(19) Deutsches Patent- und Markenamt

(10) **DE 20 2023 103 216 U1** 2023.11.02

(12) # Gebrauchsmusterschrift

(21) Aktenzeichen: **20 2023 103 216.8**
(22) Anmeldetag: **12.06.2023**
(47) Eintragungstag: **22.09.2023**
(45) Bekanntmachungstag im Patentblatt: **02.11.2023**

(51) Int Cl.: ***G06F 21/62*** (2013.01)
G06F 9/50 (2006.01)
G06F 16/00 (2019.01)

(30) Unionspriorität:

63/366,266	**13.06.2022**	**US**
18/053,956	**09.11.2022**	**US**

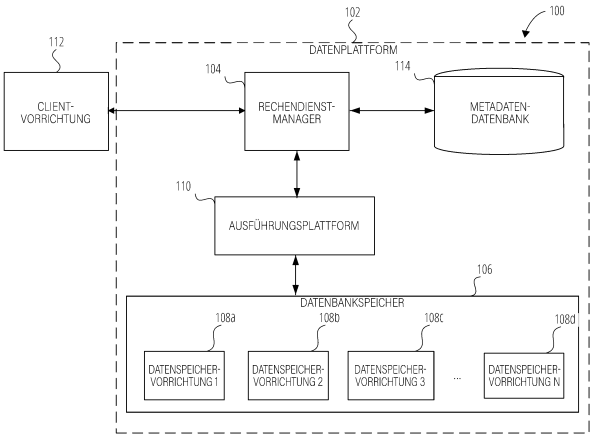
(73) Name und Wohnsitz des Inhabers:
Snowflake Inc., Bozeman, MT, US

(74) Name und Wohnsitz des Vertreters:
Betten & Resch Patent- und Rechtsanwälte PartGmbB, 80333 München, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Datenplattform mit vereinheitlichten Privilegien**

(57) Hauptanspruch: Datenplattform, die umfasst:
einen oder mehrere Prozessoren; und
wenigstens einen Speicher, der Anweisungen speichert, die dann, wenn sie durch den einen oder die mehreren Prozessoren ausgeführt werden, die Datenplattform veranlassen, Operationen auszuführen, die umfassen:
Autorisieren eines ersten Anwenders, einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen;
Autorisieren des ersten Anwenders, einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden; und
Bereitstellen der Anwenderanwendung für den zweiten Anwender basierend auf den Sicherheitsstrategien des Sicherheitskontextes durch Ausführen von Operationen, die umfassen:
Instanziieren eines Servers anwenderdefinierter Funktionen, UDF-Servers, innerhalb des Sicherheitskontextes;
Instanziieren einer Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes;
Instanziieren der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes; und
Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten basierend auf den Sicherheitsstrategien des Sicherheitskontextes.



Beschreibung

TECHNISCHES GEBIET

[0001] Beispiele der Offenbarung beziehen sich im Allgemeinen auf Datenbanken und spezifischer auf die Datenbanken-Sicherheit.

[0002] In Übereinstimmung mit den Anforderungen des Gebrauchsmustergesetzes sind nur Vorrichtungen, wie sie in den beigefügten Ansprüchen definiert sind, aber keine Verfahren als das Gebrauchsmuster zu schützen und der Gegenstand des Gebrauchsmusters. Falls in der Beschreibung auf Verfahren Bezug genommen wird, dienen diese Bezugnahmen lediglich dazu, die Vorrichtung oder die Vorrichtungen, für die mit den beigefügten Ansprüchen Schutz gesucht wird, zu veranschaulichen.

HINTERGRUND

[0003] Datenplattformen werden umfassend für die Datenspeicherung und den Datenzugriff in Computer- und Kommunikationskontexten verwendet. Bezüglich der Architektur könnte eine Datenplattform eine Vor-Ort-Datenplattform, eine netzbasierte Datenplattform (z. B. eine cloudbasierte Datenplattform), eine Kombination aus beidem sein und/oder einen weiteren Typ von Architektur enthalten. Bezüglich des Typs der Datenverarbeitung könnte eine Datenplattform eine Online-Transaktionsverarbeitung (OLTP), eine Online-Analyseverarbeitung (OLAP), eine Kombination aus beiden und/oder einen weiteren Typ der Datenverarbeitung implementieren. Überdies könnte eine Datenplattform ein relationales Datenbankmanagementsystem (RDBMS) und/oder einen oder mehrere andere Typen von Datenbankmanagementsystemen sein oder enthalten.

[0004] Die Anbieter von Daten auf einer Datenplattform können eine Möglichkeit wünschen, ihre Daten zweckmäßig zu sichern.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0005] Die vorliegende Offenbarung wird aus der im Folgenden gegebenen ausführlichen Beschreibung und aus den beigefügten Zeichnungen verschiedener Beispiele der Offenbarung umfassender erkannt; es zeigen:

Fig. 1 eine beispielhafte Computerumgebung, die eine netzbasierte Datenplattform in Kommunikation mit einem Cloud-Speicher-Anbietersystem enthält, gemäß einigen Beispielen der vorliegenden Offenbarung;

Fig. 2 einen Blockschaltplan, der Komponenten eines Rechendienstmanagers gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht;

Fig. 3 einen Blockschaltplan, der Komponenten einer Ausführungsplattform gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht;

Fig. 4A Einsatzdiagramm einer Computerumgebung zum Bereitstellen einer Anwendung gemäß einigen Beispielen der vorliegenden Offenbarung;

Fig. 4B, **Fig.** 4C und **Fig.** 4D Interaktions- und Datenflussdiagramme einer Computerumgebung zum Bereitstellen einer Anwenderanwendung gemäß einigen Beispielen der vorliegenden Offenbarung;

Fig. 5 ein Einsatzdiagramm für eine Computerumgebung einer Datenplattform, die einen vereinheitlichten Sicherheitskontext für eine Entwicklung und einen Einsatz einer Anwenderanwendung gemäß einigen Beispielen der vorliegenden Offenbarung bereitstellt;

Fig. 6 ein Aktivitätsdiagramm, das ein Verfahren zum Entwickeln und Einsetzen einer Anwenderanwendung in einem vereinheitlichten Sicherheitskontext durch eine Datenplattform 102 gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht; und

Fig. 7 eine graphische Darstellung einer Maschine in der Form eines Computersystems, innerhalb dessen ein Satz von Anweisungen ausgeführt werden kann, um die Maschine zu veranlassen, irgendeine oder mehrere der hier erörterten Methodologien auszuführen, gemäß einigen Beispielen der vorliegenden Offenbarung.

AUSFÜHRLICHE BESCHREIBUNG

[0006] Es gibt viele bestehende Lösungen zum Erstellung von Anwendungen, die auf Daten auf einer Datenplattform zugreifen, wobei aber die meisten von ihnen unterschiedliche Berechtigungsgrenzen für das Schreiben von Code für die Anwendung, das Einsetzen von Code, das Ausführen einer Anwendung, das Aktualisieren der Anwendung und das Verwenden der Anwendung beinhalten. Diese Lücken über Sicherheitsgrenzen hinweg sind Orte, an denen Fehler gemacht werden können oder an denen Autorisierungsmaterialien zum Interagieren mit Systemen gespeichert sein müssen. Es kann z. B. sein, dass ein Anwender in einem kontinuierlichen Integrationsauftrag Autorisierungsmaterialen speichern muss, um erstellte Artefakte zu einem Artefaktmanagementsystem zu schieben. Selbst in einfachen Systemen, bei denen die Erzeugung und Nutzung von Anwendungen häufig in derselben Organisation stattfinden, müssen Anmeldeinformationen für die Datenplattform gespeichert sein. Durch das Bereitstellen einer Kombination aus: einem Editor zum Bearbeiten und Erzeugen von Anwendungen innerhalb der Datenplattform; Methodologien zum Zuführen von Objekten als native Datenbankobjekte, die als Anwenderschnittstellenanwendungen (UI-Anwendungen) ausgeführt werden; Methodologien zum Verpacken und Teilen von Anwendungen; und Methodologien zum Teilen von Daten über Konten der Datenplattform, die alle innerhalb desselben Sicherheitskontextes arbeiten, kann ein Anwender Anwendungen erzeugen, einsetzen und ausführen, ohne an einem Ort außerhalb der Plattform Anmeldeinformationen oder andere Autorisierungsmaterialien zu speichern.

[0007] Gemäß einigen Beispielen teilen sich Entwicklung, Operationen und Endanwender den gleichen Sicherheitskontext mit vereinheitlichten Privilegien, die die Sicherheitsgrenzen des Sicherheitskontextes definieren. Verschiedene Komponenten des Systems können direkt auf der Datenplattform arbeiten. Dies schafft eine Autorisierungsgrenze über alle Aspekte eines Lebenszyklus einer Anwenderanwendung. Operationen, wie z. B. das Verteilen einer Anwenderanwendung, können innerhalb eines nativen Anwendungsrahmens der Datenplattform arbeiten, um eine Menge zusätzlicher Sicherheitsgrenzen zu verringern. Im Ergebnis kann ein Anwender das Sichern von Authentifikations- oder Autorisierungsmaterialien zum Integrieren von Systemen in einem Datenspeicher außerhalb der Plattform vermeiden. Jedes Mal, wenn Sicherheits-Anmeldeinformationen für Operationen, wie z. B. das Schieben von Code zwischen Systemen, das Schieben von erstellten Artefakten zwischen Systemen, das Starten von Rechenbetriebsmitteln oder das Betreiben eines laufenden Systems, gespeichert werden, gibt es potentielle Schwachstellen. Diese Schwachstellen enthalten die Möglichkeit, dass Autorisierungsmaterialien gestohlen werden, und die Möglichkeit, dass Autorisierungsmaterialien ablaufen und verursachen, dass betriebsfähige Systeme die Arbeit einstellen, bis sie ersetzt werden.

[0008] Gemäß einigen Beispielen wird der vollständige Lebenszyklus einer Anwendung von der Konzeption bis zur Auslieferung innerhalb eines vereinheitlichten Sicherheitskontextes auf der Datenplattform gemanagt. Da sämtliche Arbeit an dem Artefakt während des gesamten Lebenszyklus des Artefakts auf der Datenplattform gespeichert ist und eine Verteilungserstellung der Anwendung unter Verwendung eines Anwendungs- und Datenteilungs-Managers der Datenplattform geteilt wird, wird eine einzige Sicherheitsgrenze bereitgestellt.

[0009] Gemäß einigen Beispielen stellt die Datenplattform eine Anwendungsbearbeitungsoberfläche bereit, die Dateien in Datenspeicher innerhalb der Datenplattform schreibt, um den Quellcode für eine Anwendung zu speichern.

[0010] Gemäß einigen Beispielen wird eine interaktive Bearbeitungssitzung des Quellcodes einer Anwendung durch Verbinden mit einer Anwenderanwendung in einer Form einer gesicherten Web-Anwendung, die durch die Datenplattform bereitgestellt wird, bereitgestellt.

[0011] Gemäß einigen Beispielen wird eine interaktive Bearbeitungssitzung des Quellcodes einer Anwendung unter Verwendung eines durch die Datenplattform bereitgestellten Editors bereitgestellt.

[0012] Gemäß einigen Beispielen wird das Voranbringen des Entwicklungscodes in die Produktion durch Modifizieren einer Anwenderanwendung oder eines Datenspeichers, der den Quellcode der Anwenderanwendung speichert, erreicht, um den Quellcode zu ändern, der gegenwärtig mit einer durch die Anwenderanwendungen verwendeten Standardversion verbunden ist. Die Endanwender treffen auf den Endpunkt der Anwenderanwendung, wo ihr Authentifikationskontext zum Starten der Anwenderanwendung verwendet wird und die Anwenderanwendung in dem bereits innerhalb der Datenplattform konfigurierten Sicherheitskontext läuft.

[0013] Gemäß einigen Beispielen eine Datenplattform zum Entwickeln und Einsetzen einer Anwenderanwendung innerhalb eines vereinheitlichten Sicherheitskontextes. Die Datenplattform autorisiert einen ersten Anwender, um einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf den Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen, und autorisiert den ersten Anwender, einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden. Um die Anwenderanwendung dem zweiten Anwender bereitzustellen, setzt die Datenplattform die Anwenderanwendung durch Instanziieren eines Servers anwenderdefinierter Funktionen (UDF-Servers) und eine Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes, Instanziieren der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes und Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten der Datenplattform basierend auf den Sicherheitsstrategien des Sicherheitskontextes ein.

[0014] Gemäß einigen Beispielen enthält eine Datenplattform einen oder mehrere Prozessoren und einen Speicher, der Anweisungen speichert. Wenn die Anweisungen durch den einen oder die mehreren Prozessoren ausgeführt werden, autorisiert die Datenplattform einen ersten Anwender, einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf den Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen, wobei sie den ersten Anwender autorisiert, einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden. Die Datenplattform stellt die Anwenderanwendung dem zweiten Anwender basierend auf den Sicherheitsstrategien des Sicherheitskontextes durch Instanziieren eines Servers anwenderdefinierter Funktionen (UDF-Servers) innerhalb des Sicherheitskontextes, Instanziieren einer Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes, Instanziieren der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes und Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten der Datenplattform basierend auf den Sicherheitsstrategien des Sicherheitskontextes bereit.

[0015] Nun wird ausführlich auf spezifische Beispiele zum Ausführen des Erfindungsgegenstandes Bezug genommen. Beispiele dieser spezifischen Beispiele sind in den beigefügten Zeichnungen veranschaulicht, wobei spezifische Einzelheiten in der folgenden Beschreibung dargelegt sind, um ein umfassendes Verständnis des Gegenstands zu schaffen. Es wird erkannt, dass diese Beispiele den Schutzumfang der Ansprüche nicht auf die veranschaulichten Beispiele einschränken sollen. Im Gegenteil sollen sie derartige Alternativen, Modifikationen und Äquivalente abdecken, die innerhalb des Schutzumfangs der Offenbarung enthalten sein können.

[0016] Fig. 1 veranschaulicht eine beispielhafte Computerumgebung 100, die eine Datenplattform 102 in Kommunikation mit einer Client-Vorrichtung 112 enthält, gemäß einigen Beispielen der vorliegenden Offenbarung. Um das Verbergen des Erfindungsgegenstands mit unnötigen Einzelheiten zu vermeiden, sind verschiedene funktionale Komponenten, die für das Vermitteln eines Verständnisses des Erfindungsgegenstandes nicht relevant sind, aus **Fig.** 1 weggelassen worden. Der Fachmann erkennt jedoch leicht, dass verschiedene zusätzliche funktionale Komponenten als Teil der Computerumgebung 100 enthalten sein können, um zusätzliche Funktionalität zu ermöglichen, die hier nicht spezifisch beschrieben ist.

[0017] Wie gezeigt ist, umfasst die Datenplattform 102 einen Datenbankspeicher 106, einen Rechendienstmanager 104, eine Ausführungsplattform 110 und eine Metadaten-Datenbank 114. Der Datenbankspeicher 106 umfasst mehrere Rechenmaschinen und stellt der Datenplattform 102 auf Abruf Computersystembetriebsmittel wie z. B. Datenspeicher und Rechenleistung, bereit. Wie gezeigt ist, umfasst der Datenbankspeicher 106 mehrere Datenspeichervorrichtungen, nämlich eine Datenspeichervorrichtung 1 108a bis Datenspeichervorrichtung N 108d. Gemäß einigen Beispielen sind die Datenspeichervorrichtungen 1 bis N cloudbasierte Speichervorrichtungen, die sich an einem oder mehreren geographischen Orten befinden. Die Datenspeichervorrichtungen 1 bis N können z. B. Teil einer öffentlichen Cloud-Infrastruktur oder einer privaten Cloud-Infrastruktur sein. Die Datenspeichervorrichtungen 1 bis N können Festplattenlaufwerke (HDDs), Halbleiterlaufwerke (SSDs), Speicher-Cluster, Amazon-S3TM-Speichersysteme oder irgendeine andere Datenspeichertechnik sein. Zusätzlich kann der Datenbankspeicher 106 verteilte Dateisysteme (z. B. verteilte Hadoop-Dateisysteme (HDFS)), Objektspeichersysteme und dergleichen enthalten.

[0018] Die Datenplattform 102 wird für das Melden und die Analyse integrierter Daten aus einer oder mehreren verschiedenen Quellen verwendet, die die Speichervorrichtungen 1 bis N innerhalb des Datenbankspeichers 106 enthalten. Die Datenplattform 102 hostet und schafft Datenmelde- und Analysedienste für mehrere Kundenkonten. Administrative Anwender können Identitäten (z. B. Anwender, Rollen und Gruppen) erzeugen

und managen und Privilegien verwenden, um den Identitäten den Zugriff auf Betriebsmittel und Dienste zu erlauben oder zu verweigern. Im Allgemeinen erhält die Datenplattform 102 zahlreiche Kundenkonten für zahlreiche jeweilige Kunden aufrecht. Die Datenplattform 102 erhält jedes Kundenkonto in einer oder mehreren Speichervorrichtungen des Datenbankspeichers 106 aufrecht. Überdies kann die Datenplattform 102 Metadaten, die den Kundenkonten zugeordnet sind, in der Metadaten-Datenbank 114 aufrechterhalten. Jedes Kundenkonto enthält mehrere Datenobjekte mit Beispielen, die Anwender, Rollen, Privilegien, einen Datenspeicher oder andere Datenorte (hier als „Stufe“ oder „Stufen“ bezeichnet werden) und dergleichen enthalten.

[0019] Der Rechendienstmanager 104 koordiniert und managt die Operationen der Datenplattform 102. Außerdem führt der Rechendienstmanager 104 sowohl die Abfrageoptimierung und -kompilierung als auch das Managen von Clustern von Rechendiensten, die Rechenbetriebsmittel bereitstellen, (die außerdem als „virtuelle Warenhäuser“ bezeichnet werden) aus. Der Rechendienstmanager 104 kann irgendeine Anzahl und irgendeinen Typ von Clients, wie z. B. Endanwender, die Datenspeicherungs- und -wiedergewinnungsanforderungen bereitstellen, Systemmanager, die die hier beschriebenen Systeme und Verfahren managen, und andere Komponenten/Vorrichtungen, die mit dem Rechendienstmanager 104 interagieren, unterstützen. Der Rechendienstmanager 104 steht als ein Beispiel mit der Client-Vorrichtung 112 in Kommunikation. Die Client-Vorrichtung 112 kann durch einen Anwender eines der mehreren durch die Datenplattform 102 unterstützten Kundenkonten verwendet werden, um mit der Datenplattform 102 zu interagieren und deren Funktionalität zu nutzen. Gemäß einigen Beispielen empfängt der Rechendienstmanager 104 keine direkten Kommunikationen von der Client-Vorrichtung 112, wobei er nur Kommunikationen bezüglich der Aufträge aus einer Warteschlange innerhalb der Datenplattform 102 empfängt.

[0020] Der Rechendienstmanager 104 ist außerdem an die Metadaten-Datenbank 114 gekoppelt. Die Metadaten-Datenbank 114 speichert Daten, die verschiedene Funktionen und Aspekte betreffen, die der Datenplattform 102 und ihren Anwendern zugeordnet sind. Gemäß einigen Beispielen enthält die Metadaten-Datenbank 114 eine Zusammenfassung sowohl von Daten, die in entfernten Datenspeichersystemen gespeichert sind, als auch von Daten, die aus einem lokalen Cache verfügbar sind. Zusätzlich kann die Metadaten-Datenbank 114 Informationen hinsichtlich dessen enthalten, wie die Daten in entfernten Datenspeichersystemen (z. B. dem Datenbankspeicher 106) und den lokalen Caches organisiert sind. Die Metadaten-Datenbank 114 ermöglicht es Systemen und Diensten, zu bestimmen, ob auf einen Teil der Daten zugegriffen werden muss, ohne die eigentlichen Daten aus einer Speichervorrichtung zu laden oder auf sie zuzugreifen.

[0021] Der Rechendienstmanager 104 ist ferner an die Ausführungsplattform 110 gekoppelt, die mehrere Rechenbetriebsmittel bereitstellt, die verschiedene Datenspeicher- und Datenwiedergewinnungsaufgaben ausführen. Gemäß einigen Beispielen kommuniziert der Rechendienstmanager 104 mit der Ausführungsplattform 110 bezüglich Aufträgen und Aufgaben unter Verwendung einer Warteschlange innerhalb der Datenplattform 102. Dies isoliert die Operationen der Ausführungsplattform 110 und der Client-Vorrichtung 112. Die Ausführungsplattform 110 ist an den Datenbankspeicher 106 gekoppelt. Die Ausführungsplattform 110 umfasst mehrere Rechenknoten. Ein Satz von Prozessen in einem Rechenknoten führt einen durch den Rechendienstmanager 104 kompilierten Abfrageplan aus. Der Satz von Prozessen kann enthalten: einen ersten Prozess, um den Abfrageplan auszuführen; einen zweiten Prozess, um Mikropartitionsdateien unter Verwendung einer Am-wenigsten-vor-kurzem-verwendet-Strategie (LRU-Strategie) zu überwachen und zu löschen und einen Außerhalb-des-Speichers-Fehlerabschwächungsprozess (OOM-Fehlerabschwächungsprozess) zu implementieren; einen dritten Prozess, der Zustandsinformationen aus Prozessprotokollen und dem Status extrahiert, um sie an den Rechendienstmanager 104 zurückzusenden; einen vierten Prozess, um die Kommunikation mit dem Rechendienstmanager 104 nach einem System-Urladen herzustellen; und einen fünften Prozess, der sämtliche Kommunikation mit einem Computer-Cluster für einen gegebenen durch den Rechendienstmanager 104 bereitgestellten Auftrag abzuwickeln und um Informationen zurück zu dem Rechendienstmanager 104 und anderen Rechenknoten der Ausführungsplattform 110 zu übertragen.

[0022] Gemäß einigen Beispielen sind die Kommunikationsverbindungen zwischen den Elementen der Computerumgebung 100 über ein oder mehrere Datenkommunikationsnetze implementiert. Diese Datenkommunikationsnetze können jedes Kommunikationsprotokoll und jeden Typ von Kommunikationsmedium verwenden. Gemäß einigen Beispielen sind die Datenkommunikationsnetze eine Kombination aus zwei oder mehr Datenkommunikationsnetzen (oder Teilnetzen), die aneinandergekoppelt sind. Gemäß alternativen Beispielen sind diese Kommunikationsverbindungen unter Verwendung irgendeines Typs von Kommunikationsmedium und irgendeines Kommunikationsprotokolls implementiert.

[0023] Wie in **Fig.** 1 gezeigt ist, sind die Datenspeichereinrichtung 1 108a bis Datenspeichereinrichtung N 108d der Datenspeichervorrichtungen von den der Ausführungsplattform 110 zugeordneten Computerbetriebsmitteln entkoppelt. Diese Architektur unterstützt dynamische Änderungen an der Datenplattform 102 basierend sowohl auf den sich ändernden Datenspeicherungs-/-wiedergewinnungs-Anforderungen als auch auf den sich ändernden Anforderungen der Anwender und Systeme. Die Unterstützung dynamischer Änderungen ermöglicht, die Datenplattform 102 in Reaktion auf sich ändernde Anforderungen an die Systeme und Komponenten innerhalb der Datenplattform 102 schnell zu skalieren. Das Entkoppeln der Computerbetriebsmittel von den Datenspeichervorrichtungen unterstützt die Speicherung großer Datenmengen, ohne eine entsprechende große Menge an Computerbetriebsmitteln zu erfordern. Ähnlich unterstützt dieses Entkoppeln von Betriebsmitteln eine signifikante Erhöhung der zu einem speziellen Zeitpunkt genutzten Computerbetriebsmittel, ohne eine entsprechende Erhöhung der verfügbaren Datenspeicherbetriebsmittel zu erfordern.

[0024] Der Rechendienstmanager 104, die Metadaten-Datenbank 114, die Ausführungsplattform 110 und der Datenbankspeicher 106 sind in **Fig.** 1 als einzelne diskrete Komponenten gezeigt. Jedes des Rechendienstmanagers 104, der Metadaten-Datenbank 114, der Ausführungsplattform 110 und des Datenbankspeichers 106 kann jedoch als ein verteiltes System (z. B. verteilt über mehrere Systeme/Plattformen an mehreren geographischen Standorten) implementiert sein. Zusätzlich kann jedes des Rechendienstmanagers 104, der Metadaten-Datenbank 114, der Ausführungsplattform 110 und des Datenbankspeichers 106 (unabhängig voneinander) abhängig von den Änderungen der empfangenen Anforderungen und der sich ändernden Anforderungen der Datenplattform 102 aufwärts- oder abwärtsskaliert werden. Folglich ist die Datenplattform 102 gemäß den beschriebenen Beispielen dynamisch und unterstützt regelmäßige Änderungen, um den aktuellen Datenverarbeitungsanforderungen zu entsprechen.

[0025] Während des Betriebs verarbeitet die Datenplattform 102 mehrere Aufträge, die durch den Rechendienstmanager 104 bestimmt werden. Diese Aufträge werden durch den Rechendienstmanager 104 geplant und gemanagt, um zu bestimmen, wann und wie der Auftrag auszuführen ist. Der Rechendienstmanager 104 kann z. B. den Auftrag in mehrere diskrete Aufgaben aufteilen und bestimmen, welche Daten benötigt werden, um jede der diskreten Aufgaben auszuführen. Der Rechendienstmanager 104 kann jede der mehreren diskreten Aufgaben einem oder mehreren Knoten der Ausführungsplattform 110 zuweisen, um die Aufgabe zu verarbeiten. Der Rechendienstmanager 104 kann bestimmen, welche Daten benötigt werden, um eine Aufgabe zu verarbeiten, und ferner bestimmen, welche Knoten innerhalb der Ausführungsplattform 110 am besten geeignet sind, um die Aufgabe zu verarbeiten. Einige Knoten können die Daten, die benötigt werden, um die Aufgabe zu verarbeiten, bereits im Cache gespeichert aufweisen, wobei sie deshalb ein guter Kandidat zum Verarbeiten der Aufgabe sind. Die in der Metadaten-Datenbank 114 gespeicherten Metadaten unterstützen den Rechendienstmanager 104 beim Bestimmen, welche Knoten in der Ausführungsplattform 110 wenigstens einen Teil der Daten, die benötigt werden, um die Aufgabe zu verarbeiten, bereits im Cache gespeichert aufweisen. Ein oder mehrere Knoten in der Ausführungsplattform 110 verarbeiten die Aufgabe unter Verwendung von Daten, die durch die Knoten im Cache gespeichert sind, wobei, falls erforderlich, von aus dem Datenbankspeicher 106 wiedergewonnenen Daten. Es ist wünschenswert, so viele Daten wie möglich aus den Caches innerhalb der Ausführungsplattform 110 wiederzugewinnen, weil die Wiedergewinnungsgeschwindigkeit typischerweise schneller als das Wiedergewinnen von Daten aus dem Datenbankspeicher 106 ist.

[0026] Wie in **Fig.** 1 gezeigt ist, trennt die Computerumgebung 100 die Ausführungsplattform 110 von dem Datenbankspeicher 106. Gemäß dieser Anordnung arbeiten die Verarbeitungsbetriebsmittel und die Cache-Betriebsmittel in der Ausführungsplattform 110 unabhängig von den Datenbankspeichervorrichtungen der Datenspeichervorrichtung 1 108a bis Datenspeichervorrichtung N 108d im Datenbankspeicher 106. Folglich sind die Computerbetriebsmittel und Cache-Betriebsmittel nicht auf eine spezifische der Datenspeichervorrichtung 1 108a bis Datenspeichervorrichtung N 108d eingeschränkt. Stattdessen können alle Computerbetriebsmittel und alle Cache-Betriebsmittel Daten von jedem der Datenspeicherbetriebsmittel im Datenbankspeicher 106 wiedergewinnen und Daten dort speichern.

[0027] **Fig.** 2 ist ein Blockschaltplan, der die Komponenten des Rechendienstmanagers 104 gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht. Wie in **Fig.** 2 gezeigt ist, enthält der Rechendienstmanager 104 einen Zugriffsmanager 202 und einen Schlüsselmanager 204, die an eine Datenspeichervorrichtung 206 gekoppelt sind. Der Zugriffsmanager 202 wickelt die Authentifikations- und Autorisierungsaufgaben für die hier beschriebenen Systeme ab. Der Schlüsselmanager 204 managt die Speicherung und Authentifikation von Schlüsseln, die während der Authentifikations- und Autorisierungsaufgaben verwendet werden. Der Zugriffsmanager 202 und der Schlüsselmanager 204 managen z. B. die Schlüssel, die verwen-

det werden, um auf Daten, die in entfernten Speichervorrichtungen (z. B. den Datenspeichervorrichtungen im Datenbankspeicher 106) gespeichert sind, zuzugreifen. Wie sie hier verwendet werden, können die entfernten Speichervorrichtungen außerdem als „dauerhafte Speichervorrichtungen“ oder „gemeinsam benutzte Speichervorrichtungen“ bezeichnet werden.

[0028] Ein Anforderungsverarbeitungsdienst 208 managt empfangene Datenspeicheranforderungen und Datenwiedergewinnungsanforderungen (z. B. Aufträge, die an Datenbankdaten auszuführen sind). Der Anforderungsverarbeitungsdienst 208 kann z. B. die Daten bestimmen, die erforderlich sind, um eine empfangene Abfrage (z. B. eine Datenspeicheranforderung oder eine Datenwiedergewinnungsanforderung) zu verarbeiten. Die Daten können in einem Cache innerhalb der Ausführungsplattform 110 oder in einer Datenspeichervorrichtung im Datenbankspeicher 106 gespeichert sein.

[0029] Ein Managementkonsolendienst 210 unterstützt den Zugriff auf verschiedene Systeme und Prozesse durch Manager und andere Systemmanager. Zusätzlich kann der Managementkonsolendienst 210 eine Anforderung empfangen, um einen Auftrag auszuführen und die Arbeitsbelastung in dem System zu überwachen.

[0030] Der Rechendienstmanager 104 enthält außerdem einen Auftragskompilierer 212, einen Auftragsoptimierer 214 und einen Auftragsausführer 216. Der Auftragskompilierer 212 parst einen Auftrag in mehrere diskrete Aufgaben und erzeugt den Ausführungscode für jede der mehreren diskreten Aufgaben. Der Auftragsoptimierer 214 bestimmt basierend auf den Daten, die verarbeitet werden müssen, das beste Verfahren, um die einzelnen Aufgaben auszuführen. Außerdem wickelt der Auftragsoptimierer 214 verschiedene Datenbereinigungsoperationen und andere Datenoptimierungstechniken ab, um die Geschwindigkeit und Effizienz des Ausführens des Auftrags zu verbessern. Der Auftragsausführer 216 führt den Ausführungscode für die Aufträge aus, die von einer Warteschlange empfangen werden oder vom Rechendienstmanager 104 bestimmt werden.

[0031] Ein Auftragsplaner und -koordinator 218 sendet empfangene Aufträge an die geeigneten Dienste oder Systeme zur Kompilierung, Optimierung und Abfertigung zu der Ausführungsplattform 110. Die Aufträge können z. B. priorisiert und in dieser priorisierten Reihenfolge verarbeitet werden. Gemäß einigen Beispielen bestimmt der Auftragsplaner und -koordinator 218 eine Priorität für interne Aufträge, die durch den Rechendienstmanager 104 mit anderen „äußeren“ Aufträgen, wie z. B. Anwenderabfragen, die durch andere Systeme in der Datenbank geplant werden können, aber dieselben Verarbeitungsbetriebsmittel in der Ausführungsplattform 110 nutzen können, geplant werden. Gemäß einigen Beispielen identifiziert der Auftragsplaner und -koordinator 218 spezielle Knoten in der Ausführungsplattform 110 oder weist sie zu, um spezielle Aufgaben zu verarbeiten. Ein Manager 220 virtueller Warenhäuser managt den Betrieb mehrerer virtueller Warenhaushäuser, die in der Ausführungsplattform 110 implementiert sind. Wie im Folgenden erörtert wird, enthält jedes virtuelle Warenhaus mehrere Ausführungsknoten, die jeweils einen Cache und einen Prozessor enthalten.

[0032] Zusätzlich enthält der Rechendienstmanager 104 einen Konfigurations- und Metadaten-Manager 222, der die Informationen bezüglich der in den entfernten Datenspeicher-Vorrichtungen und in den lokalen Caches (z. B. die Caches in der Ausführungsplattform 110) gespeicherten Daten managt. Der Konfigurations- und Metadaten-Manager 222 konfiguriert die Metadaten, um zu bestimmen, auf welche Datenmikropartitionen zugegriffen werden muss, um Daten für die Verarbeitung einer speziellen Aufgabe oder eines speziellen Auftrags wiederzugewinnen. Eine Überwachungseinrichtung und ein Arbeitsbelastungsanalysator 224 überwacht die durch den Rechendienstmanager 104 ausgeführten Prozesse und managt die Verteilung der Aufgaben (z. B. der Arbeitsbelastung) über die virtuellen Warenhäusern und die Ausführungsknoten in der Ausführungsplattform 110. Die Überwachungseinrichtung und der Arbeitsbelastungsanalysator 224 verteilt außerdem bei Bedarf die Aufgaben basierend auf sich ändernden Arbeitsbelastungen überall in der Datenplattform 102 neu, und kann ferner die Aufgaben basierend auf einer Anwender- (z. B. „externen“) Abfrage-Arbeitsbelastung neu verteilen, die außerdem durch die Ausführungsplattform 110 verarbeitet werden kann. Der Konfigurations- und Metadaten-Manager 222 und die Überwachungseinrichtung und der Arbeitsbelastungsanalysator 224 sind an eine Datenspeichervorrichtung 226 gekoppelt. Die Datenspeichervorrichtung 226 nach **Fig.** 2 repräsentiert irgendeine Datenspeichervorrichtung innerhalb der Datenplattform 102. Die Datenspeichervorrichtung 226 kann z. B. Caches in der Ausführungsplattform 110, Speichervorrichtungen im Datenbankspeicher 106 oder irgendeine andere Speichervorrichtung repräsentieren.

[0033] Der Rechendienstmanager 104 validiert sämtliche Kommunikation von einer Ausführungsplattform (z. B. der Ausführungsplattform 110), um zu validieren, ob der Inhalt und der Kontext dieser Kommunikation mit

der Aufgabe (den Aufgaben) konsistent ist, die der Ausführungsplattform zuzuweisen sind. Einer Instanz der Ausführungsplattform, die eine Abfrage A ausführt, sollte z. B. nicht erlaubt sein, Zugriff auf die Datenquelle D (z. B. die Datenspeichervorrichtung 226) anzufordern, die für die Abfrage A nicht relevant ist. Ähnlich kann es sein, dass ein gegebener Ausführungsknoten (z. B. der Ausführungsknoten 1 304a) mit einem weiteren Ausführungsknoten (z. B. dem Ausführungsknoten 2 304b) kommunizieren muss, wobei ihm verweigert werden sollte, mit einem dritten Ausführungsknoten (z. B. dem Ausführungsknoten 1 316a) zu kommunizieren, wobei jede derartige unerlaubte Kommunikation (z. B. in einem Protokoll oder an anderer Stelle) aufgezeichnet werden kann. Außerdem sind die in einem gegebenen Ausführungsknoten gespeicherten Informationen auf Daten eingeschränkt, die für die aktuelle Abfrage relevant sind, wobei alle anderen Daten unbenutzbar sind und durch Zerstörung oder Verschlüsselung, wobei der Schlüssel nicht verfügbar ist, so gemacht werden.

[0034] Fig. 3 ist ein Blockschaltplan, der die Komponenten der Ausführungsplattform 110 gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht. Wie in **Fig.** 3 gezeigt ist, enthält die Ausführungsplattform 110 mehrere virtuelle Warenhäuser einschließlich des virtuellen Warenhauses 1 302a und des virtuellen Warenhause 2 302b bis virtuellen Warenhauses N 302c. Jedes virtuelle Warenhaus enthält mehrere Ausführungsknoten, die jeweils einen Daten-Cache und einen Prozessor enthalten. Die virtuellen Warenhäuser können unter Verwendung der mehreren Ausführungsknoten mehrere Aufgaben parallel ausführen. Wie hier erörtert wird, kann die Ausführungsplattform 110 basierend auf den aktuellen Verarbeitungsanforderungen der Systeme und Anwender in Echtzeit neue virtuelle Warenhäuser hinzufügen und bestehende virtuelle Warenhäuser fallenlassen. Diese Flexibilität ermöglicht es der Ausführungsplattform 110, bei Bedarf schnell große Mengen an Computerbetriebsmitteln einzusetzen, ohne gezwungen zu sein, für diese Computerbetriebsmittel weiter zu bezahlen, wenn sie nicht mehr benötigt werden. Alle virtuellen Warenhaushäuser können auf Daten von jeder Datenspeichervorrichtung (z. B. jeder Speichervorrichtung im Datenbankspeicher 106) zugreifen.

[0035] Obwohl jedes in **Fig.** 3 gezeigte virtuelle Warenhaus drei Ausführungsknoten enthält, kann ein spezielles virtuelles Warenhaus irgendeine Anzahl von Ausführungsknoten enthalten. Ferner ist die Anzahl der Ausführungsknoten in einem virtuellen Warenhaus dynamisch, so dass neue Ausführungsknoten erzeugt werden, wenn zusätzlicher Bedarf besteht, und bestehende Ausführungsknoten gelöscht werden, wenn sie nicht mehr erforderlich sind.

[0036] Jedes virtuelle Warenhaus kann auf jede der in **Fig.** 1 gezeigten Datenspeichervorrichtungen 1 bis N zuzugreifen. Folglich sind die virtuellen Warenhäuser nicht notwendigerweise einer spezifischen Datenspeichervorrichtung 1 bis N zugewiesen und können stattdessen auf Daten von jeder der Datenspeichervorrichtungen 1 bis N innerhalb des Datenbankspeichers 106 zugreifen. Ähnlich kann jeder der in **Fig.** 3 gezeigten Ausführungsknoten auf Daten von jeder der Datenspeichervorrichtungen 1 bis N zugreifen. Gemäß einigen Beispielen kann ein spezielles virtuelles Warenhaus oder ein spezieller Ausführungsknoten vorübergehend einer spezifischen Datenspeichervorrichtung zugewiesen sein, wobei aber das virtuelle Warenhaus oder der Ausführungsknoten später auf Daten von jeder anderen Datenspeichervorrichtung zugreifen kann.

[0037] Im Beispiel nach **Fig.** 3 enthält das virtuelle Warenhaus 1 302a mehrere Ausführungsknoten, wie sie durch einen Ausführungsknoten 1 304a, einen Ausführungsknoten 2 304b und einen Ausführungsknoten N 304c veranschaulicht sind. Der Ausführungsknoten 1 304a enthält einen Cache 1 306a und einen Prozessor 1 308a. Der Ausführungsknoten 2 304b enthält einen Cache 2 306b und einen Prozessor 2 308b. Der Ausführungsknoten N 304c enthält einen Cache N 306c und einen Prozessor N 308c. Jeder Ausführungsknoten 1 bis N ist der Verarbeitung einer oder mehrerer Datenspeicher- und/oder Datenwiedergewinnungsaufgaben zugeordnet. Ein virtuelles Warenhaus kann z. B. Datenspeicherungs- und Datenwiedergewinnungsaufgaben abwickeln, die einem internen Dienst, wie z. B. einem Clustering-Dienst, einem Aktualisierungsdienst für materialisierte Ansichten, einem Dateiverdichtungsdienst, einem Speicherprozedurdienst oder einem Dateiaufrüstdienst, zugeordnet sind. Gemäß anderen Implementierungen kann ein spezielles virtuelles Warenhaus Datenspeicher- und Datenwiedergewinnungsaufgaben handhaben, die einem speziellen Datenspeichersystem oder einer speziellen Datenkategorie zugeordnet sind.

[0038] Ähnlich zum oben erörterten virtuellen Warenhaus 1 302a enthält das virtuelle Warenhaus 2 302b mehrere Ausführungsknoten, die durch einen Ausführungsknoten 1 310a, einen Ausführungsknoten 2 310b und einen Ausführungsknoten N 310c veranschaulicht sind. Der Ausführungsknoten 1 304a enthält einen Cache 1 312a und einen Prozessor 1 314a. Der Ausführungsknoten 2 310b enthält einen Cache 2 312b und einen Prozessor 2 314b. Der Ausführungsknoten N 310c enthält einen Cache N 312c und einen Prozessor N 314c. Zusätzlich enthält das virtuelle Warenhaus N 302c mehrere Ausführungsknoten, die durch einen Aus-

führungsknoten 1 316a, einen Ausführungsknoten 2 316b und einen Ausführungsknoten N 316c veranschaulicht sind. Der Ausführungsknoten 1 316a enthält einen Cache 1 318a und einen Prozessor 1 320a. Der Ausführungsknoten 2 316b enthält einen Cache 2 318b und einen Prozessor 2 320b. Der Ausführungsknoten N 316c enthält einen Cache N 318c und einen Prozessor N 320c.

[0039] Gemäß einigen Beispielen sind die in **Fig.** 3 gezeigten Ausführungsknoten zustandslos in Bezug auf die Daten, die die Ausführungsknoten im Cache speichern. Diese Ausführungsknoten speichern z. B. keine Zustandsinformationen über den Ausführungsknoten oder die Daten, die durch einen speziellen Ausführungsknoten in Cache gespeichert sind, oder erhalten diese anderweitig nicht aufrecht. Folglich kann im Fall einer Ausführungsknotenstörung der ausgefallene Knoten transparent durch einen weiteren Knoten ersetzt werden. Weil dem ausgefallenen Ausführungsknoten keine Zustandsinformationen zugeordnet sind, kann der neue (Ersatz-) Ausführungsknoten den ausgefallenen Knoten ohne eine Besorgnis für das Wiederherstellen eines speziellen Zustands einfach ersetzen.

[0040] Obwohl die in **Fig.** 3 gezeigten Ausführungsknoten jeweils einen Daten-Cache und einen Prozessor enthalten, können alternative Beispiele Ausführungsknoten enthalten, die irgendeine Anzahl von Prozessoren und irgendeine Anzahl von Caches enthalten. Zusätzlich können die Größen der Caches zwischen den verschiedenen Ausführungsknoten variieren. Die in **Fig.** 3 gezeigten Caches speichern im lokalen Ausführungsknoten Daten, die von einer oder mehreren Datenspeichervorrichtungen im Datenbankspeicher 106 wiedergewonnen wurden. Folglich verringern oder eliminieren die Caches die Engpassprobleme, die bei Plattformen auftreten, die beständig Daten von entfernten Speichersystemen wiedergewinnen. Anstatt wiederholt auf Daten aus den entfernten Speichervorrichtungen zuzugreifen, greifen die hier beschriebenen Systeme und Verfahren auf Daten aus den Caches in den Ausführungsknoten zu, was signifikant schneller ist und das oben erörterte Engpassproblem vermeidet. Gemäß einigen Beispielen sind die Caches unter Verwendung von Hochgeschwindigkeits-Speichervorrichtungen implementiert, die einen schnellen Zugriff auf die im Cache gespeicherten Daten ermöglichen. Jeder Cache kann Daten von irgendeiner der Speichervorrichtungen im Datenbankspeicher 106 speichern.

[0041] Ferner können die Cache-Betriebsmittel und die Computerbetriebsmittel zwischen verschiedenen Ausführungsknoten variieren. Ein Ausführungsknoten kann z. B. signifikante Computerbetriebsmittel und minimale Cache-Betriebsmittel enthalten, was diesen Ausführungsknoten für Aufgaben nützlich macht, die signifikante Computerbetriebsmittel erfordern. Ein weiterer Ausführungsknoten kann signifikante Cache-Betriebsmittel und minimale Computerbetriebsmittel enthalten, was diesen Ausführungsknoten für Aufgaben nützlich macht, die das Speichern großer Datenmengen in einem Cache erfordern. Ein noch weiterer Ausführungsknoten kann Cache-Betriebsmittel enthalten, die schnellere Eingabe-Ausgabe-Operationen bereitstellen, was für Aufgaben nützlich ist, die ein schnelles Abtasten großer Datenmengen erfordern. Gemäß einigen Beispielen werden die Cache-Betriebsmittel und die Computerbetriebsmittel, die einem speziellen Ausführungsknoten zugeordnet sind, basierend auf den erwarteten Aufgaben, die durch den Ausführungsknoten auszuführen sind, bestimmt, wenn der Ausführungsknoten erzeugt wird.

[0042] Zusätzlich können sich die einem speziellen Ausführungsknoten zugeordneten Cache-Betriebsmittel und Computerbetriebsmittel im Lauf der Zeit basierend auf den durch den Ausführungsknoten ausgeführten sich ändernden Aufgaben ändern. Einem Ausführungsknoten können z. B. mehr Verarbeitungsbetriebsmittel zugewiesen werden, wenn die durch den Ausführungsknoten ausgeführten Aufgaben prozessorintensiver werden. Ähnlich können einem Ausführungsknoten mehr Cache-Betriebsmittel zugewiesen werden, falls die durch den Ausführungsknoten ausgeführten Aufgaben eine größere Cache-Kapazität erfordern.

[0043] Obwohl die virtuellen Warenhäuser 1, 2 und N derselben Ausführungsplattform 110 zugeordnet sind, können die virtuellen Warenhäuser unter Verwendung mehrerer Computersysteme an mehreren geographischen Orten implementiert sein. Das virtuelle Warenhaus 1 kann z. B. durch ein Computersystem an einem ersten geographischen Ort implementiert sein, während die virtuellen Warenhäuser 2 und N durch ein weiteres Computersystem an einem zweiten geographischen Ort implementiert sind. Gemäß einigen Beispielen sind diese verschiedenen Computersysteme um cloudbasierte Computersysteme, die durch eine oder mehrere verschiedene Entitäten aufrechterhalten werden.

[0044] Zusätzlich weist jedes virtuelle Warenhaus, wie in **Fig.** 3 gezeigt ist, mehrere Ausführungsknoten auf. Die mehreren Ausführungsknoten, die jedem virtuellen Warenhaus zugeordnet sind, können unter Verwendung mehrerer Computersysteme an mehreren geographischen Orten implementiert sein. Eine Instanz des virtuellen Warenhauses 1 302a implementiert z. B. den Ausführungsknoten 1 304a und den Ausführungsknoten 2 304b auf einer Computerplattform an einem geographischen Ort und implementiert den Ausführungs-

knoten N 304c auf einer anderen Computerplattform an einem weiteren geographischen Ort. Das Auswählen spezieller Computersysteme, um einen Ausführungsknoten zu implementieren, kann von verschiedenen Faktoren, wie z. B. dem Niveau der für einen speziellen Ausführungsknoten erforderlichen Betriebsmittel (z. B. Verarbeitungsbetriebsmittelanforderungen und Cache-Anforderungen), den an speziellen Computersystemen verfügbaren Betriebsmittel, den Kommunikationsfähigkeiten von Netzen innerhalb eines geographischen Orts oder zwischen geographischen Orten abhängen und davon abhängen, welche Computersysteme bereits andere Ausführungsknoten im virtuellen Warenhaus implementieren.

[0045] Eine spezielle Ausführungsplattform 110 kann irgendeine Anzahl von virtuellen Warenhäusern enthalten. Zusätzlich ist die Anzahl der virtuellen Warenhaus in einer speziellen Ausführungsplattform dynamisch, so dass neue virtuelle Warenhäuser erzeugt werden, wenn zusätzliche Verarbeitungs- und/oder das Caching-Betriebsmittel benötigt werden. Ähnlich können bestehende virtuelle Warenhäuser gelöscht werden, wenn die dem virtuellen Warenhaus zugeordneten Betriebsmittel nicht mehr benötigt werden.

[0046] Gemäß einigen Beispielen können die virtuellen Warenhaushäuser auf denselben Daten im Datenbankspeicher 106 wirken, wobei aber jedes virtuelle Warenhaus seine eigenen Ausführungsknoten mit unabhängigen Verarbeitungs- und Caching-Betriebsmitteln aufweist. Diese Konfiguration ermöglicht es, dass Anforderungen in verschiedenen virtuellen Warenhaushäusern unabhängig und ohne Störung zwischen den Anforderungen verarbeitet werden. Diese unabhängige Verarbeitung, kombiniert mit der Fähigkeit, virtuelle Warenhäuser dynamisch hinzuzufügen und zu entfernen, unterstützt die Ergänzung einer neuen Verarbeitungskapazität für neue Anwender, ohne die durch die bestehenden Anwender beobachtete Leistung zu beeinträchtigen.

[0047] **Fig.** 4A ist ein Einsatzdiagramm einer Computerumgebung 400 zum Bereitstellen einer Web-Anwendung als ein Datenbankobjekt erster Klasse gemäß einigen Beispielen. Eine Datenplattform 102 nutzt die Computerumgebung 400, um einen sicheren Rahmen für eine Anwenderanwendung 410 bereitzustellen, die durch eine Ausführungsplattform 110 der Datenplattform 102 ausgeführt werden soll. Die Anwenderanwendung 410 und alle Komponenten, die die Anwenderanwendung 410 unterstützen, wie z. B. eine Web-Anwendungs-Maschine 408 und ein Server anwenderdefinierter Funktionen (UDF-Server) 406, die hier gemeinsam als eine „Web-Anwendung“ bezeichnet werden, werden durch die Datenplattform 102 als Datenbankobjekte erster Klasse behandelt, die unter Verwendung eines oder mehrerer Befehle innerhalb einer Datenbankabfrage instanziiert werden können, wie durch das Codefragment veranschaulicht ist:

```
ERZEUGE [ODER ERSETZE] WEBAPP [FALLS NICHT EXISTIERT]
<Webapp_name>
VERSION = (
id = <Version_name>
root_location = <App_root>
file_path = <file_name>
)
[WARENHAUS = <warehouse_name>]
[KOMMENTAR = '<comment_string_literal>']
```

[0048] Wobei:

```
<Webapp_name> die Kennung für die Web-Anwendung, die für das Schema, in

dem sie erzeugt wird, eindeutig ist, spezifiziert.
<version_name> die Kennung für die Version der Web-Anwendung spezifiziert.
<app_root> Ein Verweis auf eine Stufen-URL, die auf eine Wurzel der Anwenderan-
wendung 410 zeigt. Wenn die Anwenderanwendung ausgeführt wird, sind die Da-
teien unterhalb dieser app_root für die Web-Anwendungs-Maschine 408 verfügbar.
Obwohl sich die Versionen innerhalb der Datenplattform 102 auf derselben Stufe
befinden können und nur durch Präfixe getrennt sind, kann es nützlich sein, unter-
schiedliche Stufen pro Version aufzuweisen, um die Privilegien und die Reinigung
besser zu managen.
<file_name> Ein Pfad zu einer Anwenderdatei, die als Teil der Web-Anwendungs-
Maschine 408 ausgeführt wird. Dies ist bezüglich der <app_root>.
```

```
<warehouse_name> Ein Name eines virtuellen Warenhauses, wie z. B. des virtuel-
len Warenhauses 1 302a der Datenplattform 102, um die Anwenderanwendung 410
auszuführen.
<comment_string_literal> Kommentar, der die Web-Anwenderanwendung 410 be-
schreibt.
```

[0049] Eine Teilliste von Privilegien, die durch die Sicherheitsmanagerstrategien 420 und/oder die Sandkastenstrategien 422 für die Anwenderanwendung 410 und ihre unterstützenden Komponenten durchgesetzt werden, ist in Tabelle 1 und Tabelle 2 beschrieben:

TABELLE 1

Privileg	**Verwendung**
CREATE WEBAPP	Die Fähigkeit, eine Anwenderanwendung 410 und die ihr zugeordneten Komponenten in einem Schema zu erzeugen.

TABELLE 2

Privileg	**Verwendung**
USAGE	Ermöglicht das Treffen des HTTPS-Endpunkts für die Anwenderanwendung 410 in der Standardversion. Ermöglicht das Sehen der Web-Anwendung unter Verwendung der DESCRIBE- oder SHOW-Befehle
MODIFY	Ermöglicht ALTER-Befehle für die Anwenderanwendung 410
ALL [PRIVILEGES]	Gewährt alle Privilegien mit Ausnahme von OWNERSHIP
OWNERSHIP	Gewährt vollständige Steuerung über die Web-Anwendung; erforderlich, um die Anwenderanwendung 410 fallenzulassen. Nur eine einzige Rolle kann dieses Privileg für ein spezifisches Objekt jeweils besitzen
VIEW_ANY_VERSION	Ermöglicht es einem Anwender, jede der nicht standardmäßigen Versionen zu treffen, die der Anwenderanwendung 410 zugeordnet sind. Dies kann verwendet werden, um neue Versionen zu entwickeln und um einige Anwender vorher veröffentlichte Versionen probieren zu lassen.

[0050] Gemäß einigen Beispielen gibt es Objekte der Datenplattform 102, von denen die Anwenderanwendung 410 abhängt, wie z. B., aber nicht eingeschränkt auf einen Speicherort oder eine Stufe zum Speichern von Dateien und ein virtuelles Warenhaus, wie z. B. das virtuelle Warenhaus 1 302a, in das die Anwenderanwendung 410 geladen wird. Wenn eine Anwenderanwendung 410 und ihre zugeordneten Komponenten, die auf eine Stufe verweisen, erzeugt werden, erbt die Anwenderanwendung 410 die READ-Privilegien für diese Stufe und USAGE-Privilegien für das virtuelle Warenhaus. Falls die Privilegien einer Stufe geändert werden, nachdem eine Anwenderanwendung 410 erzeugt worden ist, so dass der Eigentümer der Anwenderanwendung 410 keine Privilegien mehr für sie hat, schlagen die Anforderungen an die Anwenderanwendung 410 mit einem Fehler fehl, der darlegt, dass die Anwenderanwendung 410 nicht existiert. Falls die Privilegien eines Warenhauses geändert werden, nachdem die Anwenderanwendung 410 erzeugt worden ist, dann wirkt die Logik für das zu verwendende Warenhaus, als ob kein Warenhaus festgelegt wurde. Gemäß einigen Beispielen ist eine Stufe oder ein Datenort in die Anwenderanwendung 410 oder eine ihrer zugehörigen Komponenten eingebettet, wobei die Privilegien für die Anwenderanwendung 410 und die Privilegien der Stufe gemeinsam zugeordnet sind. Gemäß einigen Beispielen können eine Anwenderanwendung 410 und ihre zugehörigen Komponenten mit anderen Eigentümern oder Anwendern gemäß den Privilegien, die in den Sicherheitsmanagerstrategien 420 und/oder den Sandkastenstrategien 422 gespeichert sind, gemeinsam benutzt werden.

[0051] Dementsprechend erben die Anwenderanwendung 410 und alle ihre unterstützenden Komponenten, wenn sie instanziiert werden, alle Attribute eines Objekts erster Klasse innerhalb einer durch die Datenplattform 102 bereitgestellten Datenbank, einschließlich der Privilegien und Einschränkungen, die durch die Datenplattform 102 verwendet werden können, um ein Datenbankobjekt zu managen. Gemäß einigen Bei-

spielen wird die Anwenderanwendung 410 als ein Dienst durch den UDF-Server 406 unter Verwendung der Web-Anwendungs-Maschine 408 bereitgestellt, wobei auf sie über ein Netz, wie z. B. das Internet, durch eine Browser-Laufzeitkomponente 404 der Web-Anwendung zugegriffen werden kann, die in einem Browser 402 enthalten ist, der durch eine Client-Vorrichtung 112 gehostet wird, die Protokolle verwendet, die verwendet werden, um auf Dokumente und Dateien im World Wide Web zuzugreifen.

[0052] Wie bezüglich **Fig.** 2 beschrieben worden ist, implementiert der Rechendienstmanager 104 Sicherheitsstrategien, die die gesamte Kommunikation von der Ausführungsplattform 110 validieren, um zu validieren, dass der Inhalt und der Kontext dieser Kommunikation mit der Aufgabe (den Aufgaben) konsistent sind, von der (denen) bekannt ist, dass sie der Ausführungsplattform 110 zugewiesen ist (sind). Der Ausführungsplattform 110, die eine Abfrage A ausführt, ist es z. B. nicht erlaubt, den Zugriff auf eine spezielle Datenquelle (z. B. die Datenspeichervorrichtung 226 oder irgendeine der Speichervorrichtungen im Datenbankspeicher 106) anzufordern, die für die Abfrage A nicht relevant ist. Gemäß einigen Beispielen kann es sein, dass ein Ausführungsknoten 424 mit einem zweiten Ausführungsknoten kommunizieren muss, wobei aber die hier beschriebenen Sicherheitsmechanismen die Kommunikation mit einem dritten Ausführungsknoten verweigern können. Überdies kann jede derartige unerlaubte Kommunikation (z. B. in einem Protokoll 418 oder an einem anderen Ort) aufgezeichnet werden. Ferner sind die in einem bestimmten Ausführungsknoten gespeicherten Informationen auf die für die aktuelle Abfrage relevanten Daten eingeschränkt, wobei irgendwelche anderen Daten durch Zerstörung oder Verschlüsselung, wobei der Schlüssel nicht verfügbar ist, unbenutzbar sind.

[0053] Gemäß einigen Beispielen sind der UDF-Server 406 und seine Komponenten, wie z. B. die Web-Anwendungs-Maschine 408 und die Anwenderanwendung 410, in einer speziellen Programmiersprache, wie z. B. Python und dergleichen, implementiert. Gemäß einigen Beispielen ist die Browser-Laufzeitkomponente 404 der Web-Anwendung in einer anderen Programmiersprache (z. B. C oder C++) als der UDF-Server 406 implementiert, was die Sicherheit der Computerumgebung 400 unter Verwendung einer anderen Codebasis (z. B. einer ohne die gleichen oder mit weniger potentiellen Sicherheits-Exploits) weiter verbessern kann.

[0054] Der UDF-Server 406 empfängt Kommunikationen von der Browser-Laufzeitkomponente 404 der Web-Anwendung über den globalen Dienstprozess 444 der Datenplattform 102. Der globale Dienstprozess 444 ist für das Empfangen von Anforderungen von der Browser-Laufzeitkomponente 404 der Web-Anwendung verantwortlich. Der globale Dienstprozess 444 verwendet Komponenten des Rechendienstmanagers 104, um verschiedene Authentifikationsaufgaben einschließlich einer ersten Ebene der Autorisierung basierend auf den Sicherheitsstrategien 232 unter Verwendung eines Zugriffsmanagers 202 des Rechendienstmanagers 104 auszuführen. Der UDF-Server 406 führt Aufgaben einschließlich des Zuweisens von Verarbeitungs-Threads, um Anwendercode der Anwenderanwendung 410 auszuführen, und des Zurückschickens der durch die Anwenderanwendung 410 erzeugten Ergebnisse über den globalen Dienstprozess 444 zu der Browser-Laufzeitkomponente 404 der Web-Anwendung aus.

[0055] Gemäß einigen Beispielen wird der UDF-Server 406 innerhalb eines Sandkastenprozesses 414 ausgeführt, wie im Folgernden vollständiger beschrieben wird. Gemäß einigen Beispielen ist der UDF-Server 406 in Python implementiert, das durch einen Interpreter-Prozess interpretiert wird. Gemäß einigen Beispielen ist der UDF-Server 406 in einer weiteren Sprache, wie z. B. Java, implementiert, die durch eine virtuelle Maschine (JVM) ausgeführt wird. Weil der UDF-Server 406 vorteilhaft in einem separaten Prozess bezüglich des Browsers 402 ausgeführt wird, besteht ein geringeres Risiko einer bösartigen Manipulation der Anwenderanwendung 410.

[0056] Die Ergebnisse des Ausführens einer Operation können neben anderen Typen von Informationen oder Nachrichten in einem Protokoll 418 zur Überprüfung und Wiedergewinnung gespeichert sein. Gemäß einigen Beispielen kann das Protokoll 418 lokal im Speicher im Ausführungsknoten 424 oder an einem separaten Ort, wie z. B. dem Datenbankspeicher 106, gespeichert sein.

[0057] Gemäß einigen Beispielen kann ein Sicherheitsmanager 416 den Abschluss einer Operation von einer Anwenderanwendung 410 durch Auslösen einer Ausnahme (z. B. falls die Operation nicht erlaubt ist), oder von Rücksprüngen (z. B. nichts ausführen), falls die Operation erlaubt ist, verhindern. Gemäß einer Implementierung ist der Sicherheitsmanager 416 als ein Sicherheitsmanager-Objekt implementiert, das es einer Anwendung ermöglicht, eine Sicherheitsstrategie, wie z. B. die Sicherheitsmanagerstrategien 420, zu implementieren, und das es der Anwendung ermöglicht, vor dem Ausführen einer möglicherweise unsicheren oder sensiblen Operation zu bestimmen, was die Operation ist und ob sie in einem Sicherheitskontext versucht wird, der die Ausführung der Operation erlaubt. Die Sicherheitsmanagerstrategien 420 können als eine

Datei mit Privilegien implementiert sein, die dem UDF-Server 406 gewährt sind. Der UDF-Server 406 kann deshalb die Operation wenigstens teilweise basierend auf der Sicherheitsstrategie erlauben oder verbieten.

[0058] Gemäß einigen Beispielen verringert der Sandkastenprozess 414 das Risiko von Sicherheitsverletzungen durch das Einschränken der Laufumgebung von nicht vertrauenswürdigen Anwendungen unter Verwendung von Sicherheitsmechanismen, wie z. B. Namensräumen und sicheren Computerbetriebsarten (z. B. unter Verwendung eines Systemaufruffilters auf einen ausführenden Prozess und alle seine Nachkommen, wobei folglich die Angriffsfläche des Kernels eines gegebenen Betriebssystems verringert wird). Überdies ist gemäß einigen Beispielen der Sandkastenprozess 414 ein unmaßgeblicher Prozess und optimiert (z. B. eng an die Sicherheitsmechanismen eines gegebenen Betriebssystem-Kernels gekoppelt), um eine Datenbankabfrage oder eine andere Dienstanforderung in einer sicheren Weise innerhalb der Sandkastenumgebung zu verarbeiten.

[0059] Gemäß einigen Beispielen kann der Sandkastenprozess 414 eine virtuelle Netzverbindung nutzen, um mit anderen Komponenten innerhalb der Computerumgebung 400 zu kommunizieren. Ein spezifischer Satz von Regeln kann für die virtuelle Netzverbindung bezüglich anderer Komponenten der Computerumgebung 400 konfiguriert sein. Derartige Regeln für die virtuelle Netzverbindung können z. B. für einen speziellen UDF-Server 406 konfiguriert sein, um die Orte (z. B. spezielle Sites im Internet oder Komponenten, mit denen der UDF-Server 406 kommunizieren kann) einzuschränken, die durch die durch den UDF-Server 406 ausgeführten Operationen zugänglich sind. Folglich kann gemäß diesem Beispiel dem UDF-Server 406 der Zugriff auf spezielle Netzorte oder Sites im Internet verweigert werden.

[0060] Der Sandkastenprozess 414 kann als eine eingeschränkte Computerumgebung für einen Prozess (oder Prozesse) innerhalb des Sandkastens bereitstellend verstanden werden, in der diese eingeschränkten Prozesse gesteuert und eingeschränkt werden können, um den Zugriff auf bestimmte Computerbetriebsmittel zu begrenzen.

[0061] Die Beispiele der Sicherheitsmechanismen können die Implementierung von Namensräumen, in denen jede jeweilige Gruppe von Prozessen, die innerhalb der Sandkastenumgebung ausgeführt werden, Zugriff auf jeweilige Computerbetriebsmittel (z. B. Prozess-IDs, Host-Namen, Anwender-IDs, Dateinamen, Namen, die dem Netzzugang zugeordnet sind, und Interprozesskommunikation) hat, die für eine weitere Gruppe von Prozessen (die Zugriff auf eine andere Gruppe von Betriebsmitteln haben kann, die für die erstere Gruppe von Prozessen nicht zugänglich ist) nicht zugänglich ist, andere Container-Implementierungen und dergleichen enthalten. Indem der Sandkastenprozess 414 als Unterprozess ausgeführt wird, kann gemäß einigen Beispielen die Latenzzeit beim Verarbeiten einer gegebenen Datenbankabfrage im Vergleich zu anderen Techniken, die eine Lösung einer virtuellen Maschine an sich verwenden können, erheblich verringert werden.

[0062] Wie ferner veranschaulicht ist, kann der Sandkastenprozess 414 Sandkastenstrategien 422 verwenden, um eine gegebene Sicherheitsstrategie durchzusetzen. Die Sandkastenstrategien 422 können Dateien mit Informationen bezüglich einer Konfiguration des Sandkastenprozesses 414 und Einzelheiten hinsichtlich Einschränkungen, falls vorhanden, und Privilegien zum Zugreifen auf die und Verwenden der Systembetriebsmittel sein. Beispielhafte Einschränkungen können Einschränkungen des Netzzugriffs oder des Dateisystemzugriffs enthalten (z. B. Neuzuordnung des Dateisystems, um Dateien an verschiedenen Stellen anzuordnen, die nicht zugänglich sein können, andere Dateien können an verschiedenen Stellen gemountet werden und dergleichen). Gemäß einigen Beispielen schränkt der Sandkastenprozess 414 die Speicher- und Prozessornutzung (z. B. CPU-Nutzung) des UDF-Servers 406 ein, was sicherstellt, dass andere Operationen in demselben Ausführungsknoten ausgeführt werden können, ohne dass die Betriebsmittel knapp werden.

[0063] Die Browser-Laufzeitkomponente 404 der Web-Anwendung stellt ein Frontend für die Anwenderanwendung 410 bereit. Die Browser-Laufzeitkomponente 404 der Web-Anwendung führt Browser-Interaktionen mit der Datenplattform 102 für die Anwenderanwendung 410 aus. Die Komponenten der Computerumgebung 400 kommunizieren unter Verwendung eines Netzanwendungsprotokolls 412, das einen Satz von Befehlen bereitstellt, der für Interaktionen zwischen der Anwenderanwendung 410 und dem Browser 402 verwendet wird. Das Netzanwendungsprotokoll 412 interagiert logisch mit der Anwenderanwendung 410 und durchläuft physisch die Schichten der Datenplattform 102, um sicherzustellen, dass die Sicherheitseinschränkungen und -strategien in jeder Schicht durchgesetzt werden. Diese können Berechtigungen oder Laufzeitanforderungen vom Rechendienstmanager 104 enthalten. Die Browser-Laufzeitkomponente 404 der Web-Anwen-

dung sendet Nachrichten zurück, die durch die Ausführungsplattform 110 verarbeitet werden und auf die mit einer Reihe von Weiterleitungsnachrichten geantwortet wird.

[0064] Die Web-Anwendungs-Maschine 408 enthält Anweisungen, die durch Dritte definiert werden können, die aber als eine Anwendung innerhalb der Ausführungsplattform 110 ausgeführt werden. Die Web-Anwendungs-Maschine 408 stellt Programmierrahmen bereit, damit Anwender Anwendungen, wie z. B. die Anwenderanwendung 410, erstellen können. Gemäß einigen Beispielen ist die Web-Anwendungs-Maschine 408 in Python geschrieben und wird durch die Ausführungsplattform 110 als spezielle gespeicherte Python-Prozeduren behandelt. Gemäß einigen Beispielen ist die Web-Anwendungs-Maschine 408 in einer weiteren Sprache geschrieben, wie z. B., aber nicht eingeschränkt auf Java, wobei sie durch eine virtuelle Maschine innerhalb der Ausführungsplattform 110 gehostet ist. Gemäß einigen Beispielen können Dritte ihre eigenen Web-Anwendungs-Maschinen erstellen.

[0065] Die Anwenderanwendung 410 umfasst eine Anwendung, die durch einen Anwender geschrieben worden ist und durch die Web-Anwendungs-Maschine 408 ausgewertet worden ist. Gemäß einigen Beispielen umfasst die Anwenderanwendung 410 Python-Dateien, die durch einen proprietären Python-Interpreter ausgewertet werden.

[0066] Der UDF-Server 406 ist für die Ausführung von UDFs in einer gesteuerten Ausführungsumgebung, wie z. B. dem Sandkastenprozess 414, verantwortlich. Gemäß einigen Beispielen umfasst der UDF-Server 406 einen Python-UDF-Server. Gemäß einigen Beispielen verwendet der UDF-Server 406 andere Sprachen, wie z. B. Java.

[0067] Gemäß einigen Beispielen ist eine dem UDF-Server 406 zugewiesene Identifizierung eines „Uniform Resource Locator“ (URL-Identifizierung) ein eindeutiger Wert, der über Replikationen des UDF-Servers 406 stabil ist. Die URL-Identifizierung ist z. B. eine zufällig erzeugte Zeichenfolge, die innerhalb eines Kontos eines Eigentümers eindeutig ist. Die URL-Identifizierung kann unter Verwendung einer UUID4- und Base64-Codierung erzeugt werden, um ihr eine knappere Darstellung zu geben.

[0068] Gemäß einigen Beispielen wird ein Schemaobjekt der Datenplattform 102 verwendet, um die Komponenten dessen zu definieren, das die Web-Anwendung definiert, wie z. B., aber nicht eingeschränkt auf den UDF-Server 406, die Web-Anwendungs-Maschine 408, die Anwenderanwendung 410 und die Browser-Laufzeitkomponente 404 der Web-Anwendung. Der Name, der Netz-Endpunkt, die Privilegien und die Strategien basieren auf diesem Objekt. Gemäß einigen Beispielen enthält das Schemaobjekt sowohl eine spezielle Version einer zu verwendenden Web-Anwendungs-Maschine 408 als auch irgendwelche Betriebsmittel-Beschränkungen.

[0069] Gemäß einigen Beispielen ist die Version des Codes eines Anwenders spezifiziert, wobei sie eine benannte Version einer Web-Anwendung zuordnet, die sich auf einen Ort in einer Stufe bezieht, die durch die Datenplattform 102 verwendet wird, um Anwendercode auszuführen.

[0070] **Fig.** 4B, **Fig.** 4C und **Fig.** 4D sind Interaktions- und Datenflussdiagramme der Computerumgebung 400 zum Bereitstellen einer Web-Anwendung als ein Datenbankobjekt erster Klasse gemäß einigen Beispielen.

[0071] Die Computerumgebung 400 verwendet Zeilensatzoperatoren (RSOs), die als Teil eines Programms in der Ausführungsplattform 110 ausgeführt werden. Eine RSOI ist eine Instanz eines RSO, die auf einen Verarbeitungs-Thread der Ausführungsplattform 110 wirkt. Eine RSOI-Erweiterungsfunktion ist eine RSOI, die gespeicherte Prozeduren ausführt. Gemäß einigen Beispielen sind die gespeicherten Prozeduren in Python geschrieben. Gemäß einigen Beispielen sind die gespeicherten Prozeduren in Java geschrieben.

[0072] Ein Eigentümer einer RSOI-Erweiterungsfunktion 436 legt die Berechtigungen für die RSOI-Erweiterungsfunktion 436 selbst fest, die sowohl definieren, wer sie verwenden darf, als auch eine andere Rolle festlegen, um sie auszuführen. Das Festlegen dieser Berechtigungen geschieht, wenn ein Anwender die RSOI-Erweiterungsfunktion 436 definiert, wobei die Durchsetzung in dem Web-Anwendungs-Betriebsmittel 442 geschieht, wenn der Browser auf eine der Web-Anwendung zugeordnete URL zugreift. Das Web-Anwendungs-Betriebsmittel 442 startet außerdem den Auftrag, der seinen Berechtigungskontext festlegt. Die RSOI-Erweiterungsfunktion 436 arbeitet innerhalb dieses Berechtigungs-/Sitzungskontexts. Gemäß einigen Beispielen legt ein Eigentümer die Berechtigungen für jedes der Objekte fest, die instanziiert werden, wie z. B. ein UDF-Server 406, eine Web-Anwendungs-Maschine 408 und eine Anwenderanwendung 410, aber nicht

darauf eingeschränkt sind.
####

[0073] Bei der Operation 1 verwendet der durch die Client-Vorrichtung 112 gehostete Browser 402 eine WebSocket-Verbindung mit einem Web-Anwendungs-Betriebsmittel 442, um mit der durch eine Datenplattform 102 gehosteten Anwenderanwendung 410 zu kommunizieren. Der Zugriffsmanager 202 des Rechendienstmanagers 104 der Datenplattform 102 autorisiert den Zugriff auf die Anwenderanwendung 410 basierend auf einem Satz von Sicherheitsstrategien 232, die in der Datenspeichereinrichtung 206 gespeichert sind. Die Datenplattform 102, wenn sie autorisiert ist, stellt die Anwenderanwendung 410 einem Anwender bereit, der den Browser 402 verwendet. Die Browser-Laufzeitkomponente 404 der Web-Anwendung zieht Nachrichten vom WebSocket zurück und gibt geeignete Befehle aus.

[0074] Falls es keine aktuelle Sitzung mit einer Instanz der Web-Anwendungs-Maschine 408 gibt, verifiziert die Browser-Laufzeitkomponente 404 der Web-Anwendung, dass die Anforderung die Berechtigungen aufweist, um die Anwendung unter Verwendung eines für die Anwenderanwendung 410 definierten Sicherheitskontexts zu verwenden, wobei sie dann den Start eines Auftrags durch den UDF-Server 406 anfordert. Ein anfänglicher Ausführungsplan startet eine Instanz einer Web-Anwendungs-Maschine 408 für den Auftrag. Die Web-Anwendungs-Maschine 408 wird basierend auf den Sicherheitsmanagerstrategien 420 und den Sandkastenstrategien 422 instanziiert, die durch den Sicherheitsmanager 416 bzw. dem Sandkastenprozess 414 durchgesetzt werden, wobei das Ereignis im Protokoll 418 protokolliert wird. Nachdem eine Sitzung der Web-Anwendungs-Maschine 408 gestartet worden ist, sendet die Web-Anwendungs-Maschine 408 Befehle an einen Abfragekoordinator 430.

[0075] Bei der Operation 2 weist der Auftrag den Abfragekoordinator 430 auf, der dem Auftrag zugeordnet ist. Von diesem Punkt an geschieht die Kommunikation mit der Web-Anwendungs-Maschine 408 durch Hinzufügen von Abfragekoordinatorereignissen des Typs „Anwendungsinteraktion“ zum Abfragekoordinator 430. Dies führt zu einer Ausführungsanforderung, die in die Warteschlange eingereiht wird. Das Abfragekoordinatorereignis weist außerdem einen Verweis auf einen Strom auf, durch den er Vorwärtsnachrichten senden kann, um zu dem Browser 402 zu gelangen. So kann der Abfragekoordinator 430 die Antwortereignisse erhalten und sie direkt zu dem Browser 402 zurückleiten.

[0076] Bei der Operation 3 wird eine Anwendungsanforderungs-Warteschlange 426 bereitgestellt. Die Anwendungsanforderungs-Warteschlange 426 ist eine Warteschlange im Speicher, in die Rückmeldungen geschoben werden. Gemäß einigen Beispielen befindet sich die Anwendungsanforderungs-Warteschlange 426 im Speicher, um sicherzustellen, dass die Verbindung in der Operation 4 immer zu derselben globalen Dienstinstanz zurückgeht, auf der sich der Abfragekoordinator 430 befindet. Gemäß einigen Beispielen ist es im Fall einer Störung des globalen Dienstes zulässig, die Nachrichten zu verlieren und die Browser-Laufzeitkomponente 404 der Web-Anwendung den Zustand durch eine neue Ausführungsanforderung wiederherstellen zu lassen.

[0077] Bei der Operation 4 startet eine RSOI-Erweiterungsfunktion 436 eine spezielle gespeicherte Prozedur, die während einer langen Zeit läuft. Diese gespeicherte Prozedur läuft in dem Sicherheitskontext, der für die Objekte konfiguriert wurde, die in dem Ausführungsknoten 424 basierend auf den Sicherheitsmanagerstrategien 420 und den Sandkastenstrategien 422 instanziiert werden. Die gespeicherte Prozedur erzeugt einen UDF-Server 406, der Skripte sicher in der gesperrten Umgebung des Ausführungsknotens 424 der Ausführungsplattform 110 ausführt. Die RSOI-Erweiterungsfunktion 436 startet die Web-Anwendungs-Maschine 408 basierend auf der gespeicherten Prozedur durch den UDF-Server 406 unter Verwendung von RPC-Aufrufen. Die RSOI-Erweiterungsfunktion 436 ruft eine „Ausführungsprozedur“ mit Funktionsinformationen auf, die der gespeicherten Prozedur der Web-Anwendungs-Maschine 408 mitteilen, nicht zu enden. Die Web-Anwendungs-Maschine 408 wird basierend auf den Sicherheitsmanagerstrategien 420 und den Sandkastenstrategien 422 instanziiert, die durch den Sicherheitsmanager 416 bzw. den Sandkastenprozess 414 während eines Verifizierungsprozesses durchgesetzt werden, wobei das Ereignis im Protokoll 418 protokolliert wird. Die Prozedur der Web-Anwendungs-Maschine 408 verbindet sich mit einem RPC-Endpunkt einer Stromanwendungsanforderung auf dem UDF-Server 406 und gibt eine Initialisiere-die-Anwendung-Nachricht aus. Diese Nachricht wird verwendet, um die Web-Anwendungs-Maschine 408 mit den geeigneten Strategie- und Dateiinformationen urzuladen, um die Anwenderanwendung 410 auszuführen. Diese Informationen kommen als Teil des Ausführungsplans der Operation 1 herunter, die die Anwenderanwendung 410 startet. Die RSOI-Erweiterungsfunktion 436 verbindet sich mit einem Endpunkt des Web-Anwendungs-Interaktionskanals 434 in einem Ausführungsplattform-Betriebsmittel 432 und verarbeitet die Nachrichten, die durch diesen Kanal hereinkommen. Der UDF-Server 406 kann dann die Web-Anwendungs-Maschine 408 und die Anwen-

deranwendung 410 unter Verwendung der speziellen gespeicherten Prozedur für Proxy-Kommunikationen vom Browser mit geringer Latenzzeit und in einer effizienten Weise ausführen.

[0078] Gemäß einigen Beispielen basiert ein Netz-Endpunkt auf den URLs, die auf dem Konto-Lokalisierer basieren, wobei er folglich eine Domain für das Konto jedes Eigentümers bereitstellt, um als Browser-Sicherheitsgrenze zu wirken. Gemäß zusätzlichen Beispielen sind die Komponenten einer URL-Kennung eindeutige und nicht erratbare Zahlen. Die URL-Identifizierungen sind über Umbenennungen und Replikation auf andere Konten stabil. Ein Anwender des Browsers 402 kann direkt auf die URL zugreifen. Um dies auszuführen, muss er bei der Datenplattform 102 angemeldet sein, wobei er Nutzungsprivilegien für die Anwenderanwendung 410 benötigt.

[0079] Die Nachrichten, die von dem Ausführungsplattform-Betriebsmittel 432 kommen, sind anwendergesteuerte Interaktionen, die entweder aus dem Verwenden oder aus dem Bearbeiten einer Anwendung hervorgehen. Die Nachrichten, die zu dem UDF-Server 406 gehen, sind in einem Dokument definiert, das die UDF-Anwendungsanforderungen definiert. Für Anwendungs-Anwenderanforderungen leitet die RSOI-Erweiterungsfunktion 436: 1. Nachrichtenanforderungen zurück zu einem geeigneten UDF-Server 406 und hinterlässt sie, um durch die Web-Anwendungs-Maschine 408 verarbeitet zu werden; 2. Eine Aktualisiere-die-Datei-Nachrichtenanforderung befiehlt der RSOI-Erweiterungsfunktion 436: a. Dateien zu finden, die aktualisiert werden müssen; b. einen Aktualisiere-die-Datei-Startbefehl auszugeben; c. die Aktualisiere-die-Datei-Befehle zu befolgen, die erforderlich sind, um die geeigneten Dateien zu aktualisieren, die durch die Web-Anwendungs-Maschine 408 verwendet werden; und d. eine Aktualisiere-die-Dateien-Endnachricht zu erzeugen. Der gesamte Zugriff auf die Ausführungsplattform 110, wie z. B. auf die im Datenbankspeicher 106 gespeicherten Daten und die durch die Ausführungsplattform 110 ausgeführten zusätzlichen Funktionen und Prozeduren, aber nicht darauf eingeschränkt, durch die Anwenderanwendung 410 unter Verwendung der Web-Anwendungs-Maschine 408 wird durch den Sicherheitsmanager 416 und den Sandkastenprozess 414 unter Verwendung der Sicherheitsmanagerstrategien 420 bzw. der Sandkastenstrategien 422 verifiziert. Dies ermöglicht der Ausführungsplattform 110, dem Browser 402 durch die Anwenderanwendung 410 Dienste bereitzustellen, ohne zu erfordern, die Daten zwischen einem gesicherten Ort und einem ungesicherten Ort mit der Ausführungsplattform 110 zu bewegen, wobei folglich die Skalierbarkeit und die Sicherheit verbessert werden. Gemäß einigen Beispielen wird für den HTTP-Kanal für einen „Ausführungsplattform-Betriebsmittel“-Befehl der ContentType = Anwendungs-/Oktett-Strom verwendet, wobei das Protobuf-Protokoll in beiden Richtungen verwendet wird. Gemäß einigen Beispielen wird das Protobuf-Protokoll in beiden Richtungen in einer codierten Weise verwendet, die das Schieben mehrerer Nachrichten einen Strom hinunter ermöglicht, ohne die TCP-Verbindung schließen und eine Anforderung neu ausgeben zu müssen.

[0080] Bei der Operation 5 managt der UDF-Server 406 in **Fig.** 4D den Lebenszyklus der Web-Anwendungs-Maschine 408. Der UDF-Server 406 startet und managt dann Anforderungen, die durch einen Endpunkt der Stromanwendungsanforderungen hereinkommen. In **Fig.** 4D gibt es gemäß einigen Beispielen zwei grundlegende Typen von Nachrichten: Anwendungs-Anwenderanforderungen 440 und Anwendungs-Steuerebenenanforderungen 438. Die Anwendungs-Anwenderanforderungen 440 werden direkt in ein Ausführungsverfahren für die Web-Anwendungs-Maschine 408 weitergeleitet. Die Anwendungs-Steuerebenenanforderungen 438 werden an den UDF-Server 406 geleitet, um irgendeine Systemoperation auszuführen, wie z. B., aber nicht eingeschränkt auf das Aktualisieren von Dateien, das Initialisieren von Anwendungen oder das Herunterfahren von Dingen.

[0081] Bei der Operation 6 enthält eine Web-Anwendungsprozedur zusätzliche Lebenszyklusfunktionen, die aufgerufen werden. Eine beispielhafte Web-Anwendungsprozedur ist teilweise in dem Codefragment veranschaulicht:

```
// Wird aufgerufen, wenn eine InitializeApp-Nachricht hereinkommt
def start(webApp: WebApp, config: ConfigurationParameters)
// Hauptfunktion zur Verarbeitung von App-Anforderungen.
// Wird für jede BackMsg aufgerufen, die hereinkommt
def run(webApp: WebApp, message: BackMsg, responseQueue queue<Forward-
Msg>, sessionCtx: SessionCtx)
// Wird aufgerufen, wenn die Dateien aktualisiert werden

def beforeFilesChanged(webApp: WebApp, files: List<Files>)
def afterFilesChanged(webApp: WebApp, files: List<Files>)
```

```
// Wird aufgerufen, bevor die App endet

def stop(webApp: WebApp)
```

[0082] Gemäß einigen Beispielen weiß der UDF-Server 406, welche Funktionen welchen Operationen zugeordnet sind, indem sie in einem gespeicherten Datenpersistenzobjekt der Prozedur spezifiziert sind und als Teil des Startens der Web-Anwendungs-Maschine 408 weitergegeben werden. Gemäß einigen Beispielen weiß der UDF-Server 406, welche Funktionen welchen Operationen zugeordnet sind, da eine gespeicherte Prozedur DPO einen Handler als eine Startfunktion aufweist und ein Rückgabetyp der Startfunktion eine Tabelle von Funktionen zurückgibt, die andere Funktionen spezifiziert. Gemäß einigen Beispielen weiß der UDF-Server 406, welche Funktionen welchen Operationen zugeordnet sind, da eine gespeicherte Prozedur DPO einen Handler als die Startfunktion aufweist, wobei Anmerkungen bereitgestellt werden, nach denen der UDF-Server 406 suchen kann, um andere Funktionen zu finden. Gemäß einigen Beispielen markiert eine neue Eigenschaft in einer gespeicherten Prozedur die gespeicherte Prozedur als eine Web-Anwendungs-Maschine 408. Wenn dies festgelegt ist, ruft ein Handler eine Funktion auf, die eine Tabelle der Funktionen zurückschickt, die auf die verschiedenen obigen Anwendungslebenszyklusereignisse abgebildet werden (z. B. „ausführen“, „Dateien geändert“, usw.)

[0083] Bei der Operation 7 kommen die Antworten vom UDF-Server 406 durch ein Ausführungsplattform-Betriebsmittel zurück. Wenn die Anwenderanwendung 410 Nachrichten zu einer Anwendungsantwort-Warteschlange 428 hinzufügt, nimmt der UDF-Server 406 diese Antworten und leitet sie durch den RPC-Endpunkt zu der RSOI-Erweiterungsfunktion 436 zurück, die sie dann eine Langabfrage-HTTP-Verbindung hinab an den Web-App-Interaktionskanal des Ausführungsplattform-Betriebsmittels hinunterschickt. Das Ausführungsplattform-Betriebsmittel legt die Antworten in die Anwendungsantwort-Warteschlange 428 und benachrichtigt den Abfragekoordinator 430.

[0084] Bei der Operation 8 sendet der Abfragekoordinator 430 die Antwort zurück an den Browser 402. Der Abfragekoordinator 430 nimmt die Ereignisse auf und filtert irgendwelche Nachrichten heraus, die die Strategie verletzen (z. B. uneingeschränktes JS oder HTML). Da der Abfragekoordinator 430 bereits mit dem Antwortkanal versehen wurde, als er das Abfragekoordinatorereignis erhielt, verwendet er diesen, um eine Antwort zurückzusenden. Gemäß einigen Beispielen sendet das Ausführungsplattform-Betriebsmittel die Anforderung selbst, wenn sichergestellt ist, dass sich der WebSocket im gleichen globalen Dienst wie der Abfragekoordinator 430 befindet. Wenn nicht sichergestellt werden kann, dass sich der WebSocket im selben globalen Dienst wie der Abfragekoordinator 430 befindet, führt gemäß einigen Beispielen der Abfragekoordinator 430 die Operation des Zurücksendens der Antwort aus, da es bereits eine Funktionalität gibt, um die richtige globale Dienstinstanz für den Abfragekoordinator 430 zu finden.

[0085] **Fig.** 5 ist ein Einsatzdiagramm für eine Computerumgebung 500 einer Datenplattform 102, die einen vereinheitlichten Sicherheitskontext 502 für eine Entwicklung und Verwendung einer Anwenderanwendung 410 gemäß einigen Beispielen der vorliegenden Offenbarung bereitstellt. Die Computerumgebung 500 unterstützt den vereinheitlichten Sicherheitskontext 502 für die Komponenten der Datenplattform 102, die eine Software-Entwicklungsumgebung zum Entwickeln von Quellcode, wie z. B. den Anwenderanwendungs-Quellcode 512 einer Anwenderanwendung, wie z. B. der Anwenderanwendung 410, bereitstellen. Die Computerumgebung 500 unterstützt ferner als Teil des vereinheitlichten Sicherheitskontextes 502 eine Produktionsumgebung für die Bereitstellung der Anwenderanwendung 410 als Teil einer Web-Anwendung, die Anwendern ermöglicht, auf die Betriebsmittel der Datenplattform 102, wie z. B. die Datenbankobjekte 510, die in einem Datenbankspeicher 106 gespeichert sind, zuzugreifen.

[0086] Ein Zugriffsmanager 202 autorisiert den Zugriff auf Komponenten und Objekte des vereinheitlichten Sicherheitskontextes 502 basierend auf Privilegien, die in den Sicherheitsstrategien 506 spezifiziert sind, die eine oder mehrere Sicherheitsstrategien für die Komponenten und Objekte innerhalb des vereinheitlichten Sicherheitskontextes 502 umfassen. Die Sicherheitsstrategien 506 enthalten, sind aber nicht eingeschränkt auf die Sicherheitsstrategien, die auf den Anwenderanwendungs-Quellcode 512, der auf einer Stufe der Datenplattform 102 gespeichert ist, und auf die Datenbankobjekte 510, die im Datenbankspeicher 106 gespeichert sind, angewendet werden, und die Sicherheitsstrategien, die auf die Nutzung und das gemeinsame Benutzen der Anwenderanwendung 410 angewendet werden. Die Sicherheitsstrategien 506 enthalten ferner Sicherheitsmanagerstrategien 420, die durch den Sicherheitsmanager 416 durchgesetzt werden, und Sandkastenstrategien 422, die durch den Sandkastenprozess 414 durchgesetzt werden, wie bezüglich **Fig.** 4A, **Fig.** 4B, **Fig.** 4C und **Fig.** 4D beschrieben worden ist.

[0087] Die Computerumgebung 500 enthält einen Datenerforscher 228, der durch die Anwender zum Zugriff auf die im Datenbankspeicher 106 gespeicherten Datenbankobjekte 510 verwendet wird. Der Datenerforscher 228 umfasst einen Visualisierer 508, den ein Anwender verwenden kann, um auf Daten der Datenbankobjekte 510 zuzugreifen und diese anzuzeigen. Der Datenerforscher 228 umfasst ferner einen Editor, den ein Anwender verwenden kann, um Datenbankabfragen und Quellcode für Software, wie z. B. den Anwenderanwendungs-Quellcode 512 für die Anwenderanwendung 410, zu erzeugen, zu bearbeiten und/oder zu speichern, der verwendet wird, um auf Daten der Datenbankobjekte 510 zuzugreifen und diese zu modifizieren. Der Zugriffsmanager 202 verwendet eine oder mehrere Sicherheitsstrategien der Sicherheitsstrategien 506, um zu bestimmen, ob ein Anwender autorisiert ist, den Datenerforscher 228 zu verwenden, um auf eine Komponente oder ein Datenbankobjekt der Datenplattform 102 zuzugreifen, die bzw. das durch den vereinheitlichten Sicherheitskontext 502 umfasst ist. Derartige Anwender enthalten, sind aber nicht eingeschränkt auf einen Anwender der Datenplattform 102, der ein Eigentümer der Anwenderanwendung 410 und/oder der Datenbankobjekte 510 ist, einen Anwender innerhalb desselben Kontos des Eigentümers, einen Anwender eines weiteren Kontos, das sich von dem Konto des Eigentümers unterscheidet, ein externes System außerhalb der Datenplattform 102, usw.

[0088] Die Computerumgebung 500 enthält ferner einen Anwendungs- und Datenmanager 230, der es Anwendern ermöglicht, zu spezifizieren, welche anderen Entitäten die Verwendung der Anwenderanwendung 410 teilen können und auf Daten der Datenbankobjekte 510 zugreifen und/oder diese modifizieren können. Der Zugriffsmanager 202 verwendet eine oder mehrere Sicherheitsstrategien der Sicherheitsstrategien 506, um zu bestimmen, ob ein Anwender autorisiert ist, den Anwendungs- und Datenmanager 230 zu verwenden, um auf die Sicherheitsstrategien 506 zuzugreifen, um Sicherheitsstrategien festzulegen, die es einem weiteren Anwender erlauben, auf eine Komponente oder ein Datenbankobjekt der Datenplattform 102 zuzugreifen, die bzw. das durch den vereinheitlichten Sicherheitskontext 502 umfasst ist. Derartige Anwender enthalten, sind aber nicht eingeschränkt auf einen Anwender der Datenplattform 102, der ein Eigentümer der Anwenderanwendung 410 und/oder der Datenbankobjekte 510 ist, einen Anwender innerhalb desselben Kontos des Eigentümers, einen Anwender eines weiteren Kontos, das sich von dem Konto des Eigentümers unterscheidet, ein externes System außerhalb der Datenplattform 102 usw.

[0089] Der Sandkastenprozess 414, der Sicherheitsmanager 416, der UDF-Server 406 und die Web-Anwendungs-Maschine 408 arbeiten, um eine sichere Ausführungsumgebung für die Anwenderanwendung 410 bereitzustellen, wie hier bezüglich **Fig.** 4A, **Fig.** 4B, **Fig.** 4C und **Fig.** 4D beschrieben worden ist. Der Sandkastenprozess 414 und der Sicherheitsmanager 416 verwenden ihre jeweiligen Sicherheitsstrategien, die in den Sicherheitsstrategien 506 enthalten sind, um den Zugriff auf die Anwenderanwendung 410 durch einen Anwender unter Verwendung eines Browsers 402 bereitzustellen, der eine Browser-Laufzeitkomponente 404 der Web-Anwendung aufweist und durch eine Client-Vorrichtung 112 gehostet wird, wie hier bezüglich **Fig.** 4A, **Fig.** 4B, **Fig.** 4C und **Fig.** 4D beschrieben worden ist.

[0090] Gemäß einigen Beispielen verwendet der Zugriffsmanager 202 eine oder mehrere Sicherheitsstrategien der Sicherheitsstrategien 506, um zu bestimmen, ob ein Anwender auf den Anwenderanwendungs-Quellcode 512 unter Verwendung eines Editors zugreifen darf, der durch eine Web-Anwendung bereitgestellt wird, die einen (nicht gezeigten) externen Editor bereitstellt. Der externe Editor wird gemäß den in **Fig.** 4A, **Fig.** 4B, **Fig.** 4C und **Fig.** 4D beschriebenen Methodologien bereitgestellt. Gemäß einigen Beispielen umfasst eine Anwenderanwendung der Web-Anwendung einen Editor, der durch die Ausführungsplattform 110 ausgeführt wird, wobei eine UI des Editors durch eine Browser-Laufzeitkomponente der Web-Anwendung eines Browsers bereitgestellt wird, der durch eine Client-Vorrichtung gehostet wird. Gemäß zusätzlichen Beispielen umfasst eine Anwenderanwendung der Web-Anwendung einen Proxy-Server, der durch die Ausführungsplattform 110 ausgeführt wird, wobei eine Browser-Laufzeitkomponente der Web-Anwendung eines Browsers einer Client-Vorrichtung einen Editor umfasst, der auf den Anwenderanwendungs-Quellcode 512 durch die Anwenderanwendung zugreift, die als der Proxy-Server wirkt.

[0091] **Fig.** 6 ist ein Aktivitätsdiagramm, das ein Verfahren 600 zum Entwickeln und Einsetzen einer Anwenderanwendung 410 in einem vereinheitlichten Sicherheitskontext 502 durch eine Datenplattform 102 gemäß einigen Beispielen der vorliegenden Offenbarung veranschaulicht.

[0092] Bei der Operation 602 stellt die Datenplattform 102 einem ersten Anwender 614 einen Editor 504 bereit, den der erste Anwender 614 verwendet, um basierend auf den Sicherheitsstrategien 506 den Anwenderanwendungs-Quellcode 512 einer Anwenderanwendung 410 zu erzeugen, zu bearbeiten und/oder zu speichern. Der Zugriff auf den Anwenderanwendungs-Quellcode 512 durch den ersten Anwender 614 unter Verwendung des Editors 504 wird durch einen Zugriffsmanager 202 bestimmt, der den ersten Anwender 614

autorisiert, den Editor zu verwenden, um den Anwenderanwendungs-Quellcode 512 in einem Datenbankspeicher 106 der Datenplattform 102 basierend auf den Sicherheitsstrategien 506 zu erzeugen, zu bearbeiten und/oder zu speichern.

[0093] Bei der Operation 604 verwendet der erste Anwender 614 einen Anwendungs- und Datenmanager 230, um Teilungs- und Nutzungsprivilegien für die Datenbankobjekte 510 und die Anwenderanwendung 410 durch Festlegen der in den Sicherheitsstrategien 506 gespeicherten Privilegien festzulegen. Der Zugriff durch den Anwender auf die Sicherheitsstrategien 506 unter Verwendung des Anwendungs- und Datenmanagers 230 wird durch den Zugriffsmanager 202 bestimmt, der den ersten Anwender 614 autorisiert, basierend auf den Sicherheitsstrategien 506 die Teilungs- und Nutzungsprivilegien zu erzeugen, zu bearbeiten und zu speichern. Die Privilegien erlauben die Autorisierung zusätzlicher Anwender und Systeme, wie z. B. eines zweiten Anwenders 616, um die Anwenderanwendung 410 zu verwenden, um auf die Daten der Datenbankobjekte 510 und andere Komponenten der Datenplattform 102 zuzugreifen. Die Teilungs- und Nutzungsprivilegien sind in den Sicherheitsstrategien 506 gespeichert.

[0094] Bei der Operation 606 setzt die Datenplattform 102 die Anwenderanwendung 410 ein, wobei sie folglich die Anwenderanwendung 410 für einen oder mehrere Anwender, wie z. B. den zweiten Anwender 616, verfügbar macht.

[0095] Bei der Operation 608 empfängt die Datenplattform 102 eine Anforderung von dem zweiten Anwender 616, um auf Komponenten und/oder Daten der Datenbankobjekte zuzugreifen, die sich auf der Datenplattform 102 befinden. In Reaktion stellt die Datenplattform 102 dem zweiten Anwender 616 basierend auf den Sicherheitsstrategien 506 die Anwenderanwendung 410 bereit. Die Bereitstellung der Anwenderanwendung 410 für den zweiten Anwender 616 wird durch den Zugriffsmanager 202 bestimmt, der den zweiten Anwender 616 autorisiert, die Anwenderanwendung 410 zu verwenden, um auf die Komponenten und Datenbankobjekte 510 der Datenplattform 102 basierend auf den Sicherheitsstrategien 506 zuzugreifen.

[0096] Bei der Operation 610 greift die Anwenderanwendung 410 basierend auf den Sicherheitsstrategien 506 auf die Komponenten der Datenplattform 102 und die Daten der Datenbankobjekte 510 zu. Die Interaktionen der Anwenderanwendung 410 mit den Komponenten und den Datenbankobjekten 510 der Datenplattform 102 werden teilweise durch einen Sicherheitsmanager 416 basierend auf den Sicherheitsmanagerstrategien 420 der Sicherheitsstrategien 506 autorisiert. Zusätzlich werden die Interaktionen der Anwenderanwendung 410 mit den Komponenten und den Datenbankobjekten der Datenplattform 102 teilweise durch einen Sandkastenprozess 414 basierend auf den Sandkastenstrategien 422, die in den Sicherheitsstrategien 506 enthalten sind, autorisiert.

[0097] Bei der Operation 612 stellt die Anwenderanwendung 410 dem zweiten Anwender 616 die Daten der Datenbankobjekte 510 bereit. Die Bereitstellung der Daten der Datenbankobjekte 510 für den zweiten Anwender 616 über die Anwenderanwendung 410 wird durch den Zugriffsmanager 202 bestimmt, der den zweiten Anwender 616 autorisiert, um die Anwenderanwendung 410 zu verwenden, um auf die Komponenten und die Datenbankobjekte 510 der Datenplattform 102 basierend auf den Sicherheitsstrategien 506 zuzugreifen.

[0098] Gemäß einigen Beispielen wird unter Verwendung eines Editors, der durch eine (nicht gezeigte) Web-Anwendung bereitgestellt wird, auf den Anwenderanwendungs-Quellcode 512 zugegriffen und er bearbeitet.

[0099] Gemäß einigen Beispielen erlaubt eine Anwenderanwendung 410 den Zugriff auf die Datenplattform 102 durch ein externes System. Die Anwenderanwendungs-Datenplattform 102 wirkt z. B. als ein Proxy-Server, der es dem externen System ermöglicht, auf die Komponenten und die Datenbankobjekte 510 der Datenplattform zuzugreifen.

[0100] **Fig.** 7 veranschaulicht eine graphische Darstellung einer Maschine 700 in der Form eines Computersystems, in dem ein Satz von Anweisungen ausgeführt werden kann, um die Maschine 700 zu veranlassen, irgendeine oder mehrere der hier erörterten Methodologien auszuführen, gemäß einigen Beispielen der vorliegenden Offenbarung. Spezifisch zeigt **Fig.** 7 eine graphische Darstellung der Maschine 700 in der beispielhaften Form eines Computersystems, innerhalb dessen die Anweisungen 702 (z. B. Software, ein Programm, eine Anwendung, ein Applet, eine App oder ein anderer ausführbarer Code) zum Veranlassen der Maschine 700, irgendeine oder mehrere der hier erörterten Methodologien auszuführen, ausgeführt werden können. Die Anweisungen 702 können z. B. die Maschine 700 veranlassen, irgendeine oder mehrere Operationen irgendeines oder mehrerer der hier beschriebenen Verfahren auszuführen. In dieser Weise transformieren

die Anweisungen 702 eine allgemeine, nicht programmierte Maschine in eine spezielle Maschine 700 (z. B. den Rechendienstmanager 104, die Ausführungsplattform 110 und die Datenspeichervorrichtungen 1 bis N des Datenbankspeichers 106), die speziell konfiguriert ist, irgendeine der beschriebenen und veranschaulichten Funktionen in der hier beschriebenen Weise auszuführen.

[0101] Gemäß alternativen Beispielen arbeitet die Maschine 700 als eine eigenständige Vorrichtung oder kann die Maschine 700 mit anderen Maschinen gekoppelt (z. B. vernetzt) sein. In einem vernetzten Einsatz kann die Maschine 700 in der Kapazität einer Server-Maschine oder einer Client-Maschine in einer Server-Client-Netzumgebung oder als eine Peer-Maschine in einer Peer-zu-Peer-Netzumgebung (oder verteilten Netzumgebung) arbeiten. Die Maschine 700 kann einen Server-Computer, einen Client-Computer, einen Personal-Computer (PC), einen Tablet-Computer, einen Laptop-Computer, ein Netbook, ein Smartphone, eine Mobilvorrichtung, einen Netz-Router, einen Netz-Switch, eine Netz-Bridge oder jede andere Maschine, die die Anweisungen 702 sequentiell oder anderweitig ausführen kann, die die durch die Maschine 700 zu ergreifenden Maßnahmen spezifizieren, umfassen, ist aber nicht darauf eingeschränkt. Während nur eine einzige Maschine 700 veranschaulicht ist, sollte ferner der Begriff „Maschine“ außerdem so ausgelegt werden, dass er eine Sammlung von Maschinen enthält, die einzeln oder gemeinsam die Anweisungen 702 ausführen, um irgendeine oder mehrere der hier erörterten Methodologien auszuführen.

[0102] Die Maschine 700 enthält die Prozessoren 704, einen Speicher 706 und die E/A-Komponenten 708, die konfiguriert sind, wie z. B. über einen Bus 710, miteinander zu kommunizieren. Gemäß einigen Beispielen können die Prozessoren 704 (z. B. eine Zentraleinheit (CPU), ein Prozessor mit verringertem Befehlssatz (RISC-Prozessor), ein Prozessor mit komplexem Befehlssatz (CISC-Prozessor), eine Graphikverarbeitungseinheit (GPU), ein digitaler Signalprozessor (DSP), eine anwendungsspezifische integrierte Schaltung (ASIC), eine integrierte Hochfrequenz-Schaltung (RFIC), ein weiterer Prozessor oder irgendeine geeignete Kombination davon) z. B. mehrere Prozessoren enthalten, wie durch den Prozessor 712 und einen Prozessor 714 veranschaulicht ist, der die Anweisungen 702 ausführen kann. Der Begriff „Prozessor“ soll Mehrkernprozessoren enthalten, die zwei oder mehr unabhängige Prozessoren (die manchmal als „Kerne“ bezeichnet werden) umfassen können, die die Anweisungen 702 gleichzeitig ausführen können. Obwohl **Fig.** 7 mehrere Prozessoren 704 zeigt, kann die Maschine 700 einen einzelnen Prozessor mit einem einzigen Kern, einen einzelnen Prozessor mit mehreren Kernen (z. B. einen Mehrkernprozessor), mehrere Prozessoren mit einem einzigen Kern, mehrere Prozessoren mit mehreren Kernen oder irgendeine Kombination davon enthalten.

[0103] Der Speicher 706 kann einen Hauptspeicher 732, einen statischen Speicher 716 und eine Speichereinheit 718, die ein Maschinenspeichermedium 734 enthält, enthalten, die alle für die Prozessoren 704, wie z. B. über den Bus 710, zugänglich sind. Der Hauptspeicher 732, der statische Speicher 716 und die Speichereinheit 718 speichern die Anweisungen 702, die irgendeine oder mehrere der hier beschriebenen Methodologien oder Funktionen verkörpern. Die Anweisungen 702 können außerdem während ihrer Ausführung durch die Maschine 700 vollständig oder teilweise innerhalb des Hauptspeichers 732, innerhalb des statischen Speichers 716, innerhalb der Speichereinheit 718, innerhalb wenigstens eines der Prozessoren 704 (z. B. innerhalb des Cache-Speichers des Prozessors) oder in einer geeigneten Kombination davon gespeichert sein.

[0104] Die Eingabe/Ausgabe-Komponenten (E/A-Komponenten) 708 enthalten Komponenten, um eine Eingabe zu empfangen, eine Ausgabe bereitzustellen, eine Ausgabe zu erzeugen, Informationen zu übertragen, Informationen auszutauschen, Messungen zu erfassen usw. Die spezifischen E/A-Komponenten 708, die in einer speziellen Maschine 700 enthalten sind, hängen vom Typ der Maschine ab. Tragbare Maschinen, wie z. B. Mobiltelephone, enthalten z. B. wahrscheinlich eine Berührungseingabevorrichtung oder andere derartige Eingabemechanismen, während eine kopflose Server-Maschine wahrscheinlich keine derartige Berührungseingabevorrichtung enthält. Es wird erkannt, dass die E/A-Komponenten 708 viele andere Komponenten enthalten können, die in **Fig.** 7 nicht gezeigt sind. Die E/A-Komponenten 708 sind lediglich zum Vereinfachen der folgenden Erörterung gemäß der Funktionalität gruppiert, wobei die Gruppierung in keiner Weise einschränkend ist. Gemäß verschiedenen Beispielen können die E/A-Komponenten 708 die Ausgabekomponenten 720 und die Eingabekomponenten 722 enthalten. Die Ausgabekomponenten 720 können optische Komponenten (z. B. eine Anzeige, wie z. B. eine Plasmaanzeigetafel (PDP), eine Leuchtdiodenanzeige (LED-Anzeige), eine Flüssigkristallanzeige (LCD-Anzeige), einen Projektor oder eine Kathodenstrahlröhre (CRT)), akustische Komponenten (z. B. Lautsprecher), andere Signalgeneratoren usw. enthalten. Die Eingabekomponenten 722 können alphanumerische Eingabekomponenten (z. B. eine Tastatur, einen Berührungsschirm, der konfiguriert ist, eine alphanumerische Eingabe zu empfangen, eine photooptische Tastatur oder andere alphanumerische Eingabekomponenten), punktbasierte Eingabekomponenten (z. B, eine Maus, ein Tastfeld, eine Rollkugel, einen Joystick, einen Bewegungssensor oder ein weiteres Zeigeinstrument), taktile

Eingabekomponenten (z. B. eine physische Taste, einen Berührungsschirm, der den Ort und/oder die Kraft von Berührungen oder Berührungsgesten bereitstellt, oder andere taktile Eingabekomponenten), Audio-Eingabekomponenten (z. B. ein Mikrophon) und dergleichen enthalten.

[0105] Die Kommunikation kann unter Verwendung einer umfassenden Vielfalt von Techniken implementiert sein. Die E/A-Komponenten 708 können Kommunikationskomponenten 724 enthalten, die betreibbar sind, um die Maschine 700 über eine Kopplung 730 bzw. eine Kopplung 728 an ein Netz 736 oder an die Vorrichtungen 726 zu koppeln. Die Kommunikationskomponenten 724 können z. B. eine Netzschnittstellenkomponente oder eine weitere geeignete Vorrichtung, um mit dem Netz 736 eine Schnittstelle zu bilden, enthalten. Gemäß weiteren Beispielen können die Kommunikationskomponenten 724 drahtgebundene Kommunikationskomponenten, drahtlose Kommunikationskomponenten, Zellenkommunikationskomponenten und andere Kommunikationskomponenten enthalten, um eine Kommunikation über andere Modalitäten zu schaffen. Die Vorrichtungen 726 können eine weitere Maschine oder irgendeine aus einer umfassenden Vielfalt von Peripherievorrichtungen (z. B. eine Peripherievorrichtung, die über einen universellen seriellen Bus (USB) gekoppelt ist) sein. Die Maschine 700 kann z. B., wie oben erwähnt worden ist, irgendeinem des Rechendienstmanagers 104 und der Ausführungsplattform 110 entsprechen, wobei die Vorrichtungen 726 die Datenspeichervorrichtung 226 oder irgendeine andere hier beschriebene Computervorrichtung, die hier als mit der Datenplattform 102 oder dem Datenbankspeicher 106 in Verbindung stehend beschrieben wird, enthalten können.

[0106] Die verschiedenen Speicher (z. B. 706, 716, 732 und/oder der Speicher des Prozessors (der Prozessoren) 704 und/oder der Speichereinheit 718) können einen oder mehrere Sätze von Anweisungen 702 und Datenstrukturen (z. B. Software) speichern, die irgendeine oder mehrere der hier beschriebenen Methodologien oder Funktionen verkörpern oder durch diese verwendet werden. Diese Anweisungen 702 verursachen, wenn sie durch den Prozessor (die Prozessoren) 704 ausgeführt werden, verschiedene Operationen, um die offenbarten Beispiele zu implementieren.

[0107] Die Begriffe „Maschinenspeichermedium“, „Vorrichtungsspeichermedium“ und „Computerspeichermedium“, wie sie hier verwendet werden, bedeuten das gleiche und können in dieser Offenbarung synonym verwendet werden. Die Begriffe beziehen sich auf eine einzelne oder mehrere Speichervorrichtungen und/oder -medien (z. B. eine zentralisierte oder verteilte Datenbank und/oder zugeordnete Caches und Server), die ausführbare Anweisungen und/oder Daten speichern. Die Begriffe sollen dementsprechend ausgelegt werden, dass sie Halbleiterspeicher und optische und magnetische Medien, einschließlich internen oder externen Speichers für Prozessoren, enthalten, sind aber nicht darauf eingeschränkt. Spezifische Beispiele der Maschinenspeichermedien, Computerspeichermedien und/oder Vorrichtungsspeichermedien enthalten nichtflüchtigen Speicher, einschließlich beispielhaft Halbleiterspeichervorrichtungen, z. B. löschbaren programmierbaren Festwertspeicher (EPROM), elektrisch löschbaren programmierbaren Festwertspeicher (EEPROM), feldprogrammierbare Gatteranordnungen (FPGAs) und Flash-Speichervorrichtungen; Magnetplatten, wie z. B. interne Festplatten und Wechselfestplatten; magneto-optische Platten; und CD-ROM- und DVD-ROM-Platten. Die Begriffe „Maschinenspeichermedien“, „Computerspeichermedien“ und „Vorrichtungsspeichermedien“ schließen spezifisch Trägerwellen, modulierte Datensignale und derartige Medien aus, von denen wenigstens einige unter dem Begriff „Signalmedium“, der im Folgenden erörtert wird, abgedeckt sind.

[0108] Gemäß verschiedenen Beispielen können ein oder mehrere Teile des Netzes 736 ein Ad-hoc-Netz, ein Intranet, ein Extranet, ein virtuelles privates Netz (VPN), ein lokales Netz (LAN), ein drahtloses LAN (WLAN), ein Weitverkehrsnetz (WAN), ein drahtloses WAN (WWAN), ein Stadtbereichsnetz (MAN), das Internet, ein Teil des Internets, ein Teil des öffentlichen vermittelten Telephonnetzes (PSTN), ein einfaches altes Telephondienstnetz (POTS-Netz), ein Zellentelephonnetz, ein drahtloses Netz, ein Wi-Fi®-Netz, ein weiterer Typ von Netz oder eine Kombination aus zwei oder mehr derartigen Netzen sein. Das Netz 736 oder ein Teil des Netzes 736 kann z. B. ein drahtloses oder Zellennetz enthalten, wobei die Kopplung 730 eine Codemultiplex-Vielfachzugriffs-Verbindung (CDMA-Verbindung), eine Verbindung eines globalen Systems für die Mobilkommunikation (GSM-Verbindung) oder ein weiterer Typ einer Zellen- oder drahtlosen Kopplung sein kann. Gemäß diesem Beispiel kann die Kopplung 730 irgendeinen von verschiedenen Typen einer Datenübertragungstechnik, wie z. B. eine Einzelträger-Funkübertragungstechnik (1xRTT), eine evolutionsdatenoptimierte Technik (EVDO-Technik), eine Technik des allgemeinen Paketfunkdienstes (GPRS-Technik), eine Technik mit vergrößerten Datenraten für die GSM-Entwicklung (EDGE-Technik), das Partnerschaftsprojekt der dritten Generation (3GPP) einschließlich 3G, drahtlose Netze der vierten Generation (4G), drahtlose Netze der fünften Generation (5G), das universelle Mobiltelekommunikationssystem (UMTS), den Hochgeschwindigkeits-Paketzugriff (HSPA), die weltweite Zusammenarbeitsfähigkeit für den Mikrowellenzugriff (Wi-MAX), den Standard der langfristigen Entwicklung (LTE-Standard), andere, die durch verschiedene Standards festlegende

Organisationen definiert werden, andere Langstrecken-Protokolle oder eine andere Datenübertragungstechnik, implementieren.

[0109] Die Anweisungen 702 können über das Netz 736 unter Verwendung eines Übertragungsmediums über eine Netzschnittstellenvorrichtung (z. B. eine in den Kommunikationskomponenten 724 enthaltene Netzschnittstellenkomponente) und unter Verwendung irgendeines aus einer Anzahl wohlbekannter Übertragungsprotokolle (z. B. des Hypertextübertragungsprotokolls (HTTP)) übertragen oder empfangen werden. Ähnlich können die Anweisungen 702 unter Verwendung eines Übertragungsmediums über die Kopplung 728 (z. B. eine Peer-zu-Peer-Kopplung) zu den Vorrichtungen 726 übertragen oder von ihnen empfangen werden. Die Begriffe „Übertragungsmedium“ und „Signalmedium“ bedeuten das gleiche und können in dieser Offenbarung synonym verwendet werden. Die Begriffe „Übertragungsmedium“ und „Signalmedium“ sollen so ausgelegt werden, dass sie jedes immaterielle Medium enthalten, das die Anweisungen 702 für die Ausführung durch die Maschine 700 speichern, codieren oder übertragen kann, wobei sie digitale oder analoge Kommunikationssignale oder andere immaterielle Medien enthalten, um die Übertragung derartiger Software erleichtern. Folglich sollen die Begriffe „Übertragungsmedium“ und „Signalmedium“ so ausgelegt werden, dass sie jede Form eines modulierten Datensignals, einer Trägerwelle usw. enthalten. Der Begriff „moduliertes Datensignal“ bedeutet ein Signal, das eine oder mehrere seiner Eigenschaften in einer derartigen Weise festgelegt oder verändert aufweist, um Informationen in dem Signal zu codieren.

[0110] Die verschiedenen Operationen der hier beschriebenen beispielhaften Verfahren können wenigstens teilweise durch einen oder mehrere Prozessoren ausgeführt werden, die vorübergehend (z. B. durch Software) konfiguriert sind oder dauerhaft konfiguriert sind, die relevanten Operationen auszuführen. Ähnlich können die hier beschriebenen Verfahren wenigstens teilweise prozessorimplementiert sein. Wenigstens einige der Operationen der hier offenbarten Methodologien können z. B. durch einen oder mehrere Prozessoren ausgeführt werden. Die Ausführung bestimmter der Operationen kann zwischen dem einen oder den mehreren Prozessoren verteilt sein, die sich nicht nur innerhalb einer einzigen Maschine befinden, sondern außerdem über eine Anzahl von Maschinen eingesetzt werden. Gemäß einigen Beispielen können sich der Prozessor oder die Prozessoren an einem einzigen Ort (z. B. innerhalb einer Heimumgebung, einer Büroumgebung oder einer Server-Farm) befinden, während gemäß anderen Beispielen die Prozessoren über eine Anzahl von Orten verteilt sein können.

[0111] Obwohl die Beispiele der vorliegenden Offenbarung bezüglich spezifischer Beispiele beschrieben worden sind, ist es ersichtlich, dass verschiedene Modifikationen und Änderungen an diesen Beispielen vorgenommen werden können, ohne vom breiteren Schutzumfang des Erfindungsgegenstandes abzuweichen. Dementsprechend sind die Patentschrift und die Zeichnungen in einem veranschaulichenden anstatt in einem einschränkenden Sinn zu betrachten. Die beigefügten Zeichnungen, die einen Teil hiervon bilden, zeigen zur Veranschaulichung und nicht zur Einschränkung spezifische Beispiele, in denen der Gegenstand praktiziert werden kann. Die veranschaulichten Beispiele sind ausreichend ausführlich beschrieben, um es den Fachleuten auf dem Gebiet zu ermöglichen, die hier offenbarten Lehren zu praktizieren. Andere Beispiele können verwendet und davon abgeleitet werden, so dass strukturelle und logische Ersetzungen und Änderungen vorgenommen werden können, ohne vom Schutzumfang dieser Offenbarung abzuweichen. Deshalb ist diese ausführliche Beschreibung nicht in einem einschränkenden Sinn auszulegen, wobei der Schutzumfang der verschiedenen Beispiele nur durch die beigefügten Ansprüche zusammen mit dem vollständigen Bereich der Äquivalente, zu denen derartige Ansprüche berechtigt sind, definiert ist.

[0112] In diesem Dokument werden die Begriffe „ein“ oder „eine“ verwendet, wie es in Patentdokumenten üblich ist, um eines oder mehr als eines zu enthalten, unabhängig von anderen Instanzen oder Verwendungen von „wenigstens einem“ oder „einem oder mehreren“. In diesem Dokument wird der Begriff „oder“ verwendet, um sich auf ein nicht ausschließendes „oder“ zu beziehen, so dass „A oder B“ „A, aber nicht B“, „B, aber nicht A“ und „A und B“ enthält, sofern nichts anderes angegeben ist. In den beigefügten Ansprüchen werden die Ausdrücke „einschließlich“ und „in denen“ als die Äquivalente auf gut Deutsch der jeweiligen Begriffe „umfassend“ und „wobei“ verwendet. Außerdem sind in den folgenden Ansprüchen die Begriffe „einschließlich“ und „umfassend“ offen; d. h., ein System, eine Vorrichtung, ein Gegenstand oder ein Prozess, das bzw. die bzw. der zusätzlich zu jenen, die nach einem derartigen Begriff in einem Anspruch aufgeführt sind, Elemente enthält, wird dennoch als in den Schutzumfang dieses Anspruchs fallend erachtet.

[0113] Derartige Beispiele des Erfindungsgegenstandes können hier lediglich für die Zweckmäßigkeit und ohne zu beabsichtigen, den Schutzumfang dieser Anmeldung freiwillig auf irgendeine einzelne Erfindung oder irgendein einzelnes Erfindungskonzept zu einzuschränken, falls tatsächlich mehr als eines offenbart ist, einzeln und/oder gemeinsam durch den Begriff „Beispiel“ bezeichnet werden. Obwohl hier spezifischen Bei-

spiele veranschaulicht und beschrieben worden sind, sollte folglich erkannt werden, dass jede Anordnung, die berechnet worden ist, um den gleichen Zweck zu erreichen, für die gezeigten spezifischen Beispiele ersetzt werden kann. Diese Offenbarung soll sämtliche Anpassungen oder Variationen der verschiedenen Beispiele abdecken. Kombinationen der obigen Beispiele und anderer Beispiele, die hier nicht spezifisch beschrieben worden sind, sind für die Fachleute auf dem Gebiet beim Überprüfen der obigen Beschreibung ersichtlich.

Schutzansprüche

1. Datenplattform, die umfasst:
einen oder mehrere Prozessoren; und
wenigstens einen Speicher, der Anweisungen speichert, die dann, wenn sie durch den einen oder die mehreren Prozessoren ausgeführt werden, die Datenplattform veranlassen, Operationen auszuführen, die umfassen:
Autorisieren eines ersten Anwenders, einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen;
Autorisieren des ersten Anwenders, einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden; und
Bereitstellen der Anwenderanwendung für den zweiten Anwender basierend auf den Sicherheitsstrategien des Sicherheitskontextes durch Ausführen von Operationen, die umfassen:
Instanziieren eines Servers anwenderdefinierter Funktionen, UDF-Servers, innerhalb des Sicherheitskontextes;
Instanziieren einer Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes;
Instanziieren der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes; und
Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten basierend auf den Sicherheitsstrategien des Sicherheitskontextes.

2. Computerprogramm, das Anweisungen umfasst, die dann, wenn sie durch einen oder mehreren Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen, Operationen auszuführen, wobei die Operationen umfassen:
Autorisieren, durch den einen oder die mehreren Prozessoren, eines ersten Anwenders, um einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf den Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen;
Autorisieren, durch einen oder mehrere Prozessoren, des ersten Anwenders,
um einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden; und
Bereitstellen, durch einen oder mehrere Prozessoren, der Anwenderanwendung für den zweiten Anwender basierend auf den Sicherheitsstrategien des Sicherheitskontextes durch Ausführen von Operationen, die umfassen:
Instanziieren eines Servers anwenderdefinierter Funktionen, UDF-Servers, innerhalb des Sicherheitskontextes;
Instanziieren einer Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes;
Instanziieren der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes; und
Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten basierend auf den Sicherheitsstrategien des Sicherheitskontextes.

3. Computer-Speichermedium, das Anweisungen umfasst, die dann, wenn sie durch einem oder mehreren Prozessoren einer Maschine ausgeführt werden, die Maschine konfigurieren, Operationen auszuführen, die umfassen:
Autorisieren eines ersten Anwenders, um einen Editor zu verwenden, um auf den Quellcode einer Anwenderanwendung basierend auf den Sicherheitsstrategien eines Sicherheitskontextes zuzugreifen;
Autorisieren des ersten Anwenders, um einen Anwendungs- und Datenmanager zu verwenden, um Nutzungsprivilegien für einen zweiten Anwender festzulegen, um die Anwenderanwendung basierend auf den Sicherheitsstrategien des Sicherheitskontextes zu verwenden; und
Bereitstellen der Anwenderanwendung für den zweiten Anwender basierend auf den Sicherheitsstrategien des Sicherheitskontextes durch Ausführen von Operationen, die umfassen:

Instanziieren eines Servers anwenderdefinierter Funktionen, UDF-Servers, innerhalb des Sicherheitskontextes;
Instanziieren einer Anwendungsmaschine des UDF-Servers innerhalb des Sicherheitskontextes;
Instanziierung der Anwenderanwendung als eine Anwendung der Anwendungsmaschine innerhalb des Sicherheitskontextes; und
Autorisieren des Zugriffs durch die Anwenderanwendung auf die Daten basierend auf den Sicherheitsstrategien des Sicherheitskontextes.

Es folgen 9 Seiten Zeichnungen

Anhängende Zeichnungen

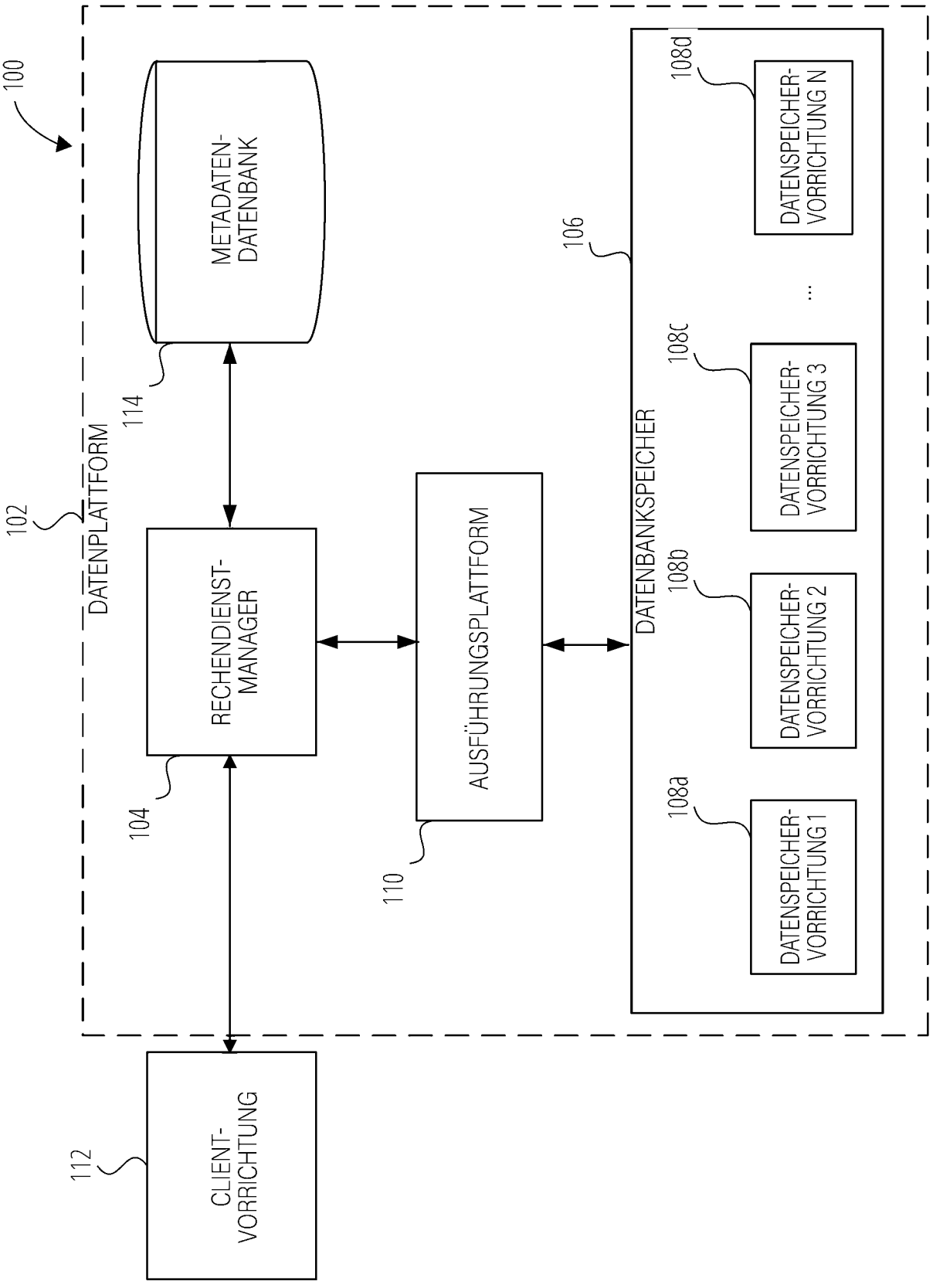


FIG. 1

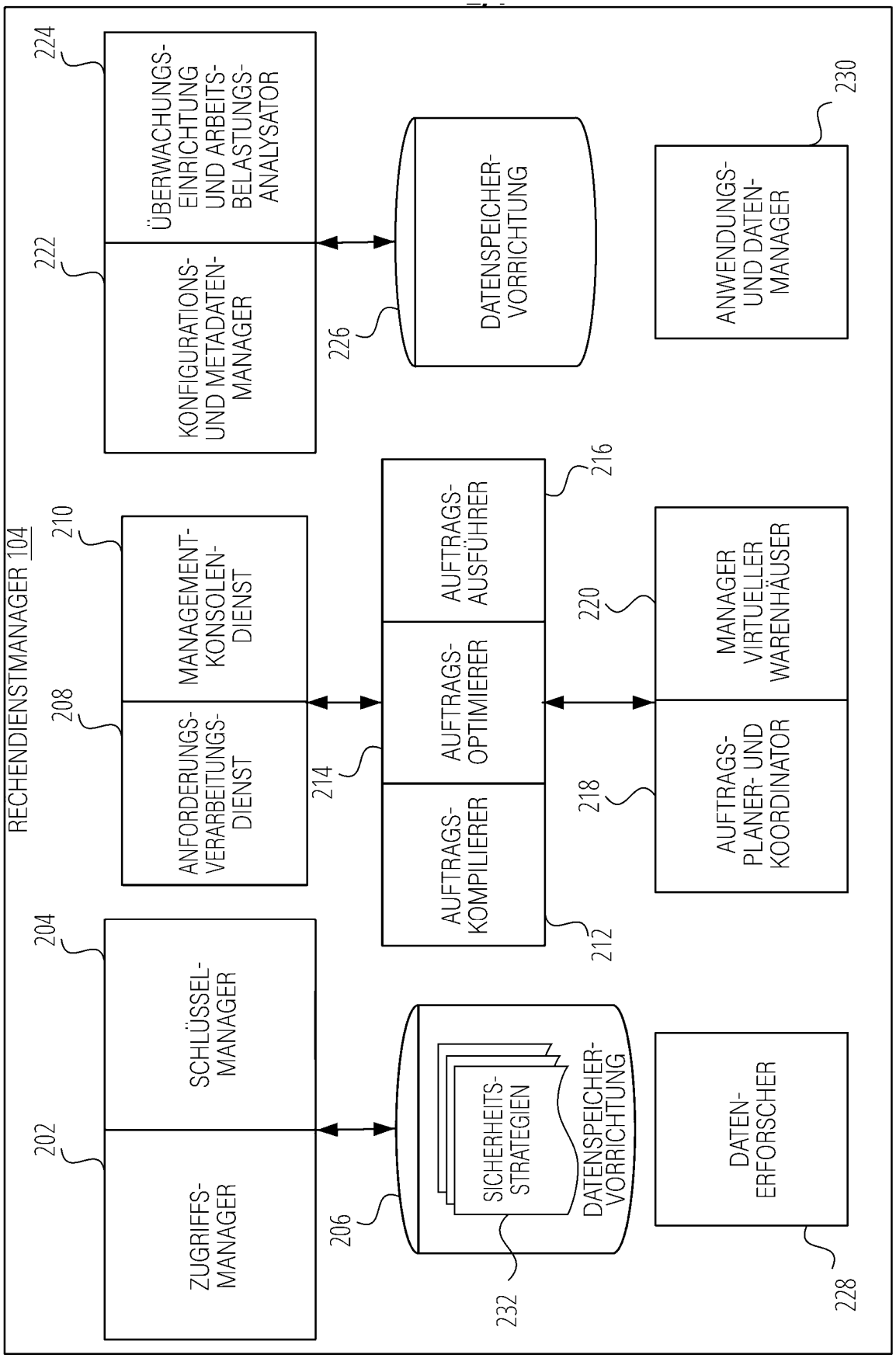

RECHENDIENSTMANAGER 104
ZUGRIFFS-MANAGER
SCHLÜSSEL-MANAGER
ANFORDERUNGS-VERARBEITUNGS-DIENST
MANAGEMENT-KONSOLEN-DIENST
AUFTRAGS-KOMPILIERER
AUFTRAGS-OPTIMIERER
AUFTRAGS-AUSFÜHRER
AUFTRAGS-PLANER- UND KOORDINATOR
MANAGER VIRTUELLER WARENHÄUSER
KONFIGURATIONS- UND METADATEN-MANAGER
ÜBERWACHUNGS-EINRICHTUNG UND ARBEITS-BELASTUNGS-ANALYSATOR
DATENSPEICHER-VORRICHTUNG
SICHERHEITS-STRATEGIEN
DATENSPEICHER-VORRICHTUNG
DATEN-ERFORSCHER
ANWENDUNGS- UND DATEN-MANAGER
202
204
206
208
210
212
214
216
218
220
222
224
226
228
230
232

FIG. 2

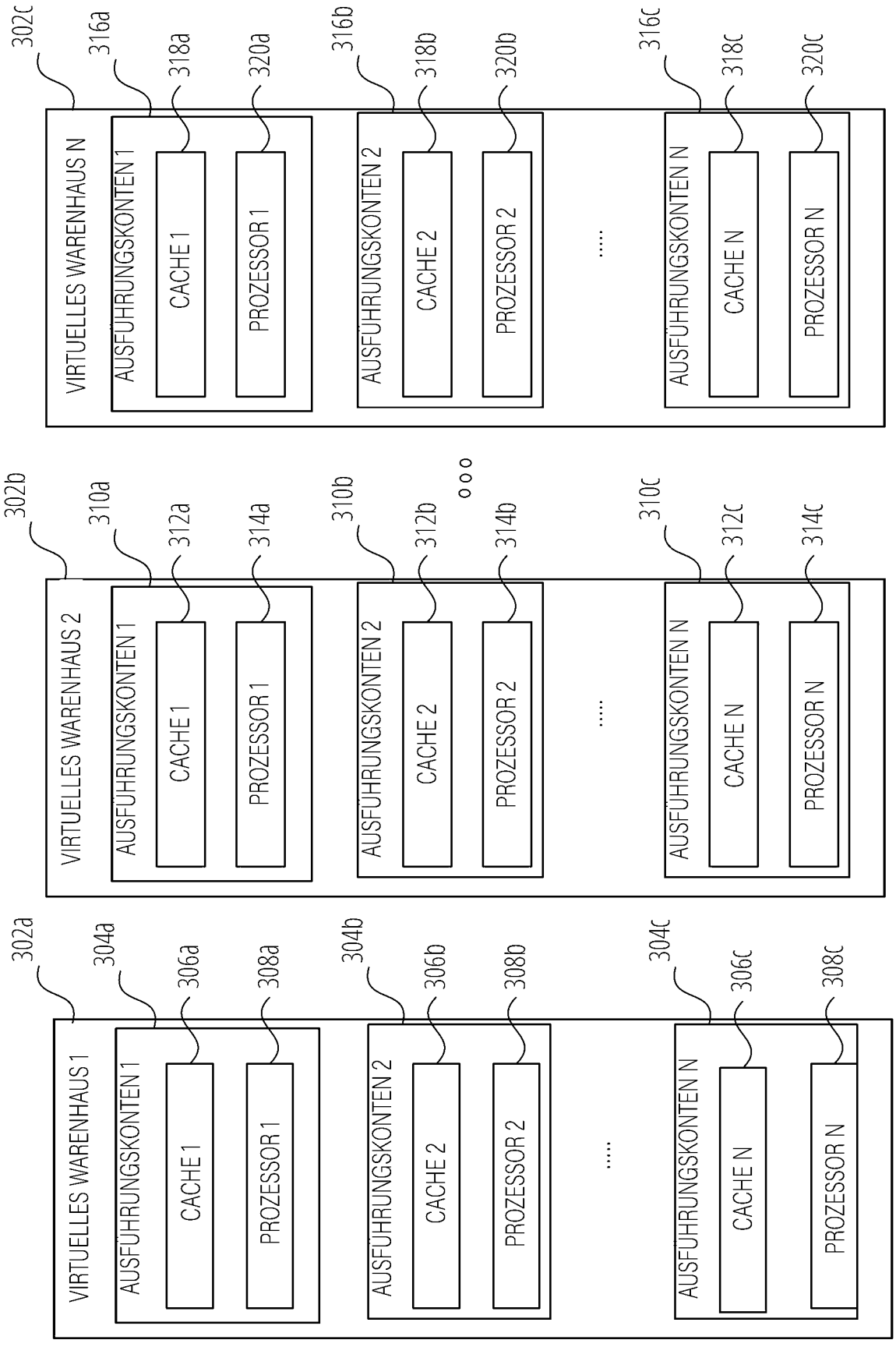

302a
304a
306a
308a
304b
306b
308b
304c
306c
308c
VIRTUELLES WARENHAUS 1
AUSFÜHRUNGSKONTEN 1
CACHE 1
PROZESSOR 1
AUSFÜHRUNGSKONTEN 2
CACHE 2
PROZESSOR 2
.....
AUSFÜHRUNGSKONTEN N
CACHE N
PROZESSOR N
302b
310a
312a
314a
310b
312b
314b
310c
312c
314c
VIRTUELLES WARENHAUS 2
302c
316a
318a
320a
316b
318b
320b
316c
318c
320c
VIRTUELLES WARENHAUS N

FIG. 3

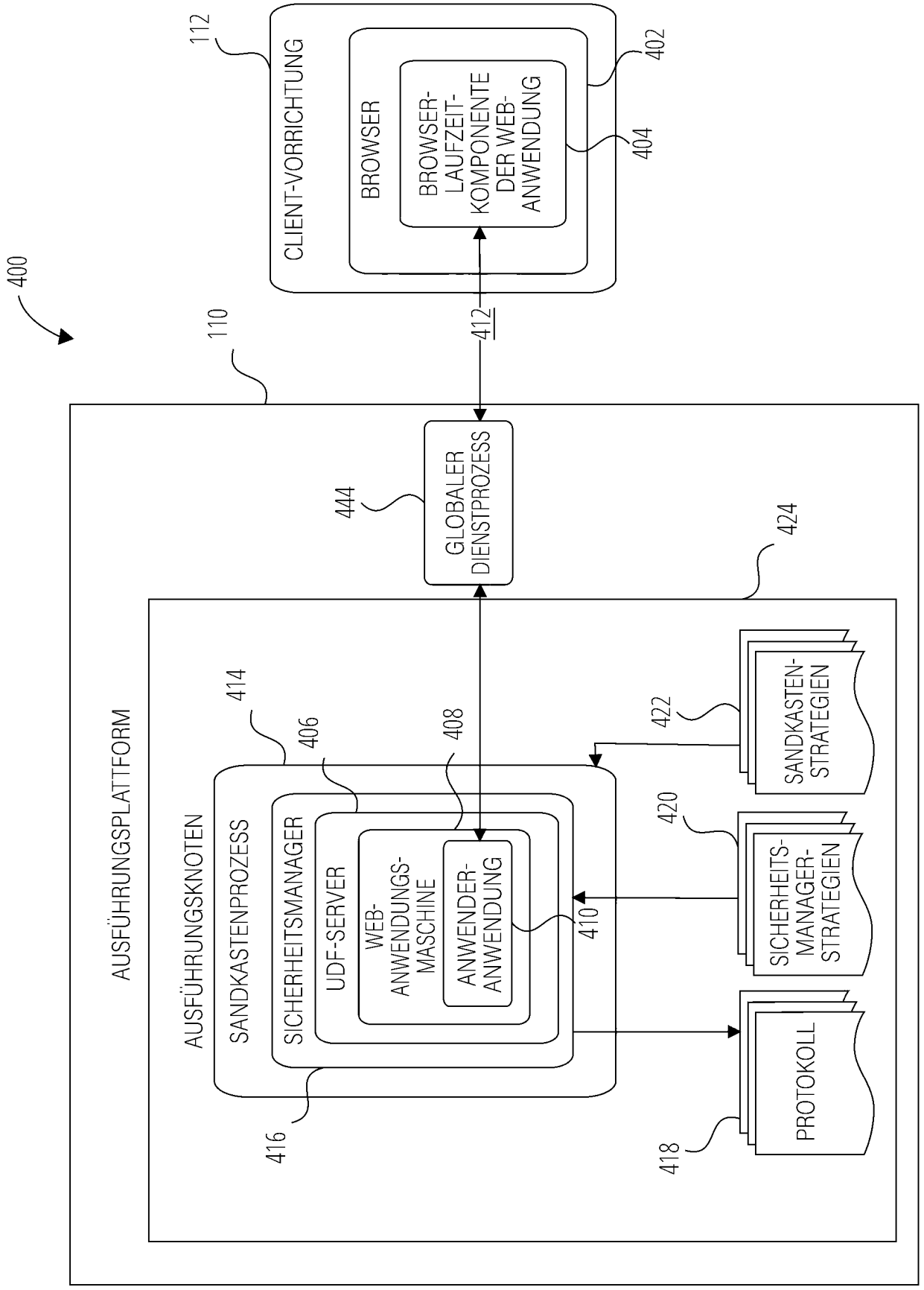

400
110
112
CLIENT-VORRICHTUNG
BROWSER
BROWSER-LAUFZEIT-KOMPONENTE DER WEB-ANWENDUNG
402
404
412
AUSFÜHRUNGSPLATTFORM
GLOBALER DIENSTPROZESS
444
424
AUSFÜHRUNGSKNOTEN
SANDKASTENPROZESS
SICHERHEITSMANAGER
UDF-SERVER
WEB-ANWENDUNGS-MASCHINE
ANWENDER-ANWENDUNG
414
406
408
410
416
422
SANDKASTEN-STRATEGIEN
420
SICHERHEITS-MANAGER-STRATEGIEN
PROTOKOLL
418

FIG. 4A

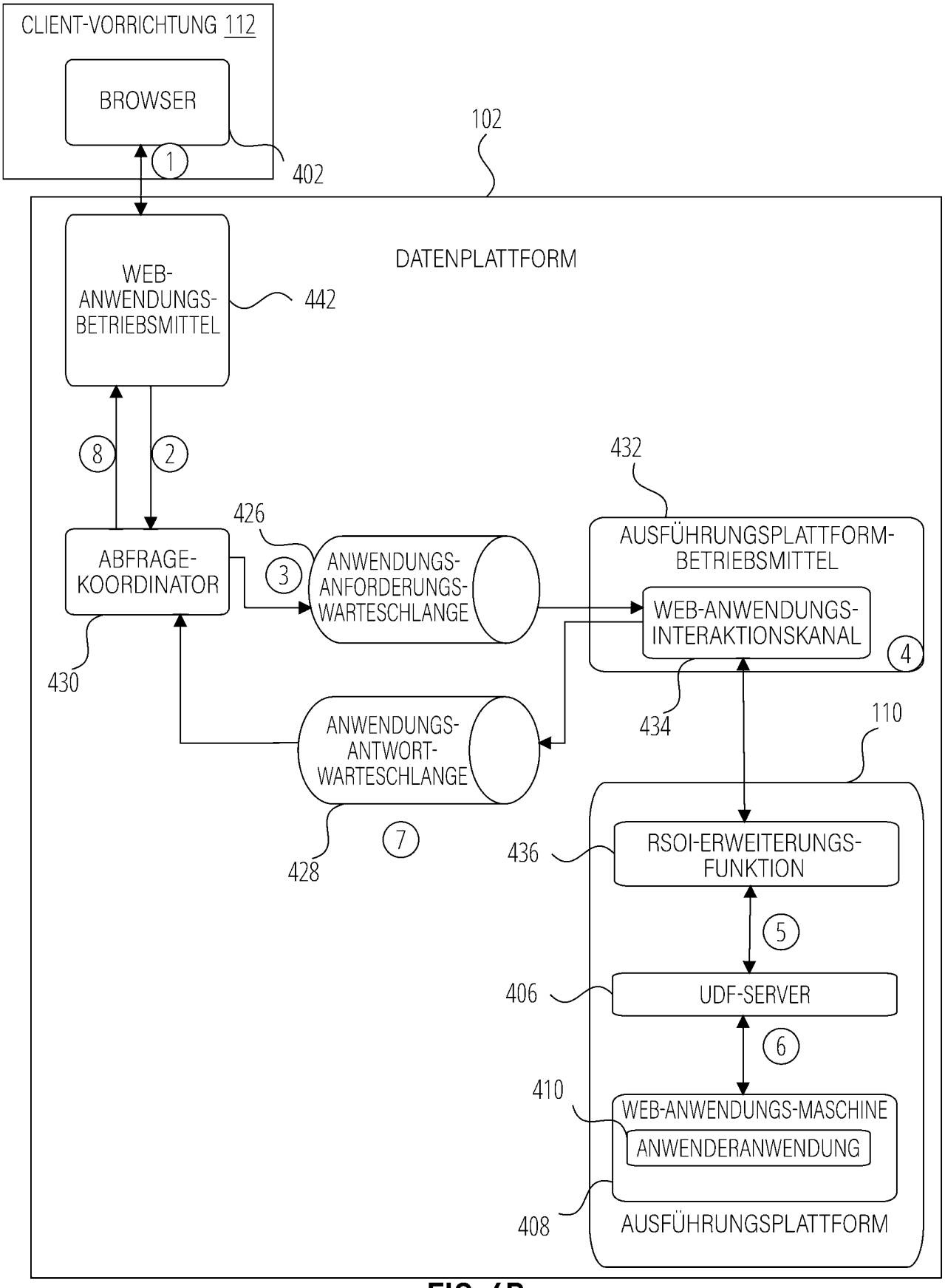

CLIENT-VORRICHTUNG 112
BROWSER
1
402
102
DATENPLATTFORM
WEB-ANWENDUNGS-BETRIEBSMITTEL
442
8
2
426
ABFRAGE-KOORDINATOR
3
ANWENDUNGS-ANFORDERUNGS-WARTESCHLANGE
430
ANWENDUNGS-ANTWORT-WARTESCHLANGE
428
7
432
AUSFÜHRUNGSPLATTFORM-BETRIEBSMITTEL
WEB-ANWENDUNGS-INTERAKTIONSKANAL
4
434
110
436
RSOI-ERWEITERUNGS-FUNKTION
5
406
UDF-SERVER
6
410
WEB-ANWENDUNGS-MASCHINE
ANWENDERANWENDUNG
408
AUSFÜHRUNGSPLATTFORM

FIG. 4B

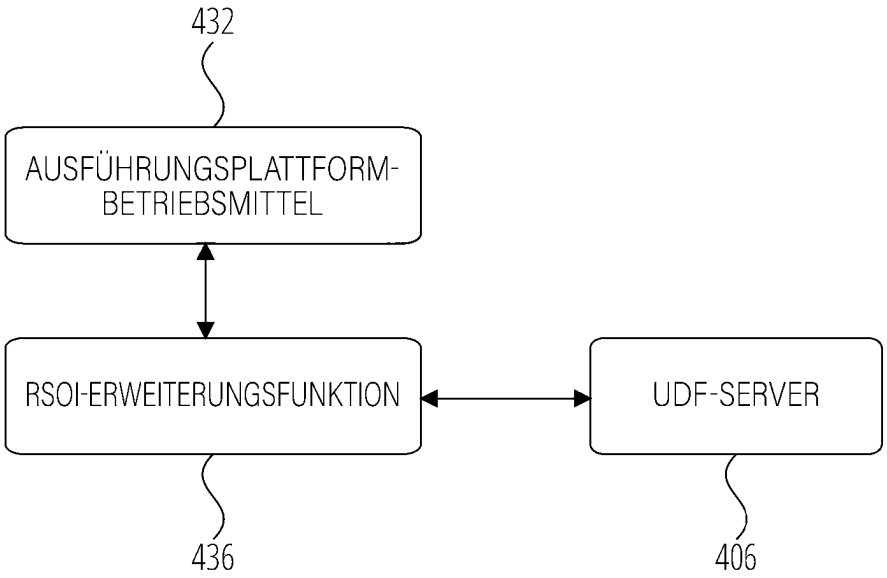

432
AUSFÜHRUNGSPLATTFORM-BETRIEBSMITTEL
RSOI-ERWEITERUNGSFUNKTION
UDF-SERVER
436
406

FIG. 4C

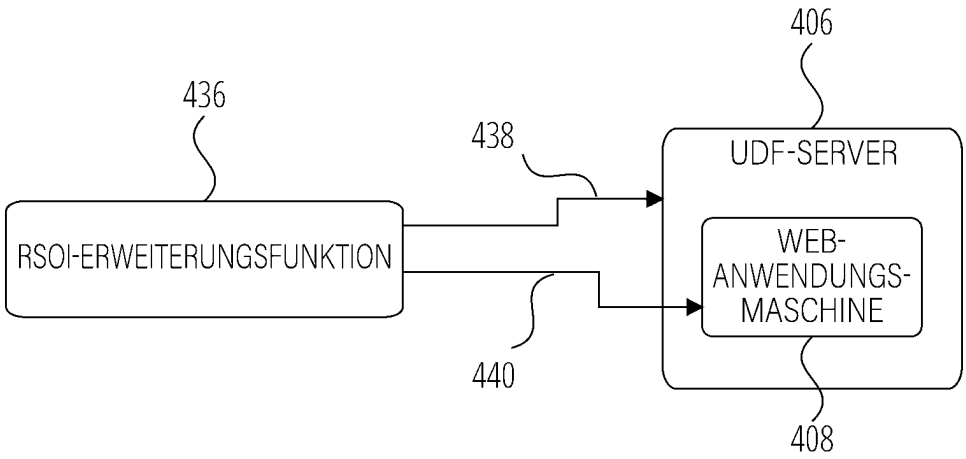

406
436
438
UDF-SERVER
RSOI-ERWEITERUNGSFUNKTION
WEB-ANWENDUNGS-MASCHINE
440
408

FIG. 4D

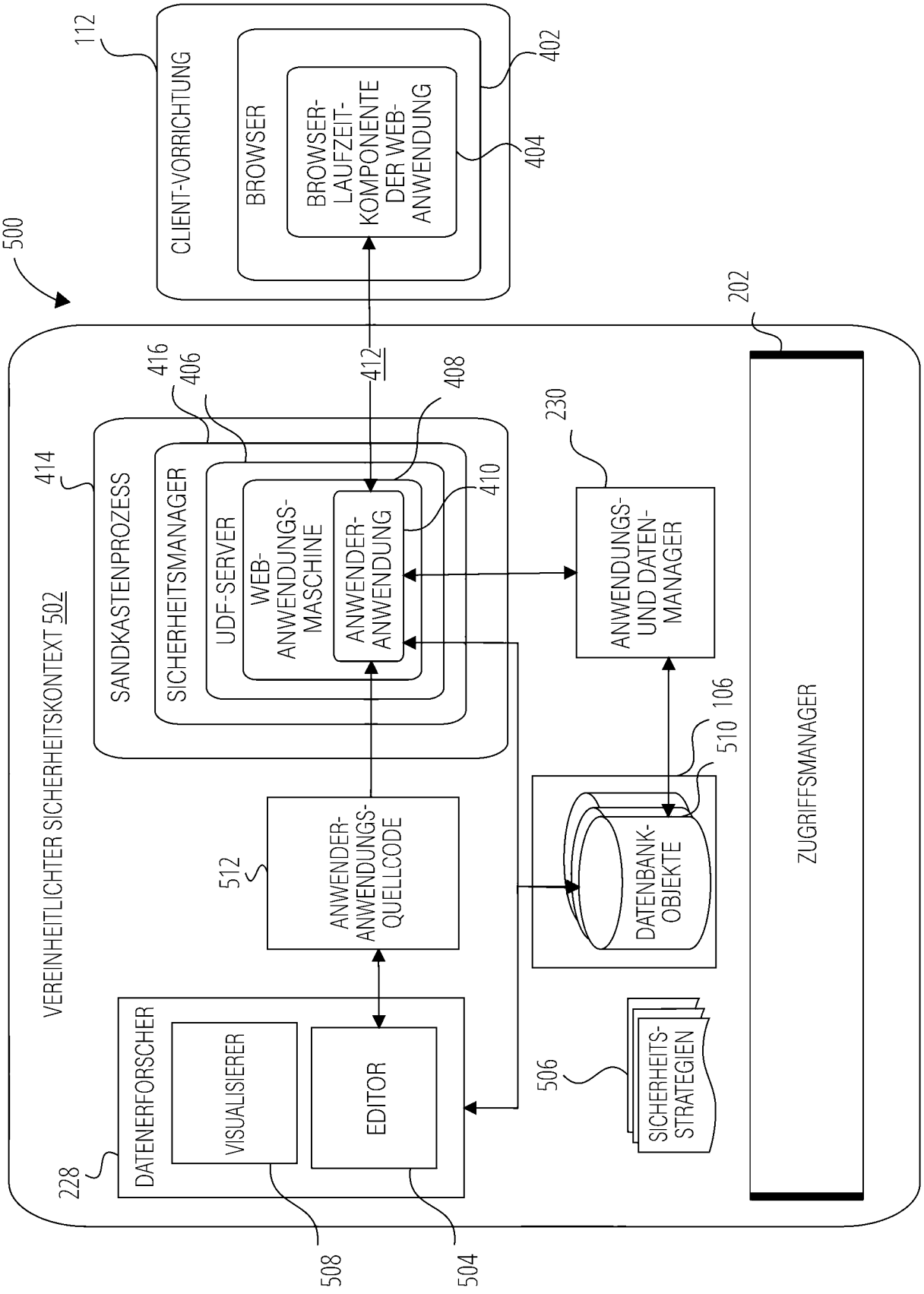

500
112
CLIENT-VORRICHTUNG
BROWSER
BROWSER-LAUFZEIT-KOMPONENTE DER WEB-ANWENDUNG
402
404
412
202
VEREINHEITLICHTER SICHERHEITSKONTEXT 502
414
SANDKASTENPROZESS
416
406
SICHERHEITSMANAGER
UDF-SERVER
WEB-ANWENDUNGS-MASCHINE
ANWENDER-ANWENDUNG
408
410
230
ANWENDUNGS- UND DATEN-MANAGER
106
510
DATENBANK-OBJEKTE
ZUGRIFFSMANAGER
512
ANWENDER-ANWENDUNGS-QUELLCODE
228
DATENERFORSCHER
VISUALISIERER
EDITOR
506
SICHERHEITS-STRATEGIEN
508
504

FIG. 5

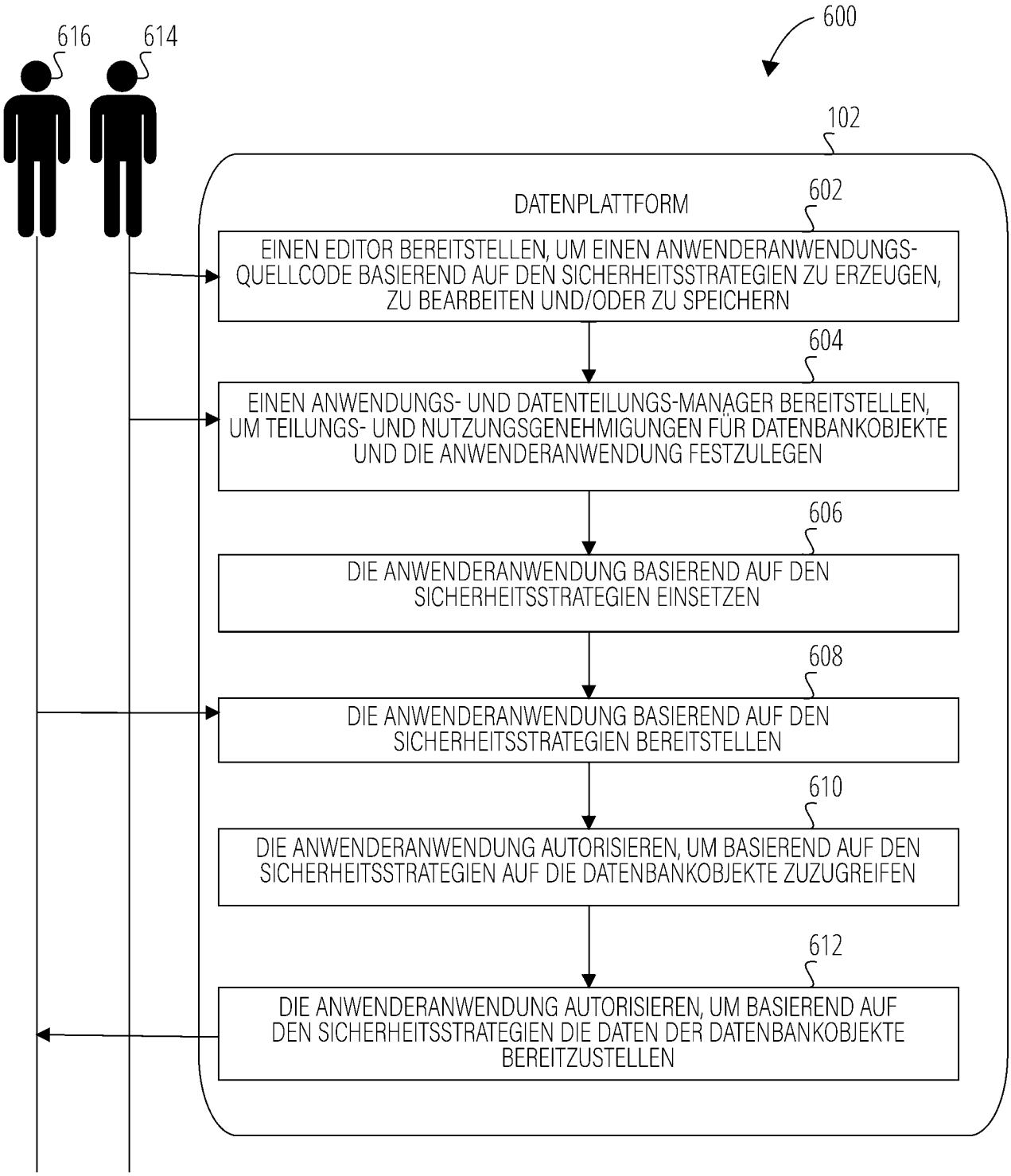

600
616
614
102
DATENPLATTFORM
602
EINEN EDITOR BEREITSTELLEN, UM EINEN ANWENDERANWENDUNGS-QUELLCODE BASIEREND AUF DEN SICHERHEITSSTRATEGIEN ZU ERZEUGEN, ZU BEARBEITEN UND/ODER ZU SPEICHERN
604
EINEN ANWENDUNGS- UND DATENTEILUNGS-MANAGER BEREITSTELLEN, UM TEILUNGS- UND NUTZUNGSGENEHMIGUNGEN FÜR DATENBANKOBJEKTE UND DIE ANWENDERANWENDUNG FESTZULEGEN
606
DIE ANWENDERANWENDUNG BASIEREND AUF DEN SICHERHEITSSTRATEGIEN EINSETZEN
608
DIE ANWENDERANWENDUNG BASIEREND AUF DEN SICHERHEITSSTRATEGIEN BEREITSTELLEN
610
DIE ANWENDERANWENDUNG AUTORISIEREN, UM BASIEREND AUF DEN SICHERHEITSSTRATEGIEN AUF DIE DATENBANKOBJEKTE ZUZUGREIFEN
612
DIE ANWENDERANWENDUNG AUTORISIEREN, UM BASIEREND AUF DEN SICHERHEITSSTRATEGIEN DIE DATEN DER DATENBANKOBJEKTE BEREITZUSTELLEN

FIG. 6

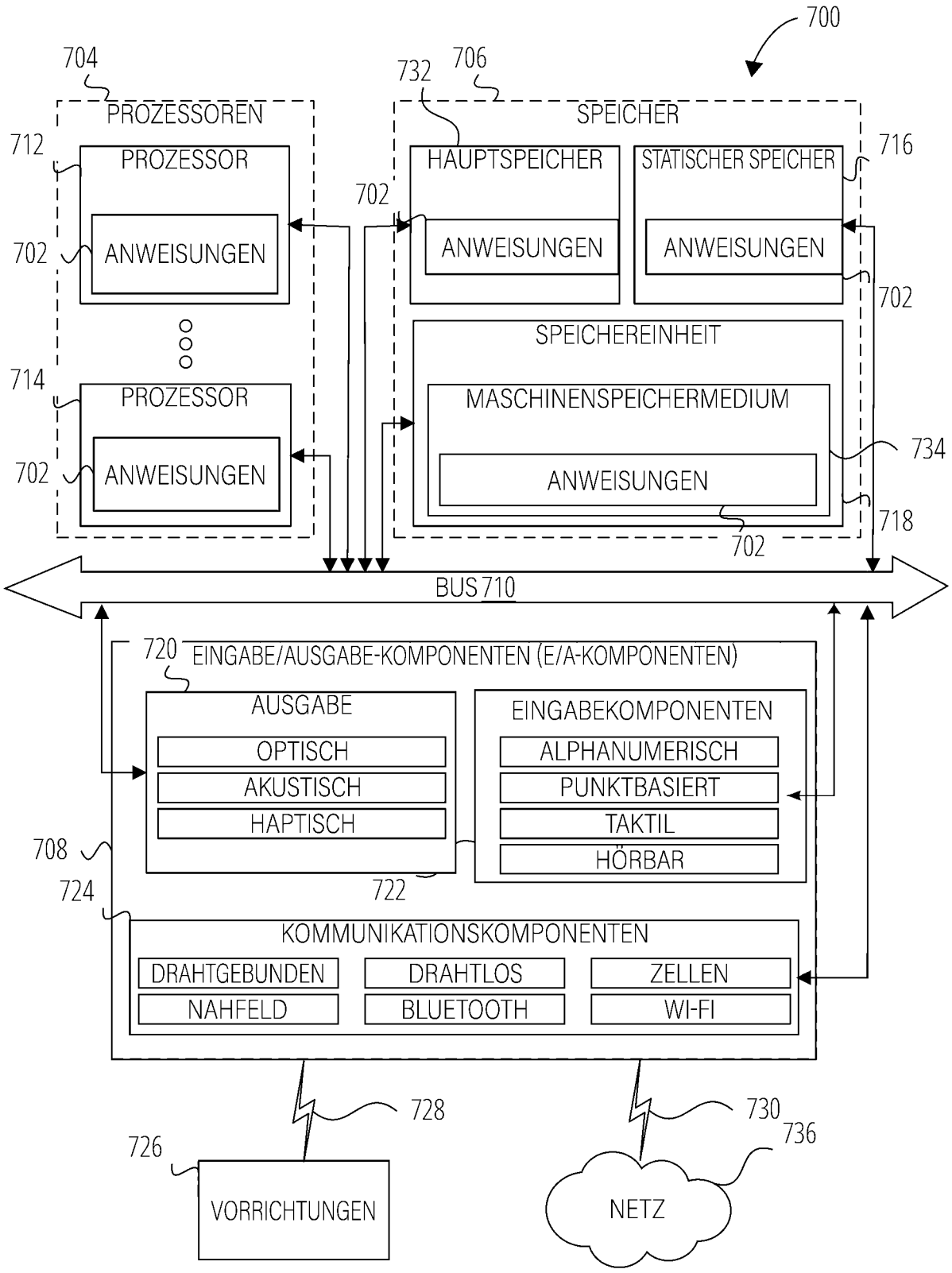


FIG. 7

 Das Dokument wurde durch die Firma Luminess hergestellt.