(19) Deutsches Patent- und Markenamt

(10) **DE 11 2022 006 297 T5** 2024.12.05

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2023/126289**
in der deutschen Übersetzung (Art. III § 8 Abs. 2 IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2022 006 297.5**
(86) PCT-Aktenzeichen: **PCT/EP2022/087386**
(86) PCT-Anmeldetag: **22.12.2022**
(87) PCT-Veröffentlichungstag: **06.07.2023**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **05.12.2024**

(51) Int Cl.: ***G06K 19/06*** (2006.01)

(30) Unionspriorität:
22150037.4 **03.01.2022** **EP**

(71) Anmelder:
Koninklijke Philips N.V., Eindhoven, NL

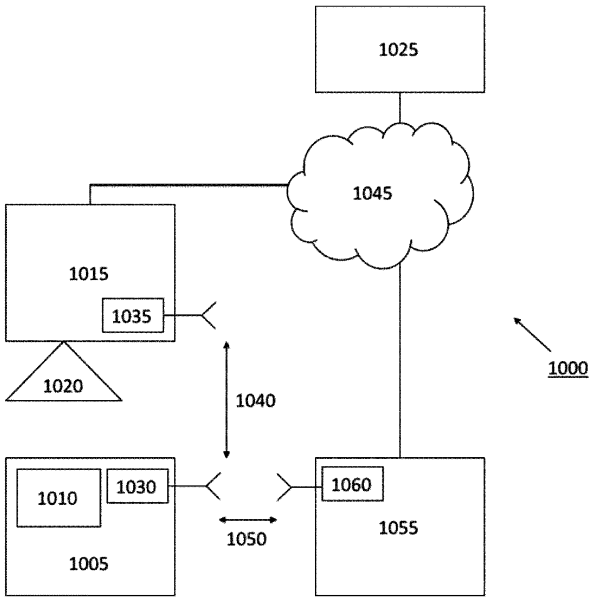
(74) Vertreter:
Meissner Bolte Patentanwälte Rechtsanwälte Partnerschaft mbB, 80538 München, DE

(72) Erfinder:
Garcia Morchon, Oscar, Eindhoven, NL

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **QR-CODE FÜR MEHRERE ANWENDUNGEN**

(57) Zusammenfassung: Die vorliegende Erfindung betrifft einen QR-Code für mehrere Anwendungen, der in der Lage ist, Informationen einer Vielzahl von Anwendungen so zu speichern, dass für jede Anwendung die jeweiligen Informationen in zuverlässiger und zeiteffizienter Weise abgerufen werden können. Die Informationen werden in verschlüsselte Daten kodiert, die mindestens einen Header und einen entsprechenden Datencontainer pro Anwendung einschließen. Der Header kann mindestens eine einzelne Kennung umfassen, die das Vorhandensein der Vielzahl von Anwendungen angibt, und eine jeweilige Anwendungskennung pro Anwendung. Die kodierten Daten werden gemäß einer Zuordnungsregel in einer Vielzahl von Datenpixeln verteilt gespeichert, die über einen Kodierbereich des QR-Codes verteilt sind. Ein QR-Code-Lesegerät kann die im QR-Code gespeicherten Informationen abrufen, indem es ausschließlich auf die der jeweiligen Anwendung zugeordneten Datenpixel zugreift und diese verarbeitet und eine für diese jeweilige Anwendung spezifische Fehlerkorrektur anwendet.



Beschreibung

GEBIET DER ERFINDUNG

[0001] Die Erfindung betrifft den Bereich der QR-Codes und insbesondere einen QR-Code für mehrere Anwendungen.

HINTERGRUND DER ERFINDUNG

[0002] Ein Quick Response-Code (QR-Code), siehe [ISO 18004], auch als zweidimensionaler Barcode bezeichnet, ist ein maschinenlesbares optisches Etikett, auf dem Informationen kodiert sind, die dann von einem QR-Code-Lesegerät oder einer Scan-Vorrichtung dekodiert werden können. Beispielsweise kann eine Handvorrichtung wie ein Smartphone mit einem QR-Code-Lesegerät und einer optischen Kamera ausgestattet sein, die ein Bild des QR-Codes aufnehmen und dekodieren kann, um die kodierten Informationen zu identifizieren. Ein QR-Code kann für viele Anwendungen verwendet werden. Beispielsweise kann ein QR-Code auf ein Plakat gedruckt sein, und wenn ein Benutzer den QR-Code scannt, öffnet beispielsweise ein Browser auf dem QR-Code-Lesegerät eine Website mit weiteren Informationen zum Plakatthema. In einem anderen Beispiel kann ein Unternehmen einen QR-Code auf sein Produkt drucken, wobei der QR-Code beispielsweise eine Seriennummer des Produkts enthält, die dann von einer Anwendung dieses Unternehmens verwendet werden kann, um das Produkt zu registrieren oder unter Verwendung der Seriennummer mit dem Produkt zu kommunizieren.

[0003] Es ist üblich, Vorrichtungen mit drahtlosen Netzwerken zu verbinden, wie unter Verwendung von Wi-Fi (d. h. basierend auf IEEE 802.11). Wenn die zu verbindenden Vorrichtungen keine oder nur eine sehr einfache Benutzeroberfläche aufweisen, wird häufig eine andere Vorrichtung zur Unterstützung beim Einrichten der Verbindung verwendet. Ein Teil des Einrichtungs- oder Konfigurationsprozesses beinhaltet Bootstrapping. Ein Beispiel hierfür ist Wi-Fi Easy Connect, das ein von der Wi-Fi Alliance entwickeltes Device Provisioning Protocol-Schema (DPP-Schema) unterstützt (siehe [DPP]) und QR-Codes zum Herstellen sicherer Wi-Fi-Verbindungen zwischen Wi-Fi-Vorrichtungen verwendet. Solche Einrichtungsprotokolle werden üblicherweise als „Inbetriebnahme“ bezeichnet.

[0004] Ein Produkt kann mehrere QR-Codes für mehrere unterschiedliche Anwendungen enthalten. Beispielsweise kann ein drahtloses Produkt, z. B. ein Wi-Fi-Produkt, auch ein Bluetooth Low Energy-Produkt (BLE-Produkt) oder ein IEEE 802.15.4-basiertes Produkt usw. sein. Ein solches Produkt benötigt möglicherweise einen ersten QR-Code, um mittels eines ersten Inbetriebnahmeprotokolls, z. B. Wi-Fi Easy Connect, eine Verbindung mit dem Wi-Fi-Netzwerk herzustellen, und einen zweiten QR-Code, der sich vom ersten QR-Code unterscheidet, um sich mittels eines zweiten Inbetriebnahmeprotokolls in einem anderen Netzwerk wie einem Smart-Home-Netzwerk (SHS) wie Matter over Wi-Fi Frames anzumelden. Allerdings kann es für einen Benutzer problematisch sein, den richtigen QR-Code für eine besondere Anwendung, z. B. ein Inbetriebnahmeprotokoll, zu scannen, da der Benutzer herausfinden muss, welcher QR-Code für welche Anwendung zu verwenden ist. Darüber hinaus implementiert das Produkt bei einem Wireless-Produkt, das sowohl Wi-Fi Easy Connect als auch SHS unterstützt, auch die Inbetriebnahmeprotokolle von Wi-Fi Easy Connect und SHS. Da beide Inbetriebnahmeprotokolle auf Wi-Fi-Frames basieren oder über diese ausgetauscht werden, muss das Produkt basierend auf der ersten, durch ein Inbetriebnahmetool ausgelösten Interaktion entscheiden, welches Inbetriebnahmeprotokoll verwendet werden soll. Dieses Verhalten ist jedoch nicht definiert. Ein ähnliches Problem tritt bei anderen drahtlosen Produkten auf, die zwei oder mehr Inbetriebnahmeprotokolle über dieselbe drahtlose Schnittstelle implementieren.

[0005] Darüber hinaus können unter solchen Umständen die in mehreren QR-Codes enthaltenen Informationen von einer der Anwendungen benötigt werden, was zur Folge hat, dass das Scannen eines einzigen QR-Codes nicht ausreicht. Tatsächlich kommt es immer häufiger vor, dass mehrere QR-Codes gescannt werden müssen, von denen jeder zusammengehörende Informationen enthält. Das Scannen jedes einzelnen QR-Codes ist ineffizient und kostet Zeit, während ein einzelner Scanvorgang viel schneller wäre. Ein solcher Fall ist beispielsweise die Dokumentenkontrolle bei Gruppenreisenden. Das Scannen eines QR-Codes für jede Person kann zu einer deutlichen Verlängerung der Warteschlangen und zu Frustration bei allen Beteiligten führen.

KURZDARSTELLUNG DER ERFINDUNG

[0006] Ziel der vorliegenden Erfindung ist es, Informationen mehrerer Anwendungen in einem QR-Code so zu speichern, dass die jeweiligen Informationen jeder Anwendung anschließend einzeln, zuverlässig und zeiteffizient abgerufen werden können. Ein weiteres Ziel der vorliegenden Erfindung besteht darin, den Betrieb von Inbetriebnahmeprotokollen, die über dieselbe Kommunikationsschnittstelle laufen, basierend auf den abgerufenen Informationen aus einem QR-Code zu koordinieren.

[0007] Dieses Ziel wird durch ein Kodierverfahren nach einem der Ansprüche 1 bis 9, durch ein Dekodierverfahren nach einem der Ansprüche 10 bis 12, durch eine Kodiereinrichtung nach Anspruch 13, durch eine Dekodiereinrichtung nach Anspruch 14, durch eine Vorrichtung nach Anspruch 15, durch ein System nach Anspruch 16 und durch ein jeweiliges Computerprogrammprodukt nach Ansprüchen 17 und 18 erreicht.

[0008] Gemäß einem ersten Gesichtspunkt, der auf das Kodieren von Informationen einer Vielzahl von Anwendungen zur Speicherung in einem ersten QR-Code gerichtet ist, umfasst ein Verfahren mindestens das Sammeln der Informationen der Vielzahl von Anwendungen und das Kodieren der Informationen in kodierte Daten, wobei die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen. Der Header umfasst mindestens eine einzelne Kennung, die das Vorhandensein der Vielzahl von Anwendungen anzeigt, und eine jeweilige Anwendungskennung pro Anwendung. Somit ermöglichen die im Header angegebenen Identifikationsinformationen für die jeweiligen Anwendungen und Datencontainer die Unterstützung mehrerer Anwendungen auf demselben (ersten) QR-Code. In der nachfolgenden Offenbarung wird der erste QR-Code, also ein QR-Code, der in der Lage ist, mehrere Anwendungen zu handhaben, synonym auch als Mastercode bezeichnet.

[0009] In einem Beispiel des ersten Gesichtspunkts wird für jeden Datencontainer pro Anwendung in den codierten Daten des jeweiligen Datencontainers pro Anwendung ein jeweiliger Fehlerkorrekturmechanismus (ECM) und/oder Erkennungsmechanismus (EDM) verwendet. Dabei ermöglicht die Bereitstellung eines ECM und/oder EDM pro Datencontainer ein schnelles Auslesen der Daten einer jeweiligen Anwendung im ersten QR-Code auf zuverlässige Weise.

[0010] In einem Beispiel des ersten Gesichtspunkts umfasst der Schritt des Kodierens der Informationen in kodierte Daten mindestens das Umwandeln eines Eingabedatenstroms der Informationen in einen jeweiligen Eingabebitstring für jede Anwendung, das Aufteilen des jeweiligen Eingabebitstrings für jede Anwendung in eine Folge von einem oder mehreren Datencodewörtern, das Unterteilen der Folge von einem oder mehreren Datencodewörtern in eine vordefinierte Anzahl von Blöcken, das Erzeugen von einem oder mehreren Fehlerkorrekturcodewörtern für jeden Block und das Einschließen der Fehlerkorrekturcodewörter in die Datencodewortfolge. Beispielsweise können die Fehlerkorrekturcodewörter an einer oder mehreren geeigneten Stellen in der Datencodewortfolge eingefügt werden. Dabei stehen je Anwendung Möglichkeiten zur Fehlerkorrektur zur Verfügung. In einem Beispiel können die Daten- und Fehlerkorrekturcodewörter eine Länge von 8 Bit oder andernfalls eine andere Bitlänge aufweisen.

[0011] In einem Beispiel des ersten Gesichtspunkts umfasst der Schritt des Kodierens der Informationen in kodierte Daten mindestens das Entnehmen eines jeweiligen Datenstroms aus dem Header für jede Anwendung, das Kombinieren des jeweiligen Datenstroms mit einem kombinierten jeweiligen Datenstrom, das Umwandeln des kombinierten jeweiligen Datenstroms in einen jeweiligen Eingabebitstring, das Aufteilen des jeweiligen Eingabebitstrings in eine Folge von einem oder mehreren Datencodewörtern, das Unterteilen der Folge von einem oder mehreren Datencodewörtern in eine vordefinierte Anzahl von Blöcken, das Erzeugen von einem oder mehreren Fehlerkorrekturcodewörtern für jeden Block und das Einschließen der Fehlerkorrekturcodewörter in die Datencodewortfolge. Beispielsweise können die Fehlerkorrekturcodewörter an einer oder mehreren geeigneten Stellen in der Datencodewortfolge eingefügt werden. Das Kombinieren der jeweiligen Datenströme kann das Verketten der jeweiligen Datenströme einschließen.

[0012] In einem Beispiel des ersten Gesichtspunkts kann das Verfahren ferner das verteilte Speichern eines Teils oder einer Gesamtheit der kodierten Daten in einer Vielzahl von Datenpixeln umfassen, die über einen Kodierbereich des ersten QR-Codes verteilt sind, gemäß einer Zuordnungsregel, wobei ein Datenpixel als Datenspeicherungseinheit im QR-Code definiert ist. Dadurch können lokale Beschädigungen des ersten QR-Codes auf die Vielzahl von Anwendungen verteilt werden, was weniger Schäden pro Anwendung und eine höhere Wahrscheinlichkeit einer vollständigen Korrektur pro Anwendung gewährleistet. Zudem ermöglicht dies einem QR-Code-Lesegerät, nur die Anwendungsdaten einer jeweiligen Anwendung zu lesen, was den Lesevorgang beschleunigt.

[0013] In einem Beispiel des ersten Gesichtspunkts umfasst das Verfahren ferner, wenn es das verteilte Speichern eines Teils der kodierten Daten umfasst, das Speichern eines anderen Teils der kodierten Daten in einem zweiten QR-Code, der sich von dem ersten QR-Code unterscheidet. Dabei kann die Speicherung der kodierten Daten für die Vielzahl von Anwendungen mit anderen QR-Codes geteilt werden, um die Datenspeicherungsressourcen zu optimieren.

[0014] In einem Beispiel des ersten Gesichtspunkts sind in jedem Datencontainer pro Anwendung jeweilige Fehlererkennungs- oder -korrekturdaten aus einem pro Anwendung verwendeten Fehlererkennungs- oder -korrekturmechanismus eingeschlossen. Dadurch können Fehler in den kodierten Daten des jeweiligen Datencontainers erkannt oder korrigiert werden.

[0015] In einem Beispiel des ersten Gesichtspunkts schließen die kodierten Daten ein: Daten, die für die Verwendung mit einem von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzten Gesamtfehlererkennungsmechanismus angeordnet sind, um basierend auf Informationen (z. B. Codewörtern) in anderen Datencontainern zu erkennen, ob der jeweilige Datencontainer Fehler enthält, und/oder Daten, die für die Verwendung mit einem von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzten Gesamtfehlerkorrekturmechanismus angeordnet sind, um basierend auf Informationen (z. B. Codewörtern) in anderen Datencontainern Fehler im jeweiligen Datencontainer zu korrigieren. Dadurch können die Fehlererkennungs- und -korrekturfunktionen verbessert werden.

[0016] In einem Beispiel des ersten Gesichtspunkts umfasst der Header ferner einen jeweiligen Anwendungslängenparameter für jede Anwendung. Dabei können Informationen darüber, wie die Daten in Bits im ersten QR-Code für jede Anwendung gespeichert werden, angegeben werden, um eine Verschwendung von Datenressourcen zu vermeiden oder mindestens zu verringern, beispielsweise in dem Fall, in dem davon ausgegangen wird, dass die Speicherungskapazität des ersten QR-Codes gleichmäßig auf die mehreren Anwendungen aufgeteilt wird, obwohl manche Anwendungen mehr als die Hälfte der Speicherungskapazität des ersten QR-Codes benötigen.

[0017] In einem Beispiel des ersten Gesichtspunkts schließt der Eingabebitstring einen Teil des Headers ein, der sich auf die jeweilige Anwendung bezieht. Dadurch können die Informationen im Header, die sich auf eine jeweilige Anwendung beziehen, in die Anwendungsdaten eingeschlossen werden, sodass der betroffene Teil des Headers mittels der Fehlerkorrekturfunktionen der jeweiligen Anwendung vor Fehlern geschützt werden kann. Somit können Paritätsbits oder Fehlerkorrekturbits in der jeweiligen Anwendung gewisse Felder im Header schützen.

[0018] In einem Beispiel des ersten Gesichtspunkts umfasst das Verfahren ferner das Erfassen von Bilddaten aus einer grafischen Darstellung, die physisch auf den ersten QR-Code gelegt ist, und das Einschließen einer Angabe eines Vorhandenseins der grafischen Darstellung und eines von der grafischen Darstellung belegten Bereichs auf dem ersten QR-Code in den Header der kodierten Daten, wobei die Datenpixel des Kodierbereichs des ersten QR-Codes, die einen Bereich im ersten QR-Code belegen, der mit dem von der grafischen Darstellung belegten Bereich übereinstimmt, von der Speicherung des Teils oder der Gesamtheit der kodierten Daten ausgeschlossen sind. Dadurch kann eine größere grafische Darstellung (z. B. Buchstabe, Figur, Bild, Logo, Symbol, Design, Muster, Modell usw.) auf den ersten QR-Code gelegt werden als bei dem Verfahren des absichtlichen Einbringens von Fehlern in die kodierten Daten, da die Fehlerkorrekturfunktionen des ersten QR-Codes zum Korrigieren der durch die überlagerte grafische Darstellung eingeführten Fehler erhalten bleiben.

[0019] In einem Beispiel des ersten Gesichtspunkts umfasst das Verfahren ferner das Konfigurieren des jeweiligen Fehlerkorrekturmechanismus pro Anwendung mit einer niedrigeren Fehlerkorrekturstufe als der einzelne Fehlerkorrekturmechanismus, der von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzt wird. Diese Konfigurationsinformationen können im Header oder in den Datencontainern gespeichert sein. Aus der Liste der bekannten Fehlerkorrekturstufen L, M, Q, H könnte der ECM pro Anwendung beispielsweise die niedrigste Fehlerkorrekturstufe L aufweisen, während der von allen Anwendungen der Vielzahl von Anwendungen gemeinsame ECM beispielsweise die höchste Fehlerkorrekturstufe H aufweisen könnte. In dem Fall, dass der erste QR-Code die Reed-Solomon-Fehlerkontrollkodierung als Fehlerkorrekturmechanismus verwendet, bieten die vier Fehlerkorrekturstufen L, M, Q, H eine jeweilige Wiederherstellungsfähigkeit von ungefähr 7 %, 15 %, 25 %, 30 %.

[0020] In einem Beispiel des ersten Gesichtspunkts umfasst der erste QR-Code Bootstrapping-Informationen zum Device Provisioning Protocol (DPP) oder Wi-Fi Easy Connect zum Speichern von Informationen auf

mindestens einem öffentlichen Schlüssel. Wenn zwischen einer Vielzahl von Konfiguratoren und einem angemeldeten Gerät, an den der erste QR-Code angehängt ist, eine entsprechende Wi-Fi-Verbindung hergestellt werden kann, kann jeder Konfigurator, der während des Bootstrapping-Prozesses den ersten QR-Code des angemeldeten Geräts scannt, seinen jeweiligen öffentlichen Schlüssel abrufen, z. B. seinen öffentlichen Identitätsschlüssel.

[0021] Gemäß einem zweiten Gesichtspunkt, der auf das Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen in einem ersten QR-Code gerichtet ist, wobei die Informationen als kodierte Daten im ersten QR-Code gespeichert sind und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung umfassen, umfasst das Verfahren mindestens: Lesen eines Headers des ersten QR-Codes, wobei der Header mindestens eine einzelne Kennung umfasst, die ein Vorhandensein der Vielzahl von Anwendungen angibt, und eine jeweilige Anwendungskennung pro Anwendung; Identifizieren des ersten QR-Codes aus dem Header unter Verwendung der einzelnen Kennung, die das Vorhandensein der Vielzahl von Anwendungen angibt; Identifizieren der jeweiligen Anwendung aus dem Header; und Identifizieren des jeweiligen Datencontainers, welcher der jeweiligen Anwendung zugeordnet ist. Alternativ kann der erste QR-Code visuell identifiziert werden, beispielsweise mittels eines für den Benutzer verständlichen gedruckten Musters (z. B. eines Buchstabens M zur Bezeichnung eines Master- oder Matter-basierten QR-Codes), sodass die Identifizierung von der Person durchgeführt werden kann, die beispielsweise das QR-Code-Lesegerät bedient.

[0022] In einem Beispiel des zweiten Gesichtspunkts umfasst das Verfahren ferner das Lesen aller Pixel, die dem Datencontainer der jeweiligen Anwendung zugeordnet sind, um mindestens ein der jeweiligen Anwendung zugeordnetes Datencodewort und Paritäts- oder Fehlererkennungs- oder Fehlerkorrekturbits für die gesamte Vielzahl von Anwendungen abzurufen, und das Erhalten fehlerfreier Anwendungsdaten der jeweiligen Anwendung. Dadurch können die Informationen einer jeweiligen Anwendung aus der Vielzahl von Anwendungen selektiv abgerufen werden, z. B. durch eine Dekodiereinrichtung wie einen QR-Code-Leser, und zwar auf zuverlässige Weise, ohne dass alle Datenpixel des ersten QR-Codes gelesen werden müssen, sondern nur die Datenpixel für die jeweilige Anwendung, was den Lesevorgang beschleunigen kann.

[0023] In einem Beispiel des zweiten Gesichtspunkts umfasst das Verfahren ferner das Auswählen einer Anwendung aus der Vielzahl von Anwendungen basierend auf einer vorkonfigurierten Anwendung oder einer Richtlinie oder einem Kontext und das Verwenden der dekodierten Informationen aus der jeweiligen Anwendung zum Starten einer Anwendung oder eines Protokolls. Beispielsweise kann es sich bei der zu startenden Anwendung oder dem zu startenden Protokoll um eine Inbetriebnahmeanwendung für ein drahtloses Produkt handeln. Um diese Vorkonfiguration zu veranschaulichen: das QR-Code-Lesegerät könnte beispielsweise von einem Hersteller stammen, der es so vorkonfiguriert, dass es nur eine spezifische Anwendung verwendet, z. B. ein SHS wie Matter. Stammt das QR-Code-Lesegerät jedoch beispielsweise von einem anderen Hersteller, der Wi-Fi verwendet, weiß dieser andere Hersteller möglicherweise nicht, ob der Endbenutzer ein Wi-Fi-Netzwerk oder ein SHS-Netzwerk verwendet, und die Wahl kann kontextabhängig sein.

[0024] In einem Beispiel des zweiten Gesichtspunkts umfasst das Verfahren ferner das Auswählen einer Anwendung aus der Vielzahl von Anwendungen als erste Inbetriebnahmeanwendung aus dem ersten QR-Code, das Abrufen von Inbetriebnahmedaten, die sich auf die ausgewählte erste Inbetriebnahmeanwendung beziehen, das Starten der ersten Inbetriebnahmeanwendung, und das Ausführen der ersten Inbetriebnahmeanwendung, bis die erste Inbetriebnahmeanwendung entweder erfolgreich ist oder fehlschlägt. Wenn die erste Inbetriebnahmeanwendung fehlschlägt, umfasst das Verfahren das Auswählen einer anderen Anwendung der Vielzahl von Anwendungen als zweite Inbetriebnahmeanwendung aus dem ersten QR-Code, das Abrufen von Inbetriebnahmedaten, die sich auf die zweite Inbetriebnahmeanwendung beziehen, und das Starten der zweiten Inbetriebnahmeanwendung. Dadurch kann die Kombination mehrerer Inbetriebnahmeanwendungen vermieden werden.

[0025] In einem Beispiel des zweiten Gesichtspunkts, wobei die Informationen Inbetriebnahmeinformationen für ein Kommunikationsprotokoll kodieren, umfasst das Verfahren ferner das Auswählen eines ersten Inbetriebnahmeprotokolls basierend auf mindestens einem vorkonfigurierten Wert, einer Richtlinie, einem Kontext und einem Teil der Daten im ersten QR-Code.

[0026] In einem Beispiel des zweiten Gesichtspunkts wird das erste Protokoll gestartet, und ein zweites Protokoll wird nicht ausgelöst, solange das erste Protokoll nicht beendet ist.

[0027] In einem Beispiel des zweiten Gesichtspunkts umfasst das Verfahren das Lesen des ersten QR-Codes, das Bestimmen der unterstützten Anwendungen der Vielzahl von Anwendungen aus dem gelesenen ersten QR-Code, das Auswählen einer Anwendung unter den unterstützten Anwendungen basierend auf mindestens einem von: einer vorkonfigurierten Auswahl, einer Richtlinie, eines Kontextes und einer Präferenz, die im ersten QR-Code eingeschlossen sind, und das Starten der ausgewählten Anwendung als erste ausgewählte Anwendung. Im Falle einer Inbetriebnahmeanwendung sollte der Konfigurator oder Inbetriebnehmer eine zweite ausgewählte Anwendung nicht starten, solange die erste ausgewählte Anwendung nicht erfolgreich war oder fehlgeschlagen ist, z. B. solange der mit der ersten ausgewählten Anwendung verbundene Inbetriebnahmeprozess eine Mindestanzahl N-mal nicht erfolgreich war oder fehlgeschlagen ist, z. B. N = 1, 2, ...

[0028] In einem Beispiel des zweiten Gesichtspunkts umfasst das Verfahren das Lesen des ersten QR-Codes, das Bestimmen der unterstützten Anwendungen der Vielzahl aus dem gelesenen ersten QR-Code, das Auswählen einer Anwendungsausführungsreihenfolge basierend auf mindestens einem von: einer vorkonfigurierten Auswahl, einer Richtlinie, einem Kontext und einer Präferenz, die im ersten QR-Code eingeschlossen sind, das Starten einer ersten Anwendung (im Fall einer Inbetriebnahmeanwendung sollte der Konfigurator oder Inbetriebnehmer keine zweite ausgewählte Anwendung starten, solange die erste ausgewählte Anwendung nicht erfolgreich war oder fehlgeschlagen ist, zumindest bis zu einem vorkonfigurierten Zielschritt, z. B. solange der mit der ersten ausgewählten Anwendung verbundene Inbetriebnahmeprozess mindestens N-mal nicht erfolgreich war oder fehlgeschlagen ist, z. B. N = 1, 2, ...), und nach erfolgreicher (möglicherweise teilweiser in dem Fall, in dem ein zweites Inbetriebnahmeprotokoll auf einem ersten Inbetriebnahmeprotokoll (DPP) als Zwischenschritt beruhen kann) Ausführung der ersten Anwendung das Starten einer zweiten Anwendung. Ein Beispiel für die letztgenannte Verwendung von zwei Anwendungen kann die Verwendung eines ersten Inbetriebnahmeprotokolls (DPP) als Zwischenschritt bei der Ausführung der ersten Anwendung zum Registrieren der Vorrichtung in einem Wi-Fi-Netzwerk sein, anstatt die zweite Anwendung hierfür zu verwenden, und das Verwenden der zweiten Anwendung zum Verbinden der Vorrichtung mit z. B. einem Cloud-Dienst oder einem Back-End in der Cloud. Ein Vorteil hiervon kann beispielsweise sein, dass die erste Anwendung eine sicherere Möglichkeit zur Verbindung mit einem Wi-Fi-Netzwerk bietet als die zweite Anwendung, während die erste Anwendung keine Verbindung zu einem oder dem richtigen Cloud-Dienst anbietet.

[0029] Gemäß einem dritten Gesichtspunkt, der sich auf eine Einrichtung zum Kodieren von Informationen einer Vielzahl von Anwendungen zur Speicherung in einem ersten QR-Code bezieht, ist die Einrichtung ausgestaltet, um das im ersten Gesichtspunkt und/oder in einem der Beispiele des ersten Gesichtspunkts beanspruchte Verfahren durchzuführen.

[0030] Gemäß einem vierten Gesichtspunkt, der sich auf eine Einrichtung zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen in einem ersten QR-Code bezieht, wobei die Informationen als kodierte Daten im ersten QR-Code gespeichert sind und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen, ist die Einrichtung ausgestaltet, um das Verfahren gemäß dem zweiten Gesichtspunkt und/oder einem der Beispiele des zweiten Gesichtspunkts durchzuführen.

[0031] Gemäß einem fünften Gesichtspunkt, der sich auf eine Vorrichtung bezieht, der ein QR-Code zugeordnet ist, wobei der QR-Code Inbetriebnahmeinformationen für zwei unterschiedliche Inbetriebnahmeprotokolle einschließt, akzeptiert die Vorrichtung nach Empfang einer ersten Inbetriebnahmenachricht eines ersten Inbetriebnahmeprotokolls von einer Inbetriebnahmevorrichtung keine erste Inbetriebnahmenachricht eines zweiten Inbetriebnahmeprotokolls, bis das erste Inbetriebnahmeprotokoll beendet ist.

[0032] Gemäß einem sechsten Gesichtspunkt, der auf ein System gerichtet ist, umfasst das System mindestens: die Einrichtung gemäß dem vierten Gesichtspunkt, umfassend eine Kamera und eine erste Kommunikationsschnittstelle; eine erste Vorrichtung mit einem zugehörigen QR-Code und einer zweiten Kommunikationsschnittstelle; eine zweite Vorrichtung mit einer dritten Kommunikationsschnittstelle; und eine dritte Vorrichtung, wobei die Einrichtung ausgestaltet ist, um eine Schnittstelle mit der dritten Vorrichtung zu bilden und ein erstes Inbetriebnahmeprotokoll mit der ersten Vorrichtung zu starten und auszuführen, und/oder wobei die dritte Vorrichtung ausgestaltet ist, um ein zweites Inbetriebnahmeprotokoll, das entweder mit dem ersten Inbetriebnahmeprotokoll identisch ist oder sich davon unterscheidet, mit der ersten Vorrichtung von der zweiten Vorrichtung aus auszulösen.

[0033] Gemäß einem siebten Gesichtspunkt, der auf eine Rechenvorrichtung gerichtet ist, umfasst ein Computerprogrammprodukt Programmanweisungen oder Codemittel, so dass, wenn das Computerprogrammprodukt auf einer Verarbeitungseinheit der Rechenvorichtung ausgeführt wird, die Rechenvorrichtung angeordnet ist, um das Verfahren gemäß dem ersten Gesichtspunkt und/oder einem der Beispiele des ersten Gesichtspunkts durchzuführen.

[0034] Gemäß einem achten Gesichtspunkt, der auf eine Rechenvorrichtung gerichtet ist, umfasst ein Computerprogrammprodukt Programmanweisungen oder Codemittel, so dass, wenn das Computerprogrammprodukt auf einer Verarbeitungseinheit der Rechenvorichtung ausgeführt wird, die Rechenvorrichtung angeordnet ist, um das Verfahren gemäß dem zweiten Gesichtspunkt und/oder einem der Beispiele des zweiten Gesichtspunkts durchzuführen.

[0035] Es wird darauf hingewiesen, dass die vorstehend genannten Einrichtungen auf der Grundlage diskreter Hardwareschaltlogiken mit diskreten Hardwarekomponenten, integrierten Chips oder Anordnungen von Chipmodulen, oder auf der Grundlage von Signalverarbeitungsvorrichtungen oder Chips implementiert werden können, die durch in Speichern abgelegte, auf computerlesbare Medien geschriebene oder von einem Netzwerk wie dem Internet heruntergeladene Softwareroutinen oder -programme gesteuert werden.

[0036] Es wird auch ein System bereitgestellt, das mindestens die hierin beschriebene Einrichtung umfasst und eine Kamera und eine erste Kommunikationsschnittstelle, einen QR-Code mit einer Vielzahl von Containern, die jeweiligen persönlichen Datenberichten jeweiliger Personen zugeordnet sind, wobei die jeweiligen Personen in einer Gruppe sind, eine zweite Vorrichtung, die eine zweite Kommunikationsschnittstelle aufweist, wobei die zweite Vorrichtung eine Datendatei aufweist, die den Datenberichten der jeweiligen Personen zugeordnet ist, wobei die Einrichtung und die zweite Vorrichtung angeordnet sind, um ein Verifizierungsprotokoll zu starten und auszuführen, wobei das Verifizierungsprotokoll das Lesen und Dekodieren des QR-Codes, das Abrufen der Datendatei und das Vergleichen des Inhalts der Datendatei mit Daten, die durch das Dekodieren der Vielzahl von Containern erhalten wurden, und das Bereitstellen einer Verifizierungsantwort umfasst.

[0037] Es versteht sich, dass eine bevorzugte Ausführungsform der Erfindung auch jede Kombination der abhängigen Ansprüche oder der vorstehenden Ausführungsformen mit dem jeweiligen unabhängigen Anspruch sein kann.

[0038] Diese und andere Gesichtspunkte der Erfindung werden aus den nachstehend beschriebenen Ausführungsformen ersichtlich und unter Bezugnahme auf diese erläutert.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0039] Zum besseren Verständnis der vorliegenden Anmeldung wird nun anhand von Beispielen auf die beigefügten Zeichnungen Bezug genommen, in denen:

Fig. 1 eine Struktur eines QR-Codesymbols der Version 7 gemäß einer herkömmlichen Ausführungsform zeigt;

Fig. 2 ein schematisches Flussdiagramm zeigt, das einen beispielhaften Kodiervorgang gemäß einer Ausführungsform der vorliegenden Erfindung beschreibt;

Fig. 3 eine schematische Zuordnung von Datenpixeln eines QR-Codes zu kodierten Daten für zwei Anwendungen zeigt, gemäß einer Ausführungsform der vorliegenden Erfindung;

Fig. 4 eine schematische Zuordnung von Datenpixeln eines QR-Codes zu kodierten Daten für drei Anwendungen zeigt, gemäß einer Ausführungsform der vorliegenden Erfindung;

Fig. 5 eine schematische Zuordnung von Datenpixeln eines QR-Codes zu kodierten Daten für zwei Anwendungen bei Vorhandensein eines Bildes zeigt, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

Fig. 6 ein schematisches Flussdiagramm zeigt, das ein Beispiel eines Dekodiervorgangs gemäß einer Ausführungsform der vorliegenden Erfindung beschreibt;

Fig. 7 ein schematisches Flussdiagramm zeigt, das ein weiteres Beispiel eines Dekodiervorgangs gemäß einer Ausführungsform der vorliegenden Erfindung beschreibt;

Fig. 8 ein schematisches Flussdiagramm zeigt, das einen weiteren beispielhaften Dekodiervorgang gemäß einer Ausführungsform der vorliegenden Erfindung beschreibt;

Fig. 9 ein schematisches Blockdiagramm einer Vorrichtung zeigt, die in der Lage ist, Informationen mehrerer Anwendungen zu kodieren und einen QR-Code zu erstellen, gemäß einer Ausführungsform der vorliegenden Erfindung;

Fig. 10 ein schematisches System zur Inbetriebnahme gemäß einer Ausführungsform der vorliegenden Erfindung zeigt.

Fig. 11 ein Inbetriebnahmeprotokoll gemäß einer herkömmlichen Ausführungsform zeigt;

Fig. 12 ein Inbetriebnahmeprotokoll gemäß einer Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 13 ein Inbetriebnahmeprotokoll gemäß einer Ausführungsform der vorliegenden Erfindung zeigt; und

Fig. 14 ein System zum Lesen und Verarbeiten mit einmaligem Scannen eines QR-Codes gemäß einer Ausführungsform der vorliegenden Erfindung zeigt.

DETAILLIERTE BESCHREIBUNG VON AUSFÜHRUNGSFORMEN

[0040] Ausführungsformen der vorliegenden Erfindung werden auf der Grundlage eines QR-Codes beschrieben, insbesondere eines QR-Codes, der in der Lage ist, mehrere Anwendungen oder Implementierungen zu handhaben.

[0041] Die Internationale Organisation für Normung (ISO) hat einen Standard ISO/IEC18004:2015 für den QR-Code mit dem Titel: "Informationstechnik - Auomatische Identifikation und Datenerfassungsverfahren - Spezifikation der Barcode - Symbologie ‚QR Code' " erstellt, der die standardisierte Symbolstruktur eines QR-Codes wie folgt definiert. Ein QR-Codesymbol besteht aus einer zweidimensionalen Anordnung heller und dunkler Quadrate, die als Module bezeichnet werden. Es gibt vierzig Größen von QR-Code-Symbolversionen, die von Version 1 bis Version 40 reichen. Jede QR-Code-Symbolversion besteht aus einer unterschiedlichen Anzahl von Modulen, und daher führen unterschiedliche QR-Code-Versionen zu unterschiedlichen Datenkapazitäten. **Abb.** 1 der vorliegenden Anwendung stammt aus **Abb.** 3 in Abschnitt 6.3.1 der Norm ISO/IEC18004:2015 und stellt eine Struktur eines QR-Codesymbols der Version 7 dar, das aus Kodierungsbereichen 1 (d. h. Formatinformationen, Versionsinformationen, Datencodewörtern und Fehlerkorrekturcodewörtern) und Funktionsmustern 2 (d. h. Sucher, Trennzeichen, Zeitmustern und Ausrichtungsmustern) besteht. Die Funktionsmuster kodieren keine Daten und das Symbol ist auf allen vier Seiten von einem leeren Bereich umgeben, der als Ruhezone 3 bezeichnet wird.

[0042] Nachrichtendaten werden als Bitstrom kodiert, der in eine Folge von Codewörtern unterteilt ist. Alle Codewörter sind 8 Bit lang. Die Codewörter werden basierend auf der QR-Code-Version und der Fehlerkorrekturstufe in mehrere Fehlerkorrekturblöcke gruppiert und für jeden Block wird eine entsprechende Anzahl von Fehlerkorrekturcodewörtern generiert. Der Fehlerkorrekturmechanismus (ECM) ermöglicht die korrekte Dekodierung der Nachricht in dem Fall, dass ein Teil des Symbols verschmutzt oder beschädigt ist. Der QR-Code der Norm ISO/IEC18004:2015 verwendet zur Fehlererkennung und -korrektur die Reed-Solomon-Fehlerkontrollkodierung. Es ist bekannt, dass der Reed-Solomon-Code ein [n, k, n - k + 1]-Code ist, d. h. ein linearer Blockcode der Länge n (über einen endlichen Körper) mit der Dimension k und einer minimalen Hamming-Distanz gleich n - k + 1. Der Reed-Solomon-Code (wie auch jeder andere MDS-Code) kann doppelt so viele Löschungen (d. h. fehlerhafte Codewörter an bekannten Stellen) wie Fehler (d. h. fehlerhafte Codewörter an unbekannten Stellen) korrigieren, und jede Kombination aus Fehlern und Löschungen kann korrigiert werden, solange die Beziehung $2E + S \leq n - k$ erfüllt ist, wobei E die Anzahl der Fehler und S die Anzahl der Löschungen im Block ist. Weitere Beispiele dieser ECMs können Hamming-Codes oder Paritätsprüfcodes mit niedriger Dichte sein. Es gibt vier Fehlerkorrekturstufen L, M, Q, H, die der Benutzer auswählen kann. Jede Stufe bietet eine unterschiedliche Fehlerkorrekturkapazität: L bis zu 7 %, M bis zu 15 %, Q bis zu 25 % und H bis zu 30 %. Höhere Fehlerkorrekturstufen verbessern die Wiederherstellungskapazität, erhöhen aber auch die Menge der kodierten Daten oder die Anzahl der Paritätssymbole. Dies bedeutet, dass bei der Kodierung derselben Nachricht mit einer höheren Fehlerkorrekturstufe möglicherweise eine größere QR-Code-Version erforderlich ist. Die Anzahl der Datencodewörter, Fehlerkorrekturblöcke und Fehlerkorrekturcodewörter hängt von der QR-Code-Version und der Fehlerkorrekturstufe ab.

[0043] In einer beispielhaften Ausführung der vorliegenden Erfindung wird ein erster QR-Code, der an einem Gegenstand, einem Produkt oder einer Vorrichtung angebracht ist, zum Speichern kodierter Daten für eine

Vielzahl von Anwendungen (oder Implementierungen) verwendet. Es ist zu beachten, dass zur Unterscheidung von einem QR-Code, der codierte Daten für eine einzelne Anwendung speichert, der Begriff „Master-QR-Code“ in der Beschreibung synonym für den ersten QR-Code verwendet werden kann, d. h. für einen QR-Code, der kodierte Daten für eine Vielzahl von Anwendungen speichert. Der erste QR-Code kann Informationen der Vielzahl von Anwendungen in verschlüsselter Form enthalten. Die Phrase „für eine Anwendung“ kann hierin als „zur Verwendung mit einer Anwendung angeordnet“ verstanden werden.

[0044] Informationen einer jeweiligen Anwendung können beliebige Zeichenfolgen sein, die beispielsweise in nicht abschließender Weise mindestens eines darstellen können von: dem Namen der Anwendung (z. B. „URL zu Unternehmen X“ oder „Wi-Fi-Netzwerkkonfiguration“), einer Kennung, welche die jeweilige Anwendung eindeutig identifiziert, Konfigurationsdaten für den OA, die URL zu der für diesen QR-Code und diese Anwendung zu verwendenden App oder dem zu verwendenden Programm, und den Speicherort des QR für die jeweilige Anwendung (z. B. „der QR-Code, auf den der grüne Pfeil zeigt“, oder „der QR-Code, der durch das blaue Rechteck angezeigt wird“, oder „der QR-Code rechts neben dem Ein-/Aus-Schalter“).

[0045] Die kodierten Daten der kodierten Informationen können mindestens einen Header und einen entsprechenden Datencontainer pro Anwendung einschließen. Der Header kann eine einzelne Kennung, die das Vorhandensein der Vielzahl von Anwendungen anzeigt, und eine Zählung der Vielzahl von Anwendungen und eine jeweilige Anwendungskennung pro Anwendung umfassen. Die einzelne Kennung kann implizit das Vorhandensein einer Vielzahl (spezifischer) Anwendungen angeben, z. B. zwei Anwendungen im Zusammenhang mit Wi-Fi Easy Connect und einem SHS. Ein Datencontainer kann die Anwendungsdaten einer Anwendung enthalten und auf diejenigen Datenpixel des ersten QR-Codes verweisen, die für die Speicherung der Anwendungsdaten reserviert sind. In einer weiteren Ausführungsform kann der Datencontainer auch auf die physischen Orte im ersten QR-Code verweisen, die zum Speichern der Anwendungsdaten dienen. Ein Datencontainer kann sich auch auf einen Teil eines Bitstrings beziehen, der die einer Anwendung zugeordneten Daten enthält.

[0046] Es ist zu beachten, dass sich der Begriff „Datenpixel“ in der vorliegenden Erfindung auf eine Speicherungseinheit in einem QR-Code bezieht. Wenn beispielsweise ein „Datenpixel“ die kleinstmögliche Speicherungseinheit im QR-Code ist, dann kann es nur eine gewisse Menge von Bits speichern, beispielsweise 1 Bit. In dem Beispielfall, in dem sich ein „Datenpixel“ auf eine Datenspeicherungseinheit beziehen würde, die b Bits in einem QR-Code speichern kann, der in der Lage ist, D Bits zu speichern, dann könnte der QR-Code D/b „Datenpixel“ speichern. Der Begriff „Datenpixel“ sollte daher nicht mit dem Begriff „Bildpixel“ verwechselt werden. Beispielsweise besteht das Bild eines QR-Codes bei der Aufnahme durch eine Kamera aus Bildpixeln, während Gruppen von einem oder mehreren Bildpixeln jeweilige Datenpixel darstellen können.

[0047] Ein entsprechender ECM (z. B. Reed-Solomon-Codes, Hamming-Codes, Low-Density-Parity-Check-Codes und dergleichen) kann in jedem Datencontainer pro Anwendung eingeschlossen sein, um etwaige Fehler in den kodierten Daten des jeweiligen Datencontainers pro Anwendung zu korrigieren und dadurch ein schnelles Lesen der Anwendungsdaten im ersten QR-Code zu ermöglichen, der kodierte Daten zur Verwendung mit mehr als einer einzigen Anwendung enthält. Insbesondere wenn eine jeweilige Anwendung, gekennzeichnet mit OAi, die Kodierung von Anwendungsdaten Di erfordert, wird ein ECM verwendet, um Di einem Codewort Ci zuzuordnen. Der ECM kann Di von Ci abrufen, auch wenn Ci einige Fehler aufweist. Die Anwendungsdaten Di können zunächst in b Blöcke aufgeteilt werden, so dass insgesamt b Codewörter erhalten werden. Um das Risiko einer Beschädigung der Anwendungsdaten zu verringern, können die Bits oder Symbole dieser unterschiedlichen Codewörter verschachtelt, d. h. gemischt, werden, sodass beim Auftreten einer Fehlerhäufung (z. B. aufgrund einer Beschädigung eines physischen Bereichs des QR-Codes) die Fehler auf mehrere Codewörter verteilt werden.

[0048] In einer optionalen Ausführungsform kann in jedem Datencontainer pro Anwendung ein entsprechender Fehlererkennungsmechanismus (EDM) (z. B. zyklische Redundanzprüfung (CRC), Paritätsprüfung, Prüfsumme und dergleichen) oder ein entsprechender Fehlerkorrekturmechanismus (ECM) zum Erkennen oder Korrigieren von Fehlern in den kodierten Daten des jeweiligen Datencontainers pro Anwendung eingeschlossen sein.

[0049] In einer optionalen Ausführungsform können die kodierten Daten ein von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzter Gesamt-EDM zum Erkennen von Fehlern in einem Datencontainer basierend auf Informationen in einem oder mehreren der anderen Datencontainer einschließen.

[0050] In einer optionalen Ausführungsform können die kodierten Daten einen Gesamt-ECM einschließen, der von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzt wird, um alle Fehler in einem Datencontainer basierend auf Informationen in einem oder mehreren der anderen Datencontainer zu korrigieren.

[0051] Fig. 2 zeigt ein schematisches Flussdiagramm 200, das ein beispielhaftes Kodierverfahren zum Kodieren von Informationen der Vielzahl von Anwendungen zur Speicherung im ersten QR-Code gemäß einer beispielhaften Ausführungsform der vorliegenden Erfindung beschreibt.

[0052] In Schritt 210 werden Informationen zu einer Vielzahl von Anwendungen gesammelt.

[0053] In Schritt 220 werden die Informationen in verschlüsselte Daten kodiert, wobei die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen. Der Header kann eine einzelne Kennung, die ein Vorhandensein der Vielzahl von Anwendungen angibt, und eine Zählung der Vielzahl von Anwendungen und eine jeweilige Anwendungskennung pro Anwendung umfassen.

[0054] In einem detaillierten Beispiel, das sich auf das obige Kodierverfahren aus **Fig.** 2 bezieht, können die Informationen der Vielzahl von Anwendungen, wobei jede Anwendung einem Datencontainer zugeordnet ist, zur Speicherung im ersten QR-Code gemäß dem folgenden beispielhaften Verfahren unter Verwendung des von allen Datencontainern gemeinsam genutzten ECM und optional des EDM und des Headers in die oben genannten kodierten Daten kodiert werden. Es ist zu beachten, dass in diesem Beispiel ein einzelner QR-Code Informationen für mehrere Anwendungen enthalten kann.

[0055] In einem ersten Schritt werden Eingabedaten, die sich auf den ersten QR-Code (also den Master-QR-Code) beziehen, gesammelt. Die Eingabedaten können somit mindestens eine Zählung der Vielzahl von Anwendungen und eine jeweilige Anwendungskennung pro Anwendung einschließen. Im Beispielfall von N Anwendungen, die als OA1, ..., OAN bezeichnet werden, können diese als Header-Daten (HD) bezeichneten Eingabedaten wie folgt strukturiert sein: N, OA1, ..., OAN. Es ist zu beachten, dass die jeweiligen Anwendungskennungen jeweils eindeutig sind, z. B. durch Registrierung oder Standardisierung. Im Beispielfall zweier Anwendungen mit den Kennungen OA1 und OA2 könnte der Inhalt des Headers explizit als „2, OA1, OA2“ strukturiert sein, oder alternativ könnte im Header eine einzelne Kennung (ID) eingeschlossen sein, die implizit „2, OA1, OA2“ bedeutet.

[0056] Es ist zu beachten, dass die obigen Eingabedaten keinen Aufschluss darüber geben, wie viele Daten in Bits für die jeweilige Anwendung gespeichert werden. In diesem Fall muss das QR-Code-Lesegerät, wie bereits erwähnt, bei einem QR-Code mit einer gegebenen Speicherungskapazität möglicherweise davon ausgehen, dass die Speicherungskapazität gleichmäßig auf die Anzahl der jeweiligen verfügbaren Anwendungen aufgeteilt ist. Dies kann zu einer Verschwendung von Ressourcen führen, wenn beispielsweise eine Anwendung nur sehr wenig Datenspeicherplatz benötigt und eine andere Anwendung deutlich mehr, insbesondere etwas mehr als die Hälfte der Speicherungskapazität eines bestimmten QR-Code-Typs. In diesem Fall müssten, wenn die Länge nicht eingeschlossen ist, die jeweiligen beiden Anwendungen in einem größeren QR-Code kodiert werden. Um dieses Problem zu vermeiden, können in einer optionalen Ausführungsform die Header-Daten (HD) auch Längenparameter für jede Anwendung einschließen und somit wie folgt strukturiert sein: N, OA1, ..., OAN, L1, ..., LN.

[0057] In einem zweiten Schritt werden für jede der N jeweiligen Anwendungen Anwendungsdaten Di (d. h. die Zeichen in Di) der jeweiligen Anwendung OAi gesammelt.

[0058] In einem ersten Unterschritt des zweiten Schritts werden entsprechende Bitstromdaten (BSD) aus Di (d. h. ein Eingabebitstring BSDi aus Di) berechnet. Es ist zu beachten, dass der Eingabestring auch einen Teil der Header-Daten (HD) einschließen kann, die sich auf die jeweilige Anwendung OAi beziehen, z. B. die OAi-Kennung und die Position der jeweiligen Anwendung OAi in der Liste der N Anwendungen, oder die Länge Li für die jeweilige Anwendung OAi. Der Nutzen des Einschließens dieser HD-Informationen in den Eingabebitstring BSDi liegt darin, dass sie mithilfe des ECM und optional des der jeweiligen Anwendung zugeordneten EDM vor Fehlern geschützt sind. Mit anderen Worten können Paritätsbits in den Anwendungsdaten der jeweiligen Anwendung gewisse Felder im Header schützen. Um die Zuverlässigkeit des Headers zu verbessern, können darüber hinaus zwei Optionen einzeln oder in Kombination vorgeschlagen werden. In einer ersten Option kann der Header beispielsweise seine eigenen Fehlerkorrekturfunktionen in einer Weise einschließen, dass alle Fehler korrigiert werden können, bevor der Header auf den Eingabebitstring BSDi zugreift. In einer zweiten Option können die Header-Informationen beispielsweise auch physisch in Pixeln in

der Nähe von einem, zwei oder drei „Augen“ oder „Positionsmarkierungen“ oder „dem Quadrat innerhalb von Quadraten an drei Ecken“ gespeichert werden. Der Grund liegt darin, dass diese Positionsmarkierungen zur Identifizierung und Positionierung eines QR-Codes erforderlich sind. Dies ähnelt den in **Fig.** 1 dargestellten Format- und Versionsinformationen. Werden die Positionsmarkierungen entfernt oder beschädigt, ist der QR-Code somit in jedem Fall nicht lesbar. Wenn die Header-Informationen jedoch beispielsweise an bekannten physischen Orten in der Nähe einer oder mehrerer Positionsmarkierungen platziert werden, erhöht sich die Wahrscheinlichkeit, dass die Header-Informationen gelesen werden können.

[0059] In einem zweiten Unterschritt des zweiten Schritts wird optional ein Fehlererkennungsbitstring EDi, z. B. ein CRC, aus dem Eingabebitstring BSDi berechnet, wenn der EDM auf CRC basiert.

[0060] In einem dritten Unterschritt des zweiten Schritts wird optional ein verschachtelter Eingabebitstring BSDi, bezeichnet mit IBSDi, durch Verschachteln von EDi und BSDi erhalten, um potenzielle Fehler zu verteilen. Es ist zu beachten, dass das QR-Codelesegerät wissen muss, ob und wie eine Verschachtelung erfolgt. Dies kann auf einem Standard basieren oder explizit bzw. implizit, beispielsweise im Header, gespeichert sein. Dies gilt auch für andere Verschachtelungsschritte in anderen Ausführungsformen der vorliegenden Erfindung.

[0061] In einem vierten Unterschritt des zweiten Schritts werden ein oder mehrere 8-Bit-Codewörter Ci berechnet, indem der verschachtelte Eingabebitstring BSDi in eine Folge von einem oder mehreren 8-Bit-Datencodewörtern aufgeteilt wird, indem die Datencodewortfolge in eine erforderliche Anzahl von Blöcken unterteilt wird (abhängig von der QR-Codeversion und von der Fehlerkorrekturstufe L, M, Q, H), um die Verarbeitung des ECM (z. B. des Reed-Solomon-Codes) zu ermöglichen, indem ein oder mehrere 8-Bit-Fehlerkorrekturcodewörter für jeden Block generiert werden und indem die Fehlerkorrekturcodewörter an das Ende der Datencodewortfolge angehängt werden oder indem die Fehlerkorrekturcodewörter an einer oder mehreren geeigneten Stellen in der Datencodewortfolge eingefügt werden.

[0062] Es ist zu beachten, dass dieser vierte Unterschritt aufzeigt, dass pro OA Fehlerkorrekturfähigkeiten vorhanden sind. Wenn wir Anwendungsdaten von zwei Anwendungen OA1, OA2 speichern und dabei einen einzigen Fehlerkorrekturbitstring für alle Anwendungsdaten verwenden würden, läge der Gesamtspeicherbedarf bei D = D1 + D2 + PB, wobei sich D1 und D2 jeweils auf die Anwendungsdaten von OA1 und OA2 beziehen und PB sich auf die Paritätsbits für alle beiden Anwendungen OA1 und OA2 bezieht und für die Fehlerkorrektur erforderlich ist. Wenn wir hingegen im vorliegenden vierten Unterschritt beispielsweise Anwendungsdaten von zwei Anwendungen OA1, OA2 speichern müssen, wird es einen Fehlerkorrekturbitstring geben, der auf die jeweiligen Anwendungsdaten von OA1 und OA2 angewendet wird. Dies bedeutet, dass der gesamte Speicherbedarf nicht mehr D, sondern D' = D1 + D2 + PB 1 + PB2 beträgt, wobei sich D1 und D2 jeweils auf die Anwendungsdaten von OA1 und OA2 beziehen, PB1 sich auf die Paritätsbits für die Anwendung OA1 bezieht, und PB2 sich auf die Paritätsbits für die Anwendung OA2 bezieht. Obwohl D' größer als D sein kann, muss ein QR-Code-Lesegerät nur D1 + PB1 lesen, wenn es an der Anwendung OA1 interessiert ist, oder D2 + PB2, wenn es an der Anwendung OA2 interessiert ist, und diese Summe D1 + PB1 oder D2 + PB2 wird voraussichtlich kleiner als D sein.

[0063] In einem fünften Unterschritt des zweiten Schritts werden die ein oder mehreren 8-Bit-Codewörter Ci optional mit sich selbst oder mit IBSDi oder mit einer Kombination davon verschachtelt, um verschachtelte Codewörter ICi für die jeweilige Anwendung OAi zu erhalten.

[0064] Es ist anzumerken, dass die vorstehend genannten Unterschritte in einer unterschiedlichen Reihenfolge durchgeführt werden können. Beispielsweise kann der vierte Unterschritt auf den zweiten Unterschritt folgen, und der dritte Unterschritt kann auf den vierten Unterschritt folgen.

[0065] In einem dritten Schritt wird, ausgehend von den jeweiligen verschachtelten Codewörtern ICi für jede der N Anwendungen OA1, ..., OAN und optional den Header-Daten (HD) in dem Fall, wenn der obige Eingabezeichenstring auch einen Teil der HD einschließt, die sich auf die jeweilige Anwendung OAi beziehen, die Verkettung TIC von IC1, ..., ICN und optional von HD erhalten.

[0066] In einem ersten Unterschritt des dritten Schritts wird EDTIC optional erhalten, indem ein zweiter EDM angewendet wird, um einen Fehlererkennungsbitstring ED (z. B. einen CRC) zu berechnen, wodurch Fehler in TIC erkannt werden können, und indem der Fehlererkennungsbitstring ED an die ICs angehängt wird, um zu erhalten: EDTIC = TIC | ED.

[0067] In einem zweiten Unterschritt des dritten Schritts wird IEDTIC optional durch Verschachteln von ED und TIC erhalten, um mögliche Fehler zu verteilen. In einem Beispielfall kann durch Verschachteln ein Teil des ED in jedem IC und/oder der gesamte ED in jedem IC verschachtelt werden.

[0068] In einem dritten Unterschritt des dritten Schritts werden optional Paritätsbits (PB) erhalten, indem ein zweiter ECM angewendet wird, um ein oder mehrere 8-Bit-Codewörter über IEDTIC und optional über HD| IEDTIC zu berechnen. Es ist zu beachten, dass der zweite ECM die ICi selbst nicht ändern darf. Beispielsweise kann der zweite ECM aus der Berechnung von Paritätsbits (PB) bestehen, die an IEDTIC angehängt werden können.

[0069] In einem vierten Unterschritt des dritten Schritts werden die vom zweiten ECM berechneten Paritätsbits (PB) optional (z. B. mittels Verschachtelung) auf die jeweiligen verschachtelten Codewörter ICi für jede Anwendung OAi verteilt. Angenommen beispielsweise, es werden k Paritätsbits (PB) erhalten und es gibt zwei Anwendungen OA1 und OA2, dann können die k Paritätsbits auf die Datencontainer verteilt werden, die diesen beiden Anwendungen OA1 und OA2 zugeordnet sind (z. B. k-x Bits für den Datencontainer, der OA1 zugeordnet ist, und x Bits für den Datencontainer, der OA2 zugeordnet ist). In einer Beispielvariante können die k Paritätsbits in beiden Datencontainern gespeichert werden. Diese ausgewählten Paritätsbits (PB), die pro Anwendung in einem Datencontainer gespeichert werden sollen, sollen über die ICi der jeweiligen Anwendung verteilt werden.

[0070] In einer beispielhaften Ausführung kann der ECM pro Anwendung (d. h. der ECM pro Datencontainer) mit einer niedrigeren Fehlerkorrekturstufe konfiguriert sein als der einzelne ECM, der von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzt wird (z. B. der zweite ECM). Beispielsweise könnte aus der Liste der bekannten Fehlerkorrekturstufen L, M, Q, H der ECM pro Anwendung die niedrigste Fehlerkorrekturstufe L aufweisen, während der allen Anwendungen der Vielzahl von Anwendungen gemeinsame ECM die höchste Fehlerkorrekturstufe H aufweisen könnte. In dem Fall, dass der erste QR-Code eine Reed-Solomon-Fehlerkontrollcodierung verwendet, bieten die vier Fehlerkorrekturstufen L, M, Q, H eine jeweilige Wiederherstellungsfähigkeit von ungefähr 7 %, 15 %, 25 %, 30 %. Dies kann ein schnelles Lesen der Anwendungsdaten einer jeweiligen Anwendung ermöglichen, da weniger Anwendungsdaten beteiligt sind.

[0071] In einer optionalen Ausführungsform kann ein Teil oder die Gesamtheit der kodierten Daten gemäß einer Zuordnungsregel in einer Vielzahl von Datenpixeln, die über den ersten QR-Code verteilt sind, genauer gesagt über den Kodierbereich des ersten QR-Codes, verteilt gespeichert sein.

[0072] In einem Beispielfall, in dem die Datenspeicherungsressourcen des ersten QR-Codes begrenzt sind, können die kodierten Daten dann in einer Vielzahl unterschiedlicher QR-Codes gespeichert werden, z. B. im ersten QR-Code und in einem zweiten QR-Code, sodass ein Teil der kodierten Daten im ersten QR-Code und der verbleibende Teil im zweiten QR-Code gespeichert wird, anstatt dass die Gesamtheit der kodierten Daten im ersten QR-Code gespeichert werden. In diesem Fall kann der erste QR-Code als Master-QR-Code ein oder mehrere für den Benutzer verständliche gedruckte Muster einschließen, welche die Position der anderen QR-Codes (z. B. des zweiten QR-Codes) in Bezug auf den Master-QR-Code angeben. In einem Beispiel können die für den Benutzer verständlichen Muster ein oder mehrere (farbige) Pfeile sein, die auf jeden der anderen QR-Codes zeigen (z. B. den zweiten QR-Code), oder können eine Art Schachbrett mit (farbigen oder nummerierten) Rechtecken sein, die den Master-QR-Code und alle anderen QR-Codes kennzeichnen (z. B. den zweiten QR-Code).

[0073] Die Bestimmung dieser Datenpixel zum verteilten Speichern der kodierten Daten über den Kodierbereich des ersten QR-Codes kann gemäß unterschiedlichen Zuordnungsregeln erfolgen. Es ist zu beachten, dass die Header-Daten möglicherweise ähnlich wie die kodierten Daten verteilt sind.

[0074] Unter der Annahme, dass der erste QR-Code die Anwendungsdaten zweier Anwendungen, bezeichnet mit A und B, einschließt und die Datenkapazität gleichmäßig zwischen den jeweiligen Datencontainern für die Anwendungen A und B aufgeteilt ist, kann eine beispielhafte Zuordnungsregel darin bestehen, die codierten Daten der jeweiligen Datencontainer (d. h. nicht nur die codierten Anwendungsdaten, sondern auch die Paritäts-, EDM- oder ECM-Bits) für die Anwendungen A und B in abwechselnden Datenpixeln im (Kodierbereich des) ersten QR-Code zu speichern, wie in **Abb.** 3 gezeigt.

[0075] Unter der Annahme, dass der erste QR-Code die Anwendungsdaten von drei Anwendungen, bezeichnet mit A, B und C, einschließt und die Datenkapazität gleichmäßig zwischen den drei jeweiligen Datencontainern für die Anwendungen A, B und C aufgeteilt ist, kann die beispielhafte Zuordnungsregel

darin bestehen, die kodierten Daten der jeweiligen Datencontainer (d. h. nicht nur die kodierten Anwendungsdaten, sondern auch die Paritäts-, EDM- oder ECM-Bits) für die Anwendungen A, B und C in Datenpixeln (des Kodierbereichs) des ersten QR-Codes zu speichern, wie in **Fig.** 4 gezeigt.

[0076] Diese beispielhafte Zuordnungsregel gewährleistet, dass eine Beschädigung in einem gewissen Teil des QR-Codes nicht die kodierten Daten für eine jeweilige Anwendung oder einen hohen Prozentsatz dieser kodierten Daten beschädigt.

[0077] In einer weiteren beispielhaften Zuordnungsregel können die spezifischen Datenpixel im ersten QR-Code, welche die kodierten Daten der jeweiligen Datencontainer pro Anwendung speichern, beispielsweise wie folgt berechnet werden:

(1) wenn Datenpixel durch ihre kartesischen Koordinaten (i,j) identifiziert werden,

(2) Datenpixel mit Kennung (i,j) im Koordinatensystem (I,J) zur Speicherung der kodierten Daten der jeweiligen Datencontainer pro Anwendung verwendet werden,

(3) der erste QR-Code MxM Datenpixel (oder Elemente) aufweist und

(4) die gleiche Menge von Datenpixeln (oder Elementen) den kodierten Daten für alle Anwendungen zugewiesen wird,

(5) dann können die der Anwendungszahl k zugeordneten kodierten Daten, wobei $0 \leq k \leq N-1$, in den Datenpixeln gespeichert werden, die beispielsweise die folgende Beziehung (1) erfüllen:

$$(i * M + j)(\text{mod } N) = k \text{ mit } (i,j) \text{ in } (I,J) \qquad (1)$$

[0078] Wenn die jeweiligen Anwendungen einen unterschiedlichen Speicherbedarf aufweisen, kann dies bei der Datenpixelzuordnung berücksichtigt werden. Insbesondere wenn N Anwendungen Längenparameter von L1, L2, ..., LN Bits für die kodierten Daten der jeweiligen Datencontainer pro Anwendung erfordern, dann ist es erforderlich, die Längen zu normalisieren, indem alle durch die kleinste geteilt werden und das Ergebnis mit der Funktion RoundUp() aufgerundet wird, wie durch die folgende Beziehung (2) gegeben:

$$NL1, NL2, \ldots, NLN = \text{RoundUp}(L1/Lmin), \text{RoundUp}(L2/Lmin), \ldots, \text{RoundUp}(LN/Lmin) \qquad (2)$$

[0079] Es ist zu beachten, dass die Verwendung der RoundUp()-Funktion bedeutet, dass einigen oder allen entsprechenden Anwendungen möglicherweise einige Dummy-Bits angehängt werden müssen. Der Inhalt der Dummy-Bits könnte beispielsweise so gewählt werden, dass die Fehlerkorrekturmöglichkeiten verbessert werden oder dass die Taktregeneration verbessert wird. Zur Klarstellung: Taktregeneration bedeutet, zu bestimmen, wo das Bild abgetastet werden soll, das einen QR-Code enthält, der möglicherweise (optisch) verzerrt und möglicherweise gedreht ist, um alle Pixel des QR-Codes zu erhalten. Durch die Durchsetzung von Lauflängenbeschränkungen in beide Richtungen des QR-Codes kann die Schätzung der Grenzen zwischen den Datenpixeln des QR-Codes verbessert werden. Daher sollten die Dummy-Bits so gewählt werden, dass lange Lauflängen in beide Richtungen über einem gewissen Schwellenwert verhindert werden, oder noch besser so, dass die Lauflängen in beide Richtungen minimiert werden, z. B. indem in den Dummy-Bitbereichen ein Schachbrettmuster der Datenpixel des QR-Codes erstellt wird. Dies ist besonders hilfreich, wenn der QR-Code auf einer gewölbten Oberfläche gedruckt ist.

[0080] In einer weiteren beispielhaften Zuordnungsregel können die Datenpixel des ersten QR-Codes verwendet werden, um in aufeinanderfolgenden Runden die codierten Daten der jeweiligen Datencontainer für die N verschiedenen Anwendungen (indexiert von 1 bis N) zu speichern:

Runde 1:

die ersten NL1-Bits kodierter Daten für eine erste Anwendung werden in den ersten NL1-Bits des ersten QR-Codes gespeichert,

die ersten NL2-Bits kodierter Daten für eine zweite Anwendung werden in den nachfolgenden NL2-Bits des ersten QR-Codes gespeichert, ...,

die ersten NLN-Bits kodierter Daten für eine N-te Anwendung werden in den nachfolgenden NLN-Bits des ersten QR-Codes gespeichert;

Runde 2:

Die nachfolgenden NL1-Bits kodierter Daten für die erste Anwendung werden in den nachfolgenden NL1-Bits des ersten QR-Codes gespeichert.

die nachfolgenden NL2-Bits kodierter Daten für die zweite Anwendung werden in den nachfolgenden NL2-Bits des ersten QR-Codes gespeichert. ...,

die nachfolgenden NLN-Bits kodierter Daten für die N-te Anwendung werden in den nachfolgenden NLN-Bits des ersten QR-Codes gespeichert. ...;

Runde i:

Die nachfolgenden NL1-Bits kodierter Daten für die erste Anwendung werden in den nachfolgenden NL1-Bits des ersten QR-Codes gespeichert,

die nachfolgenden NL2-Bits kodierter Daten für die zweite Anwendung werden in den nachfolgenden NL2-Bits des ersten QR-Codes gespeichert, ...,

Die nachfolgenden NLN-Bits kodierter Daten für die N-te Anwendung werden in den nachfolgenden NLN-Bits des ersten QR-Codes gespeichert.

[0081] In einer weiteren beispielhaften Zuordnungsregel können die spezifischen Datenpixel im ersten QR-Code, welche die kodierten Daten der jeweiligen Datencontainer pro Anwendung speichern, beispielsweise wie folgt berechnet werden:

(1) wenn Datenpixel durch ihre kartesischen Koordinaten (i,j) identifiziert werden,

(2) Datenpixel mit Kennung (i,j) im Koordinatensystem (I,J) zur Speicherung der kodierten Daten der jeweiligen Datencontainer pro Anwendung verwendet werden,

(3) der QR-Code MxM Pixel (oder Elemente) aufweist,

(4) definieren: SNL_k = NL1 + NL2 + ...+ NLk, wobei k = 1, 2, ..., N und NL1, NL2, ..., NLN die normalisierten Längen der kodierten Daten für die jeweiligen Anwendungen sind, wie durch die Beziehung (2) gegeben,

(5) dann werden, basierend auf SNL_k, Datenpixel (oder Elemente) (i,j) in (I,J) den der Anwendungszahl k zugeordneten kodierten Daten gemäß der folgenden Beziehung (3) zugewiesen:

$$SNL_k \le (i * M + j) (\mathrm{mod}\ SNL_N) < SNL_\{k+1\} \qquad (3)$$

[0082] Es ist zu beachten, dass die Beziehung (2) bedeutet, dass die normalisierte Länge des kleinsten Werts 1 wird. Das obige Schema bezüglich der Beziehung (1) verschachtelt ebenfalls in Gruppen von 1. Es ist zu beachten, dass die Bits auch in Gruppen der Länge k mit k > 1 verschachtelt werden könnten, wobei jedes der k Bits aus einer Anwendung stammt. Die Bits der jeweiligen Anwendung können auf Bereiche, beispielsweise quadratische oder rechteckige Bereiche, im QR-Code verteilt sein. Beispielsweise könnten die Quadrate in den obigen Figuren einzelne Datenpixel oder rechteckige Bereiche darstellen, die Datenpixel einer Anwendung enthalten. Die oben genannten Formeln könnten sich auf Rechtecke i,j mit s Datenpixeln statt nur auf ein einziges Datenpixel beziehen. Es ist auch zu beachten, dass die Bereiche nicht unbedingt rechteckig sein müssen, sondern vielmehr eine andere Form aufweisen können, z. B. dreieckig oder rautenförmig (d. h. eine Raute mit 4 Ecken und Achsen in horizontaler und vertikaler Richtung). Es ist zu beachten, dass bei Rechtecken, die größer als 1 Pixel sind (d. h. 1 x 1), auch die Lauflänge in beiden Dimensionen begrenzt werden kann, wodurch Abtastfehler auf gekrümmten oder zerknitterten Oberflächen minimiert werden.

[0083] Es ist zu beachten, dass ein anderer Weg zum Umgang mit den Anwendungen, die unterschiedliche Längenparameter von L1, ..., LN Bits für die kodierten Daten der jeweiligen Datencontainer pro Anwendung erfordern, wie folgt aussehen kann: Angenommen, g ist der größte gemeinsame Teiler von L1, ..., LN, also die größte ganze Zahl, durch die alle L1, L2, ..., LN geteilt werden können. Der Bitstrom wird in g Teile aufgeteilt, wobei jeder Teil L1/g Bits aus einer ersten Anwendung, L2/g Bits aus einer zweiten Anwendung usw. enthält. In jedem Teil entsprechen die ersten L1/g-Bits der ersten Anwendung, die nächsten L2/g-Bits der zweiten Anwendung und so weiter. Wenn beispielsweise N = 3 und L1 = 20, L2 = 30 und L3 = 40, dann ist g = 10. Jede der 10 Gruppen weist Bits AABBBCCCC auf, wobei A, B und C ein Bit aus der jeweiligen ersten, zweiten und dritten Anwendung angeben. Eine Variante wäre, dass in jedem Teil zuerst ein Symbol aus jeder Anwendung platziert wird, dann ein zweites Symbol aus jeder Anwendung, wobei in jedem Teil mehr als ein Symbol platziert wird, und so weiter. In diesem Fall würde sich statt der obigen Bitfolge AABBBCCCC

die Bitfolge ABCABCBCC ergeben. Der Vorteil dieser anderen Methode besteht darin, dass keine Dummy-Bits durch Aufrunden entstehen.

[0084] In einer beispielhaften Ausführung können die in diesen oben berechneten Datenpixeln zu speichernden kodierten Daten für die jeweiligen Anwendungen einem kombinierten Bitstring entsprechen, wie im Folgenden definiert. Der aus dem fünften Unterschritt des zweiten Schritts abgeleitete Gesamtausgabebitstring IC ist derart, dass er eine Anzahl von ICs enthält, die der Gesamtzahl der verschachtelten Codewörter ICi pro Anwendung OAi (wobei i = 1, 2, ..., N) für alle N Anwendungen entspricht. Im zweiten und dritten Unterschritt des dritten Schritts werden die allen Anwendungen zugeordneten Fehlererkennungsinformationen (EDI) und Fehlerkorrekturinformationen (ECI) auf alle N Anwendungen verteilt. Wenn der IC eine Länge L und der ECI/EDI eine Länge H aufweist, dann ist die Gesamtlänge Z = L + H. Die Bits des ECI/EDI können gleichmäßig in dem Ausgabebitstring IC verteilt werden. Dies kann wie oben erwähnt durch Normalisierung von L und H durch Division durch J = Min(L,H) erfolgen, um so zu erhalten: 1 = RoundUp(L/J) und h = RoundUp(H/J). IC und ECI/EDI werden dann in Blöcke von 1 und h Bits unterteilt. Der kombinierte Bitstring wird durch abwechselndes Verketten von 1-Bitblöcken und h-Bitblöcken von IC und ECI/EDI erhalten. Der in den oben berechneten Datenpixeln zu speichernde kombinierte Bitstring kann dann den IC eines OA und den ECI/EDI des gesamten QR-Codes enthalten. Es ist zu beachten, dass unter der Annahme, dass Fehler gehäuft auftreten, z. B. Fehler aufgrund eines defekten QR-Codes oder aufgrund eines Lochs im QR-Code, die Fehlerkorrekturfähigkeiten eine hohe Wahrscheinlichkeit einer vollständigen Korrektur pro zu wartender Anwendung ermöglichen, da nur ein Bruchteil jeder Anwendung betroffen ist.

[0085] Eine gängige Praxis besteht darin, einen QR-Code mit einer grafischen Darstellung (z. B. Buchstabe, Zahl, Bild, Logo, Symbol, Design, Muster, Modell usw.) zu überlagern, z. B. mit dem Bild eines Firmenlogos auf dem QR-Code. Auf diese Weise können die Benutzer den Zweck des QR-Codes leichter erkennen, z. B. ob es sich um einen Master-QR-Code handelt, indem ein für den Benutzer verständliches gedrucktes Muster, z. B. der Buchstabe M, überlagert wird. Es ist machbar, eine grafische Darstellung (z. B. ein Bild) zu überlagern und trotzdem einen funktionierenden QR-Code zu haben, da QR-Codes auf einem Fehlerkorrekturmechanismus basieren und die Fehlerkorrekturfunktionen daher verwendet werden, um alle durch die überlagerte grafische Darstellung verursachten Fehler zu korrigieren. Dies funktioniert jedoch bis zu einem gewissen Grad, obwohl es einige Einschränkungen gibt. Beispielsweise kann das überlagerte Bild nicht beliebig groß sein, da sonst zu viele Fehler auftreten würden, die nicht korrigiert werden könnten, selbst wenn der stärkste verfügbare Fehlerkorrekturcode angewendet würde. Eine beispielhafte Ausführungsform, die sich mit diesem Problem befasst, bezieht sich auf den ersten QR-Code, der eine Angabe einschließt, dass dem ersten QR-Code in einem gegebenen physischen Bereich des ersten QR-Codes eine grafische Darstellung überlagert ist. Eine Lösung kann dann darin bestehen, die Datenpixel des Kodierbereichs des ersten QR-Codes, die einen Bereich im ersten QR-Code belegen, der mit dem Bereich übereinstimmt, den die grafische Darstellung belegt, von der Datenspeicherung auszuschließen. In einem Beispiel kann die Angabe in den Header-Informationen eingeschlossen sein, die beispielsweise physisch in Datenpixeln in der Nähe von einem, zwei oder drei „Augen“ oder „Positionsmarkierungen“ oder „dem Quadrat innerhalb von Quadraten an drei Ecken“ des ersten QR-Codes gespeichert sein können, d. h. in Datenpixeln, die sich nicht im zentralen Bereich des ersten QR-Codes befinden, wo sich normalerweise das Bild der grafischen Darstellung befindet.

[0086] In einem Beispiel kann der Bereich spezifiziert werden, indem die Datenpixel des ersten QR-Codes, die von der überlagerten grafischen Darstellung betroffen sind, aufgelistet werden, und indem mindestens eines von dem Mittelpunkt des Bereichs, der Bereichsform (z. B. ein Kreis, Quadrat, Rechteck, Dreieck, eine Raute usw.) und der Größe oder des Radius der Bereichsform spezifiziert wird.

[0087] In einem Beispiel kann die Angabe aus einem Bit, also 0 oder 1, bestehen, das angibt, ob die grafische Darstellung vorhanden ist oder nicht, und wenn eine überlagerte grafische Darstellung vorhanden ist, kann die Angabe zusätzlich aus der Form der grafischen Darstellung bestehen, z. B. 0 für einen Kreis und 1 für ein Quadrat, und aus dem (skalierten) Radius r der grafischen Darstellung, der in wenigen Bits, z. B. 3 Bits, kodiert ist. Der Radius kann beispielsweise je nach QR-Code-Version mit einem festen Wert skaliert werden. Beispielsweise kann ein Radius mit dem Wert r mit einem Skalierungsfaktor F multipliziert werden, sodass der tatsächliche Radius der grafischen Darstellung in Pixeln r*F beträgt. Diese beispielhafte Ausführungsform weist mehrere Vorteile auf, wobei einer davon ist, dass größere grafische Darstellungen auf QR-Codes gelegt werden können.

[0088] Es ist zu beachten, dass, wenn der Header das Vorhandensein einer grafischen Darstellung angibt, die Datenpixel im angegebenen physischen Bereich des ersten QR-Codes keine Daten speichern. Somit verwendet der erste QR-Code diese Datenpixel nicht zum Speichern von Daten, und wenn der erste QR-Code

von einem QR-Code-Lesegerät gescannt wird, werden diese Datenpixel vom QR-Code-Lesegerät ignoriert. Dies wirkt sich zwar auf die Menge der kodierten Daten aus, die im ersten QR-Code gespeichert werden können, beschleunigt aber auch den Leseprozess für das QR-Code-Lesegerät. Basierend auf dem Beispiel von **Fig.** 3 zeigt **Fig.** 5, wie bei Vorhandensein eines Bildes die Datenpixel abwechselnd den Anwendungsdaten der Anwendungen A und B zugeordnet werden. Es ist zu beachten, dass diese Technik auch auf QR-Codes anwendbar ist, die eine einzelne Anwendung speichern. Wie zu sehen ist, gibt **Fig.** 5 das Vorhandensein eines Bildes an, dessen Fläche mit der von 25 Datenpixeln belegten Fläche in der Mitte des ersten QR-Codes übereinstimmt, wobei die Mitte des Bildes durch die Buchstabengruppe „CI“ und andere Datenpixel, deren Fläche vom Bild belegt wird, durch den Buchstaben „I“ gekennzeichnet sind. Es ist zu erkennen, dass das Vorhandensein des Bildes nichts an der ursprünglichen Zuordnung der Datenpixel ändert, deren Fläche nicht mit der Fläche des Bildes übereinstimmt, auch wenn dies dazu führen kann, dass zwei „aufeinanderfolgende“ Datenpixel mit dem Bild dazwischen den Anwendungsdaten derselben Anwendung A oder B zugeordnet werden.

[0089] Wie oben erwähnt, wurde ein beispielhaftes Verfahren unter Verwendung des ECM und optional des EDM in jedem Datencontainer beschrieben, um die Informationen der Vielzahl von Anwendungen zur Speicherung im ersten QR-Code in die kodierten Daten zu kodieren.

[0090] In einer beispielhaften Ausführung der vorliegenden Erfindung kann ein QR-Code-Lesegerät diese kodierten Daten, die im ersten QR-Code (d. h. dem Master-QR-Code) gespeichert sind, für jede Anwendung scannen und selektiv lesen. Dies kann durch ein beispielhaftes Verfahren 600 zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen erreicht werden, wobei die Informationen als kodierte Daten im ersten QR-Code gespeichert werden und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen. Das in **Fig.** 6 schematisch beschriebene beispielhafte Verfahren 600 kann mindestens die folgenden Schritte umfassen:

in Schritt 602, Lesen des Headers des ersten QR-Codes, wobei der Header eine einzelne Kennung einschließt, die das Vorhandensein einer Vielzahl von Anwendungen angibt, sowie eine Zählung der Vielzahl von Anwendungen und eine jeweilige Anwendungskennung pro Anwendung;

in Schritt 604, Identifizieren des ersten QR-Codes aus dem Header unter Verwendung der einzelnen Kennung, die das Vorhandensein der Vielzahl von Anwendungen angibt;

in Schritt 606, Identifizieren der jeweiligen Anwendung aus dem Header. Beispielsweise kann das QR-Code-Lesegerät eine vorkonfigurierte Anwendungskennung speichern, die der jeweiligen Anwendungskennung entspricht;

in Schritt 608, Identifizieren des jeweiligen Datencontainers, welcher der jeweiligen Anwendung zugeordnet ist, z. B. basierend auf einer Position, an der die jeweilige Anwendungskennung auf einem Display des QR-Code-Lesegeräts erscheint;

in Schritt 610, Lesen aller Pixel, die dem Datencontainer der jeweiligen Anwendung zugeordnet sind, um mindestens ein Datencodewort (z. B. mit einer Länge von 8 Bit) abzurufen, das der jeweiligen Anwendung sowie den Paritätsbits (PB), Fehlererkennungsinformationsbits (EDM-Bits) oder Fehlerkorrekturinformationsbits (ECM-Bits) für die jeweilige Anwendung zugeordnet ist. Das QR-Code-Lesegerät kann mithilfe des mit der jeweiligen Anwendung verknüpften EDM oder ECM mindestens ein Datencodewort (z. B. mit einer Länge von 8 Bit) lesen, um etwaige Fehler zu erkennen oder zu korrigieren;

in Schritt 612, Erhalten der fehlerfreien Anwendungsdaten D (z. B. Di) der jeweiligen Anwendung (z. B. OAi);

in Schritt 614, wenn der ECM/EDM auch über die Fähigkeit verfügt, zu erkennen, ob alle Fehler korrigiert wurden, dann Erkennen unter Verwendung des ECM/EDM, ob in den abgerufenen Anwendungsdaten D der jeweiligen Anwendung Fehler vorhanden sind. Dies ist möglich, wenn D einen CRC einschließt oder wenn der ECM Fehler erkennen kann, wie dies bei Reed-Solomon-Codes der Fall ist; und

in Schritt 616, wenn ein Fehler erkannt wird, dann Lesen der gesamten Anwendungsdaten aller Datencontainer durch das QR-Code-Lesegerät. Das QR-Code-Lesegerät kann dann die Paritätsbits (PB), Fehlererkennungsinformationsbits (EDM-Bits) oder Fehlerkorrekturinformationsbits (ECM-Bits) für die verbleibenden Anwendungen abrufen (also die Paritätsbits aus den anderen Datencontainern) und diese Bits sowie die Anwendungsdaten in allen Datencontainern verwenden, um zu versuchen, die noch vorhandenen Fehler in der jeweiligen Anwendung von Interesse zu korrigieren.

[0091] Es ist zu beachten, dass das QR-Code-Lesegerät nach dem Lesen der Header-Daten (HD) nur dann ECM/EDM auf den Header anwenden kann, wenn der Header eigene Funktionen zur Fehlererkennung/-korrektur aufweist.

[0092] Es ist zu beachten, dass beim Schritt des Lesens aller dem Datencontainer der jeweiligen Anwendung zugeordneten Pixel und beim Schritt des Erkennens unter Verwendung von ECM/EDM, ob in den abgerufenen Anwendungsdaten D der jeweiligen Anwendung Fehler vorhanden sind, das QR-Code-Lesegerät ECM/EDM auch auf den Header anwenden kann, wenn die Fehlererkennungs-/Korrekturfähigkeiten nicht nur auf die jeweilige Anwendung, sondern auf die Verkettung des Headers und der Anwendungsdaten angewendet werden.

[0093] Es ist zu beachten, dass das QR-Code-Lesegerät beim Schritt des Lesens der Header-Daten (HD) mehrere Kopien des Headers lesen kann (vorausgesetzt, im ersten QR-Code sind mehrere Kopien des Headers verfügbar), um sicherzustellen, dass die Header-Daten (HD) ordnungsgemäß gelesen werden.

[0094] In einer anderen beispielhaften Ausführungsform der vorliegenden Erfindung kann ein QR-Code-Lesegerät diese kodierten Daten, die im ersten QR-Code (d. h. dem Master-QR-Code) gespeichert sind, für jede Anwendung scannen und selektiv lesen. Dies kann durch ein Beispielverfahren 700 zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen erreicht werden, wobei die Informationen als kodierte Daten im ersten QR-Code gespeichert werden und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen. Das in **Fig.** 7 schematisch beschriebene beispielhafte Verfahren 700 kann mindestens die folgenden Schritte umfassen:

in Schritt 702, Identifizieren (z. B. aus einigen Header-Dateninformationen wie einer einzelnen Kennung) des ersten QR-Codes als Master-QR-Code, d. h. als ein QR-Code, der eine Vielzahl von Anwendungen handhabt (z. B. zwei Anwendungen im Zusammenhang mit Easy Connect und einem SHS);

in Schritt 704, Lesen der Symbole im QR-Code und Extrahieren eines Bitstrings aus dem QR-Code nach der Fehlerkorrektur, wobei der Bitstring Daten enthält, die sich auf eine Vielzahl von Anwendungen beziehen; und

in Schritt 706, Lokalisieren eines Teils des Bitstrings, auch als Unterbitstring bezeichnet (d. h. Datencontainer), der sich auf eine erste Anwendung aus dem extrahierten Bitstring bezieht, wobei die Position des Unterbitstrings (d. h. des Datencontainers) aus den Header-Daten oder einer vorkonfigurierten Richtlinie oder einem vorkonfigurierten Kontext bestimmt wird (wenn beispielsweise die beiden Anwendungen Easy Connect und SHS unterschiedliche Semantiken verwenden, könnten sie sich einfach darauf einigen, dass ein Teil des Bitstrings für Easy Connect und der verbleibende Teil für SHS bestimmt ist).

[0095] Diese Implementierung von **Fig.** 7 weist den Vorteil auf, dass gegenüber dem QR-Code-Standard nur wenige Änderungen erforderlich sind.

[0096] In einer anderen beispielhaften Ausführungsform der vorliegenden Erfindung kann ein QR-Code-Lesegerät diese kodierten Daten, die im ersten QR-Code (d. h. dem Master-QR-Code) gespeichert sind, für jede Anwendung scannen und selektiv lesen. Dies kann durch ein beispielhaftes Verfahren 800 zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen erreicht werden, wobei die Informationen als die kodierten Daten im ersten QR-Code gespeichert werden und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen. Das in **Fig.** 8 schematisch beschriebene beispielhafte Verfahren 800 kann mindestens die folgenden Schritte umfassen:

in Schritt 802, Extrahieren eines Strings (z. B. eines Strings, der einer gegebenen Kodierung folgt);

in Schritt 804, Überprüfen, ob der QR-Code eine Vielzahl von Anwendungen enthält (also zwei oder mehr Anwendungen), indem der Header (z. B. die

einzelne Kennung) in dem String betrachtet wird, z. B. die ersten N Zeichen in dem String;

in Schritt 806, wenn der Header übereinstimmt, Zugreifen auf einen Unterstring, der die Informationen der jeweiligen Anwendung enthält.

[0097] Diese Implementierung von **Fig.** 8 weist den Vorteil auf, dass keine Änderung gegenüber dem QR-Code-Standard erforderlich ist.

[0098] Um den ersten QR-Code als Master-QR-Code zu identifizieren, d. h. als einen QR-Code, der eine Vielzahl von Anwendungen handhabt, kann die Person, die das QR-Code-Lesegerät bedient, in einem Beispiel den Master-QR-Code visuell identifizieren, z. B. mittels eines für den Benutzer verständlichen gedruckten Musters (z. B. dem Buchstaben M), das den Master-QR-Code überlagert. In einem anderen Beispiel, wenn die Person, die das QR-Code-Lesegerät bedient, nicht weiß, welcher QR-Code unter einer Vielzahl von QR-Codes der richtige Master-QR-Code ist, und das QR-Code-Lesegerät daher auf die Vielzahl dieser QR-Codes richtet, sodass sie gleichzeitig erfasst werden, dann kann das QR-Code-Lesegerät alle diese QR-Codes identifizieren, das Vorhandensein eines Master-QR-Codes bestimmen, z. B. durch Überwachen eines Unterscheidungsmerkmals (z. B. einiger Pixel), das angibt, dass es sich um einen Master-QR-Code handelt, oder durch Lesen der Header-Daten, um die einzelne Kennung zu identifizieren, die das Vorhandensein einer Vielzahl von Anwendungen angibt.

[0099] **Fig.** 9 zeigt ein schematisches Blockdiagramm 900 einer Vorrichtung 905, die gemäß einer Ausführungsform der vorliegenden Erfindung in der Lage ist, Informationen mehrerer Anwendungen zu kodieren und einen QR-Code 910 zu erstellen. Die Vorrichtung 905 kann eine Rechenvorrichtung sein, z. B. ein Computer, der als Eingabe Informationen aus einer Vielzahl von Anwendungen verwendet und dann die Informationen der Vielzahl von Anwendungen wie in den oben genannten Ausführungsformen mit Bezug auf **Fig.** 2 beschrieben kodiert, um einen ersten QR-Code 910 zu erstellen, d. h. einen QR-Code, der mehrere Anwendungen handhaben kann.

[0100] **Fig.** 10 zeigt ein schematisches System 1000 zur Inbetriebnahme gemäß einer Ausführungsform der vorliegenden Erfindung. Das System 1000 umfasst mindestens eine erste Vorrichtung 1005, der ein erster QR-Code 1010 zugeordnet ist, und ein QR-Code-Lesegerät 1015, das mit einer Kamera 1020 ausgestattet ist. Eine zweite Vorrichtung 1025 und/oder eine dritte Vorrichtung 1055 können ebenfalls vorhanden sein. Der QR-Code 1010 kann der ersten Vorrichtung 1005 auf verschiedene Weisen zugeordnet werden, wie durch physisches Anbringen des QR-Codes 1010 an der ersten Vorrichtung 1005 und/oder Drucken des QR-Codes 1010 auf die erste Vorrichtung 1005 oder auf die Verpackung, in der die erste Vorrichtung 1005 geliefert wird, oder auf jede andere Weise.

[0101] Die erste Vorrichtung 1005, welcher der erste QR-Code 1010 zugeordnet ist, weist eine erste Kommunikationsschnittstelle 1030 auf. Die Kamera 1020 des QR-Code-Lesegeräts 1015 dient zum Scannen und Lesen des ersten QR-Codes 1010. Nach dem Dekodieren des ersten QR-Codes 1010 trifft das QR-Code-Lesegerät 1015 eine Entscheidung über die ausgewählte Anwendung aus dem ersten QR-Code 1010, z. B. basierend auf einer lokalen Richtlinie oder einem Kontext, und auf dem Inbetriebnahmeprotokoll, das mit der ersten Vorrichtung 1005 ausgeführt werden soll. In einer ersten Alternative umfasst das QR-Code-Lesegerät 1015 eine zweite Kommunikationsschnittstelle 1035, die zum Starten und Ausführen eines ersten Inbetriebnahmeprotokolls 1040 mit der ersten Vorrichtung 1005 verwendet wird und die möglicherweise über ein dediziertes Kommunikationsnetzwerk 1045 oder direkt zwischen den Kommunikationsschnittstellen 1035 und 1060 mit der dritten Vorrichtung 1055, d. h. einer zusätzlichen Netzwerkvorrichtung 1055, die beispielsweise ein Wi-Fi-Zugangspunkt (AP) oder eine Gateway-Vorrichtung sein kann, über eine Schnittstelle verbunden ist. Das dedizierte Kommunikationsnetzwerk 1045 kann beispielsweise ein lokales Netzwerk (LAN) wie Ethernet, Wi-Fi, Zigbee usw. oder ein Weitverkehrsnetz (WAN) wie ein Mobilfunknetz oder das Internet oder eine beliebige Kombination von Netzwerken sein. Das erste Protokoll kann ein Protokoll zum Einrichten einer (sicheren) Kommunikation zwischen der ersten Vorrichtung 1005 und der zusätzlichen Netzwerkvorrichtung 1055 sein. In einer zweiten Alternative löst das QR-Code-Lesegerät 1015 ein zweites Inbetriebnahmeprotokoll 1050 aus, das entweder mit dem ersten Inbetriebnahmeprotokoll identisch ist oder sich von diesem unterscheidet, zwischen der ersten Vorrichtung 1005 und der zweiten Vorrichtung 1025, wobei letztere beispielsweise ein Server eines Cloud-Dienstes sein kann, oder zwischen einer „Administrator“-Vorrichtung über die dritte Vorrichtung 1055, d. h. eine zusätzliche Netzwerkvorrichtung 1055, einschließlich einer dritten Kommunikationsschnittstelle 1060, und dem Kommunikationsnetzwerk 1045.

[0102] Eine dritte Alternative kann eine Art Kombination der beiden vorherigen Alternativen sein, wobei das QR-Code-Lesegerät 1015 basierend auf Informationen im ersten QR-Code 1010 das erste Inbetriebnahmeprotokoll auslöst, um die (sichere) Kommunikation zwischen der ersten Vorrichtung 1005 und der dritten Vorrichtung 1055 einzurichten, die beispielsweise ein Wi-Fi-AP oder eine Gateway-Vorrichtung sein kann, und das zweite Inbetriebnahmeprotokoll auslöst, das entweder mit dem ersten Inbetriebnahmeprotokoll identisch ist oder sich von diesem unterscheidet, um die (sichere) Kommunikation über die dritte Vorrichtung 1055 und das Netzwerk 1045 zwischen der ersten Vorrichtung 1005 und der zweiten Vorrichtung 1025 einzurichten, wobei letztere beispielsweise ein Server eines Cloud-Dienstes oder eine „Administrator“-Vorrichtung sein kann.

[0103] **Fig.** 11 zeigt ein Inbetriebnahmeprotokoll 1100 gemäß einer herkömmlichen Ausführungsform zwischen einer ersten Vorrichtung 1110, umfassend eine erste Kommunikationsschnittstelle 1120, und einer zweiten Vorrichtung 1130, umfassend eine zweite Kommunikationsschnittstelle 1140. Die erste Vorrichtung 1110 kann ihre Anwesenheit mittels Inbetriebnahmenachrichten 1150 in Bezug auf das Inbetriebnahmeprotokoll 1100 ankündigen. Die zweite Vorrichtung 1130 startet den Inbetriebnahmeablauf des Inbetriebnahmeprotokolls 1100 durch Rücksenden einer Nachricht 1160.

[0104] **Fig.** 12 zeigt ein Inbetriebnahmeprotokoll 1200 gemäß einer Ausführungsform der vorliegenden Erfindung zwischen einer ersten Vorrichtung 1210, umfassend eine erste Kommunikationsschnittstelle 1220, und einer zweiten Vorrichtung 1230, umfassend eine zweite Kommunikationsschnittstelle 1240. Im Gegensatz zur ersten Vorrichtung 1110 von **Fig.** 11 ist die erste Vorrichtung 1210 von **Fig.** 12 mit einem angehängten ersten QR-Code 1250 versehen. Dieser erste QR-Code 1250 enthält Inbetriebnahmeinformationen für zwei unterschiedliche Inbetriebnahmeprotokolle, die über dieselbe (erste) Kommunikationsschnittstelle 1220 der ersten Vorrichtung 1210 laufen können. Die erste Vorrichtung 1210 kann ihre Anwesenheit sowie ihre Inbetriebnahmefähigkeiten mittels Inbetriebnahmenachrichten 1260 und 1270 ankündigen, die sich auf ein jeweiliges Inbetriebnahmeprotokoll der beiden Inbetriebnahmeprotokolle beziehen. Nach dem Dekodieren des ersten QR-Codes 1250, der an der ersten Vorrichtung 1210 angebracht ist, startet die zweite Vorrichtung 1230 ein ausgewähltes Inbetriebnahmeprotokoll unter den beiden Inbetriebnahmeprotokollen, indem sie eine Nachricht 1280 zurücksendet.

[0105] Bei einem vorrichtungsgesteuerten Konfigurationsverfahren wie dem Device Provisioning Protocol (DPP), das in Netzwerken mit Wi-Fi oder IEEE 802.11 verwendet wird, kann eine als DPP-Konfigurator fungierende Vorrichtung jede Wi-Fi-fähige Vorrichtung, die als angemeldetes Gerät fungiert, sicher konfigurieren, um beispielsweise eine Verbindung mit einem Wi-Fi-Zugangspunkt (AP) herzustellen. In der vorliegenden Erfindung ist der erste QR-Code ein Master-QR-Code, der in der Lage ist, die Informationen einer Vielzahl von Anwendungen zu speichern. In einer beispielhaften Ausführungsform kann eine entsprechende Wi-Fi-Verbindung zwischen einer Vielzahl von Konfiguratoren und der ersten Wi-Fi-fähigen Vorrichtung hergestellt werden, die als angemeldetes Gerät fungiert und an die ein erster QR-Code angehängt ist. Dadurch kann der erste QR-Code DPP-Bootstrapping-Informationen, z. B. einen öffentlichen Schlüssel in verschlüsselter Form, der ersten Wi-Fi-fähigen Vorrichtung enthalten, wobei jeder Konfigurator während des Bootstrapping-Prozesses den ersten QR-Code der ersten Wi-Fi-fähigen Vorrichtung scannt, um seinen jeweiligen öffentlichen Schlüssel abzurufen.

[0106] Bei DPP beginnt der Prozess mit den beiden Vorrichtungen 1210, 1230 in einem nicht verbundenen Zustand und eine der Vorrichtungen ist nicht für die Verbindung mit dem betreffenden Wi-Fi-Netzwerk konfiguriert. Eine Vorrichtung 1230 wird verwendet, um die andere Vorrichtung 1210 für den Beitritt zum drahtlosen Netzwerk zu konfigurieren, d. h. für die Verbindung mit dem Netzwerkzugangspunkt (AP), und kann als Konfigurator (oder Inbetriebnehmer) und die zweite Vorrichtung als angemeldetes Gerät (oder in Betrieb genommenes Gerät) bezeichnet werden.

[0107] Die erste Phase ist „Bootstrapping“, bei der eine Vorrichtung den öffentlichen Bootstrapping-Schlüssel (BR) einer anderen Vorrichtung zum Konfigurieren der Vorrichtung erhält. Dies geschieht auf andere Weise als über die drahtlose Kommunikationstechnologie und wird daher allgemein als „Out-of-Band“-Kommunikation (OOB) bezeichnet, wobei der Benutzer bewirkt, dass eine Vorrichtung einen QR-Code auf der zweiten Vorrichtung liest. Wenn das Bootstrapping erfolgreich war, werden die Vorrichtungen „urgeladen“, andernfalls kehren sie in den Zustand „Start“ zurück. Häufig sind einfache oder sogenannte Headless-Vorrichtungen (also solche mit wenig oder keiner Benutzeroberfläche) so programmiert, dass sie nach dem Einschalten oder Zurücksetzen aufwachen und ihren Funk auf dem in ihrem QR-Code angegebenen Kanal zur Konfiguration einschalten und auf eine Authentifizierungsanforderungsnachricht warten. (Eine nicht-einfache Vorrichtung, z. B. ein Smartphone oder ein Laptop, würde über seine Benutzeroberfläche in diesen Modus versetzt werden, wenn der Benutzer eine Konfiguration wünscht.)

[0108] Die Vorrichtungen führen ein Authentifizierungsverfahren durch, bei dem die Vorrichtungen „Vertrauen“ herstellen, d. h. der Benutzer kann sicher sein, dass es sich bei den Vorrichtungen um die handelt, für die er sie hält, und nicht um andere unbekannte (und möglicherweise bösartige) Vorrichtungen, die „vorgeben“, eine der fraglichen Vorrichtungen zu sein. Von einer Vorrichtung wird eine Nachricht mit der Aufforderung gesendet, eine Authentifizierung zu starten. Diese Nachricht kann entweder von der Vorrichtung gesendet werden, welche die Konfiguration durchführt (Konfigurator), oder von der zu konfigurierenden Vorrichtung, dem angemeldeten Gerät. Der Konfigurator/Inbetriebnehmer kann mit dem Wi-Fi-Netzwerk verbunden sein, obwohl dies für das Funktionieren der Ausführungsformen nicht unbedingt erforderlich ist. Die Vorrichtung,

welche die drahtlose Kommunikation startet, wird als Initiator bezeichnet, und die Vorrichtung, die antwortet, wird als Responder bezeichnet. Insbesondere das DPP-Protokoll ermöglicht es sowohl einem Konfigurator als auch einem angemeldeten Gerät, als Initiator des DPP-Protokolls zu fungieren, wobei die andere Vorrichtung automatisch zum Responder wird. Einfache Vorrichtungen oder kopflose Vorrichtungen übernehmen normalerweise die Responder-Rolle.

[0109] Die andere Vorrichtung antwortet auf diese Nachricht. Wenn die Authentifizierungsanforderungsnachricht korrekt dekodiert wird und die Information enthält, dass es sich beim Initiator um die Vorrichtung handelt, für die der Benutzer sie hält, und dass sie über die erforderlichen Fähigkeiten verfügt, zeigt die Antwortnachricht an, dass die Nachricht „akzeptiert“ wurde und Informationen enthält, die der Initiator benötigt, um die Anmeldeinformationen des Responders zu überprüfen, und dass dieser ebenfalls über die erforderlichen Fähigkeiten verfügt. Wenn die beiden Vorrichtungen nicht die erforderlichen Informationen von der jeweils anderen Vorrichtung empfangen, wird der Vorgang abgebrochen und die Vorrichtung kehrt in den Bootstrap-Zustand zurück.

[0110] Im Falle des DPP-Protokolls ist die erste Nachricht die Authentifizierungsanforderungsnachricht und die Antwortnachricht die DPP-Authentifizierungsantwort. Der Responder prüft, ob die DPP-Authentifizierungsanforderungsnachricht einen korrekt generierten kryptografischen Hash des öffentlichen Bootstrapping-Schlüssels des Responders enthält und optional, ob er eine Kopie des öffentlichen Bootstrapping-Schlüssels des Initiators aufweist. Der Responder sendet eine DPP-Authentifizierungsantwortnachricht, die angibt, ob die Authentifizierung fortgesetzt werden kann. Wenn nicht, weil beispielsweise der Versuch, einen verschlüsselten Nonce in der DPP-Authentifizierungsanforderungsnachricht zu entschlüsseln, fehlschlägt, wird der Vorgang abgebrochen. Die DPP-Authentifizierungsantwort enthält einen kryptografischen Hash des öffentlichen Bootstrapping-Schlüssels des Responders und kann den Hash des öffentlichen Bootstrapping-Schlüssels des Initiators enthalten. Ebenso kann es sein, dass das angemeldete Gerät diesen öffentlichen Schlüssel durch eine OOB-Kommunikation erhalten hat. Der öffentliche Bootstrapping-Schlüssel des Initiators kann dann zur gegenseitigen Authentifizierung verwendet werden. Ohne den öffentlichen Bootstrapping-Schlüssel des Initiators kann nur der Initiator das angemeldete Gerät authentifizieren, nicht jedoch umgekehrt.

[0111] Wenn die Authentifizierungsantwortnachricht angibt, dass der Responder die Authentifizierungsanforderungsnachricht akzeptiert hat und die Antwort die durch die Einrichtung des Initiators auferlegten Kriterien erfüllt, gibt der Initiator eine Authentifizierungsbestätigungsnachricht aus. Wenn die Authentifizierungswerte in der Authentifizierungsantwort und der Bestätigungsnachricht von den entsprechenden Vorrichtungen als korrekt befunden werden, ist dieser Teil des Protokolls, der Authentifizierungsteil, erfolgreich und die Konfiguration kann beginnen. Die Bestätigungsnachricht kann auch eine Angabe zum Ergebnis eines vorherigen Konfigurationsversuchs enthalten, wobei das angemeldete Gerät auch der Initiator ist.

[0112] Im Fall des DPP-Protokolls ist die Authentifizierungsbestätigungsnachricht die DPP-Authentifizierungsbestätigungsnachricht.

[0113] Anschließend sendet die angemeldete Vorrichtung eine Konfigurationsanforderungsnachricht mit Informationen darüber, welche Art von Konfiguration das angemeldete Gerät wünscht. Wenn der Konfigurator der Anforderung stattgeben kann, sendet er eine Nachricht mit den vom angemeldeten Gerät benötigten Informationen, wie einen Netzwerkschlüssel. Der Prozess wird dann mit der erfolgreichen Konfiguration des angemeldeten Geräts abgeschlossen.

[0114] Im Fall von DPP ist die Anforderungsnachricht die DPP-Konfigurationsanforderung und die Antwort des Konfigurators die DPP-Konfigurationsantwortnachricht. Die DPP-Konfigurationsantwort kann den Service Set Identifier (SSID) des Netzwerks enthalten, mit dem sich das angemeldete Gerät verbinden soll, und kann einen DPP-Connector enthalten. Der DPP-Connector kann als vom Konfigurator digital signiertes Zertifikat betrachtet werden und enthält unter anderem den öffentlichen Netzwerkzugangsschlüssel des angemeldeten Geräts. Die DPP-Konfigurationsantwortnachricht enthält auch den öffentlichen Signaturschlüssel des Konfigurators. Andere Vorrichtungen, die vom gleichen Konfigurator konfiguriert wurden, können dadurch prüfen, ob sie dem öffentlichen Netzwerkzugangsschlüssel anderer Vorrichtungen vertrauen können. Die DPP-Konfigurationsantwortnachricht kann auch die Wi-Fi-Passphrase oder den Pre Shared Key (PSK) des Netzwerks enthalten. Das angemeldete Gerät sendet eine DPP-Konfigurationsergebnisnachricht (abhängig von der DPP-Version) an den Konfigurator, um ihm mitzuteilen, ob es die Konfiguration akzeptiert oder nicht. Wenn der Konfigurator diese Nachricht nicht erhält, kann dies für den Konfigurator ein Hinweis darauf sein, dass zwischen dem Konfigurator und dem angemeldeten Gerät ein Wi-Fi-Problem aufgetreten ist. Anschließend kann das „angeblich konfigurierte“ angemeldete Gerät seinen Connector an einen DPP-konfigurierten AP

senden. Wenn die Signatur des Connectors korrekt ist und der AP einen passenden Connector aufweist, d. h. einen Connector für dasselbe Netzwerk, der vom gleichen Konfigurator signiert wurde, sendet der AP seinen eigenen Connector an das angemeldete Gerät. Das angemeldete Gerät und der AP können auf Grundlage des Netzwerkzugriffsschlüssels des jeweils anderen in den Connectors und ihres eigenen privaten Netzwerkzugangsschlüssels im Diffie-Hellman-Verfahren einen symmetrischen Schlüssel berechnen.

[0115] Falls das angemeldete Gerät ein Wi-Fi-Passwort oder einen Wi-Fi-Pre-Shared-Key (PSK) empfangen hat, versucht das angemeldete Gerät auf normalem Weg über den 4-Wege-Handshake, wie im IEEE 802.11-Standard angegeben, eine Verbindung mit dem AP herzustellen. Ein Protokoll zum Konfigurieren oder Inbetriebnehmen einer Vorrichtung für den Beitritt zu einem SHS-Netzwerk kann einen Schlüsselaustausch umfassen, wobei der Inbetriebnahmevorrichtung ein Code bereitgestellt wird. Dieser Code, der als Passwort fungiert, wird durch ein Out-of-Band-Verfahren bereitgestellt. Im vorliegenden Beispiel kann dieser Passcode in den Teil des QR-Codes der angemeldeten Vorrichtung integriert werden, der sich auf die SHS-Inbetriebnahmeanwendung bezieht.

[0116] In einem Beispielfall der vorliegenden Erfindung kann eine Vorrichtung, z. B. eine drahtlose Vorrichtung oder eine drahtlose Wi-Fi-Vorrichtung, in der Lage sein, einem Wi-Fi-Netzwerk mittels DPP oder einem SHS-Netzwerk mittels des SHS-Bootstrapping-Protokolls beizutreten. In diesem Fall kann die Vorrichtung einen ersten QR-Code wie in der vorliegenden Erfindung definiert aufweisen, der Informationen in Bezug auf DPP und das SHS-Bootstrapping-Protokoll speichern kann. In einer beispielhaften Ausführungsform implementiert ein Konfigurator oder Inbetriebnehmer ein QR-Code-Lesegerät (oder kommuniziert damit), das den ersten an einer Vorrichtung angebrachten QR-Code liest, ein DPP- oder SHS-Bootstrapping-Protokoll basierend auf mindestens einem von Folgendem auslösend: (1) einer Vorkonfiguration im QR-Code-Lesegerät, (2) einer im QR-Code-Lesegerät verfügbaren Richtlinie, und (3) den kontextuellen Anforderungen, die in der Umgebung verfügbar sind, in der die Vorrichtung eingesetzt wird. Somit können die für diese beispielhafte Ausführungsform erforderlichen Schritte Folgendes umfassen:

a) Lesen des ersten QR-Codes;

b) Ermitteln der unterstützten Anwendungen aus dem gelesenen ersten QR-Code, insbesondere der unterstützten Inbetriebnahmeprotokolle;

c) Auswählen einer Anwendung (z. B. einer bevorzugten Anwendung) basierend auf mindestens einem von: (1) einer vorkonfigurierten Auswahl, (2) einer Richtlinie, (3) einem Kontext und (4) einer Präferenz, die im ersten QR-Code eingeschlossen sind. Beispielsweise kann das QR-Code-Lesegerät ein QR-Code-Lesegerät für die DPP-basierte Inbetriebnahme sein und nur in der Lage sein, DPP auszuführen. Beispielsweise kann die Richtlinie die Verwendung einer Anwendung bevorzugen, wenn diese verfügbar ist, und diese verwenden, wenn sie unterstützt wird (angegeben durch den ersten QR-Code). Beispielsweise kann das QR-Code-Lesegerät erkennen, ob der Kontext oder die Umgebung (z. B. ein Zuhause oder ein Krankenhaus), in der die Vorrichtung eingesetzt werden soll, nur eine bestimmte Anwendung unterstützt (z. B. nur einen hochsicheren DPP-basierten Wi-Fi-Konfigurationsprozess) und kann dann auf dieser Grundlage eine Anwendungsauswahl treffen. Die Erkennung kann beispielsweise auf dem Vorhandensein oder Inhalt der von den Vorrichtungen übertragenen Beacons basieren. Beispielsweise kann ein erster QR-Code ein Feld einschließen, das dem QR-Code-Lesegerät je nach Kontext die bevorzugte Anwendungsauswahl angibt; und

d) Starten der ausgewählten Anwendung (z. B. der ausgewählten bevorzugten Anwendung) als erste ausgewählte Anwendung. Im Falle einer Inbetriebnahmeanwendung sollte der Konfigurator oder Inbetriebnehmer eine zweite ausgewählte Anwendung nicht starten, solange die erste ausgewählte Anwendung nicht erfolgreich war oder fehlgeschlagen ist, z. B. solange der mit der ersten ausgewählten Anwendung verbundene Inbetriebnahmeprozess eine Mindestanzahl N-mal nicht erfolgreich war oder fehlgeschlagen ist, z. B. N = 1, 2, ...

[0117] In einer beispielhaften Ausführungsform der vorliegenden Erfindung kann ein erster QR-Code (möglicherweise an einem Produkt oder einer Vorrichtung angebracht) wie in der vorliegenden Erfindung definiert Informationen enthalten, die sich auf zwei oder mehr Anwendungen beziehen. Ein QR-Code-Lesegerät, das ihn liest, kann beide Anwendungen basierend auf einer Vorkonfiguration, einer Richtlinie oder einem Kontext ausführen. Die Ausführung beider Anwendungen kann sequenziell erfolgen, z. B. in einer durch eine Richtlinie definierte Reihenfolge. Die Ausführung einer der Anwendungen kann auch teilweise erfolgen (z. B. in dem Fall, in dem ein zweites Inbetriebnahmeprotokoll auf einem ersten Inbetriebnahmeprotokoll (DPP) als Zwischenschritt basiert). Die Anwendungen oder unterschiedlichen Phasen der Anwendungen können auch mehrfach ausgeführt werden. Als ein weiteres Beispiel, kann, wenn der erste QR-Code an einer Vorrichtung

angebracht ist, z. B. einer drahtlosen Vorrichtung oder einer drahtlosen Wi-Fi-Vorrichtung, die Vorrichtung möglicherweise einem Smart-Home-System (SHS) beitreten, indem sie sich auf Inbetriebnahmeprotokolle (1) der zugrunde liegenden drahtlosen Netzwerktechnologie, z. B. Wi-Fi, und (2) des SHS, verlässt. In diesem Fall kann die Vorrichtung den ersten QR-Code aufweisen, wie in der vorliegenden Erfindung definiert, der in der Lage ist, Informationen im Zusammenhang mit der drahtlosen Netzwerktechnologie und dem Bootstrapping-Protokoll des SHS zu speichern. In einer beispielhaften Ausführungsform implementiert ein Konfigurator oder Inbetriebnehmer ein QR-Code-Lesegerät (oder kommuniziert damit), das den ersten an einer Vorrichtung angebrachten QR-Code liest, ein DPP-Bootstrapping-Protokoll und ein SHS-Bootstrapping-Protokoll basierend auf mindestens einem der Folgenden auslösend: (1) einer Vorkonfiguration im QR-Code-Lesegerät, (2) einer im QR-Code-Lesegerät verfügbaren Richtlinie und (3) kontextbezogenen Anforderungen, die in der Umgebung verfügbar sind, in der die Vorrichtung eingesetzt wird. Somit können die für diese beispielhafte Ausführungsform erforderlichen Schritte Folgendes umfassen:

a) Lesen des ersten QR-Codes;

b) Ermitteln der unterstützten Anwendungen aus dem gelesenen ersten QR-Code, insbesondere der unterstützten Inbetriebnahmeprotokolle;

c) Auswählen einer Anwendungsausführungsreihenfolge (z. B. einer bevorzugten Anwendungsausführungsreihenfolge) basierend auf mindestens einem von: (1) einer vorkonfigurierten Auswahl, (2) einer Richtlinie, (3) einem Kontext und (4) einer Präferenz, die im ersten QR-Code eingeschlossen sind. Beispielsweise kann das QR-Code-Lesegerät ein QR-Code-Lesegerät für das SHS sein, das die Interoperabilität mit DPP-basierten Wi-Fi-Netzwerken ermöglicht. Beispielsweise kann das QR-Code-Lesegerät erkennen, ob der Kontext oder die Umgebung (z. B. ein Zuhause oder ein Krankenhaus), in der die Vorrichtung eingesetzt werden soll, beide Anwendungen erfordert (z. B. einen sicheren DPP-basierten Wi-Fi-Konfigurationsprozess und einen SHS-Konfigurationsprozess) und kann dann auf dieser Grundlage eine Anwendungsauswahl treffen;

d) Starten einer ersten Anwendung. Im Falle einer Inbetriebnahmeanwendung sollte der Konfigurator oder Inbetriebnehmer eine zweite ausgewählte Anwendung nicht starten, solange die erste ausgewählte Anwendung nicht erfolgreich war oder fehlgeschlagen ist, mindestens bis zu einem vorkonfigurierten Zielschritt, z. B. solange der mit der ersten ausgewählten Anwendung verbundene Inbetriebnahmeprozess mindestens N-mal nicht erfolgreich war oder fehlgeschlagen ist, z. B. N = 1, 2, ...;

e) nach erfolgreicher (möglicherweise teilweiser im Fall, dass sich ein zweites Inbetriebnahmeprotokoll auf ein erstes Inbetriebnahmeprotokoll (DPP) als Zwischenschritt verlassen kann) Ausführung der ersten Anwendung, Starten einer zweiten Anwendung, z. B. im Fall eines Inbetriebnahmeprotokolls, eines zweiten Inbetriebnahmeprotokolls, das eine weitere Konfiguration für die Vorrichtung bereitstellt, um sie an das Netzwerk anzuschließen; und

f) optional, nach (nicht) erfolgreicher Ausführung von Schritt e) und der zweiten Anwendung, Rückkehr zur ersten Anwendung, um diese zu beenden.

[0118] Diese beispielhafte Ausführungsform wird anhand von **Fig.** 13 veranschaulicht, die ein Inbetriebnahmeprotokoll 1300 zwischen einer ersten Vorrichtung 1310, die einen angebrachten ersten QR-Code 1330 und eine erste Kommunikationsschnittstelle 1340 umfasst, und einer zweiten Vorrichtung 1320, die eine zweite Kommunikationsschnittstelle 1350 und eine Kamera 1360 zum Scannen des QR-Codes umfasst, zeigt. Die zweite Vorrichtung 1320 scannt den ersten QR-Code 1330 und entscheidet, basierend auf beispielsweise einer Richtlinie, die Inbetriebnahmeschritte 1371, 1372, 1373, die zu den im ersten QR-Code 1330 verfügbaren ersten und zweiten Anwendungen gehören, über die Kommunikationsschnittstellen 1350 und 1340 mit der ersten Vorrichtung 1310 auszuführen. In einem beispielhaften Szenario kann sich Schritt 1371 auf ein erstes Inbetriebnahmeprotokoll beziehen, Schritt 1372 kann sich auf ein zweites Inbetriebnahmeprotokoll beziehen, wenn Schritt 1371 ein gewisses Stadium erreicht, und Schritt 1373 kann sich nach (nicht) erfolgreicher Ausführung von Schritt 1372 erneut auf ein erstes Inbetriebnahmeprotokoll beziehen und die Ausführung von Schritt 1371 fortsetzen.

[0119] Wenn in der vorliegenden Erfindung eine Vorrichtung, z. B. ein Wi-Fi-Produkt, einen ersten angehängten QR-Code aufweist, der z. B. zwei Anwendungen enthält, wie eine DPP-Anwendung und eine SHS-Anwendung, dann kann die Inbetriebnahmevorrichtung sich auch dafür entscheiden, Wi-Fi Easy Connect für die Wi-Fi-Inbetriebnahme und das SHS-Protokoll zum Verbinden der Vorrichtung, an die der erste QR-Code angehängt ist, mit den Cloud-Diensten und allen anderen höherwertigen Dienstleistungen zu verwenden.

[0120] Wenn in der vorliegenden Erfindung eine Vorrichtung, z. B. ein Wi-Fi-Produkt, einen ersten angehängten QR-Code aufweist, der z. B. zwei Anwendungen enthält, wie eine DPP-Anwendung und eine SHS-Anwendung, dann muss die Vorrichtung, an die der erste QR-Code angehängt ist, beim Lesen des ersten QR-Codes möglicherweise auf die Anfrage des Konfigurators (z. B. DPP) oder Inbetriebnehmers (z. B. Matter) reagieren. Sobald die Vorrichtung eine erste Anfrage für eine erste Anwendung empfängt, die im ersten QR-Code identifiziert wird, sollte die Vorrichtung keine nachfolgende Anfrage für eine zweite Anwendung akzeptieren, die im ersten QR-Code enthalten ist, solange die erste Anwendung nicht abgeschlossen ist.

[0121] Wenn eine Vorrichtung in der vorliegenden Erfindung eine gegebene Anwendung unterstützt, kann der erste an die Vorrichtung angehängte QR-Code, der Informationen über die Anwendung einschließt, angeben, ob diese Informationen gesendet werden (im Klartext oder nicht) oder nicht. Beispielsweise können einige Felder aus Datenschutzgründen möglicherweise nicht gesendet werden (stattdessen wird ein 0-Bitstring gesendet), dies kann jedoch den Inbetriebnehmer (ausgelöst durch das erste QR-Lesegerät) daran hindern, eine Inbetriebnahmenachricht von der richtigen Vorrichtung auszuwählen, z. B. eine Nachricht, die den Service Set Identifier (SSID) der richtigen Vorrichtung enthält, wenn die Vorrichtung in einen bestimmten Konfigurationszustand wechselt, z. B. indem sie ein Soft Access Point (softAP) wird. Alternativ kann der erste QR-Code eine zufällige Bitmaske einschließen, z. B. eine zufällige Bitmaske, die pro Vorrichtung eindeutig ist und die zum XOR (Verschlüsseln) einiger der im ersten QR-Code eingeschlossenen Felder vor der Übertragung verwendet wird, z. B. wenn sie in einem Beacon eingeschlossen ist oder Teil der SSID der Vorrichtung ist. Eine Vorrichtung, die diese Informationen, z. B. eine zufällige Bitmaske, aus dem QR-Code liest, weiß dann, wie die in den von der Vorrichtung empfangenen Wireless Beacons enthaltenen Informationen zu entschlüsseln sind.

[0122] Zusammenfassend bezieht sich die vorliegende Erfindung auf einen QR-Code für mehrere Anwendungen, gekennzeichnet durch einen ersten oder Master-QR-Code, der in der Lage ist, die Informationen einer Vielzahl von Anwendungen so zu speichern, dass für jede Anwendung die jeweiligen Informationen zuverlässig und zeiteffizient abgerufen werden können. Die Informationen werden in verschlüsselte Daten kodiert, die pro Anwendung mindestens einen Header (z. B. eine einzelne Kennung) und einen entsprechenden Datencontainer (z. B. einen Teil eines Bitstrings oder Unter-Bitstrings in einem vollständigen Bitstring) einschließen. Der Header kann mindestens eine einzelne Kennung umfassen, die das Vorhandensein der Vielzahl von Anwendungen angibt, und eine jeweilige Anwendungskennung pro Anwendung. In jedem Datencontainer kann pro Anwendung ein entsprechender Fehlerkorrekturmechanismus eingeschlossen sein. Die kodierten Daten werden gemäß einer Zuordnungsregel verteilt in einer Vielzahl von Datenpixeln gespeichert, die über einen Kodierbereich des ersten/Master-QR-Codes verteilt sind. Ein QR-Code-Lesegerät kann die im QR-Code gespeicherten Informationen abrufen, indem es ausschließlich auf die der jeweiligen Anwendung zugeordneten Datenpixel zugreift und diese verarbeitet und eine für diese jeweilige Anwendung spezifische Fehlerkorrektur anwendet.

[0123] Auch wenn die Erfindung im Detail in den Zeichnungen und der vorangehenden Beschreibung veranschaulicht und beschrieben wurde, sind eine derartige Veranschaulichung und Beschreibung als veranschaulichend oder beispielhaft und nicht als einschränkend zu betrachten. Darüber hinaus kann die Erfindung in jedem Produkt angewendet werden, das eine Netzwerkschnittstelle (z. B. Wi-Fi, Bluetooth Low Energy oder ein anderer Typ) oder ein Inbetriebnahmeprotokoll (z. B. Wi-Fi Easy Connect oder ein Smart-Home-System wie Matter) implementiert. Das QR-Code-Lesegerät kann direkt mit dem Produkt kommunizieren, indem es dieselbe vom Produkt implementierte Netzwerkschnittstelle verwendet, oder indirekt, z. B. über einen Access Point (AP) oder ein Gateway.

[0124] In ähnlicher Weise kann ein System 1400 zum Lesen und Verarbeiten mit einem einzigen Scannen eines QR-Codes gemäß einer Ausführungsform der vorliegenden Erfindung eine personenbezogene Daten-App (wie sie beispielsweise auf einem Mobiltelefon verwendet wird) einschließen, mit der ein Benutzer personenbezogene Daten anfordern, speichern und anzeigen kann, die einer einzelnen Person oder mehreren Personen zugeordnet sind. Beispiele für solche Daten sind Impfberichte, Infektionstests oder Zertifikate über die Genesung von einer besonderen Infektion.

[0125] Fig. 14 zeigt das System 1400, das mindestens einen personenbezogenen Bericht einschließt, der mittels eines QR-Codes 1410 kodiert ist und mittels einer personenbezogenen Daten-App (1431) angezeigt wird, die auf einem mobilen QR-Code-Lesegerät 1420 (z. B. einem Mobiltelefon) läuft. Der QR-Code 1410 kann mittels des QR-Code-Lesegeräts 1420 mit einer Kamera 1430 gescannt werden. Eine erste Verwaltungseinheit 1440 ist für die Übermittlung personenbezogener Berichte zuständig, z. B. Berichte über den Status einer Person im Hinblick auf eine Krankheit, nach Identifizierung/Authentifizierung einer Person

anhand gegebener Identifikationsdaten. Diese Verwaltungseinheit verwaltet möglicherweise auch geheimes Schlüsselmaterial, z. B. einen privaten Schlüssel zum Signieren der personenbezogenen Berichte. Optional kann eine zweite Verwaltungseinheit 1450 für die Überprüfung der personenbezogenen Berichte zuständig sein, z. B. durch Überprüfen der digitalen Signatur auf dem personenbezogenen Bericht oder durch Abgleichen der mehreren Berichte mit einer Richtlinie. Die zweite Verwaltungseinheit 1450 und die personenbezogene Daten-App 1431 interagieren, um die personenbezogenen Berichte von der ersten Verwaltungseinheit 1440 an die personenbezogene Daten-App 1431 zu übermitteln. Die personenbezogene Daten-App 1431 könnte als Teil der zweiten Verwaltungseinheit 1450 im QR-Code-Lesegerät 1420 implementiert sein, oder das QR-Code-Lesegerät 1420 könnte die abgerufenen personenbezogenen Berichte zur Überprüfung an die Verwaltungseinheit 1450 weiterleiten.

[0126] Mittels eines QR-Codes für mehrere Anwendungen können die personenbezogenen Daten einer oder mehrerer Personen kodiert und in der mobilen App gespeichert werden.

[0127] Sofern es sich bei den personenbezogenen Daten um Daten amtlicher Natur handelt, wie die oben genannten Impf-/Genesungs-/Testberichte, kann eine Verwaltungseinheit mit der Aufgabe betraut werden, die personenbezogenen Datensätze für die betreffende(n) Person(en) nach Identifizierung/Authentifizierung der Person(en) unter Verwendung eines Identifikationsmerkmals anzulegen, d. h. entsprechend der ersten Verwaltungseinheit 1440. Diese Verwaltungseinheit 1440 verwaltet möglicherweise auch geheimes Schlüsselmaterial, z. B. einen privaten Schlüssel, um die personenbezogenen Datenberichte zu signieren. In einem solchen Fall kann es eine zweite Verwaltungseinheit (also die oben erwähnte zweite Verwaltungseinheit 1450) geben, die für die Überprüfung der Berichte zu personenbezogenen Daten zuständig ist, z. B. durch Überprüfung der digitalen Signatur auf dem Bericht oder durch Abgleich der mehreren Berichte mit einer Richtlinie.

[0128] Personal, das für die zweite Verwaltungseinheit 1450 tätig ist, kann ein QR-Code-Lesegerät 1420 verwenden (das z. B. als mobile App läuft), um die personenbezogenen Datenberichte 1410 zu lesen und Richtlinien basierend auf den Ergebnissen des Lesens der personenbezogenen Datenberichte umzusetzen. Solche Richtlinien können, ohne jegliche Einschränkung, offizielle Vorschriften wie Zugangs- oder Durchgangsbeschränkungen einschließen.

[0129] Ein QR-Code, der einen mit einer einzelnen Person verknüpften Bericht über personenbezogene Daten enthält, kann eine Gesamtsignatur einschließen, die es dem QR-Code-Lesegerät (oder der Überprüfungsstelle, welche die aus dem QR-Code gelesenen Daten mithilfe des QR-Code-Lesegeräts abruft und überprüft) ermöglicht, zu prüfen, ob alle Berichte über personenbezogene Daten miteinander verknüpft sind.

[0130] Um dies zu erreichen, muss jeder Bericht mit personenbezogenen Daten, wie oben beschrieben, in einen unterschiedlichen Datencontainer aufgenommen werden. Darüber hinaus wird eine Gesamtsignatur beispielsweise in einem letzten Datencontainer oder als Teil des Headers eingefügt.

[0131] Ein QR-Code, der personenbezogene Datenberichte einschließt, die mehreren Personen zugeordnet sind, kann eine Gesamtsignatur einschließen, die es dem QR-Code-Lesegerät (oder der Überprüfungsstelle personenbezogener Daten, welche die aus dem QR-Code gelesenen Daten mithilfe des QR-Code-Lesegeräts abruft und überprüft) ermöglicht, zu prüfen, ob alle personenbezogenen Datenberichte miteinander verknüpft sind.

[0132] Um dies zu erreichen, können sich mehrere Benutzer über dieselbe personenbezogene Daten-App identifizieren/authentifizieren, beispielsweise indem sie ihre offiziellen Anmeldeinformationen verwenden. Jedes Mal, wenn sich ein Benutzer über dieselbe App identifiziert/authentifiziert, wird ein neuer QR-Code erstellt, der die personenbezogenen Datenberichte für alle Benutzer einschließt, die bisher in dieser personenbezogenen Daten-App identifiziert/authentifiziert wurden. Jeder Bericht zu personenbezogenen Daten ist in einem unterschiedlichen Datencontainer eingeschlossen. Eine von der für die Zertifizierung der Covid-19-Meldungen (Impfung/Genesung/Test) zuständigen Zertifizierungsstelle generierte Gesamtsignatur wird angehängt und beispielsweise in einen letzten Datencontainer eingefügt, der die Überprüfung ermöglicht, dass alle personenbezogenen Berichte im QR-Code zusammen gelesen werden können, also beispielsweise zu einer Gruppe von Personen gehören, die zusammen leben.

[0133] Wenn ein QR-Code personenbezogene Datenberichte für Inlands- oder Auslandsreisen einschließt, kann das QR-Code-Lesegerät die bevorzugte Anwendung kontextbezogen, beispielsweise basierend auf dem Standort, automatisch auswählen. Der QR-Code könnte beispielsweise im Header das Heimatland bein-

halten. Anschließend kann das QR-Code-Lesegerät basierend auf dem Standort des QR-Code-Lesegeräts einen spezifischen Datencontainer auswählen (z. B. einen personenbezogenen Datenbericht zur Verwendung in einer spezifischen Region).

[0134] Wenn ein QR-Code mehrere personenbezogene Datenberichte einschließt, die mit regulären Impfungen verknüpft sind, akzeptiert das QR-Code-Lesegerät die personenbezogenen Datenberichte möglicherweise nur dann, wenn sie aktuell sind (innerhalb eines gegebenen Zeitraums ausgestellt) und der personenbezogene Datenbericht einer von der zweiten Verwaltungsstelle festgelegten Richtlinie entspricht. Ein Beispiel für eine solche Richtlinie könnte beispielsweise lauten, dass die meisten Impfungen in den letzten X Monaten (z. B. in den letzten 3 Monaten) verabreicht wurden oder dass keine Person in der (Familien-) Gruppe zugelassen werden sollte, wenn eines der Mitglieder keinen positiven Genesungsbericht aufweist.

[0135] Wenn ein QR-Code personenbezogene Datenberichte einschließt, die mehreren Personen zugeordnet sind, z. B. Familienmitgliedern, wird erwartet, dass diese Familienmitglieder ihre Identität gemeinsam bestätigen, indem sie ihre Ausweise unter Verwendung eines einzigen QR-Codes vorzeigen. Daher wird ein QR-Code für mehrere Anwendungen, der personenbezogene Datenberichte einschließt, die mehreren Personen zugeordnet sind, vom QR-Code-Lesegerät möglicherweise nur auf der Grundlage einer komplexen Richtlinie akzeptiert, z. B.:

> kann ein QR-Code für mehrere Anwendungen vom QR-Code-Lesegerät nur dann akzeptiert werden, wenn mindestens 2 Personen mit gültigen Ausweisen anwesend sind.

[0136] Die gruppenbezogene Verarbeitung personenbezogener Daten von Gruppen stellt erhebliche Vorteile dar. Erstens ergibt sich eine erhebliche Zeit- und Aufwandsersparnis, wenn das Personal der zweiten Verwaltungsstelle die personenbezogenen Datenberichte liest und prüft. Dieses Personal steht häufig unter Druck, die Personen schnell abzufertigen, was aufgrund der räumlichen Gegebenheiten oft schwierig sein kann. Mit einem einzigen Lese- und Prüfvorgang kann eine ganze Gruppe überprüft und zum Fortfahren autorisiert werden (oder nicht) - das ist viel schneller und weniger fehleranfällig, als jeden QR-Code einzeln verarbeiten zu müssen. Darüber hinaus erschwert das Verfahren gemäß dieser Ausführungsform den Missbrauch der personenbezogenen Datenberichte, indem eine zweite Zertifizierungsebene vorhanden ist: es erfolgt eine Zertifizierung der einzelnen Container und eine Zertifizierung des gesamten QR-Codes.

[0137] Weitere Variationen der offenbarten Ausführungsformen können durch Fachleute, welche die beanspruchte Erfindung umsetzen, anhand eines Studiums der Zeichnungen, der Offenbarung und der beiliegenden Ansprüche gefolgert und bewirkt werden. In den Ansprüchen schließt das Wort „umfassen“ andere Elemente oder Schritte nicht aus, und der unbestimmte Artikel „ein“ oder „eine“ schließt eine Mehrzahl nicht aus. Ein einzelner Prozessor oder eine weitere Einheit kann die Funktionen mehrerer in den Ansprüchen angegebener Punkte erfüllen. Die bloße Tatsache, dass gewisse Maße in voneinander unterschiedlichen abhängigen Ansprüchen angegeben sind, gibt nicht an, dass eine Kombination dieser Maße nicht vorteilhaft verwendet werden kann.

[0138] Die vorstehende Beschreibung beschreibt gewisse Ausführungsformen der Erfindung im Detail. Es ist jedoch offensichtlich, dass die Erfindung, ungeachtet dessen, wie detailliert das Vorstehende im Text erscheint, auf viele Arten umgesetzt werden kann und daher nicht auf die offenbarten Ausführungsformen beschränkt ist. Es ist zu beachten, dass die Verwendung einer besonderen Terminologie bei der Beschreibung gewisser Merkmale oder Gesichtspunkte der Erfindung nicht so zu verstehen ist, dass die Terminologie hierin so neu definiert wird, dass sie auf spezifische Eigenschaften der Merkmale oder Gesichtspunkte der Erfindung beschränkt ist, die dieser Terminologie zugeordnet sind.

[0139] Eine einzelne Einheit oder Vorrichtung kann die Funktionen mehrerer in den Ansprüchen genannter Elemente erfüllen. Die bloße Tatsache, dass gewisse Maße in voneinander unterschiedlichen abhängigen Ansprüchen angegeben sind, gibt nicht an, dass eine Kombination dieser Maße nicht vorteilhaft verwendet werden kann.

[0140] Die beschriebenen Vorgänge können als Programmcodemittel eines Computerprogramms und/oder als dedizierte Hardware implementiert werden. Das Computerprogramm kann auf einem geeigneten Medium, wie einem optischen Speichermedium oder einem Festkörpermedium, das zusammen mit oder als Teil anderer Hardware geliefert wird, gespeichert und/oder verteilt werden, kann jedoch auch in anderen Formen verteilt werden, wie über das Internet oder andere drahtgebundene oder drahtlose Telekommunikationssysteme.

Erwähnte Dokumente:

[DPP] Device Provisioning Protocol - Technical Specification - Version 2.0, Wi-Fi Alliance, 2020,

(https://www.wi-fi.org/downloads-public/Wi-Fi_Easy_Connect_Specification_v2.0.pdf/35330)

[ISO 18004] ISO/IEC 18004:2015: Informationstechnik - Auomatische Identifikation und Datenerfassungsverfahren - Spezifikation der Barcode - Symbologie „QR Code"

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- Device Provisioning Protocol - Technical Specification - Version 2.0, Wi-Fi Alliance, 2020 [0140]
- https://www.wi-fi.org/downloads-public/Wi-Fi_Easy_Connect_Specification_v2.0.pdf/35330 [0140]
- ISO/IEC 18004:2015: Informationstechnik - Auomatische Identifikation und Datenerfassungsverfahren - Spezifikation der Barcode - Symbologie „QR Code" [0140]

Patentansprüche

1. Verfahren zum Kodieren von Informationen einer Vielzahl von Anwendungen zur Speicherung in einem QR-Code, wobei das Verfahren mindestens Folgendes umfasst:
- Sammeln der Informationen der Vielzahl von Anwendungen; und
- Kodieren der Informationen in kodierte Daten, wobei die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen, wobei:
- der Header mindestens eine einzelne Kennung, die das Vorhandensein der Vielzahl von Anwendungen angibt, und eine jeweilige Anwendungskennung pro Anwendung umfasst.

2. Verfahren nach Anspruch 1, wobei für jeden Datencontainer pro Anwendung in den codierten Daten des jeweiligen Datencontainers pro Anwendung ein jeweiliger Fehlerkorrektur- und/oder Fehlererkennungsmechanismus verwendet wird.

3. Verfahren nach Anspruch 1 oder 2, wobei das Kodieren der Informationen in kodierte Daten mindestens Folgendes umfasst:
- Umwandeln eines Eingangsdatenstroms der Informationen in einen jeweiligen Eingangsbitstring für jede Anwendung;
- Aufteilen des jeweiligen Eingangsbitstrings für jede Anwendung in eine Folge von einem oder mehreren Datencodewörtern;
- Unterteilen der Folge eines oder mehrerer Datencodewörter in eine vordefinierte Anzahl von Blöcken;
- Erzeugen eines oder mehrerer Fehlerkorrekturcodewörter für jeden Block; und
- Einschließen der Fehlerkorrekturcodewörter in die Datencodewortfolge.

4. Verfahren nach Anspruch 1 oder 2, wobei das Kodieren der Informationen in kodierte Daten mindestens Folgendes umfasst:
- Entnehmen eines jeweiligen Datenstroms aus dem Header für jede Anwendung;
- Kombinieren der jeweiligen Datenströme in einen kombinierten jeweiligen Datenstrom;
- Umwandeln des kombinierten jeweiligen Datenstroms in einen jeweiligen Eingangsbitstring;
- Aufteilen des jeweiligen Eingangsbitstrings in eine Folge von einem oder mehreren Datencodewörtern;
- Unterteilen der Folge eines oder mehrerer Datencodewörter in eine vordefinierte Anzahl von Blöcken;
- Erzeugen eines oder mehrerer Fehlerkorrekturcodewörter für jeden Block; und
- Einschließen der Fehlerkorrekturcodewörter in die Datencodewortfolge.

5. Verfahren nach einem der Ansprüche 1 bis 4, ferner umfassend:
- verteiltes Speichern eines Teils oder der Gesamtheit der kodierten Daten in einer Vielzahl von Datenpixeln, die über einen Kodierbereich des QR-Codes verteilt sind, gemäß einer Zuordnungsregel.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei in jedem Datencontainer pro Anwendung Daten eingeschlossen sind, die für die Verwendung mit einem jeweiligen Fehlererkennungsmechanismus pro Anwendung angeordnet sind.

7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die kodierten Daten einschließen:
- Daten, die zur Verwendung mit einem von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzten Gesamtfehlererkennungsmechanismus angeordnet sind, um basierend auf Codewörtern in anderen Datencontainern zu erkennen, ob im jeweiligen Datencontainer Fehler vorhanden sind; und/oder
- Daten, die für die Verwendung mit einem von allen Anwendungen der Vielzahl von Anwendungen gemeinsam genutzten Gesamtfehlerkorrekturmechanismus zum Korrigieren von Fehlern im jeweiligen Datencontainer basierend auf Codewörtern in anderen Datencontainern angeordnet sind.

8. Verfahren nach Anspruch 5, ferner umfassend:
- Sammeln von Bilddaten aus einer grafischen Darstellung, die physisch über den QR-Code gelegt wird; und
- Einschließen, in den Header der kodierten Daten, einer Angabe über das Vorhandensein der grafischen Darstellung und einen von der grafischen Darstellung belegten Bereich auf dem QR-Code, wobei die Datenpixel des Kodierbereichs des QR-Codes, die einen Bereich im QR-Code belegen, der mit dem von der grafischen Darstellung belegten Bereich übereinstimmt, vom Speichern des Teils oder der Gesamtheit der kodierten Daten ausgeschlossen sind.

9. Verfahren nach einem der Ansprüche 1 bis 8, wobei der QR-Code DPP-Bootstrapping-Informationen zum Speichern von Informationen mindestens eines öffentlichen Schlüssels umfasst.

10. Verfahren zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen in einem QR-Code, wobei die Informationen als kodierte Daten im QR-Code gespeichert sind und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen, wobei das Verfahren mindestens umfasst:
- Lesen eines Headers des QR-Codes, wobei der Header mindestens eine einzelne Kennung einschließt, die das Vorhandensein der Vielzahl von Anwendungen und eine jeweilige Anwendungskennung pro Anwendung angibt;
- Identifizieren, aus dem Header, des QR-Codes unter Verwendung der einzelnen Kennung, die das Vorhandensein der Vielzahl von Anwendungen angibt;
- Identifizieren, aus dem Header, der jeweiligen Anwendung; und
- Identifizieren des jeweiligen Datencontainers, welcher der jeweiligen Anwendung zugeordnet ist.

11. Verfahren nach Anspruch 10, ferner umfassend:
- Lesen aller Pixel, die dem Datencontainer der jeweiligen Anwendung zugeordnet sind, um mindestens ein der jeweiligen Anwendung zugeordnetes Datencodewort und Paritäts- oder Fehlererkennungs- oder Fehlerkorrekturbits für die gesamte Vielzahl von Anwendungen abzurufen; und
- Erhalten fehlerfreier Anwendungsdaten der jeweiligen Anwendung.

12. Verfahren nach Anspruch 10 oder 11, wobei die Informationen Inbetriebnahmeinformationen für ein Kommunikationsprotokoll kodieren, wobei das Verfahren ferner umfasst:
- Auswählen eines ersten Inbetriebnahmeprotokolls basierend auf mindestens einem von einem vorkonfigurierten Wert, einer Richtlinie, einem Kontext und einem Teil der Daten im QR-Code.

13. Einrichtung zum Kodieren von Informationen einer Vielzahl von Anwendungen zur Speicherung in einem QR-Code, wobei die Einrichtung ausgestaltet ist, um das Verfahren nach einem der Ansprüche 1 bis 9 durchzuführen.

14. Einrichtung zum Dekodieren von Informationen einer jeweiligen Anwendung unter einer Vielzahl von Anwendungen in einem QR-Code, wobei die Informationen als kodierte Daten im QR-Code gespeichert sind und die kodierten Daten mindestens einen Header und einen jeweiligen Datencontainer pro Anwendung einschließen, wobei die Einrichtung ausgestaltet ist, um das Verfahren nach einem der Ansprüche 10 bis 12 durchzuführen.

15. System, mindestens umfassend:
- die Einrichtung nach Anspruch 14, umfassend eine Kamera und eine erste Kommunikationsschnittstelle;
- einen QR-Code mit einer Vielzahl von Containern, die jeweiligen personenbezogenen Datenberichten jeweiliger Personen zugeordnet sind, wobei die jeweiligen Personen zu einer Gruppe gehören;
eine zweite Vorrichtung, die eine zweite Kommunikationsschnittstelle aufweist, wobei die zweite Vorrichtung eine Datendatei aufweist, die den Datenberichten der jeweiligen Personen zugeordnet ist,
wobei die Einrichtung ausgestaltet ist zum: Verbinden mit der zweiten Vorrichtung über eine Schnittstelle; und
wobei die Einrichtung und die zweite Vorrichtung angeordnet sind, um ein Verifizierungsprotokoll zu starten und auszuführen, wobei das Verifizierungsprotokoll das Lesen und Dekodieren des QR-Codes, das Abrufen der Datendatei und das Vergleichen des Inhalts der Datendatei mit Daten, die durch das Dekodieren der Vielzahl von Behältern erhalten wurden, und das Bereitstellen einer Verifizierungsantwort umfasst.

16. Computerprogrammprodukt für eine Rechenvorrichtung, wobei das Computerprogrammprodukt umfasst:
- Programmanweisungen oder Codemittel, sodass, wenn das Computerprogrammprodukt auf einer Verarbeitungseinheit der Rechenvorrichtung läuft, die Rechenvorrichtung angeordnet ist, das Verfahren nach einem der Ansprüche 1 bis 9 durchzuführen.

17. Computerprogrammprodukt für eine Rechenvorrichtung, wobei das Computerprogrammprodukt umfasst:
- Programmanweisungen oder Codemittel, sodass, wenn das Computerprogrammprodukt auf einer Verar-

beitungseinheit der Rechenvorrichtung läuft, die Rechenvorrichtung angeordnet ist, das Verfahren nach einem der Ansprüche 10 bis 12 durchzuführen.

Es folgen 10 Seiten Zeichnungen

Anhängende Zeichnungen

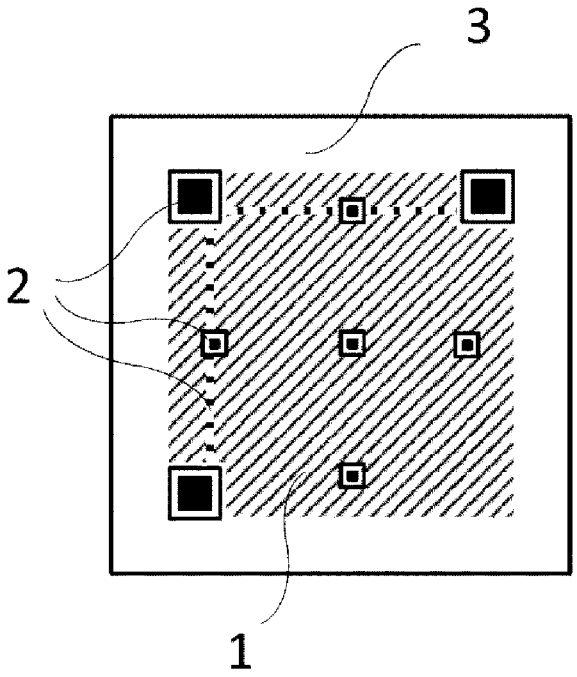


Fig. 1

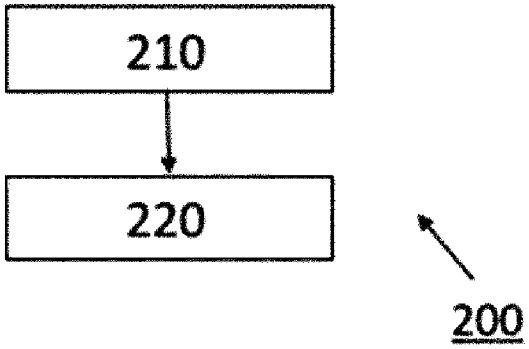


Fig. 2

…	…	…	…	…	…	…
…	A	B	A	B	A	…
…	B	A	B	A	B	…
…	A	B	A	B	A	…
…	B	A	B	A	B	…
…	A	B	A	B	A	…
…	…	…	…	…	…	…

Fig. 3

…	…	…	…	…	…	…
…	A	B	C	A	B	…
…	B	C	A	B	C	…
…	C	A	B	C	A	…
…	A	B	C	A	B	…
…	B	C	A	B	C	…
…	…	…	…	…	…	…

Fig. 4

...	...	...	...	...	...	...	...	...	...	...
...	A	B	A	B	A	B	A	B	A	...
...	B	A	B	A	B	A	B	A	B	...
...	A	B	I	I	I	I	I	B	A	...
...	B	A	I	I	I	I	I	A	B	...
...	A	B	I	I	Cl	I	I	B	A	...
...	B	A	I	I	I	I	I	A	B	...
...	A	B	I	I	I	I	I	B	A	...
...	B	A	B	A	B	A	B	A	B	...
...	A	B	A	B	A	B	A	B	A	...
...	...	...	...	...	...	...	...	...	...	...

Fig. 5

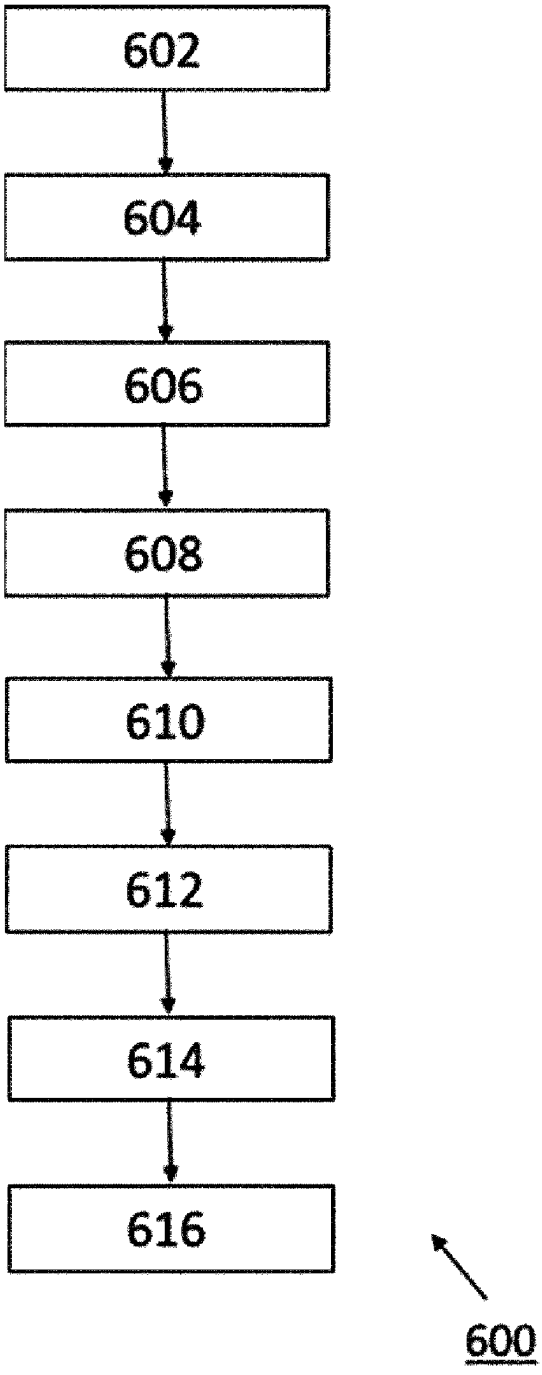


Fig. 6

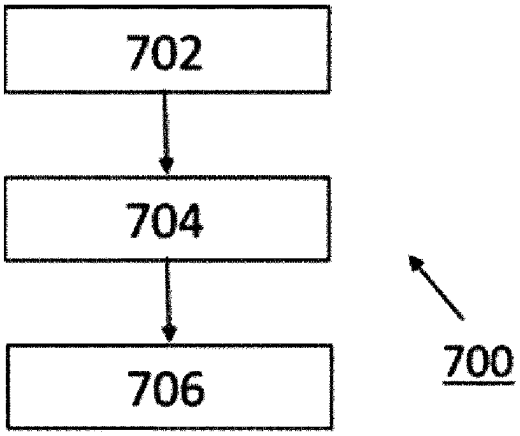


Fig. 7

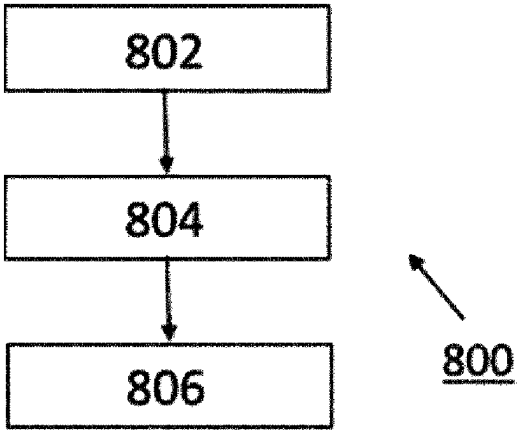


Fig. 8

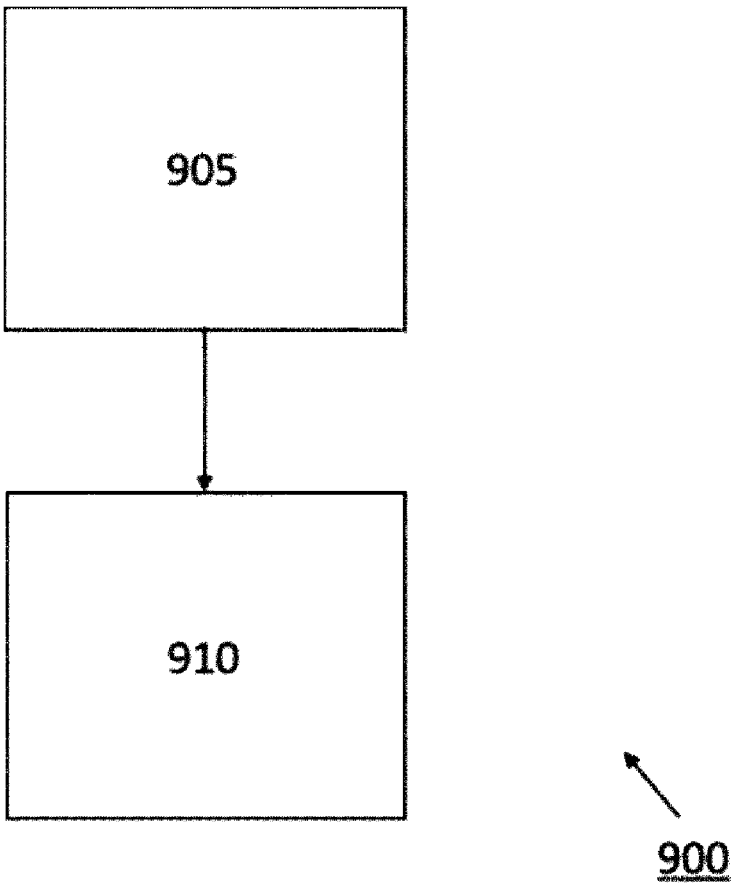


Fig. 9

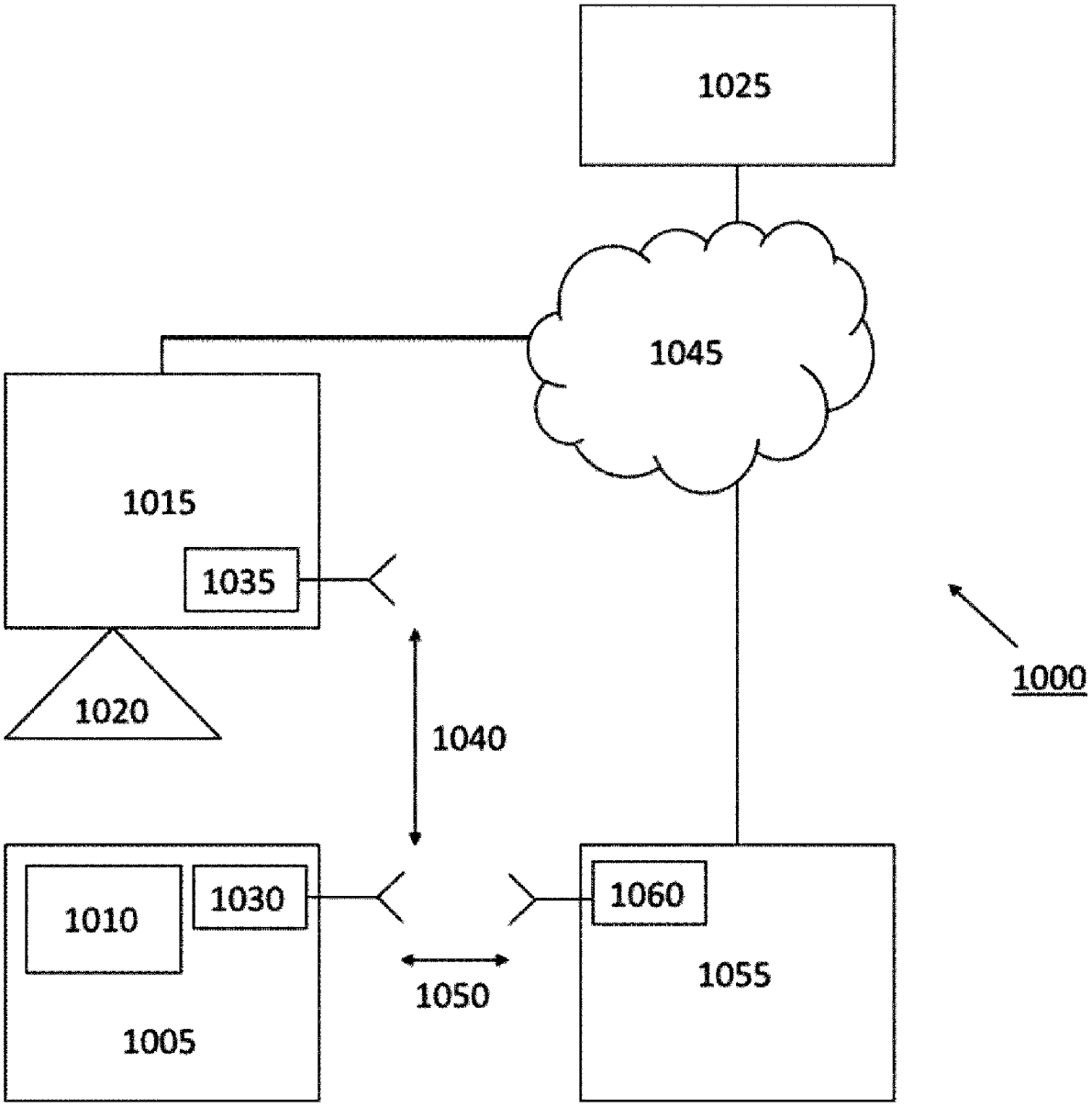

1025
1045
1015
1035
1020
1040
1000
1010
1030
1060
1055
1050
1005

Fig. 10

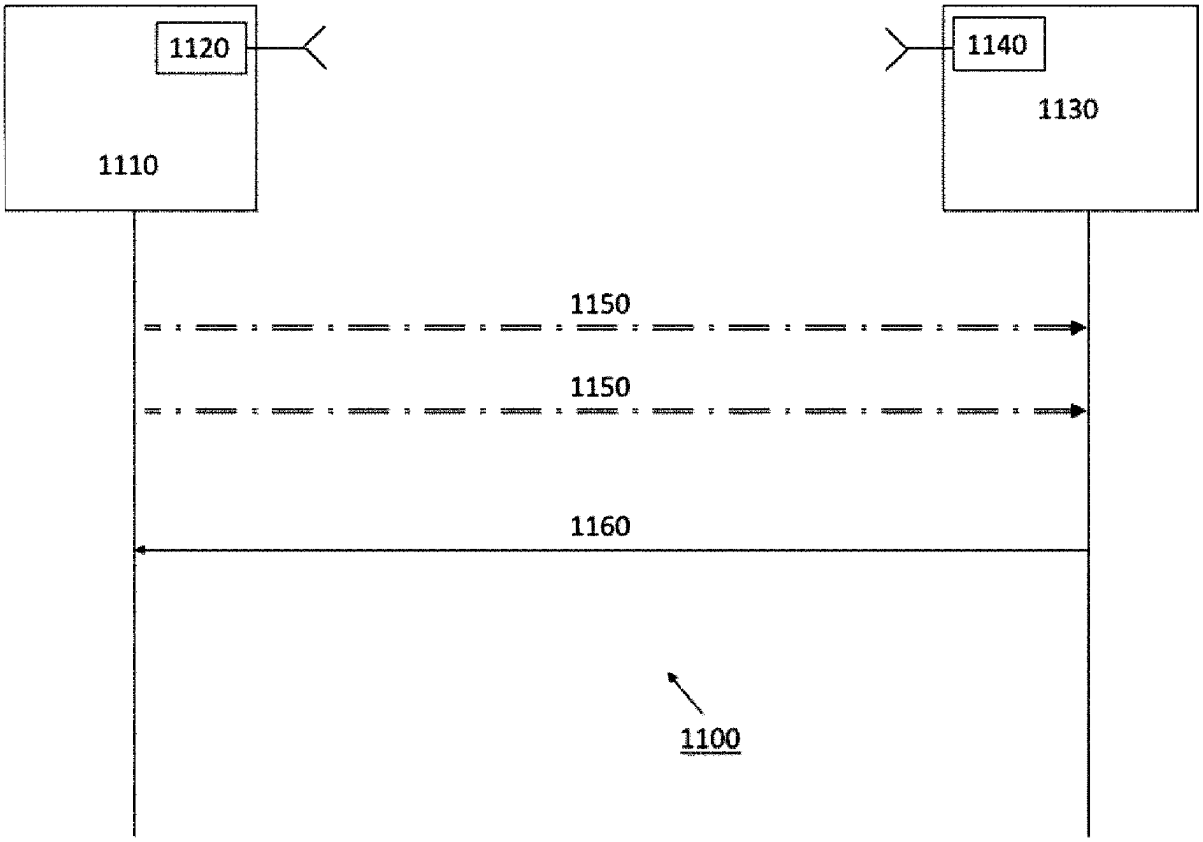


Fig. 11

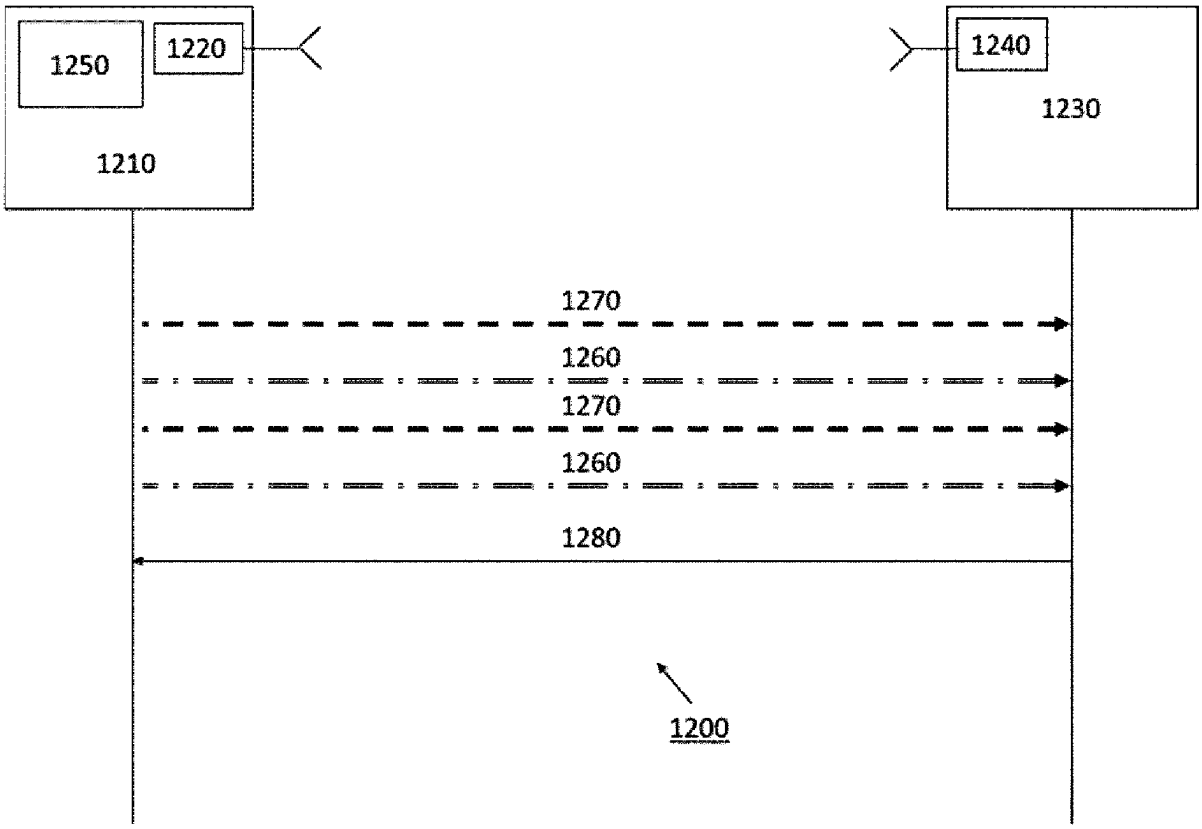


Fig. 12

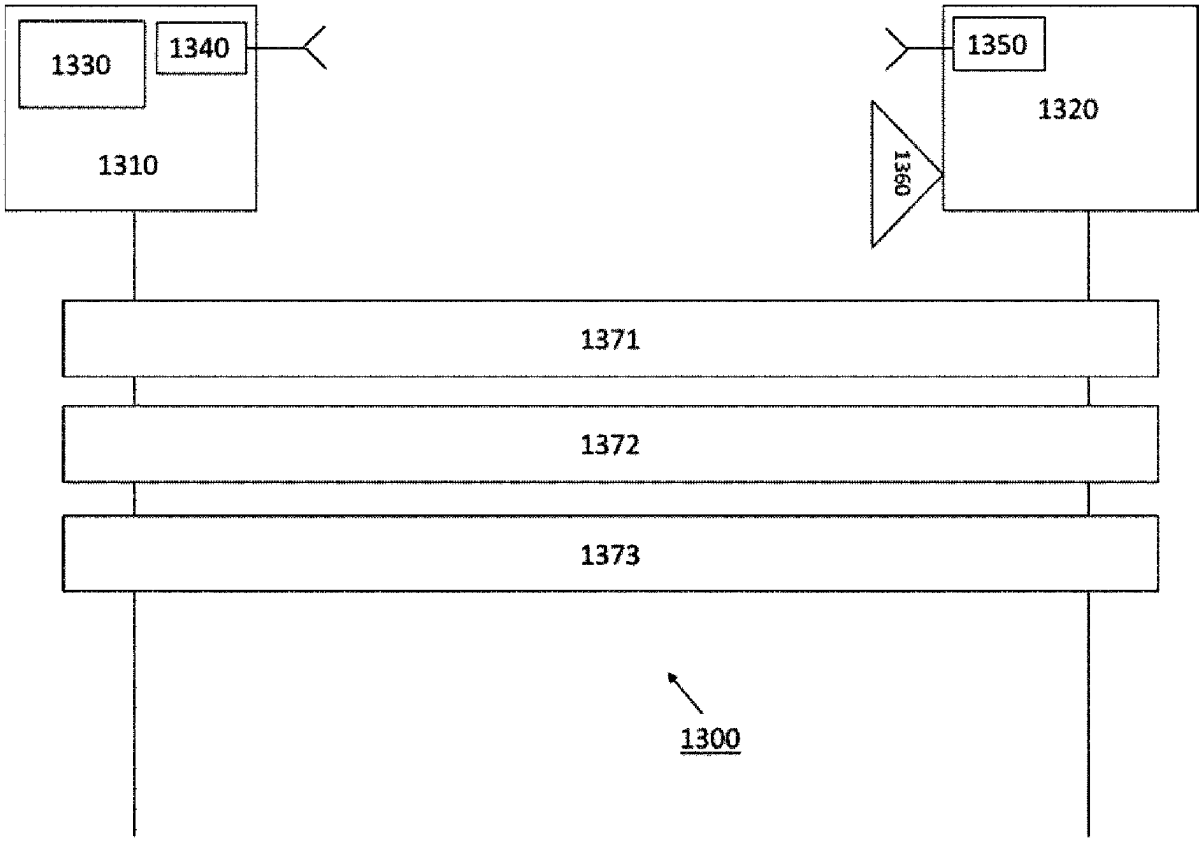


Fig. 13

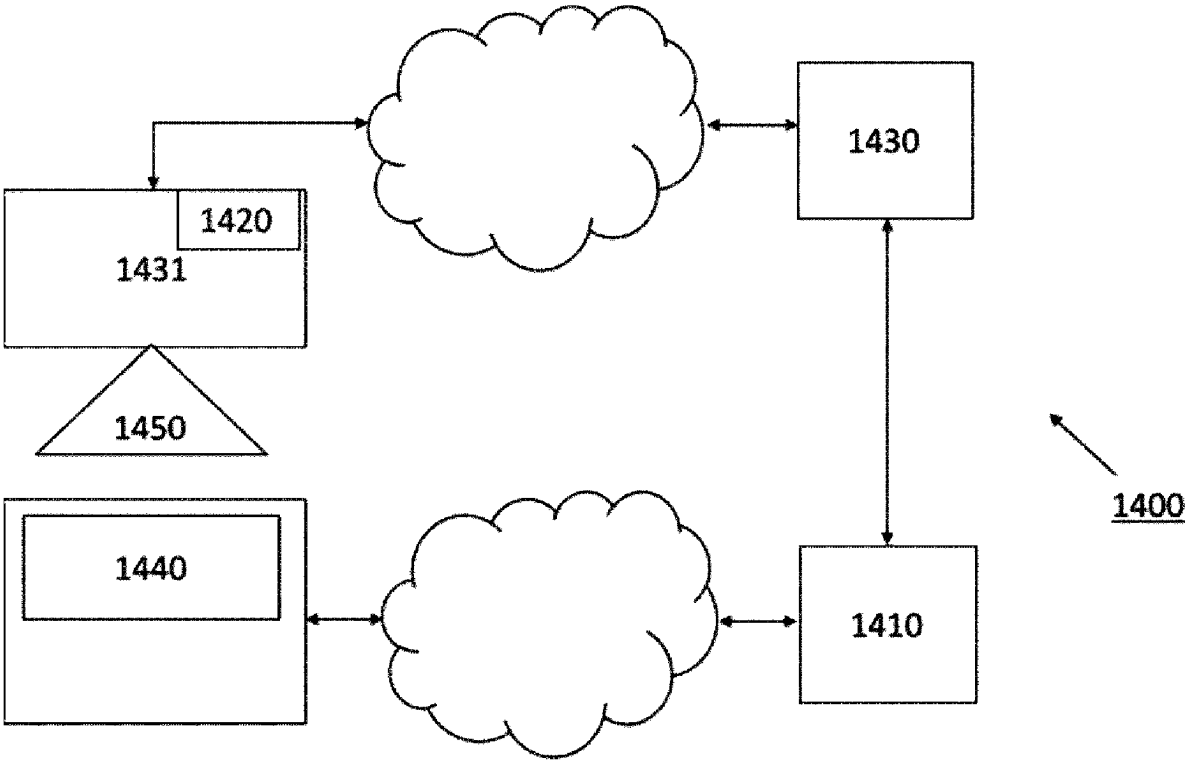


Fig. 14

 Das Dokument wurde durch die Firma Luminess hergestellt.