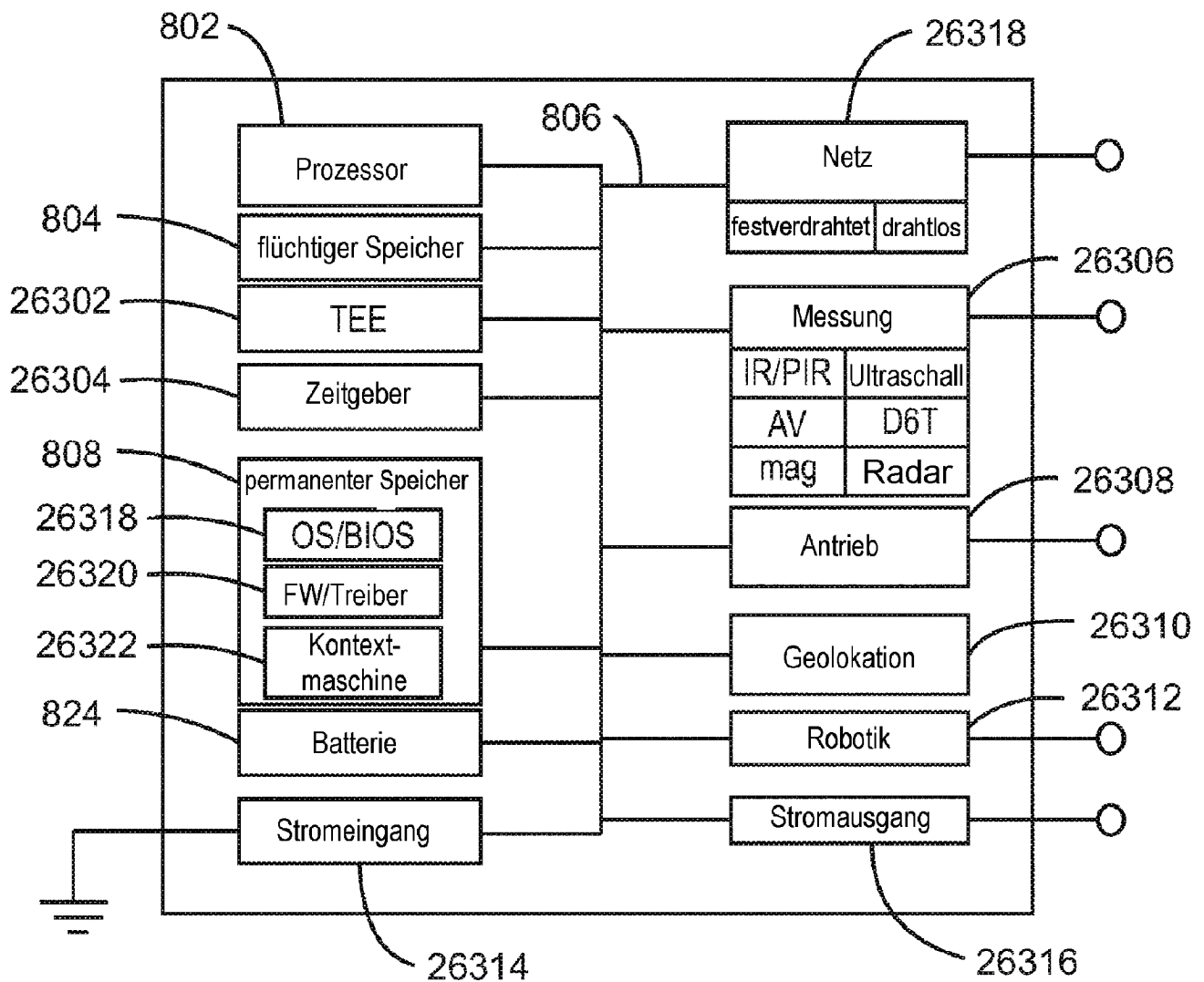


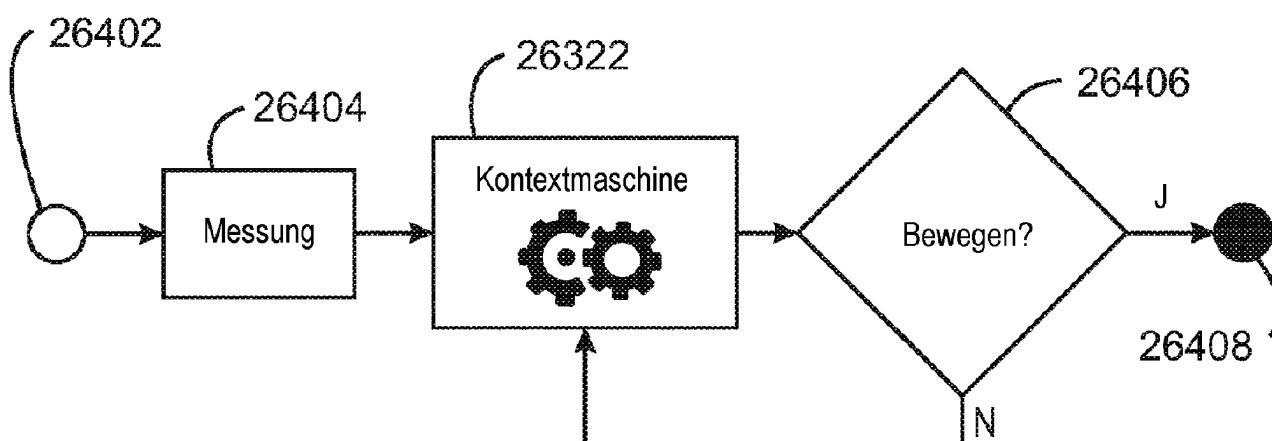
FIG. 261



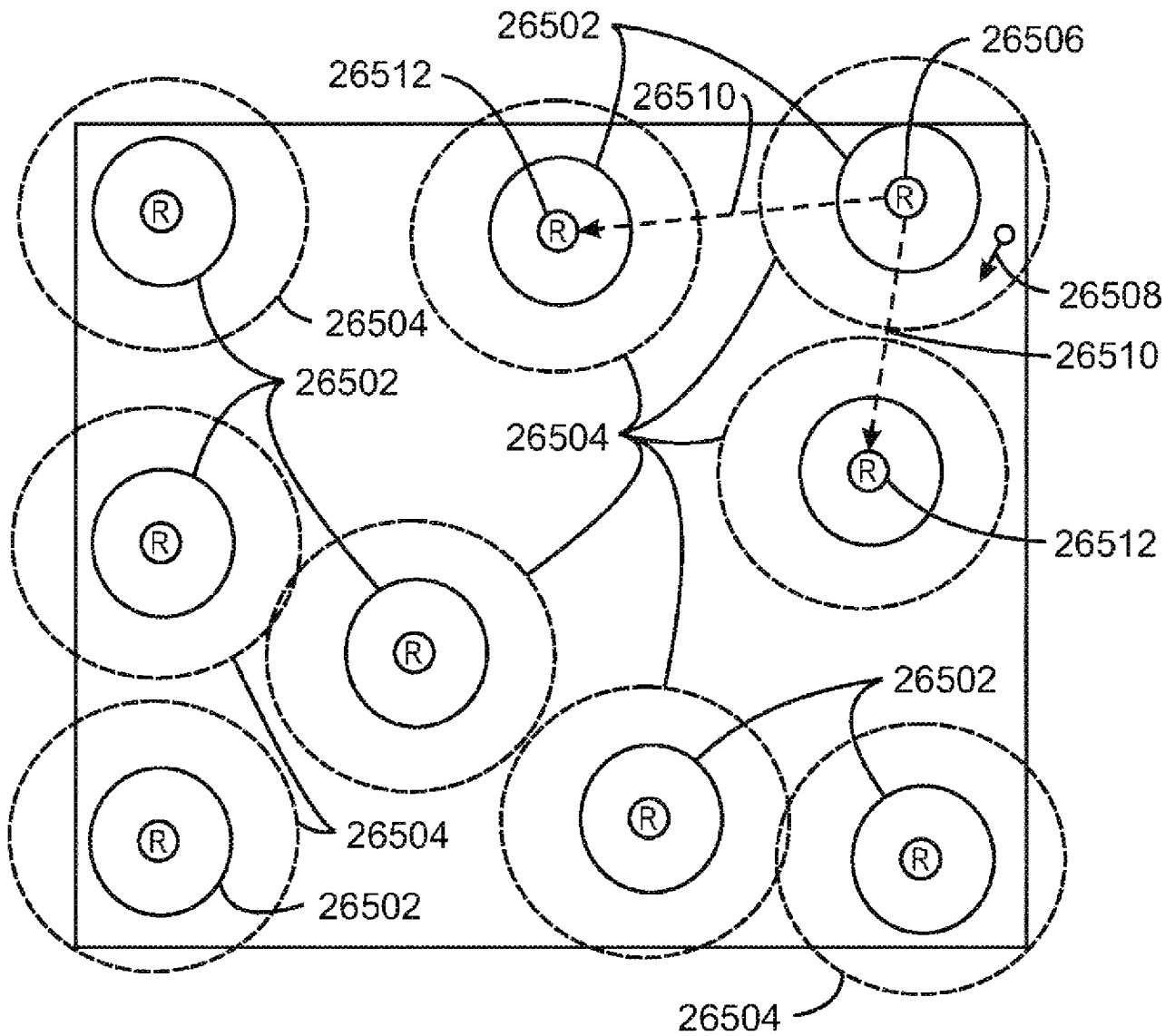
26200
FIG. 262



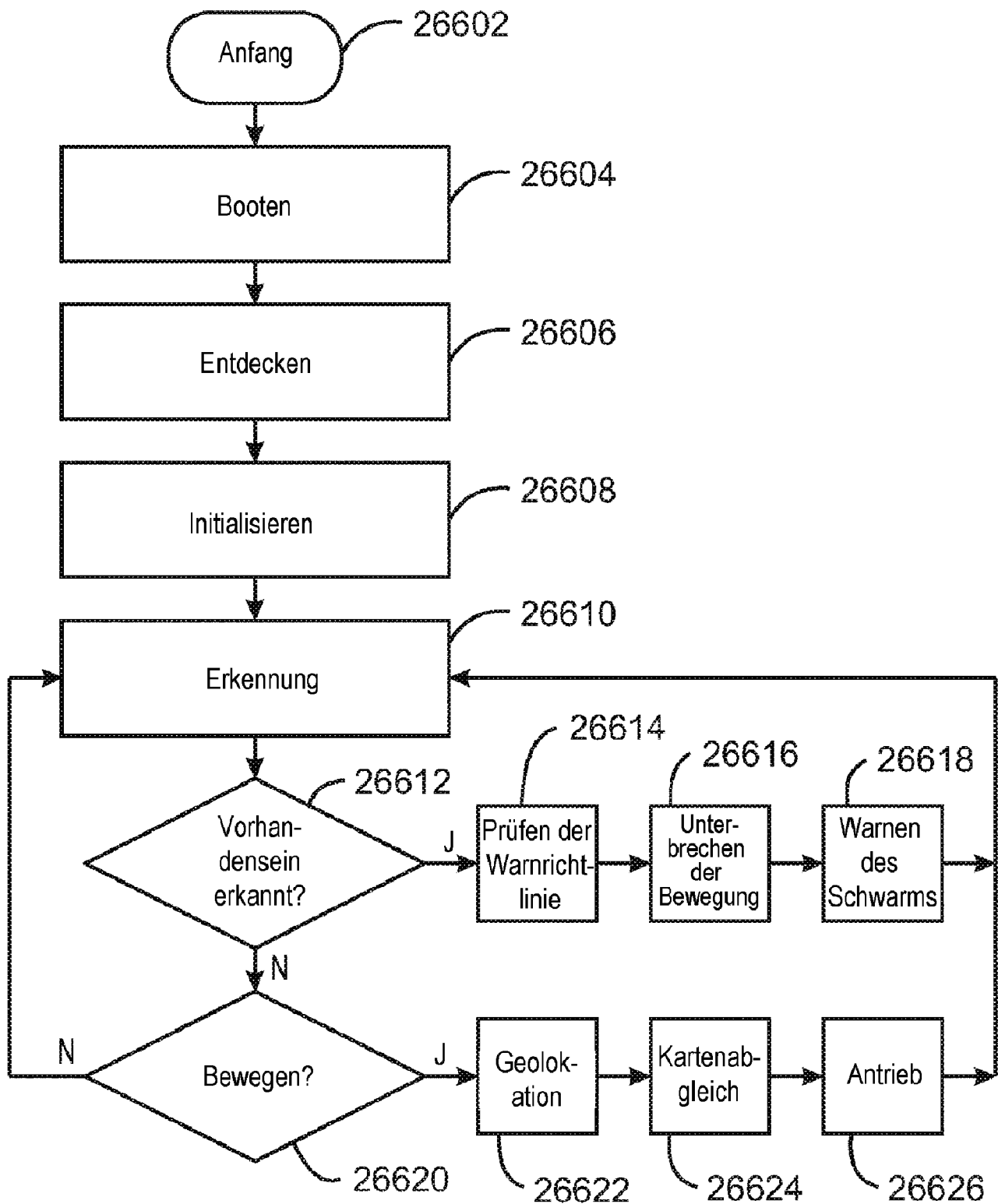
26300
FIG. 263



26400
FIG. 264



26500
FIG. 265



26600
FIG. 266

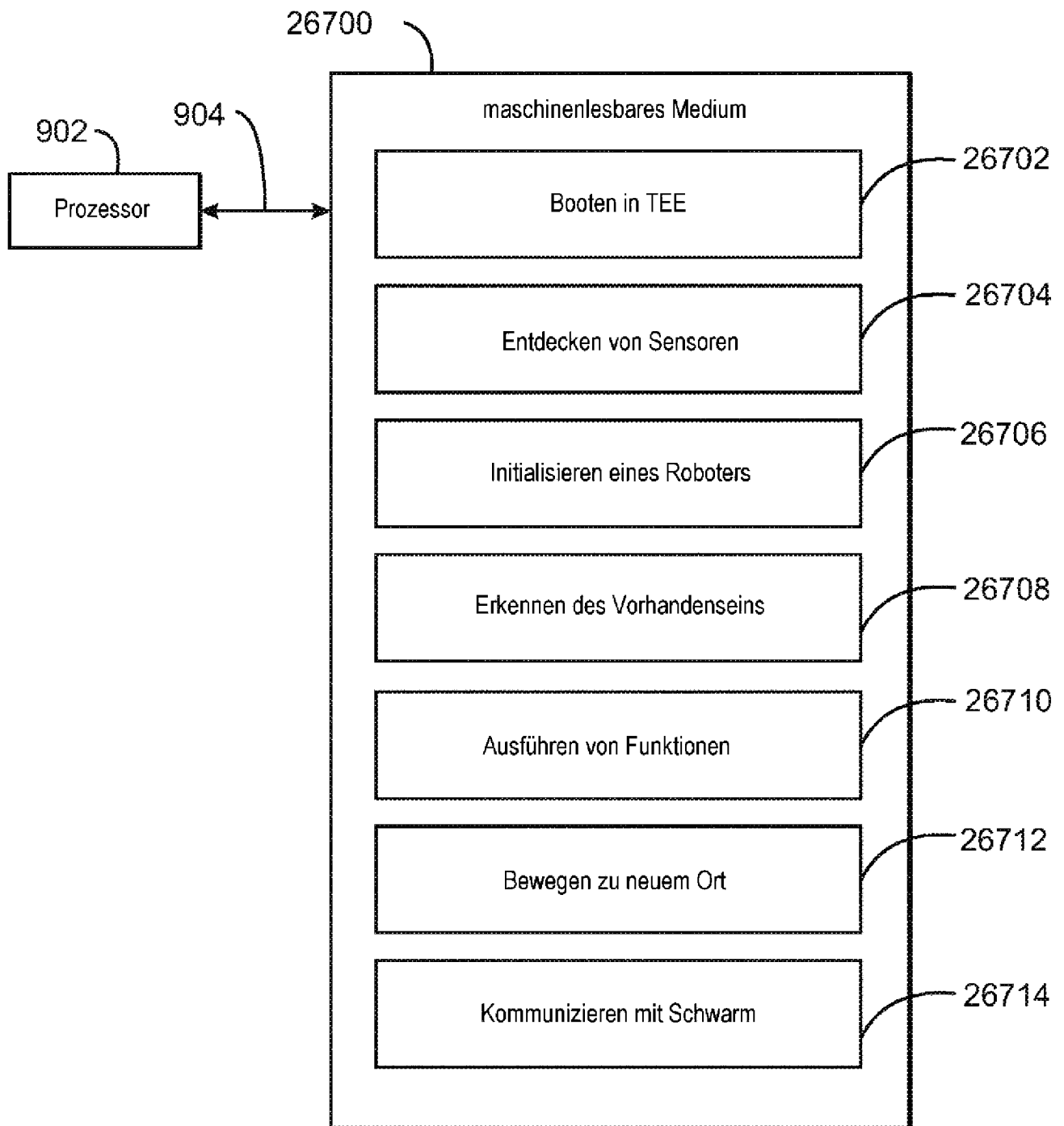
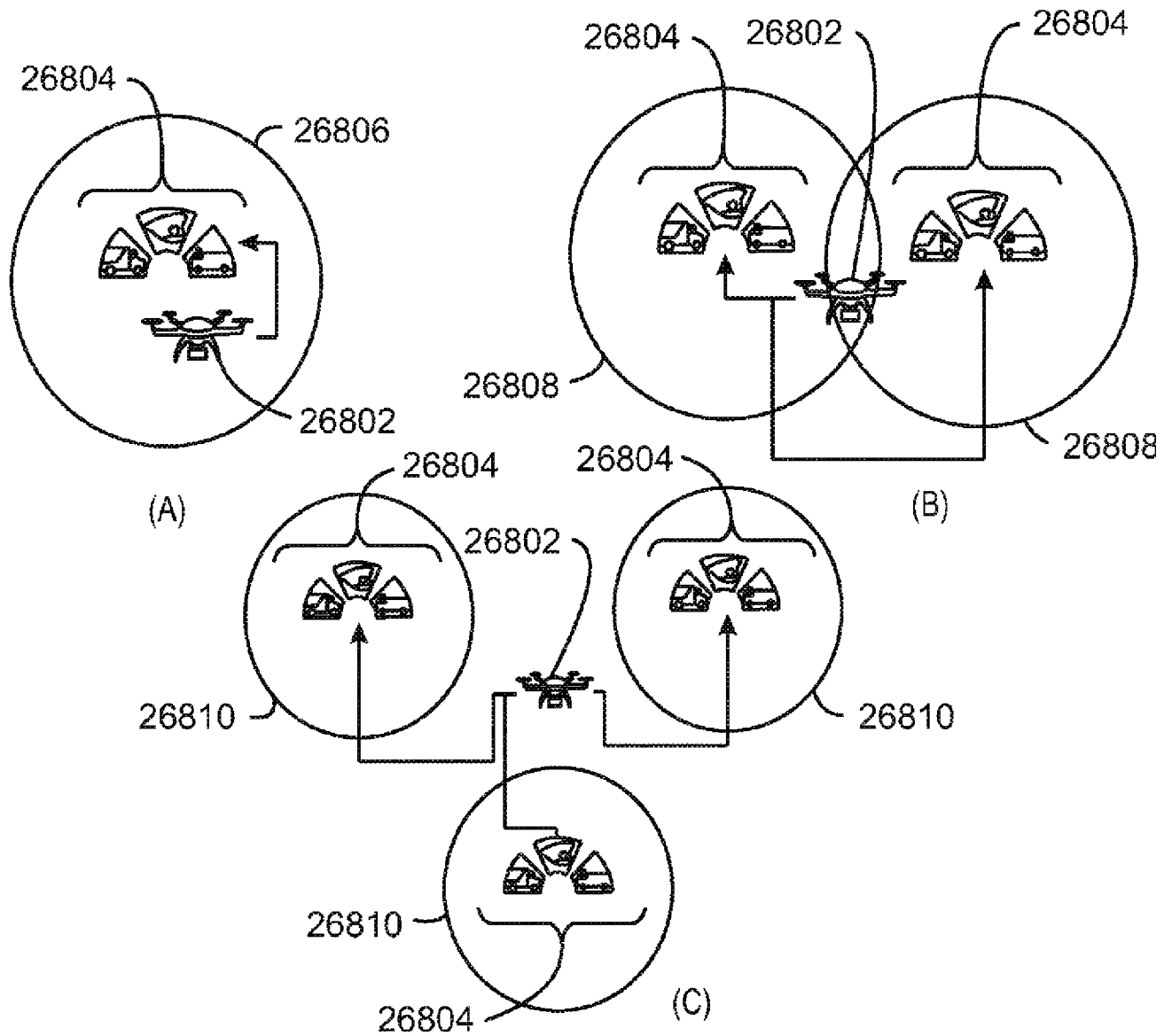
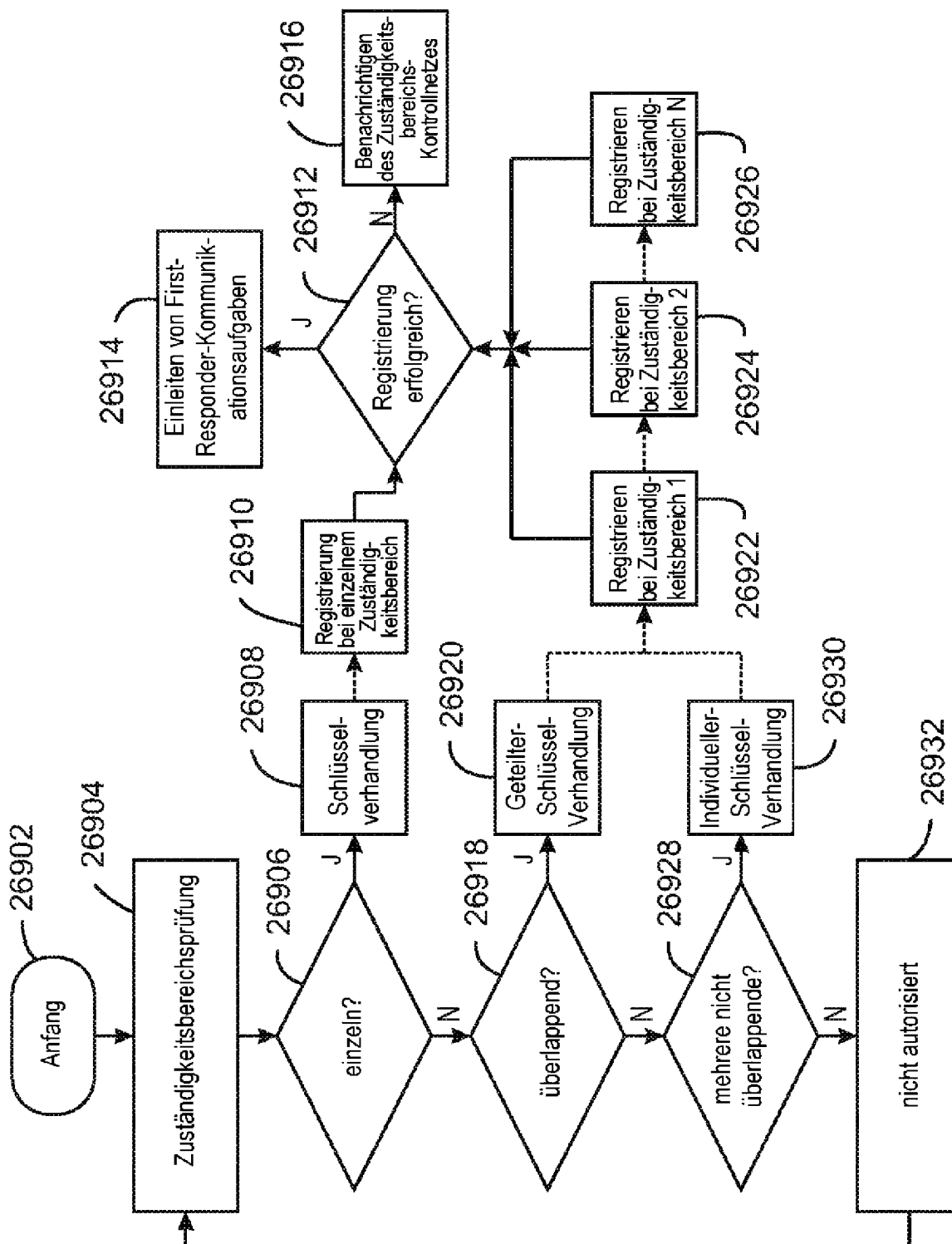


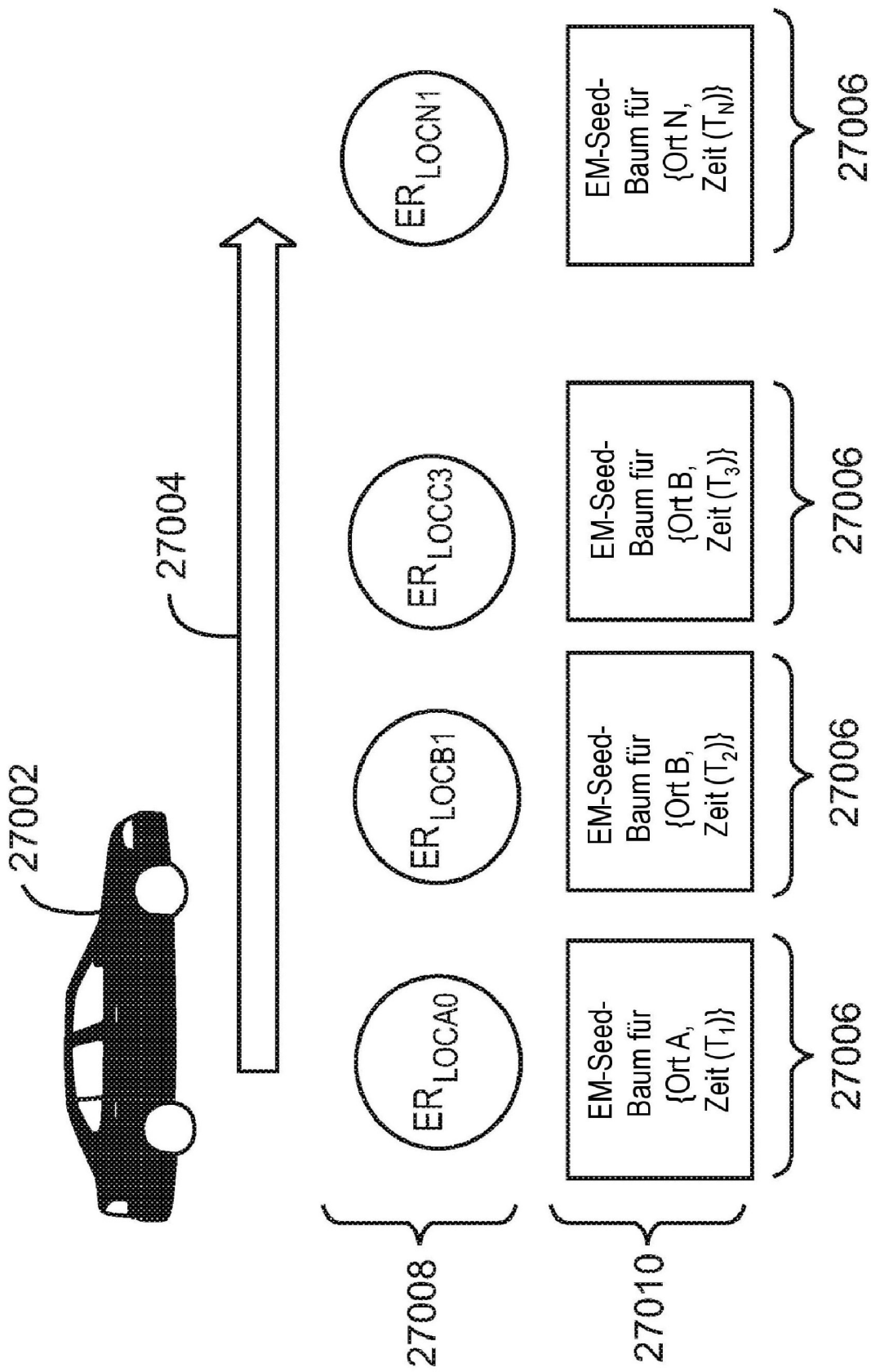
FIG. 267



26800
FIG. 268

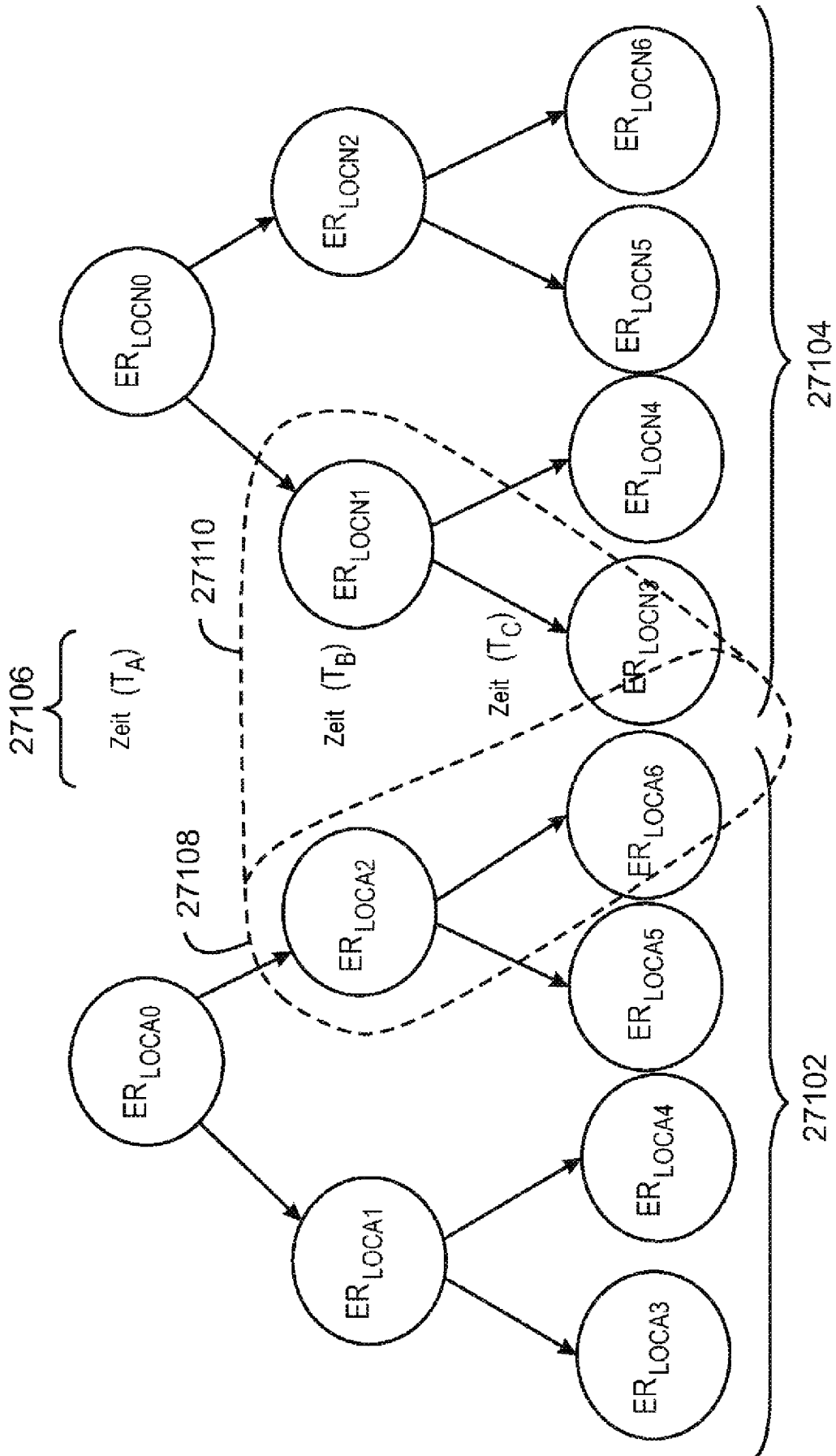


26900
FIG. 269

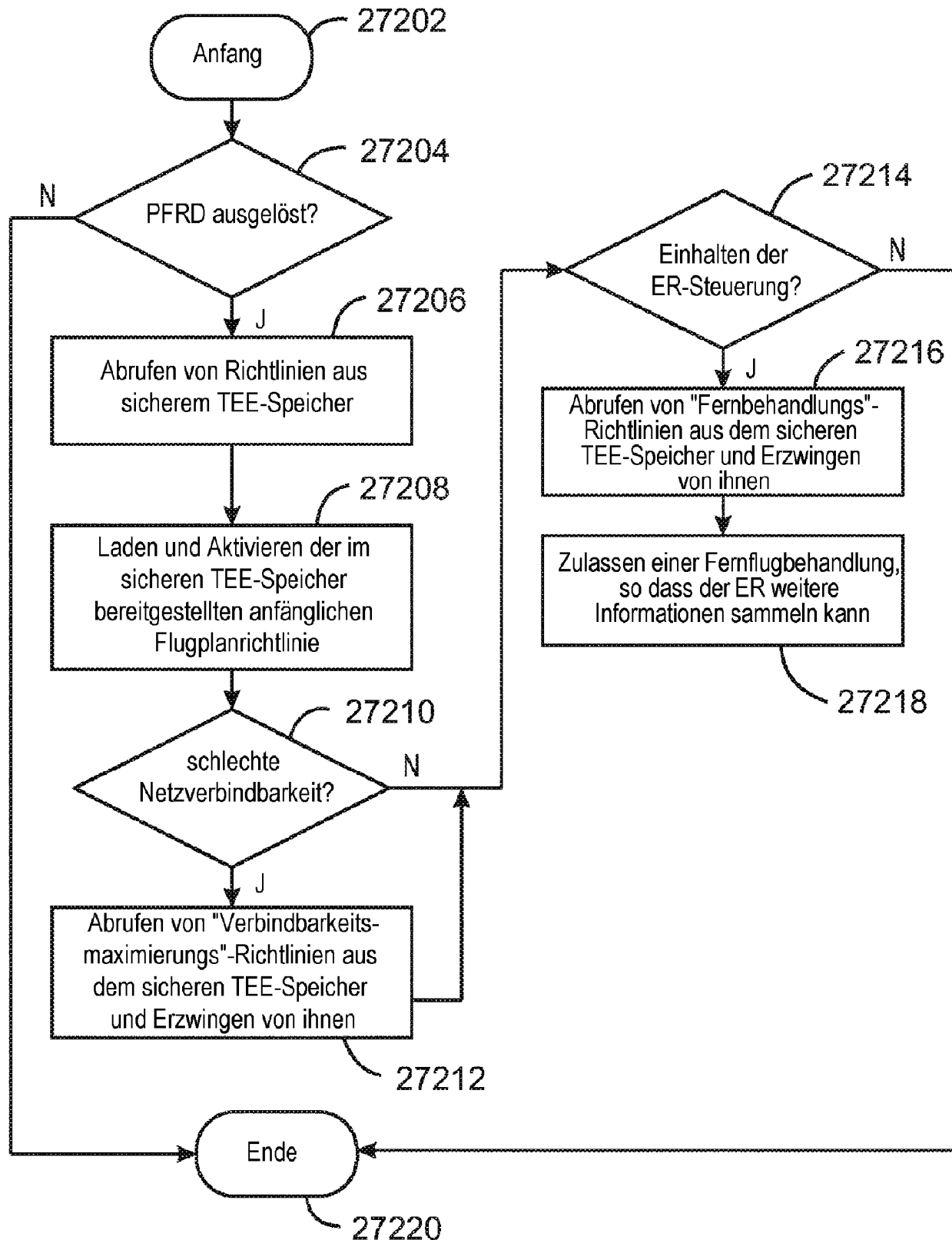


27000

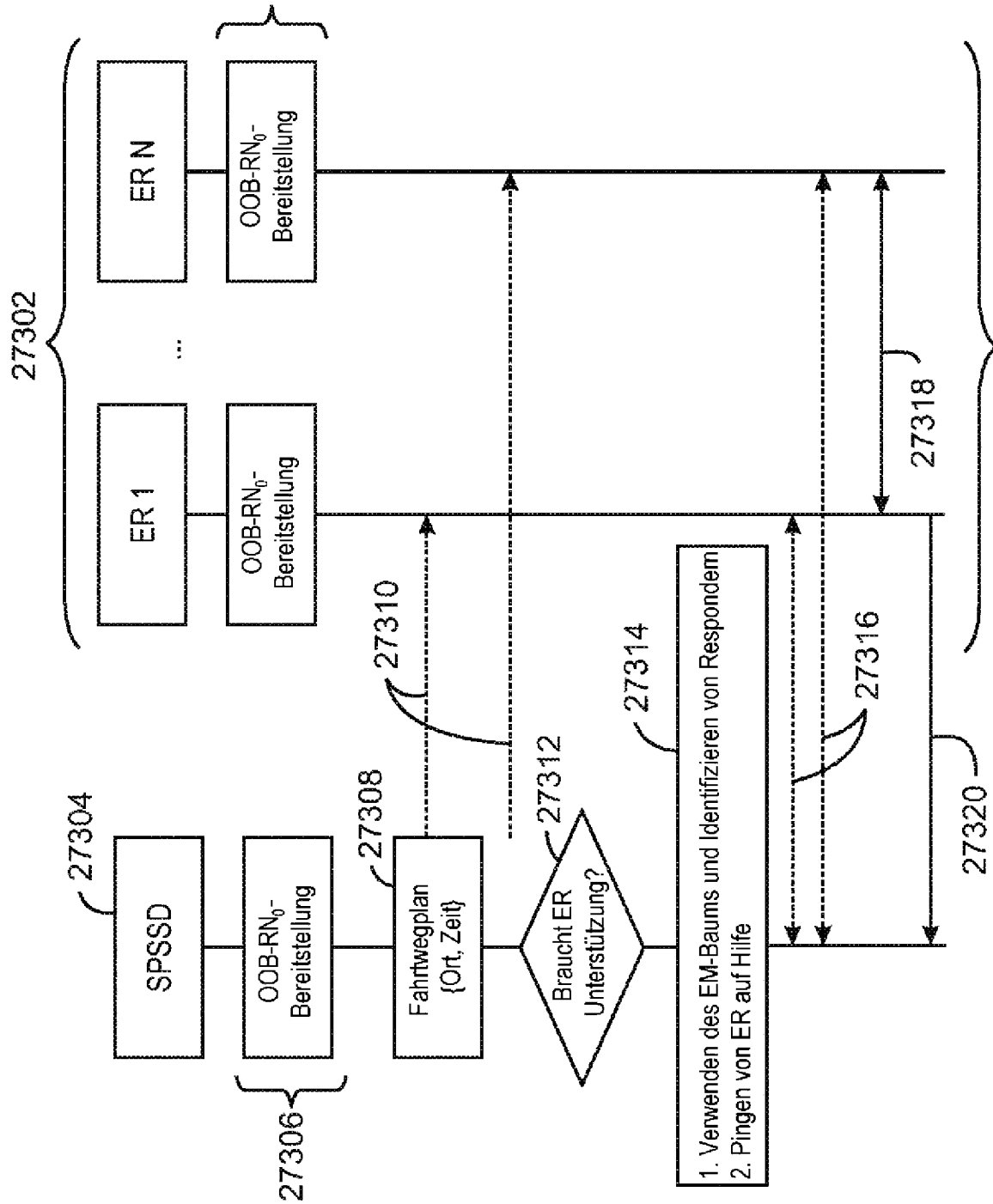
FIG. 270



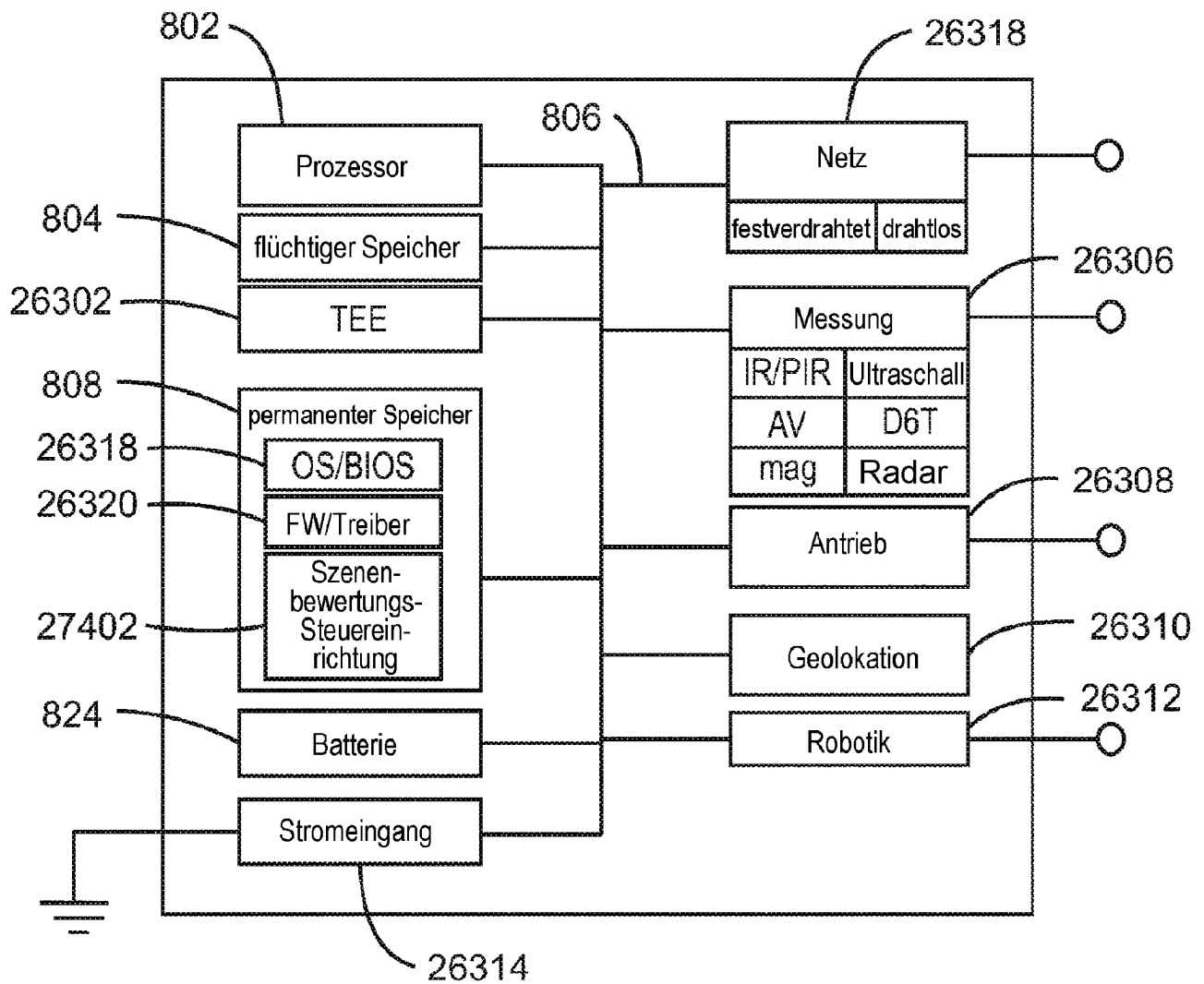
27100
FIG. 271



27200
FIG. 272



27300
FIG. 273



27400
FIG. 274

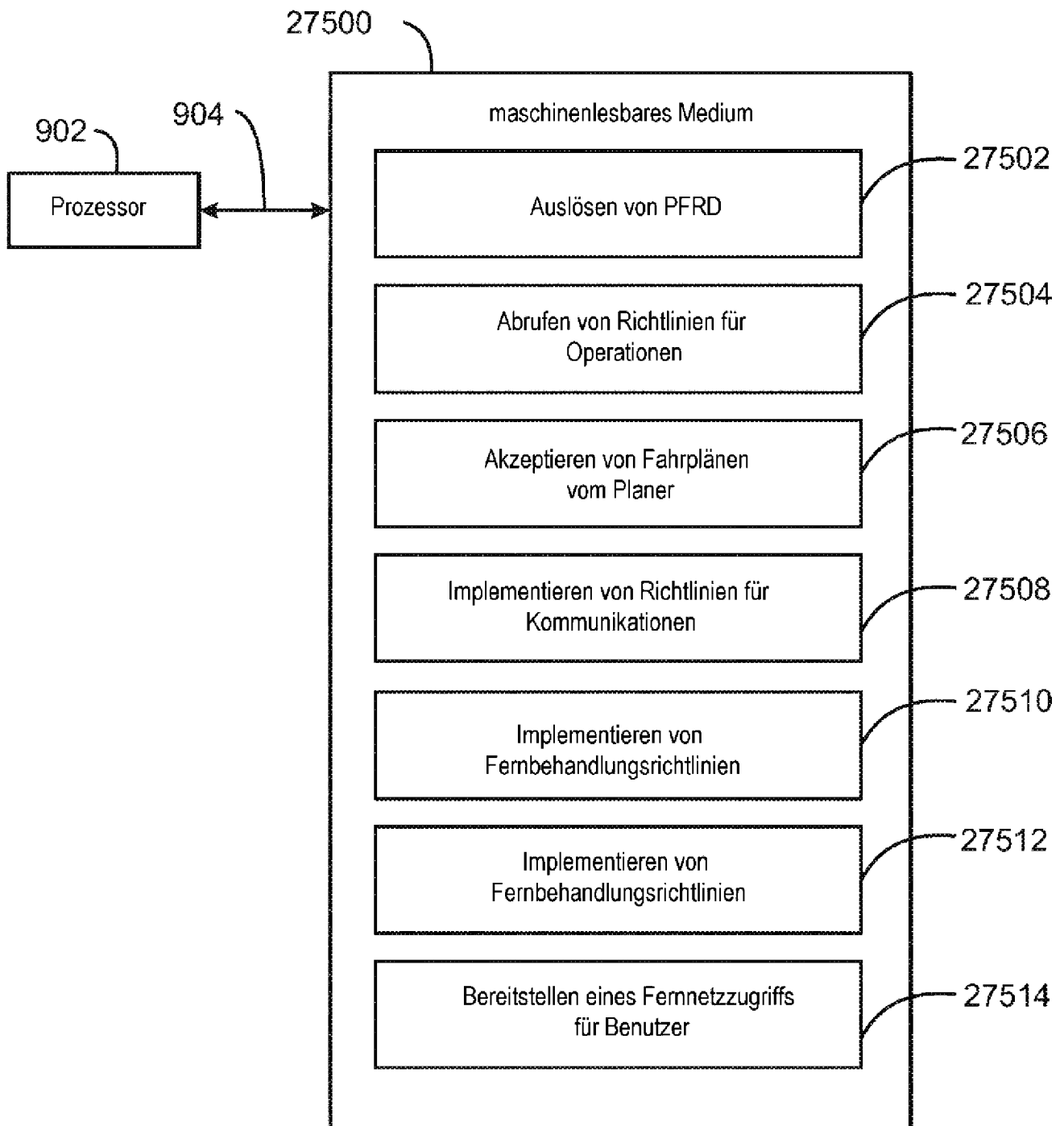


FIG. 275