(19) Deutsches Patent- und Markenamt

(10) **DE 20 2019 005 483 U1** 2020.12.03

(12) # Gebrauchsmusterschrift

(21) Aktenzeichen: **20 2019 005 483.9**
(22) Anmeldetag: **23.04.2019**
(47) Eintragungstag: **28.10.2020**
(45) Bekanntmachungstag im Patentblatt: **03.12.2020**

(51) Int Cl.: ***G06F 16/27*** (2019.01)
G06F 16/23 (2019.01)
G06F 11/07 (2006.01)

(30) Unionspriorität:
62/694,656 **06.07.2018** **US**

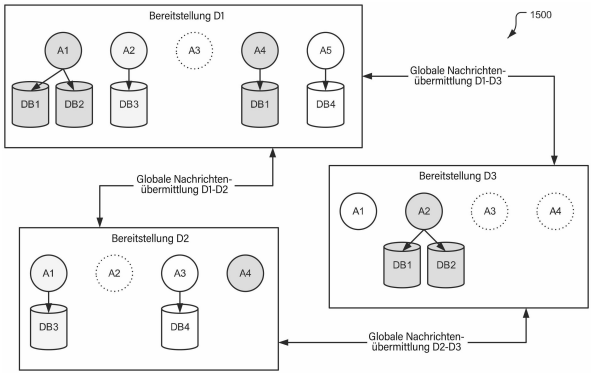
(73) Name und Wohnsitz des Inhabers:
Snowflake Inc., San Mateo, CA, US

(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: **Datenreplikation und Datenausfallsicherung in Datenbanksystemen**

(57) Hauptanspruch: System, das umfasst:
einen Speicher, um Datenbankdaten zu speichern; und
einen Prozessor, der betriebstechnisch mit dem Speicher gekoppelt ist, um:
die in einer primären Bereitstellung gespeicherten Datenbankdaten zu replizieren, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind;
zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist; und
eine oder mehrere Transaktionen an den Datenbankdaten in der sekundären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, auszuführen.



Beschreibung

QUERVERWEIS AUF VERWANDTE ANMELDUNGEN

[0001] Diese Anmeldung beansprucht den Vorteil der vorläufigen US-Anmeldung, laufende Nr. 62/694.656, mit dem Titel „SYSTEMS, METHODS, AND DEVICES FOR DATABASE REPLICATION“, eingereicht am 6. Juli 2018, deren Offenbarung durch Bezugnahme hier insgesamt mit aufgenommen ist, einschließlich, aber nicht eingeschränkt auf, jener Teile, die im Folgenden spezifisch erscheinen, wobei die Aufnahme durch Bezugnahme mit der folgenden Ausnahme erfolgt: In dem Fall, dass irgendein Teil der vorläufigen Anmeldung, auf die oben Bezug genommen wird, nicht mit dieser Anmeldung konsistent ist, ersetzt diese Anmeldung die vorläufige Anmeldung, auf die oben Bezug genommen wird.

[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.

TECHNISCHES GEBIET

[0003] Die vorliegende Offenbarung bezieht sich auf Datenbanken und bezieht sich insbesondere auf die Datenreplikation und Datenausfallsicherung in Datenbanksystemen.

HINTERGRUND

[0004] Datenbanken werden für die Datenspeicherung und den Datenzugriff in Rechenanwendungen umfassend verwendet. Es ist ein Ziel der Datenbankspeicherung, enorme Informationssummen in einer organisierten Weise bereitzustellen, so dass auf sie zugegriffen, sie gemanagt und sie aktualisiert werden kann. In einer Datenbank können Daten in Zeilen, Spalten und Tabellen organisiert sein. Verschiedene Datenbank-Speichersysteme können zum Speichern verschiedener Arten von Inhalt, wie z. B. bibliographischen, Volltext-, numerischen und/oder Bildinhalt, verwendet werden. Ferner können im Computerbereich verschiedene Datenbanksysteme gemäß der Organisationsherangehensweise der Datenbank klassifiziert werden. Es gibt viele verschiedene Typen von Datenbanken einschließlich relationalen Datenbanken, verteilten Datenbanken, Cloud-Datenbanken, objektorientierte und andere.

[0005] Datenbanken werden durch verschiedene Entitäten und Unternehmen zum Speichern von Informationen verwendet, auf die vielleicht zugegriffen werden muss oder die vielleicht analysiert werden müssen. In einem Beispiel kann ein Einzelhandelsunternehmen eine Auflistung aller Verkaufstransaktionen in einer Datenbank speichern. Die Datenbank kann Informationen darüber enthalten, wann eine Transaktion stattgefunden hat, wo sie stattgefunden hat, die Gesamtkosten der Transaktion, eine Kennung und/oder Beschreibung aller Artikel, die in der Transaktion gekauft wurden, usw. Dasselbe Einzelhandelsunternehmen kann z. B. außerdem Mitarbeiterinformationen in derselben Datenbank speichern, die die Namen der Mitarbeiter, Kontaktinformationen der Mitarbeiter, die Arbeitshistory der Mitarbeiter, den Lohnsatz der Mitarbeiter usw. enthalten können. Abhängig von den Notwendigkeiten dieses Einzelhandelsunternehmens können die Mitarbeiterinformationen und die Transaktionsinformationen in verschiedenen Tabellen derselben Datenbank gespeichert sein. Das Einzelhandelsunternehmen kann eine Notwendigkeit aufweisen, seine Datenbank „abzufragen“, wenn es die Informationen erfahren möchte, die in der Datenbank gespeichert sind. Dieses Einzelhandelsunternehmen kann z. B. die Daten über die Namen aller Mitarbeiter, die in einem bestimmten Laden arbeiten, aller Mitarbeiter, die an einem bestimmten Datum arbeiten, alle Transaktionen für ein bestimmtes Produkt, die in einem bestimmten Zeitraum getätigt wurden, usw. finden wollen.

[0006] Wenn das Einzelhandelsunternehmen seine Datenbank abfragen möchte, um bestimmte organisierte Informationen aus der Datenbank zu extrahieren, wird eine Abfrageanweisung gegen die Datenbankdaten ausgeführt. Die Abfrage gibt bestimmte Daten gemäß einem oder mehreren Abfrageprädikaten zurück, die angeben, welche Informationen durch die Abfrage zurückgegeben werden sollten. Die Abfrage extrahiert spezifische Daten aus der Datenbank und formatiert diese Daten in eine lesbare Form. Die Abfrage kann in einer Sprache geschrieben sein, die durch die Datenbank verstanden wird, wie z. B. einer strukturierten Abfragesprache („SQL“), so dass die Datenbanksysteme bestimmen können, welche Daten lokalisiert werden sollten und wie sie zurückgegeben werden sollten. Die Abfrage kann irgendwelche relevanten Informationen anfordern, die innerhalb der Datenbank gespeichert sind. Falls die geeigneten Daten gefunden werden können, um auf die

Abfrage zu antworten, hat die Datenbank das Potential, komplexe Trends und Aktivitäten aufzudecken. Diese Leistung kann nur durch die Verwendung einer erfolgreich ausgeführten Abfrage nutzbar gemacht werden.

[0007] Das herkömmliche Datenbankmanagement erfordert, dass die Unternehmen die Infrastruktur und Betriebsmittel vorhalten, um die Datenbank in einem Datenzentrum zu managen. Das Management einer herkömmlichen Datenbank kann sehr teuer sein und erfordert die Beaufsichtigung durch mehrere Personen mit einem weiten Bereich von technischen Fähigkeiten. Herkömmliche Managementsysteme relationaler Datenbanken (RDMS) erfordern umfangreiche Rechen- und Speicherbetriebsmittel und weisen eine begrenzte Skalierbarkeit auf. Große Datensummen können über mehrere Rechenvorrichtungen gespeichert sein. Ein Server kann die Daten so managen, dass sie für Kunden mit Vor-Ort-Operationen zugänglich sind. Für eine Entität, die einen hausinternen Datenbank-Server haben möchte, muss die Entität signifikante Betriebsmittel für eine Anlageinvestition in Hardware und Infrastruktur für die Datenbank zusammen mit signifikantem physischen Raum zum Lagern der Datenbankinfrastruktur aufwenden. Ferner kann die Datenbank während eines Leistungsausfalls oder anderen Katastrophensituationen im hohen Grade anfällig für einen Datenverlust sein. Derartige herkömmliche Datenbanksysteme weisen signifikante Nachteile auf, die durch ein cloud-basiertes Datenbanksystem gemildert werden können.

[0008] Ein Cloud-Datenbanksystem kann durch eine Cloud-Plattform eingesetzt und geliefert werden, die es Organisationen und Endanwendern ermöglicht, Daten in der Cloud zu speichern, in der Cloud zu managen und aus der Cloud abzurufen. Einige Cloud-Datenbanksysteme enthalten eine herkömmliche Datenbankarchitektur, die durch die Installation von Datenbank-Software auf einer Rechen-Cloud implementiert ist. Auf die Datenbank kann durch einen Web-Browser oder eine Anwendungsprogrammierschnittstelle (API) für die Anwendungs- und Dienstintegration zugegriffen werden. Einige Cloud-Datenbanksysteme werden durch einen Anbieter betrieben, der die Backend-Prozesse der Datenbankinstallations-, Datenbankbereitstellungs- und Betriebsmittelzuweisungsaufgaben im Auftrag eines Kunden direkt managt. Der Kunde kann mehrere Endanwender aufweisen, die über einen Web-Browser und/oder eine API auf die Datenbank zugreifen. Die Cloud-Datenbanken können einigen Kunden durch das Abschwächen des Risikos des Verlierens von Datenbankdaten und das Erlauben, dass auf die Daten durch mehrere Anwender über mehrere geographische Gebiete zugegriffen wird, signifikante Vorteile bereitstellen.

[0009] Es gibt mehrere Architekturen für herkömmliche Datenbanksysteme und Cloud-Datenbanksysteme. Eine beispielhafte Architektur ist ein System mit gemeinsam benutzten Platten. In dem System mit gemeinsam benutzten Platten sind alle Daten in einer gemeinsam benutzten Speichervorrichtung gespeichert, die von allen Verarbeitungsknoten in einem Daten-Cluster zugänglich ist. In diesem Typ von System werden alle Datenänderungen in die gemeinsam benutzte Speichervorrichtung geschrieben, um sicherzustellen, dass alle Verarbeitungsknoten im Daten-Cluster auf eine konsistente Version der Daten zugreifen. Wenn die Anzahl der Verarbeitungsknoten in einem System mit gemeinsam benutzten Platten zunimmt, wird die gemeinsam benutzte Speichervorrichtung (und die Kommunikationsverbindungen zwischen den Verarbeitungsknoten und der gemeinsam benutzten Speichervorrichtung) ein Engpass, der die Datenlese- und -schreiboperationen verlangsamt. Dieser Engpass wird durch das Hinzufügen weiterer Verarbeitungsknoten noch verstärkt. Folglich weisen vorhandene Systeme mit gemeinsam benutzten Platten aufgrund dieses Engpassproblems eine eingeschränkte Skalierbarkeit auf.

[0010] In einigen Fällen kann es vorteilhaft sein, die Datenbankdaten an mehreren Orten oder in mehreren Speichervorrichtungen zu replizieren. Das Replizieren von Daten kann vor Systemausfällen schützen, die die Daten über ein Cloud-Netz unzugänglich machen können und/oder verursachen können, dass die Daten verloren werden oder permanent unlesbar werden. Das Replizieren der Datenbankdaten kann zusätzliche Vorteile und Verbesserungen bereitstellen, wie hier offenbart wird.

[0011] Angesichts des Vorangehenden werden hier Systeme, Verfahren und Vorrichtungen zur Datenbankreplikation offenbart.

Figurenliste

[0012] Nicht einschränkende und nicht vollständige Implementierungen der vorliegenden Offenbarung werden bezüglich der folgenden Figuren beschrieben, wobei sich überall in den verschiedenen Ansichten gleiche Bezugszeichen auf gleiche oder ähnliche Teile beziehen, wenn es nicht anderweitig spezifiziert ist. Die Vorteile der vorliegenden Offenbarung werden bezüglich der folgenden Beschreibung und der beigefügten Zeichnungen besser verstanden, wobei:

Fig. 1 einen Blockschaltplan von Komponenten eines Wiedergewinnungs- und Datenspeichersystems in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 2 einen Blockschaltplan veranschaulicht, der eine Ausführungsform eines Betriebsmittelmanagers in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung darstellt;

Fig. 3 einen Blockschaltplan veranschaulicht, der eine Ausführungsform einer Ausführungsplattform in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung darstellt;

Fig. 4 einen Blockschaltplan von Komponenten einer Betriebsumgebung in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 5 eine schematische graphische Darstellung eines Prozessablaufs zum Erzeugen einer Datenbank-Momentaufnahme in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 6 eine schematische graphische Darstellung eines Prozessablaufs zum Erzeugen eines Transaktionsprotokolls zum Replizieren einer Datenbank in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 7 eine schematische graphische Darstellung eines Prozessablaufs für zwei verschiedene Transaktionen zum Replizieren einer Datenbank in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 8 eine schematische graphische Darstellung eines Prozessablaufs veranschaulicht, der eine Vorbereitungsphase für die Replikation einer Datenbank in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung enthält;

Fig. 9 die Erzeugung und Übermittlung einer Auffrischungsanforderung in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 10 die Erzeugung und Übertragung einer Momentaufnahmeantwort in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 11 den Import einer Momentaufnahmeantwort in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 12 eine schematische graphische Darstellung einer Bereitstellungarchitektur in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 13 eine schematische graphische Darstellung eines Prozessablaufs zum Senden von Nachrichten in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 14 eine schematische graphische Darstellung eines Prozessablaufs zum Empfangen einer Nachricht in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht;

Fig. 15 eine schematische graphische Darstellung einer globalen Bereitstellungsgruppe in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschau licht;

Fig. 16 eine schematische graphische Darstellung eines Verschlüsselungssystems in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschau licht;

Fig. 17 eine schematische graphische Darstellung eines Verschlüsselungssystems in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschau licht;

Fig. 18 einen schematischen Ablaufplan eines Verfahrens für die Datenbank-Ausfallsicherung in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht; und

Fig. 19 ein Beispiel einer Rechenvorrichtung in Übereinstimmung mit den Lehren und Prinzipien der Offenbarung veranschaulicht.

AUSFÜHRLICHE BESCHREIBUNG

[0013] Es werden die Replikation und Ausfallsicherung von Datenbankdaten offenbart. Ein Verfahren enthält das Replizieren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind. Das Verfahren enthält das Ausführen einer oder mehrerer Aktualisierungen der Datenbankdaten in der sekundären Bereitstellung, wenn die primäre Bereitstellung nicht verfügbar ist, und das Übertragen der einen oder der mehreren Aktualisierungen zur primären Bereitstellung, wenn die primäre Bereitstellung wieder verfügbar wird. Das Verfahren enthält das Aus-

führen von Abfragen an den Datenbankdaten in der primären Bereitstellung, wenn die primäre Bereitstellung verfügbar ist.

[0014] Hier werden Systeme, Verfahren und Vorrichtungen für die Stapel-Datenbankreplikation und -ausfallsicherung zwischen mehreren Datenbankbereitstellungen oder Datenbankanbietern offenbart. Ein System der Offenbarung bewirkt, dass Datenbankdaten in einer primären Bereitstellung gespeichert und in einer oder mehreren sekundären Bereitstellungen repliziert werden. In dem Fall, dass die Daten in der primären Bereitstellung nicht verfügbar sind, können die Transaktionen an einer oder mehreren der sekundären Bereitstellungen ausgeführt werden. Wenn die ursprüngliche primäre Bereitstellung wieder verfügbar wird, können irgendwelche an den sekundären Bereitstellungen ausgeführten Transaktionen zu der primären Bereitstellung übertragen werden. Das System kann so konfiguriert sein, dass die Abfragen an die Datenbankdaten in der primären Bereitstellung zu irgendeinem Zeitpunkt ausgeführt werden, wenn die primäre Bereitstellung verfügbar ist.

[0015] In einigen Fällen ist es wünschenswert, die Datenbankdaten über mehrere Bereitstellungen zu replizieren. Für einige Datenbankkunden ist es unerlässlich, dass die in irgendwelchen sekundären Bereitstellungen gespeicherten Daten eine nicht veraltete und aktuelle Kopie der in der primären Bereitstellung gespeicherten Daten darstellen. Eine replizierte Datenbank kann für die Zwecke der Katastrophenwiederherstellung wünschenswert sein. Die eine oder die mehreren sekundären Bereitstellungen können als Reserve dienen, um die Operationen anzunehmen, falls die primäre Bereitstellung ausfällt oder anderweitig nicht verfügbar wird. Zusätzlich kann eine replizierte Datenbank zum Verbessern der Leseleistung wünschenswert sein. Die Leseleistung kann verbessert werden, indem eine Anforderung zu einer Bereitstellung weitergeleitet wird, die sich geographisch am nächsten bei dem Kundenkonto befindet, um die Gesamtlatenzzeit der Anforderungsverarbeitung zu verringern. In Anbetracht des Vorhergehenden stellen die hier offenbarten Systeme, Verfahren und Vorrichtungen Mittel bereit, um eine transaktionell konsistente Kopie einer primären Bereitstellung bereitzustellen und zu aktualisieren, so dass die eine oder die mehreren sekundären Bereitstellungen jederzeit mit der primären Bereitstellung synchronisiert sind.

[0016] In einer Ausführungsform werden die Datenbankdaten zwischen einer primären Bereitstellung und einer oder mehreren sekundären Bereitstellungen repliziert. In einer weiteren Ausführungsform wird ein Ausfallsicherung von der primären Bereitstellung zu einer sekundären Bereitstellung ausgeführt, wobei ein Failback von der sekundären Bereitstellung zurück zur ursprünglichen primären Bereitstellung ausgeführt werden kann.

[0017] In einer Ausführungsform wird ein Verfahren zur Ausfallsicherung von Datenbankdaten zwischen mehreren Bereitstellungen offenbart. Das Verfahren enthält das Replizieren von in einer primären Bereitstellung gespeicherten Datenbankdaten, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind. Das Verfahren enthält in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, das Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung. Das Verfahren enthält in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger verfügbar ist, das Übertragen der einen oder der mehreren Transaktionen an den Datenbankdaten zu der primären Bereitstellung. Das Verfahren enthält, während die primäre Bereitstellung verfügbar ist, das Ausführen von Abfragen an den Datenbankdaten in der primären Bereitstellung.

[0018] Die Datenbankdaten können in einem cloud-basierten Speicher gespeichert sein, der über geographische Gebiete zugänglich ist. Dieser cloud-basierte Speicher bezieht sich auf Datenbankdaten, die in einem Speichersystem außerhalb des Standorts gespeichert sind, das in einigen Implementierungen durch einen Dritten aufrechterhalten werden kann. Ein Kunde kann z. B. wählen, die Daten bei einem Cloud-Speicheranbieter bereitzustellen, anstatt die Daten auf einer lokalen Computerfestplatte oder einer anderen lokalen Speichervorrichtung, die dem Kunden gehört, zu speichern. Der Kunde kann über eine Internetverbindung zwischen den Computerbetriebsmitteln des Kunden und den Speicherbetriebsmitteln außerhalb des Standorts, die die Daten des Kunden speichern, auf die Daten zugreifen. Die Cloud-Speicherung von Datenbankdaten kann gegenüber der herkömmlichen lokalen Speicherung vor Ort mehrere Vorteile bereitstellen. Wenn die Datenbankdaten in einem Cloud-Speicher gespeichert sind, kann auf die Informationen an irgendeinem Ort zugegriffen werden, der eine Internetverbindung aufweist. Deshalb ist es nicht erforderlich, dass ein Datenbankkunde physische Speichervorrichtungen bewegt oder denselben Computer verwendet, um die Datenbankinformationen zu speichern, zu aktualisieren oder wiederzugewinnen. Ferner kann durch mehrere Anwender an verschiedenen geographischen Orten gleichzeitig auf die Datenbankinformationen zugegriffen werden und können die Datenbankinformationen durch mehrere Anwender an verschiedenen geographischen Orten gleichzeitig aktualisiert und gespeichert werden. Der Kunde kann Kopien von Dateien über das Internet an einen Daten-Server senden, der dem Cloud-Speicheranbieter zugeordnet ist und der die Dateien aufzeichnet. Der Kunde kann durch das Zugreifen über eine web-basierte Schnittstelle oder eine andere Anwenderschnittstelle auf den dem Cloud-

Speicheranbieter zugeordneten Daten-Server die Daten wiedergewinnen. Der dem Cloud-Speicheranbieter zugeordnete Daten-Server kann dann die Dateien an den Kunden zurücksenden oder dem Kunden erlauben, auf die Dateien auf dem Daten-Server selbst zuzugreifen und sie zu manipulieren.

[0019] Die Cloud-Speichersysteme enthalten in der Regel Hunderte oder Tausende von Daten-Servern, die mehrere Clients bedienen können. Weil die Computer gelegentlich eine Wartung oder eine Reparatur erfordern und weil Computer gelegentlich ausfallen, ist es wichtig, die gleichen Informationen auf mehreren Maschinen zu speichern. Diese Redundanz kann sicherstellen, dass Kunden sogar im Fall eines Server-Ausfalls zu jedem gegebenen Zeitpunkt auf ihre Daten zugreifen können.

[0020] In einer Ausführungsform der Offenbarung sind die Datenbankdaten über mehrere Cloud-Speicherbereitstellungen gespeichert. Derartige Cloud-Speicherbereitstellungen können sich an verschiedenen geographischen Orten befinden, wobei die Datenbankdaten über mehrere Maschinen und/oder Server in jeder der Bereitstellungen gespeichert sein können. Die Cloud-Speicherbereitstellungen können sich an einem einzigen geographischen Ort befinden, aber mit verschiedenen Leistungsversorgungen verbunden sein und/oder unterschiedliche Rechenmaschinen zum Speichern der Daten verwenden. Die Cloud-Speicherbereitstellungen können durch verschiedene Cloud-Speicheranbieter betrieben werden. In derartigen Ausführungsformen werden die Datenbankdaten über die mehreren Bereitstellungen repliziert, so dass in dem Fall, dass eine Bereitstellung nicht verfügbar wird oder ausfällt, auf die Datenbankdaten weiterhin zugegriffen werden kann und sie weiterhin aktualisiert und gespeichert werden können. In einer Ausführungsform sind die Datenbankdaten in einer primären Bereitstellung gespeichert, wobei sie ferner in einer oder mehreren sekundären Bereitstellungen gespeichert sind. Die primäre Bereitstellung kann jederzeit, wenn die primäre Bereitstellung verfügbar ist, zum Zugreifen, Abfragen und Aktualisieren von Daten verwendet werden. Die eine oder die mehreren sekundären Bereitstellungen können Operationen übernehmen, falls und wenn die primäre Bereitstellung nicht verfügbar wird. Wenn die primäre Bereitstellung wieder verfügbar wird, kann die primäre Bereitstellung mit irgendwelchen Änderungen aktualisiert werden, die an der einen oder den mehreren sekundären Bereitstellungen geschehen sind, als die primäre Bereitstellung nicht verfügbar war. Die aktualisierte primäre Bereitstellung kann dann die Operationen wiederaufnehmen, einschließlich des Zugreifens, des Abfragens und des Aktualisierens von Daten.

[0021] Wenn Daten über mehrere Bereitstellungen gespeichert sind, ist es wichtig, sicherzustellen, dass die Daten über jede der Bereitstellungen konsistent sind. Wenn Daten in einer primären Bereitstellung aktualisiert, modifiziert oder hinzugefügt werden, können die Aktualisierungen über die eine oder die mehreren sekundären Bereitstellungen übertragen werden, um sicherzustellen, dass alle Bereitstellungen eine konsistente und aktuelle Version der Daten aufweisen. In dem Fall, dass eine primäre Bereitstellung nicht verfügbar wird, kann jede der aktuellen sekundären Bereitstellungen den Betrieb der Daten übernehmen, ohne dass die Daten veraltet oder falsch sind. Wenn irgendeine der mehreren Bereitstellungen nicht verfügbar wird, kann ferner die Bereitstellung später mit allen Änderungen aktualisiert werden, die während des Zeitraums, in dem die Bereitstellung nicht verfügbar war, vorgenommen wurden. Wenn die Bereitstellung aktualisiert wird, nachdem sie „offline“ oder nicht verfügbar war, kann es vorteilhaft sein, sicherzustellen, dass die Bereitstellung nur mit jenen Änderungen aktualisiert wird, die während des Zeitraums ausgeführt wurden, in dem die Bereitstellung nicht verfügbar war.

[0022] Vorhandene Herangehensweisen zur Datenreplikation sind typischerweise durch eine Momentaufnahmestrategie oder eine Protokollierungsstrategie implementiert. Die Momentaufnahmestrategie erzeugt eine serialisierte Repräsentation des aktuellen Zustands der Quelldatenbank, nachdem eine Änderung an der Quelldatenbank vorgenommen worden ist. Die Zieldatenbank wird dann basierend auf der Momentaufnahme neu befüllt, wobei dies für jede an der Quelldatenbank ausgeführte Änderung geschieht. Die Protokollierungsstrategie beginnt mit einem anfänglichen (d. h. leeren) Datenbankzustand und zeichnet eine Änderung auf, die durch jede erfolgreiche Transaktion gegen die Quelldatenbank vorgenommen worden ist. Die Reihenfolge der Änderungen definiert das „Transaktionsprotokoll“ der Quelldatenbank, wobei jede Änderung im Transaktionsprotokoll in genau der gleichen Reihenfolge gegen die Zieldatenbank wiedergegeben wird.

[0023] Die Momentaufnahmestrategie löst die Replikation durch das Aufnehmen einer Momentaufnahme der Quelldatenbank und das Instanziieren der Zieldatenbank aus der Momentaufnahme. Bei der Momentaufnahmestrategie hängt jedoch das Erzeugen oder das Verbrauchen einer Momentaufnahme grob von der in der Anzahl der zu replizierenden Objekte und potentiell der Anzahl der gespeicherten Bytes gemessenen Größe der Datenbank ab. Die Momentaufnahmestrategie erfordert potentiell eine O(Größe der Datenbank)-Operation für jede Transaktion, um eine aktuelle Zieldatenbank aufrechtzuerhalten. Das Ausführen einer O(Größe der

Datenbank)-Operation nach jeder erfolgreichen Transaktion an der Quelldatenbank kann für alle außer kleine oder relativ statische Datenbanken unpraktisch sein.

[0024] Die Protokollierungsstrategie versucht, die Probleme bei der Momentaufnahmestrategie durch das Verringern der Kosten des Übertragens der durch eine einzelne Transaktion vorgenommenen Änderungen herunter auf nur etwa die Größe der Transaktion selbst zu lösen. Das Ausführen einer O(Größe der Transaktion)-Operation nach jeder erfolgreichen Transaktion, die die Datenbank modifiziert, kann weniger Rechenbetriebsmittel erfordern. Die Protokollierungsstrategie erfordert jedoch einen Protokolldatensatz für jede auf die Quelldatenbank angewendete Transaktion, seit sie erzeugt worden ist, um eine Kopie der Zieldatenbank zu erzeugen. Das Ausführen einer O(Größe des Transaktionsprotokolls)-Operation zum Bootstrapping einer Zieldatenbank kann teurer als das Bootstrapping aus einer Momentaufnahme sein. Zusätzlich kann das Protokollieren weniger widerstandsfähig gegenüber Fehlern in der Replikationslogik sein, weil die Fehler in der Replikationslogik zu Inkonsistenz oder Drift zwischen der Quelldatenbank und der Zieldatenbank führen können. Wenn eine Drift auftritt, ist es unerlässlich, dass sie so schnell wie möglich korrigiert wird. Wenn sich der Fehler in der Quelldatenbank befindet (d. h., bei der Erzeugung der Protokollaufzeichnungen), dann ist er bereits im Transaktionsprotokoll selbst eingebacken, und dies schwierig einzustellen oder zu korrigieren sein kann. Wenn sich der Fehler in der Zieldatenbank befindet (d. h., beim Verbrauch von Protokolldatensätzen), dann könnte das Ziel alternativ durch Wiedergeben des Transaktionsprotokolls von Anfang an neu erzeugt werden, wobei dies aber signifikante Rechenbetriebsmittel erfordern kann.

[0025] In bestimmten Implementierungen ist weder die Momentaufnahmestrategie noch die Protokollierungsstrategie zum Replizieren von Datenbankdaten praktisch oder durchführbar. Die hier offenbarte Hybridstrategie kombiniert Momentaufnahmen mit einem Transaktionsprotokoll.

[0026] Die hier offenbarte Hybridherangehensweise für die Datenbankreplikation kombiniert die Verwendung von Momentaufnahmen mit der Verwendung eines Transaktionsprotokolls. Diese hier offenbarte Herangehensweise ermöglicht die Transaktionsprotokollierung in der Quelldatenbank und ermöglicht die periodische Erzeugung von Momentaufnahmen in der Quelldatenbank. Die Hybridherangehensweise führt ferner eine anfängliche Instanziierung an der Zieldatenbank basierend auf der neuesten Momentaufnahme der Quelldatenbank aus. Die Hybridherangehensweise enthält das Wiedergeben (nach der Momentaufnahme) eines Transaktionsprotokolldatensatzes in der Zieldatenbank in der gleichen Reihenfolge, in der er in der Quelldatenbank angewendet wurde. Die Hybridherangehensweise enthält ferner ein periodisches Auffrischen der Zieldatenbank basierend auf einer neueren Momentaufnahme und wendet Transaktionsprotokolldatensätze nach der Momentaufnahme weiterhin an. Wie hier offenbart ist, ist die Hybridherangehensweise konfiguriert, sicherzustellen, dass sowohl die Protokolldatensätze als auch die Momentaufnahmen verfügbar sind, wobei sie ferner konfiguriert ist, die anfängliche Bootstrapping-Zeit auf einem Minimum zu halten, um sicherzustellen, dass der anfängliche Zielzustand bezüglich der Quelldatenbank angemessen aktuell ist. Die Hybridherangehensweise ermöglicht ferner eine kostengünstige Herangehensweise, um die Zieldatenbank mit der Quelldatenbank auf den neuesten Stand zu bringen und zu halten. Die Hybridherangehensweise ermöglicht ferner sowohl eine schnelle Korrektur der Drift als auch einen schnellen Aufholpfad für irgendwelche Kopien, die z. B. aufgrund von Kopieausfallzeiten, kleinen Dienst- oder Netzproblemen, die zu Verarbeitungsverzögerungen führen, usw., weit hinter die Quelle zurückgefallen sein können.

[0027] In einer Ausführungsform werden die in einer primären Bereitstellung gespeicherte Datenbankdaten so repliziert, dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind. Die primäre Bereitstellung kann z. B. aufgrund einer geplanten Ausfallzeit für eine Wartung oder Aktualisierungen, eines Leistungsausfalls, eines Systemausfalls, eines Ausfalls des Rechenzentrums, eines Fehlers, der zu einer ungeeigneten Änderung oder einer Löschung von Datenbankdaten führt, eines Ausfalls des Cloud-Anbieters usw. nicht verfügbar werden. In Reaktion, dass die primäre Bereitstellung nicht verfügbar wird, werden eine oder mehrere Transaktionen an den Datenbankdaten in der sekundären Bereitstellung ausgeführt. Die primäre Bereitstellung kann wieder verfügbar werden, wobei die eine oder mehrere Transaktionen, die in der sekundären Bereitstellung ausgeführt wurden, zu der primären Bereitstellung übertragen werden. Die Abfragen an den Datenbankdaten können in der primären Bereitstellung ausgeführt werden, wenn die primäre Bereitstellung verfügbar ist.

[0028] Eine Datenbanktabelle kann in Reaktion auf eine Anweisung einer Datenmanipulationssprache (DML), wie z. B. einen Einfügebefehl, einen Löschbefehl, einen Verschmelzungsbefehl usw., geändert werden. Derartige Modifikationen können als eine Transaktion bezeichnet werden, die in der Datenbanktabelle stattgefunden hat (wobei die Modifikation hier alternativ als eine „Aktualisierung“ bezeichnet werden kann). In einer Ausführungsform enthält jede Transaktion einen Zeitstempel, der angibt, wann die Transaktion empfangen wurde

und/oder wann die Transaktion vollständig ausgeführt war. In einer Ausführungsform enthält eine Transaktion mehrere Änderungen, die an einer Tabelle vorgenommen wurden, wobei sich derartige Änderungen auf eine oder mehrere Mikropartitionen in der Tabelle auswirken können.

[0029] Eine Datenbanktabelle kann Daten in mehreren Mikropartitionen speichern, wobei die Mikropartitionen unveränderliche Speichervorrichtungen sind. Wenn eine Transaktion in einer derartigen Tabelle ausgeführt wird, werden alle betroffenen Mikropartitionen neu erzeugt, um neue Mikropartitionen zu erzeugen, die die Modifikationen der Transaktion widerspiegeln. Nachdem eine Transaktion vollständig ausgeführt worden ist, können dann irgendwelche ursprünglichen Mikropartitionen, die neu erzeugt wurden, aus der Datenbank entfernt werden. Nach jeder Transaktion, die in der Tabelle ausgeführt wird, wird eine neue Version der Tabelle erzeugt. Die Tabelle kann während eines Zeitraums viele Versionen erfahren, falls die Daten in der Tabelle viele Änderungen erfahren, wie z. B. Einfügungen, Löschungen und/oder Verschmelzungen. Jede Version der Tabelle kann Metadaten enthalten, die angeben, welche Transaktion die Tabelle erzeugt hat, wann die Transaktion bestellt wurde, wann die Transaktion vollständig ausgeführt war und wie die Transaktion eine oder mehrere Zeilen in der Tabelle geändert hat. Die offenbarten Systeme, Verfahren und Vorrichtungen zur kostengünstigen Versionierung von Tabellen können wirksam eingesetzt werden, um effiziente Mittel zum Erzeugen einer umfassenden Änderungsverfolgungs-Zusammenfassung bereitzustellen, die alle Zwischenänderungen angibt, die an einer Tabelle zwischen einem ersten und einem zweiten Zeitstempel ausgeführt worden sind. In einer Ausführungsform gibt der erste Zeitstempel einen Zeitpunkt an, zu dem eine primäre Bereitstellung nicht verfügbar wird, während der zweite Zeitstempel einen Zeitpunkt angibt, zu dem die primäre Bereitstellung zur Verfügbarkeit zurückgekehrt ist.

[0030] In einer Ausführungsform werden alle Daten in den Tabellen automatisch in eine unveränderliche Speichervorrichtung aufgeteilt, die als eine Mikropartition bezeichnet wird. Die Mikropartition kann als eine Stapeleinheit betrachtet werden, wobei jede Mikropartition zusammenhängende Speichereinheiten aufweist. Beispielhaft kann jede Mikropartition zwischen 50 MB und 500 MB unkomprimierte Daten enthalten (es wird angegeben, dass die tatsächliche Größe des Speichers kleiner sein kann, da die Daten komprimiert gespeichert werden können). Gruppen von Zeilen in den Tabellen können in einzelne Mikropartitionen abgebildet sein, die spaltenweise organisiert sind. Diese Größe und Struktur ermöglichen ein extrem granulares Kürzen sehr großer Tabellen, die Millionen oder sogar Hunderte von Millionen von Mikropartitionen umfassen können. Die Metadaten können automatisch über alle in einer Mikropartition gespeicherten Zeilen einschließlich: des Wertebereichs für jede der Spalten in der Mikropartition, der Anzahl der verschiedenen Werte und/oder zusätzlicher Eigenschaften, die sowohl zur Optimierung als auch zur effizienten Abfrageverarbeitung verwendet werden, gesammelt werden. In einer Ausführungsform kann die Mikropartitionierung automatisch an allen Tabellen ausgeführt werden. Die Tabellen können z. B. unter Verwendung der Reihenfolge, die auftritt, wenn die Daten eingefügt/geladen werden, transparent partitioniert werden.

[0031] Das Abfragen der Auflistung der dazwischenliegenden Modifikationen stellt ein effizientes und kostengünstiges Mittel zum Bestimmen einer umfassenden Auflistung der inkrementellen Änderungen bereit, die an einer Datenbanktabelle zwischen zwei Zeitpunkten vorgenommen worden sind. Dies ist den in der Technik bekannten Verfahren überlegen, wo jede einer Reihe von folgenden Tabellenversionen manuell verglichen werden muss, um zu bestimmen, wie die Tabelle im Laufe der Zeit modifiziert worden ist. Derartige in der Technik bekannten Verfahren erfordern umfangreiche Speicherbetriebsmittel und Rechenbetriebsmittel, um ausgeführt zu werden.

[0032] In einer Ausführungsform sind die Datei-Metadaten innerhalb eines Metadatenspeichers gespeichert. Die Datei-Metadaten enthalten Tabellenversionen und Informationen über jede Tabellendaten-Mikropartition. Der Metadatenspeicher kann einen veränderlichen Speicher (einen Speicher, der überschrieben oder an Ort und Stelle geschrieben werden kann), z. B. ein lokales Dateisystem, ein lokales System, einen lokalen Speicher oder dergleichen, enthalten. In einer Ausführungsform bestehen die Mikropartitions-Metadaten aus zwei Datensätzen: Tabellenversionen und Mikropartitions-Informationen. Der Datensatz der Tabellenversionen enthält eine Abbildung der Tabellenversionen auf Listen von hinzugefügten Mikropartitionen und entfernten Mikropartitionen. Die Mikropartitions-Informationen bestehen aus Informationen über Daten innerhalb der Mikropartition, einschließlich z. B. des Mikropartitionspfads, der Mikropartitionsgröße, der Mikropartitionsschlüssel-ID und der Zusammenfassungen aller Zeilen und Spalten, die in der Mikropartition gespeichert sind. Jede Modifikation der Tabelle erzeugt neue Mikropartitionen und neue Mikropartitions-Metadaten. Die Einfügungen in die Tabelle erzeugen neue Mikropartitionen. Die Löschungen aus der Tabelle entfernen Mikropartitionen und fügen potentiell neue Mikropartitionen mit den verbleibenden Zeilen in einer Tabelle hinzu, falls nicht alle Zeilen in einer Mikropartition gelöscht wurden. Die Aktualisierungen entfernen Mikropartitionen und ersetzen sie durch neue Mikropartitionen mit Zeilen, die die geänderten Datensätze enthalten.

[0033] In einer Ausführungsform können die Metadaten in Metadaten-Mikropartitionen in einem unveränderlichen Speicher gespeichert sein. In einer Ausführungsform kann ein System für jede Modifikation einer Datenbanktabelle Metadaten-Mikropartitionen in den Cloud-Speicher schreiben. In einer Ausführungsform kann ein System Metadaten-Mikropartitionen herunterladen und lesen, um die Abtastmenge zu berechnen. Die Metadaten-Mikropartitionen können parallel heruntergeladen und gelesen werden, wie sie empfangen werden, um die Berechnung der Abtastmenge zu verbessern. In einer Ausführungsform kann ein System die Metadaten-Mikropartitionen im Hintergrund periodisch konsolidieren. In einer Ausführungsform können Leistungsverbesserungen, einschließlich des Vorab-Abrufens, des Caching, der Spaltenanordnung und dergleichen enthalten sein. Weiterhin sind Sicherheitsverbesserungen einschließlich der Verschlüsselung und Integritätsprüfung außerdem bei Metadaten-Dateien mit einer Spaltenanordnung möglich.

[0034] In einer Ausführungsform ist die Initialisierung und Wartung einer Kopie über eine Kombination aus Erzeugung/Verbrauch von Datenbank-Momentaufnahmen und Erzeugung/Verbrauch von Transaktionsprotokolldatensätzen implementiert. Eine Kopie kann aus einer Momentaufnahme erzeugt und inkrementell auf einzelne Transaktionsdatensätze angewendet werden, so dass die Kopie mit der Quelldatenbank synchronisiert ist. In einer Ausführungsform wird die Kopie periodisch basierend auf einer Momentaufnahme aufgefrischt, selbst wenn die Kopie basierend auf inkrementellen Transaktionsaktualisierungen als aktuell betrachtet wird. Die periodische Auffrischung basierend auf der Momentaufnahme kann die Probleme der Drift aufgrund von Fehlern und andere Probleme behandeln. In einer Ausführungsform werden die Momentaufnahmen und die Transaktionsprotokolldatensätze für die Sichtbarkeit der Kreuz-Bereitstellung in einen entfernten Speicher geschrieben. Die Modifikationen an der Transaktionsverarbeitungsinfrastruktur können benutzt werden, um die Transaktionskonsistenz zwischen dem Transaktionszustand der Quelldatenbank und dem Aussehen der Transaktionsprotokolldatensätze im entfernten Speicher sicherzustellen. In einer Ausführungsform können die Modifikationen an der Verarbeitungslogik der Datendefinitionssprache (DDL) in einen Transaktionsverarbeitungs-Arbeitsablauf integriert sein, um die Konsistenz der DDL-Anwendung und das Aussehen des Transaktionsprotokolldatensatzes im entfernten Speicher sicherzustellen.

[0035] In der folgenden Beschreibung der Offenbarung wird auf die beigefügten Zeichnungen Bezug genommen, die einen Teil davon bilden und in denen zur Veranschaulichung spezifische Implementierungen gezeigt sind, in denen die Offenbarung praktiziert werden kann. Es wird erkannt, dass andere Implementierungen verwendet werden können und strukturelle Änderungen vorgenommen werden können, ohne vom Schutzumfang der Offenbarung abzuweichen.

[0036] Beim Beschreiben und Beanspruchen der Offenbarung wird die folgende Terminologie in Übereinstimmung mit den im Folgenden dargelegten Definitionen verwendet.

[0037] Es muss angegeben werden, dass die Einzahlformen „ein“, „eine“ und „der/die/das“, wie sie in dieser Beschreibung und den beigefügten Ansprüchen verwendet werden, Verweise auf die Mehrzahl enthalten, es sei denn, der Kontext schreibt es deutlich anders vor.

[0038] Die Verweise überall in dieser Beschreibung auf „eine einzige Ausführungsform“, „eine Ausführungsform“, „eine einzige Implementierung“, „eine Implementierung“, „ein einziges Beispiel“ oder „ein Beispiel“ bedeuten, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder ein bestimmtes Merkmal, das im Zusammenhang mit der Ausführungsform, der Implementierung oder dem Beispiel beschrieben wird, in wenigstens einer Ausführungsform der vorliegenden Offenbarung enthalten ist. Folglich beziehen sich nicht notwendigerweise alle Auftritte der oben identifizierten Ausdrücke an verschiedenen Stellen überall in dieser Beschreibung auf dieselbe Ausführungsform, dieselbe Implementierung oder dasselbe Beispiel. Zusätzlich sollte erkannt werden, dass die hiermit bereitgestellten Figuren für Erklärungszwecke für die Durchschnittsfachleute auf dem Gebiet sind.

[0039] Die Begriffe „umfassen“, „einschließen“, „enthalten“ und deren grammatikalische Äquivalente, wie sie hier verwendet werden, sind einschließende oder nicht abschließende Begriffe, die zusätzliche, nicht aufgeführte Elemente oder Verfahrensschritte nicht ausschließen.

[0040] Eine „Tabelle“, wie sie hier verwendet wird, ist als eine Sammlung von Datensätzen (Zeilen) definiert. Jeder Datensatz enthält eine Sammlung von Werten von Tabellenattributen (Spalten). Die Tabellen sind typischerweise physisch in mehreren kleineren Speichereinheiten (mit variierender Größe oder fester Größe), z. B. Dateien oder Blöcken, gespeichert.

[0041] Eine „Partitionierung“, wie sie hier verwendet wird, ist als Datensätze mit unterschiedlichen Daten physisch trennend definiert, um Datenpartitionen zu trennen. Eine Tabelle kann z. B. Daten basierend auf dem Länderattribut partitionieren, was zu einer Partition pro Land führt.

[0042] Die Ausführungsformen in Übereinstimmung mit der vorliegenden Offenbarung können als eine Vorrichtung, ein Verfahren oder ein Computerprogrammprodukt verkörpert sein. Entsprechend kann die vorliegende Offenbarung die Form einer vollständig aus Hardware bestehenden Ausführungsform, einer vollständig aus Software bestehenden Ausführungsform (einschließlich Firmware, residenter Software, Mikrocode usw.) oder einer Ausführungsform annehmen, die Software- und Hardware-Aspekte kombiniert, die hier alle im Allgemeinen als eine „Schaltung“, ein „Modul“ oder ein „System“ bezeichnet werden können. Weiterhin können Ausführungsformen der vorliegenden Offenbarung die Form eines Computerprogrammprodukts annehmen, das in irgendeinem greifbaren Ausdrucksmedium verkörpert ist, das einen in dem Medium verkörperten computerverwendbaren Programmcode enthält.

[0043] Es kann irgendeine Kombination aus einem oder mehreren computerverwendbaren oder computerlesbaren Medien verwendet werden. Ein computerlesbares Medium kann z. B. eine oder mehrere einer tragbaren Computerdiskette, einer Festplatte, einer Schreib-Lese-Speicher-Vorrichtung (RAM-Vorrichtung), einer Festwertspeicher-Vorrichtung (ROM-Vorrichtung), einer löschbaren programmierbaren Festwertspeicher-Vorrichtung (EPROM- oder Flash-Speicher-Vorrichtung), eines tragbaren Kompaktplatten-Festwertspeichers (CDROM), einer optischen Speichervorrichtung und einer magnetischen Speichervorrichtung enthalten. Der Computerprogrammcode zum Ausführen der Operationen der vorliegenden Offenbarung kann in irgendeiner Kombination aus einer oder mehrerer Programmiersprachen geschrieben sein. Derartiger Code kann aus dem Quellcode in eine computerlesbare Assemblersprache oder in einen Maschinencode kompiliert werden, der für die Vorrichtung oder den Computer, in der bzw. dem der Code ausgeführt wird, geeignet ist.

[0044] Die Ausführungsformen können außerdem in Cloud-Rechenumgebungen implementiert sein. In dieser Beschreibung und den folgenden Ansprüchen kann der „Cloud-Computerbereich“ als ein Modell zum Ermöglichen eines allgegenwärtigen, zweckmäßigen Netzzugangs auf Anforderung zu einem gemeinsam benutzten Pool konfigurierbarer Rechenbetriebsmittel (z. B. Netze, Server, Speicher, Anwendungen und Dienste) definiert sein, die durch Virtualisierung schnell bereitgestellt und mit minimalem Managementsaufwand oder minimaler Dienstanbieter-Wechselwirkung freigegeben und dann entsprechend skaliert werden können. Ein Cloud-Modell kann sich aus verschiedenen Eigenschaften (z. B. Selbstbedienung auf Anforderung, breiter Netzzugang, Betriebsmittel-Zusammenlegung, schnelle Elastizität und gemessener Dienst), Dienstmodellen (z. B. Software als ein Dienst („SaaS“), Plattform als ein Dienst („PaaS“) und Infrastructure als ein Dienst („IaaS“)) und Bereitstellungsmodellen (z. B. private Cloud, Gemeinschafts-Cloud, öffentliche Cloud und Hybrid-Cloud) zusammensetzen.

[0045] Die Ablaufpläne und Blockschaltpläne in den beigefügten Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Implementierungen der Systeme, Verfahren und Computerprogrammprodukte gemäß den verschiedenen Ausführungsformen der vorliegenden Offenbarung. In dieser Hinsicht kann jeder Block in den Ablaufplänen oder Blockschaltplänen ein Modul, ein Segment oder einen Teil des Codes darstellen, der einen oder mehrere ausführbare Befehle zum Implementieren der spezifizierten logischen Funktion(en) umfasst. Es wird außerdem angegeben, dass jeder Block der Blockschaltpläne und/oder Ablaufpläne und Kombinationen aus Blöcken in den Blockschaltplänen und/oder Ablaufplänen durch hardwarebasierte Spezialsysteme, die die spezifizierten Funktionen oder Handlungen ausführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen implementiert sein kann. Diese Computerprogrammanweisungen können außerdem in einem computerlesbaren Medium gespeichert sein, das einen Computer oder eine andere programmierbare Datenverarbeitungsvorrichtung steuern kann, in einer speziellen Weise zu funktionieren, so dass die in dem computerlesbaren Medium gespeicherten Anweisungen einen Herstellungsartikel erzeugen, der Anweisungsmittel enthält, die die in dem Block oder den Blöcken im Ablaufplan und/oder im Blockschaltplan spezifizierte Funktion/Handlung implementieren.

[0046] Die hier beschriebenen Systeme und Verfahren können in einem flexiblen und skalierbaren Datenlager unter Verwendung einer neuen Datenverarbeitungsplattform arbeiten. In einigen Ausführungsformen setzen die beschriebenen Systeme und Verfahren eine Cloud-Infrastruktur wirksam ein, die cloud-basierte Speicherbetriebsmittel, Rechenbetriebsmittel und dergleichen unterstützt. Beispielhafte cloud-basierte Speicherbetriebsmittel bieten eine signifikante Speicherkapazität, die auf Anforderung verfügbar ist, zu geringen Kosten an. Ferner können diese cloud-basierten Speicherbetriebsmittel fehlertolerant und im hohen Grade skalierbar sein, was in privaten Datenspeichersystemen kostspielig zu erreichen sein kann. Beispielhafte cloud-basierte Rechenbetriebsmittel sind auf Anforderung verfügbar und können basierend auf den tatsächlichen Nutzungsni-

veaus der Betriebsmittel mit einem Preis versehen werden. Typischerweise wird die Cloud-Infrastruktur schnell dynamisch eingesetzt, rekonfiguriert und stillgelegt.

[0047] In den beschriebenen Systemen und Verfahren verwendet ein Datenspeichersystem eine SQL-basierte (auf einer strukturierten Abfragesprache basierende) relationale Datenbank. Diese Systeme und Verfahren sind jedoch unter Verwendung irgendeiner Datenspeicherarchitektur und unter Verwendung irgendeiner Sprache, um die Daten innerhalb der Datenspeicher- und -wiedergewinnungsplattform zu speichern und wiederzugewinnen, auf irgendeinen Typ von Datenbank und irgendeinen Typ von Datenspeicher- und -wiedergewinnungsplattform anwendbar. Die hier beschriebenen Systeme und Verfahren stellen ferner ein mandantenfähiges System bereit, das die Isolierung von Rechenbetriebsmitteln und Daten zwischen verschiedenen Kunden/Klienten und zwischen verschiedenen Anwendern innerhalb desselben Kunden/Klienten unterstützt.

[0048] In **Fig. 1** ist nun ein Computersystem zum Ausführen der hier offenbarten Verfahren veranschaulicht. Wie in **Fig. 1** gezeigt ist, kann der Betriebsmittelmanager **102** an mehrere Anwender **104**, **106** und **108** gekoppelt sein. In speziellen Implementierungen kann der Betriebsmittelmanager **102** irgendeine Anzahl von Anwendern unterstützen, die Zugang zur Datenverarbeitungsplattform **100** wünschen. Die Anwender **104**, **106**, **108** können z. B. Endanwender, die Datenspeicher- und -wiedergewinnungsanforderungen bereitstellen, Systemmanager, die die hier beschriebenen Systeme und Verfahren managen, und andere Komponenten/Vorrichtungen, die mit dem Betriebsmittelmanager **102** wechselwirken, enthalten.

[0049] Der Betriebsmittelmanager **102** stellt verschiedene Dienste und Funktionen bereit, die den Betrieb aller Systeme und Komponenten innerhalb der Datenverarbeitungsplattform **100** unterstützen. Der Betriebsmittelmanager **102** kann an die Metadaten **110** gekoppelt sein, die der Gesamtheit der überall in der Datenverarbeitungsplattform **100** gespeicherten Daten zugeordnet sind. In einigen Ausführungsformen können die Metadaten **110** eine Zusammenfassung sowohl der Daten, die in entfernten Datenspeichersystemen gespeichert sind, als auch der Daten, die aus einem lokalen Cache verfügbar sind, enthalten. Zusätzlich können die Metadaten **110** Informationen hinsichtlich dessen enthalten, wie die Daten in den entfernten Datenspeichersystemen und den lokalen Caches organisiert sind. Die Metadaten **110** können es den Systemen und Diensten ermöglichen, zu bestimmen, ob ein Teil der Daten verarbeitet werden muss, ohne die tatsächlichen Daten aus einer Speichervorrichtung zu laden oder auf sie zuzugreifen.

[0050] Der Betriebsmittelmanager **102** kann ferner an die Ausführungsplattform **112** gekoppelt sein, die mehrere Rechenbetriebsmittel bereitstellt, die verschiedene Datenspeicherungs- und Datenwiedergewinnungsaufgaben ausführen, wie im Folgenden ausführlicher erörtert wird. Die Ausführungsplattform **112** kann an mehrere Datenspeichervorrichtungen **116**, **118** und **120** gekoppelt sein, die ein Teil einer Speicherplattform **114** sind. Obwohl in **Fig. 1** drei Datenspeichervorrichtungen **116**, **118** und **120** gezeigt sind, kann die Ausführungsplattform **112** mit irgendeiner Anzahl von Datenspeichervorrichtungen kommunizieren. In einigen Ausführungsformen sind die Datenspeichervorrichtungen **116**, **118** und **120** cloud-basierte Speichervorrichtungen, die sich an einem oder mehreren geographischen Orten befinden. Die Datenspeichervorrichtungen **116**, **118** und **120** können z. B. ein Teil einer öffentlichen Cloud-Infrastruktur oder einer privaten Cloud-Infrastruktur sein. Die Datenspeichervorrichtungen **116**, **118** und **120** können Festplattenlaufwerke (HDDs), Festkörperlaufwerke (SSDs), Speicher-Cluster oder irgendeine andere Datenspeichertechnik sein. Zusätzlich kann die Speicherplattform **114** verteilte Dateisysteme (wie z. B. verteilte Hadoop-Dateisysteme (HDFS)), Objektspeichersysteme und dergleichen enthalten.

[0051] In speziellen Ausführungsformen sind die Kommunikationsverbindungen zwischen dem Betriebsmittelmanager **102** und den Anwendern **104**, **106**, **108**, den Metadaten **110** und der Ausführungsplattform **112** über ein oder mehrere Datenkommunikationsnetze implementiert. Ähnlich sind die Kommunikationsverbindungen zwischen der Ausführungsplattform **112** und den Datenspeichervorrichtungen **116**, **118**, **120** in der Speicherplattform **114** über ein oder mehrere Datenkommunikationsnetze implementiert. Diese Datenkommunikationsnetze können irgendein Kommunikationsprotokoll und irgendeinen Typ von Kommunikationsmedium verwenden. In einigen Ausführungsformen sind die Datenkommunikationsnetze eine Kombination aus zwei oder mehr miteinander gekoppelten Datenkommunikationsnetzen (oder Teilnetzen). In alternativen Ausführungsformen sind diese Kommunikationsverbindungen unter Verwendung irgendeines Typs eines Kommunikationsmediums und irgendeines Kommunikationsprotokolls implementiert.

[0052] Wie in **Fig. 1** gezeigt ist, sind die Datenspeichervorrichtungen **116**, **118** und **120** von den der Ausführungsplattform **112** zugeordneten Rechenbetriebsmitteln entkoppelt. Diese Architektur unterstützt dynamische Änderungen der Datenverarbeitungsplattform **100** basierend sowohl auf den sich ändernden Erfordernissen der Datenspeicherung/-wiedergewinnung als auch auf den sich ändernden Erfordernisse der Anwender und

Systeme, die auf die Datenverarbeitungsplattform **100** zugreifen. Die Unterstützung dynamischer Änderungen ermöglicht, dass die Datenverarbeitungsplattform **100** in Reaktion auf die sich ändernden Anforderungen an die Systeme und Komponenten innerhalb der Datenverarbeitungsplattform **100** schnell skaliert wird. Das Entkoppeln der Rechenbetriebsmittel von den Datenspeichervorrichtungen unterstützt die Speicherung großer Datenmengen, ohne dass eine entsprechende große Menge an Rechenbetriebsmittel benötigt wird. Ähnlich unterstützt dieses Entkoppeln der Betriebsmittel eine signifikante Zunahme der zu einem bestimmten Zeitpunkt benutzten Rechenbetriebsmittel, ohne eine entsprechende Zunahme der verfügbaren Datenspeicherbetriebsmittel zu erfordern.

[0053] Der Betriebsmittelmanager **102**, die Metadaten **110**, die Ausführungsplattform **112** und die Speicherplattform **114** sind in **Fig. 1** als einzelne Komponenten gezeigt. Der Betriebsmittelmanager **102**, die Metadaten **110**, die Ausführungsplattform **112** und die Speicherplattform **114** können jedoch jeweils als ein verteiltes System (z. B. verteilt über mehrere Systeme/Plattformen an mehreren geographischen Orten) implementiert sein. Zusätzlich können der Betriebsmittelmanager **102**, die Metadaten **110**, die Ausführungsplattform **112** und die Speicherplattform **114** jeweils (unabhängig voneinander) abhängig von den Änderungen der von den Anwendern **104**, **106**, **108** empfangenen Anforderungen und den sich ändernden Erfordernissen der Datenverarbeitungsplattform **100** aufwärts oder abwärts skaliert werden. Folglich ist die Datenverarbeitungsplattform **100** dynamisch und unterstützt regelmäßige Änderungen, um den aktuellen Datenverarbeitungserfordernissen zu entsprechen.

[0054] **Fig. 2** ist ein Blockschaltplan, der eine Ausführungsform des Betriebsmittelmanagers **102** darstellt. Wie in **Fig. 1** gezeigt ist, enthält der Betriebsmittelmanager **102** einen Zugriffsmanager **202** und einen Schlüsselmanager **204**, die an eine Datenspeichervorrichtung **206** gekoppelt sind. Der Zugriffsmanager **202** kann Authentifizierungs- und Autorisierungsaufgaben für die hier beschriebenen Systeme abwickeln. Der Schlüsselmanager **204** kann die Speicherung und Authentifizierung von Schlüsseln managen, die während der Authentifizierungs- und Autorisierungsaufgaben verwendet werden. Ein Anforderungsverarbeitungsdienst **208** managt die empfangenen Datenspeicheranforderungen und Datenwiedergewinnungsanforderungen. Ein Managementkonsolendienst **210** unterstützt den Zugriff auf verschiedene Systeme und Prozesse durch Administratoren und andere Systemmanager.

[0055] Der Betriebsmittelmanager **102** kann außerdem einen SQL-Kompilierer **212**, einen SQL-Optimierer **214** und einen SQL-Ausführer **210** enthalten. Der SQL-Compiler **212** parst die SQL-Abfragen und erzeugt den Ausführungscode für die Abfragen. Der SQL-Optimierer **214** bestimmt basierend auf den Daten, die verarbeitet werden müssen, das beste Verfahren, um die Abfragen auszuführen. Der SQL-Ausführer **216** führt den Abfragecode für die durch den Betriebsmittelmanager **102** empfangenen Abfragen aus. Ein Abfrageplaner und -koordinator **218** kann die empfangenen Abfragen an die geeigneten Dienste oder Systeme zur Kompilierung, Optimierung und Abfertigung an die Ausführungsplattform **112** senden. Ein Manager **220** der virtuellen Lager managt den Betrieb mehrerer virtueller Lager, die in einer Ausführungsplattform implementiert sind.

[0056] Zusätzlich enthält der Betriebsmittelmanager **102** einen Konfigurations- und Metadatenmanager **222**, der die auf die in den entfernten Datenspeichervorrichtungen und in den lokalen Caches gespeicherten Daten bezogenen Informationen managt. Eine Überwachungseinrichtung und ein Arbeitsbelastungsanalysator **224** überwacht die durch den Betriebsmittelmanager **102** ausgeführten Prozesse und managt die Verteilung der Aufgaben (z. B. der Arbeitsbelastung) über die virtuellen Lager und Ausführungsknoten in der Ausführungsplattform. Der Konfigurations- und Metadaten-Manager **222** und die Überwachungseinrichtung und der Arbeitsbelastungsanalysator **224** sind an eine Datenspeichervorrichtung **226** gekoppelt.

[0057] Der Betriebsmittel-Manager **102** enthält außerdem einen Replikations- und Ausfallsicherungsmanager **228**, der die Datenreplikationsanforderungen, eine Datenbank-Ausfallsicherung und ein Datenbank-Failback managt. Der Replikations- und Ausfallsicherungsmanager **228** managt und plant z. B. die Stapeldatenreplikation zwischen mehreren Datenbank-Speicherbetriebsmitteln und Datenbankbereitstellungen. In einer Ausführungsform kann der Replikations- und Ausfallsicherungsmanager **228** die Replikation der Daten managen, die innerhalb einer primären Bereitstellung gespeichert sind, so dass sie eine Replikation innerhalb einer oder mehrerer sekundärer oder Sicherungsbereitstellungen sind. Ferner kann der Replikations- und Ausfallsicherungsmanager **228** das Verlagern von Datenbankoperationen von einer primären Bereitstellung zu einer sekundären Bereitstellung managen, wenn die primäre Bereitstellung ausfällt, und/oder das Verlagern von Datenbankoperationen von der sekundären Bereitstellung zurück zur primären Bereitstellung managen, wenn die primäre Bereitstellung wieder verfügbar wird. Der Replikations- und Ausfallsicherungsmanager **228** kann eine konsistente Datenreplikation zwischen den mehreren Bereitstellungen sicherstellen und kann ferner sicherstellen, dass irgendwelche Aktualisierungen, die an einer ersten Bereitstellung vorgenommen werden, während

eine zweite Bereitstellung nicht verfügbar ist, zu der zweiten Bereitstellung übertragen werden, wenn die zweite Bereitstellung wieder verfügbar wird.

[0058] **Fig. 3** ist ein Blockschaltplan, der eine Ausführungsform einer Ausführungsplattform darstellt. Wie in **Fig. 3** gezeigt ist, enthält die Ausführungsplattform **112** mehrere virtuelle Lager **302**, **304** und **306**. Jedes virtuelle Lager enthält mehrere Ausführungsknoten, die jeder einen Cache und einen Prozessor enthalten. Obwohl jedes virtuelle Lager **302**, **304**, **306**, wie in **Fig. 3** gezeigt ist, drei Ausführungsknoten enthält, kann ein spezielles virtuelles Lager irgendeine Anzahl von Ausführungsknoten enthalten, ohne vom Schutzumfang der Offenbarung abzuweichen. Ferner ist die Anzahl der Ausführungsknoten in einem virtuellen Lager dynamisch, so dass neue Ausführungsknoten erzeugt werden, wenn zusätzlicher Bedarf besteht, und vorhandene Ausführungsknoten gelöscht werden, wenn sie nicht länger notwendig sind.

[0059] Jedes virtuelle Lager **302**, **304**, **306** kann auf irgendeine der in **Fig. 1** gezeigten Datenspeichervorrichtungen **116**, **118**, **120** zugreifen. Folglich sind die virtuellen Lager **302**, **304**, **306** nicht notwendigerweise einer spezifischen Datenspeichervorrichtung **116**, **118**, **120** zugeordnet und können stattdessen auf die Daten von irgendeiner der Datenspeichervorrichtungen **116**, **118**, **120** zugreifen. Ähnlich kann jeder der in **Fig. 3** gezeigten Ausführungsknoten auf Daten von irgendeiner der Datenspeichervorrichtungen **116**, **118**, **120** zugreifen. In einigen Ausführungsformen kann ein spezielles virtuelles Lager oder ein spezieller Ausführungsknoten vorübergehend einer bestimmten Datenspeichervorrichtung zugewiesen werden, wobei aber das virtuelle Lager oder der Ausführungsknoten später auf Daten von irgendeiner anderen Datenspeichervorrichtung zugreifen kann.

[0060] Im Beispiel nach **Fig. 3** enthält das virtuelle Lager **302** drei Ausführungsknoten **308**, **310** und **312**. Der Ausführungsknoten **308** enthält einen Cache **314** und einen Prozessor **316**. Der Ausführungsknoten **310** enthält einen Cache **318** und einen Prozessor **320**. Der Ausführungsknoten **312** enthält einen Cache **322** und einen Prozessor **324**. Jeder Ausführungsknoten **308**, **310**, **312** ist der Verarbeitung einer oder mehrerer Datenspeicher- und/oder Datenwiedergewinnungsaufgaben zugeordnet. Ein spezielles virtuelles Lager kann z. B. Datenspeicher- und Datenwiedergewinnungsaufgaben abwickeln, die einem speziellen Anwender oder Kunden zugeordnet sind. In anderen Implementierungen kann ein spezielles virtuelles Lager Datenspeicher- und Datenwiedergewinnungsaufgaben abwickeln, die einem bestimmten Datenspeichersystem oder einer bestimmten Datenkategorie zugeordnet sind.

[0061] Ähnlich zu dem oben erörterten virtuellen Lager **302** enthält das virtuelle Lager **304** drei Ausführungsknoten **326**, **328** und **330**. Der Ausführungsknoten **326** enthält einen Cache **332** und einen Prozessor **334**. Der Ausführungsknoten **328** enthält einen Cache **336** und einen Prozessor **338**. Der Ausführungsknoten **330** enthält einen Cache **340** und einen Prozessor **342**. Zusätzlich enthält das virtuelle Lager **306** drei Ausführungsknoten **344**, **346** und **348**. Der Ausführungsknoten **344** enthält einen Cache **350** und einen Prozessor **352**. Der Ausführungsknoten **346** enthält einen Cache **354** und einen Prozessor **356**. Der Ausführungsknoten **348** enthält einen Cache **358** und einen Prozessor **360**.

[0062] Obwohl die in **Fig. 3** gezeigten Ausführungsknoten jeder einen Cache und einen Prozessor enthalten, können alternative Ausführungsformen Ausführungsknoten enthalten, die irgendeine Anzahl von Prozessoren und irgendeine Anzahl von Caches enthalten. Zusätzlich können die Caches zwischen den verschiedenen Ausführungsknoten in der Größe variieren. Die in **Fig. 3** gezeigten Caches speichern im lokalen Ausführungsknoten Daten, die von einer oder mehreren Datenspeichervorrichtungen in einer Speicherplattform **114** wiedergewonnen wurden (siehe **Fig. 1**). Folglich verringern oder eliminieren die Caches potentielle Engpassprobleme, die in Plattformen auftreten, die Daten konsistent aus entfernten Speichersystemen wiedergewinnen. Anstatt wiederholt auf Daten von den entfernten Speichervorrichtungen zuzugreifen, greifen die hier beschriebenen Systeme und Verfahren auf die Daten aus den Caches in den Ausführungsknoten zu, was signifikant schneller ist und das Engpassproblem vermeidet. In einigen Ausführungsformen sind die Caches unter Verwendung von Hochgeschwindigkeitsspeichervorrichtungen implementiert, die einen schnellen Zugriff auf die gecacheten Daten bereitstellen. Jeder Cache kann die Daten von irgendeiner der Speichervorrichtungen in der Speicherplattform **114** speichern.

[0063] Ferner können die Cache-Betriebsmittel und Rechenbetriebsmittel zwischen verschiedenen Ausführungsknoten variieren. Ein Ausführungsknoten kann z. B. signifikante Rechenbetriebsmittel und minimale Cache-Betriebsmittel enthalten, was den Ausführungsknoten für Aufgaben nützlich macht, die signifikante Rechenbetriebsmittel erfordern. Ein weiterer Ausführungsknoten kann signifikante Cache-Betriebsmittel und minimale Rechenbetriebsmittel enthalten, dies macht diesen Ausführungsknoten für Aufgaben nützlich, die das Caching großer Datenmengen erfordern. In einigen Ausführungsformen werden die einem speziellen Ausführungsknoten zugeordneten Cache-Betriebsmittel und Rechenbetriebsmittel basierend auf der erwarteten Auf-

gaben, die durch den Ausführungsknoten auszuführen sind, bestimmt, wenn der Ausführungsknoten erzeugt wird.

[0064] Zusätzlich können sich die einem bestimmten Ausführungsknoten zugeordneten Cache-Betriebsmittel und Rechenbetriebsmittel im Lauf der Zeit basierend auf den durch den Ausführungsknoten ausgeführten Aufgaben ändern. Einem speziellen 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.

[0065] Obwohl die virtuellen Lager **302**, **304**, **306** derselben Ausführungsplattform **112** nach **Fig. 1** zugeordnet sind, können die virtuellen Lager unter Verwendung mehrerer Rechensysteme an mehreren geographischen Orten implementiert sein. Das virtuelle Lager **302** z. B. durch ein Rechensystem an einem ersten geographischen Ort implementiert sein, während die virtuellen Lager **304** und **306** durch ein weiteres Rechensystem an einem zweiten geographischen Ort implementiert sind. In einigen Ausführungsformen sind diese verschiedenen Rechensystemen cloud-basierte Rechensysteme, die durch eine oder mehrere verschiedenen Entitäten aufrechterhalten werden.

[0066] Zusätzlich ist jedes virtuelle Lager in **Fig. 3** als mehrere Ausführungsknoten aufweisend gezeigt. Die mehreren Ausführungsknoten, die jedem virtuellen Lager zugeordnet sind, können unter Verwendung mehrerer Rechensysteme an mehreren geographischen Orten implementiert sein. Eine spezielle Instanz des virtuellen Lagers **302** implementiert z. B. die Ausführungsknoten **308** und **310** in einer Rechenplattform an einem speziellen geographischen Ort und implementiert den Ausführungsknoten **312** in einer anderen Rechenplattform an einem weiteren geographischen Ort. Das Auswählen spezieller Rechensysteme zum Implementieren eines Ausführungsknotens kann von verschiedenen Faktoren, z. B. von dem Niveau der Betriebsmittel, die für einen speziellen Ausführungsknoten benötigt werden (z. B. die Verarbeitungsbetriebsmittelanforderungen und die Cache-Anforderungen), den Betriebsmitteln, die in speziellen Rechensystemen verfügbar sind, den Kommunikationsfähigkeiten der Netze innerhalb eines geographischen Orts oder zwischen geographischen Orten und davon, welche Rechensysteme bereits andere Ausführungsknoten im virtuellen Lager implementieren, abhängen. Die Ausführungsplattform **112** ist außerdem fehlertolerant. Falls z. B. ein virtuelles Lager ausfällt, wird dieses virtuelle Lager schnell durch ein anderes virtuelles Lager an einem anderen geographischen Ort ersetzt.

[0067] Eine spezielle Ausführungsplattform **112** kann irgendeine Anzahl virtueller Lager **302**, **304**, **306** enthalten. Zusätzlich ist die Anzahl der virtuellen Lager in einer speziellen Ausführungsplattform dynamisch, so dass neue virtuelle Lager erzeugt werden, wenn zusätzliche Verarbeitungs- und/oder Caching-Betriebsmittel benötigt werden. Ähnlich können bestehende virtuelle Lager gelöscht werden, wenn die mit dem virtuellen Lager zugeordneten Betriebsmittel nicht länger erforderlich sind.

[0068] **Fig. 4** ist ein Blockschaltplan, der eine Ausführungsform einer Betriebsumgebung **400** mit mehreren Anwendern, die durch eine Lastausgleichseinrichtung auf mehrere Datenbanken zugreifen, und mehreren virtuellen Lagern, die in einer Gruppe virtueller Lager enthalten sind, darstellt. Die Umgebung **400** enthält einen Betriebsmittelmanager **408** der virtuellen Lager und mehrere virtuelle Lager **410**, **412** und **414**, die in einer Gruppe **416** virtueller Lager angeordnet sind. Der Betriebsmittelmanager **408** der virtuellen Lager kann im Betriebsmittelmanager **102** enthalten sein. Insbesondere greifen mehrere Anwender **402**, **404** und **406** durch den Betriebsmittelmanager **408** der virtuellen Lager und die Gruppe **416** virtueller Lager auf mehrere Datenbanken **418**, **420**, **422**, **424**, **426** und **428** zu. In einigen Ausführungsformen greifen die Anwender **402-406** durch den Betriebsmittelmanager **102** (**Fig. 1**) auf den Betriebsmittelmanager **408** der virtuellen Lager zu. In einigen Ausführungsformen ist der Betriebsmittelmanager **408** der virtuellen Lager innerhalb des Betriebsmittelmanagers **102** implementiert.

[0069] Die Anwender **402-406** können Datenwiedergewinnungs- und Datenspeicheranforderungen bei dem Betriebsmittelmanager **408** der virtuellen Lager einreichen, der die Datenwiedergewinnungs- und Datenspeicheranforderungen an ein geeignetes virtuelles Lager **410-414** in der Gruppe **416** virtueller Lager weiterleitet. In einigen Implementierungen stellt der Betriebsmittelmanager **408** der virtuellen Lager eine dynamische Zuweisung der Anwender **402-406** zu den virtuellen Lagern **410-414** bereit. Wenn die Anwender **402-406** eine Datenwiedergewinnungs- oder Datenspeicheranforderung einreichen, können sie die Gruppe **416** virtueller Lager spezifizieren, um die Anforderung zu verarbeiten, ohne das spezielle virtuelle Lager **410-414** zu spezifizieren, das die Anforderung verarbeitet. Diese Anordnung ermöglicht es dem Betriebsmittelmanager **408** der virtuellen Lager, basierend auf dem Wirkungsgrad, den verfügbaren Betriebsmitteln und der Verfügbarkeit von

gecacheten Daten innerhalb der virtuellen Lager **401-414** mehrere Anforderungen über die virtuellen Lager **410-414** zu verteilen. Wenn der Betriebsmittelmanager **408** der virtuellen Lager bestimmt, wie die Datenverarbeitungsanforderungen zu leiten sind, berücksichtigt er die verfügbaren Betriebsmittel, die aktuellen Betriebsmittellasten, die Anzahl der aktuellen Anwender und dergleichen.

[0070] In einigen Ausführungsformen erzeugen Fehlertoleranzsysteme in Reaktion auf den Ausfall eines virtuellen Lagers neue virtuelle Lager. Das neue virtuelle Lager kann sich in derselben Gruppe virtueller Lager befinden oder kann in einer anderen Gruppe virtueller Lager an einem anderen geographischen Ort erzeugt werden.

[0071] Jedes virtuelle Lager **410-414** ist konfiguriert, mit einer Teilmenge aller Datenbanken **418-428** zu kommunizieren. In einer Umgebung **400** ist z. B. das virtuelle Lager **410** konfiguriert, mit den Datenbanken **418**, **420** und **422** zu kommunizieren. Ähnlich ist das virtuelle Lager **412** konfiguriert, mit den Datenbanken **420**, **424** und **426** zu kommunizieren. Das virtuelle Lager **414** ist konfiguriert, mit den Datenbanken **422**, **426** und **428** zu kommunizieren. In alternativen Ausführungsformen können die virtuellen Lager **410-414** mit irgendeiner (oder allen) der Datenbanken **418-428** kommunizieren.

[0072] Obwohl die Umgebung **400** eine Gruppe **416** virtueller Lager zeigt, können alternative Ausführungsformen irgendeine Anzahl von Gruppen virtueller Lager enthalten, die jeweils irgendeiner Anzahl von virtuellen Lagern zugeordnet sind. Für jede Kunden- oder Anwendergruppe können z. B. verschiedene virtuelle Lager erzeugt werden. Zusätzlich können verschiedene virtuelle Lager für verschiedene Entitäten oder irgendeine andere Gruppe, die auf verschiedene Datensätze zugreift, erzeugt werden. Mehrere virtuelle Lagergruppen können unterschiedliche Größen und Konfigurationen aufweisen. Die Anzahl der Gruppen virtueller Lager in einer bestimmten Umgebung ist dynamisch und kann sich basierend auf dem sich ändernden Bedarf der Anwender und anderer Systeme in der Umgebung ändern.

[0073] **Fig. 5** ist eine schematische graphische Darstellung, die einen Prozessablauf **500** zum Erzeugen einer Datenbank-Momentaufnahme veranschaulicht. Die Datenbank-Momentaufnahme ermöglicht das Instanziieren einer Kopie einer Quelldatenbank an einem anderen Ort, z. B. das Kopieren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, in eine sekundäre Bereitstellung. Die Momentaufnahme erfasst ein oder mehrere Objekte der Datenbank, z. B. die Struktur der Datenbank (z. B. Schemata, Tabellen, Sichten usw.) und/oder die Inhalte der Datenbank (d. h., die Zeilen). In bestimmten Ausführungsformen findet die konzeptionell sauberste Herangehensweise statt, wo die Momentaufnahme eine transaktional konsistente Sicht der Datenbank zu einem spezifischen Zeitpunkt widerspiegelt. In einer Ausführungsform ist eine transaktional konsistente Momentaufnahme zu einem Zeitpunkt keine strenge Anforderung, wobei es ausreichend ist, eine Momentaufnahme zu erzeugen, die durch die Anwendung einer Menge von Transaktionsprotokolldatensätzen in einen transaktional konsistenten Zustand gebracht werden kann.

[0074] Der Prozessablauf **500** veranschaulicht eine Zeitachse, die eine Momentaufnahme darstellt, die zum Zeitpunkt t_1 eingeleitet wird und zum Zeitpunkt t_6 abgeschlossen wird. Bei **502** beginnt der Prozessablauf **500** und wird eine Momentaufnahme eingeleitet. Eine Momentaufnahme des Objekts X wird bei **504** zum Zeitpunkt t_2 erzeugt, während die Momentaufnahme des Objekts Y bei **510** zum Zeitpunkt ts erzeugt wird. Es sollte erkannt werden, dass das Objekt X und das Objekt Y irgendwelche zwei Objekte in der Datenbank repräsentieren können. Wie veranschaulicht ist, wird das Objekt X bei **506** zum Zeitpunkt t_3 modifiziert, während das Objekt Y bei **508** zum Zeitpunkt t_4 modifiziert wird. Das Objekt X wird bei **506** modifiziert, nachdem die Momentaufnahme des Objekts X bei **504** erzeugt worden ist. Das Objekt Y wird bei **508** modifiziert, bevor bei **510** die Momentaufnahme des Objekts Y erzeugt wird. Die Momentaufnahme endet bei **512**.

[0075] Abhängig von der Semantik dessen, wie einzelne Objekt-Momentaufnahmen erzeugt werden, kann der in **Fig. 5** veranschaulichte Prozessablauf **500** eine transaktional konsistente Zeitpunkt-Darstellung der Objekte X und Y erzeugen oder nicht. Falls die Momentaufnahmedarstellung eines Objekts basierend auf dem Zustand des Objekts zum Zeitpunkt, zu dem die Momentaufnahme eingeleitet wurde, erzeugt wird, ist die Momentaufnahme selbst eine transaktional konsistente Darstellung der Datenbank an dem Punkt, an dem die Momentaufnahme begann. Im Kontext nach **Fig. 5** würde z. B. die Semantik z. B. der Erzeugung von Momentaufnahmedarstellungen sowohl von X als auch von Y basierend auf dem Zustand zum Zeitpunkt t_1 entsprechen und zu einer Momentaufnahme führen, die eine transaktional konsistente Sicht sowohl von X als auch von Y zum Zeitpunkt t_1 bereitstellt. Falls jedoch die Momentaufnahmedarstellung eines Objekts basierend auf dem Objekt zu dem Zeitpunkt, zu dem die Momentaufnahmedarstellung des Objekts erzeugt wird, erzeugt wird, ist die Momentaufnahme nicht notwendigerweise zu jedem Zeitpunkt eine transaktional konsistente Darstellung der Datenbank. Im Kontext nach **Fig. 5** würde die Semantik z. B. dem Erzeugen einer Momentaufnahmedarstel-

lung von X basierend auf seinem Zustand zum Zeitpunkt t_2 und einer Momentaufnahmedarstellung von Y basierend auf seinem Zustand zum Zeitpunkt t_5 entsprechen. Diese Kombination würde eine Momentaufnahme erzeugen, die einem Datenbankzustand entspricht, der nie vorhanden war und gegebenenfalls abhängig von der Beziehung zwischen den beiden Modifikationen bei **506** und **508** potentiell ungültig ist.

[0076] Ein potentiell anomaler Zustand kann z. B. auftreten, wenn eine Modifikation zum Zeitpunkt t_3 eine Spalte zur Tabelle X hinzufügt und die Modifikation zum Zeitpunkt t_4 die Tabelle Y basierend auf eines CTAS erzeugt, das die neue Spalte in Tabelle X einbezieht. Das CTAS erzeugt im Wesentlichen durch das Ausführen eine Auswahlabfrage gegen die Datenbank ein neues Tabellenobjekt. Diese Auswahlabfrage könnte mehrere Tabellen in der Datenbank einbeziehen. In einer beispielhaften Implementierung könnte es eine Abhängigkeit zwischen den Daten und der Struktur der Objekte X und Y geben. In einer derartigen Implementierung kann es ein Szenario geben, in dem das Objekt X keine Spalte aufweist, selbst wenn das Objekt Y basierend sowohl auf den Daten als auch auf der Struktur des Objekts X erzeugt wurde. Dieses Szenario kann die Möglichkeit erzeugen, dass strukturelle Änderungen und Inhaltsänderungen in subtilen Weisen wechselwirken. Es können andere Szenarien vorhanden sein, die zu garantierten Inkonsistenzen führen. Wenn z. B. die Modifikationen an X und Y Teil einer einzigen Transaktion sind, dann würde das Erzeugen einer Momentaufnahme basierend auf dem aktuellen Zustand zu einer zerrissenen Transaktion führen, bei der ein Teil der Transaktion in der Momentaufnahme widergespiegelt wird und ein weiterer Teil der Transaktion in der Momentaufnahme nicht widergespiegelt wird.

[0077] In einer Ausführungsform ist es ungeachtet dessen, wie die Momentaufnahme erzeugt wird, durch das Beginnen mit der Momentaufnahme und dann das Anwenden irgendwelcher während des Zeitrahmens der Momentaufnahme erzeugten Protokolldatensätze in der serialisierten Reihenfolge der Protokolldatensätze möglich, das Ziel am Ende der Momentaufnahmezeit in einen transaktional konsistenten Zustand zu bringen. In einer derartigen Ausführungsform nimmt die vorhergehende Aussage an, dass das Anwenden eines Protokolldatensatzes eine idempotente Operation ist, bei der die Zieldatenbank bereits die durch einen speziellen Protokolldatensatz vorgenommene Aktualisierung widerspiegelt und das Anwenden des Protokolldatensatzes eine Nulloperation ist. Im Kontext eines derartigen Beispiels führt das Anwenden der Protokolldatensätze, die mit den Modifikationen zum Zeitpunkt t_3 und zum Zeitpunkt t_4 verbunden sind, auf die erzeugten Momentaufnahme ungeachtet dessen, wie die einzelnen Objekt-Momentaufnahmen erzeugt wurden, zu einem Endzustand, der ab dem Zeitpunkt t_6 konsistent ist.

[0078] In einer Ausführungsform der Datenbankreplikation stellt die Momentaufnahme den Anfangszustand für die Zieldatenbank bereit, auf den alle nachfolgenden Änderungen angewendet werden. In einer Ausführungsform wird eine Momentaufnahme für Datenbankdaten erzeugt, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten in eine oder mehrere sekundäre Bereitstellungen kopiert werden können. In einer weiteren Ausführungsform wird eine Momentaufnahme für eine sekundäre Bereitstellung erzeugt, um irgendwelche Aktualisierungen zu erfassen, die an den in der sekundären Bereitstellung gespeicherten Datenbankdaten vorgenommen wurden, während eine primäre Bereitstellung oder eine oder mehrere andere sekundäre Bereitstellungen nicht verfügbar waren. Falls die Momentaufnahme mit der Quelldatenbank inkonsistent ist, ist außerdem die Zieldatenbank mit der Quelldatenbank inkonsistent. Das Anwenden weiterer Änderungen auf den inkonsistenten Anfangspunkt wird die Inkonsistenz im Allgemeinen nicht korrigiert. Falls z. B. der Ausfall eines Kundenkontos von einer Quelldatenbank (in einer Ausführungsform ist die Quelldatenbank die primäre Bereitstellung) zu einer sekundären Kopie-Bereitstellung, die von der Quelldatenbank (in diesem Fall der primären Bereitstellung) abgewichen ist, gesichert wird, ist der Nettoeffekt eine Datenverfälschung und/oder ein Datenverlust. Weil eine Ausfallsicherung jederzeit stattfinden kann, kann das Sicherstellen der Transaktionskonsistenz zwischen einer Quelldatenbank (z. B. der primären Bereitstellung) und einer Zieldatenbank (z. B. der sekundären Bereitstellung) für die Nutzendarstellung der Datenbankreplikation entscheidend sein. In einer Ausführungsform ist das Sicherstellen der Konsistenz der aus einer Momentaufnahme konstruierten Datenbank ein Baustein zum Herstellen und Aufrechterhalten der Konsistenz zwischen einer Quelldatenbank und einer Zieldatenbank zu jeder Zeit.

Erzeugen einer Datenbank-Momentaufnahme

[0079] In einer Ausführungsform enthalten die verschiedenen Teile der Informationen, die eine Datenbank umfassen, Metadaten-Dateien. Eine Implementierung der Metadaten-Dateien kann hier als Ausdruckseigenschafts- „EP“-Dateien bezeichnet werden. Die EP-Dateien können spezifisch kumulative Tabellen-Metadaten enthalten, die Informationen über alle Daten enthalten, die überall in einer Tabelle in der Datenbank gespeichert sind. Die EP-Dateien können ferner Gruppierungsausdruckseigenschaften enthalten, die Informationen über die Daten enthalten, die in einer Gruppierung von Mikropartitionen innerhalb der Tabelle gespeichert sind.

Die EP-Dateien können ferner Mikropartitions-Statistiken enthalten, die Informationen über die Daten enthalten, die in einer spezifischen Mikropartition der Tabelle gespeichert sind, wie z. B. Minimal-/Maximalwerte, ein Nullzählerstand, die Anzahl der Einträge usw. Die EP-Dateien können ferner Spaltenausdruckseigenschaften enthalten, die Informationen über die Daten enthalten, die in einer bestimmten Spalte einer Mikropartition der Tabelle gespeichert sind. Die hier offenbarten Metadaten-Dateien können spezifisch EP-Dateien enthalten oder können irgendeine andere Datei enthalten, die Informationen über Datenbankdaten enthält.

[0080] Die Metadaten-Dateien enthalten Informationen, die die Struktur der Datenbank beschreiben, und können die Liste und die Eigenschaften irgendwelcher Schemata in der Datenbank, die Liste und die Eigenschaften der Tabellen und Sichten in jedem Schema, die Liste und die Eigenschaften der in jeder Tabelle oder Sicht vorhandenen Spalten usw. enthalten. Die einzelnen Tabelleninhalte können durch eine Kombination aus EP-Dateien und irgendeiner anderen Form von Metadaten-Dateien definiert sein. Die einzelnen Tupelwerte der einzelnen Tabelleninhalte können in Mikropartitionen gespeichert sein. In einer Ausführungsform ist der genaue Satz von Mikropartitionen, der die Inhalte einer speziellen Tabelle zu einem speziellen Zeitpunkt in der Transaktionszeit enthält, in den Inhalten eines Satzes von Metadaten-Dateien enthalten. In einer Ausführungsform kann eine Metadaten-Datei so betrachtet werden, dass sie eine Liste von Mikropartitionen enthält. Sowohl die Mikropartitionen als auch die Metadaten-Dateien sind unveränderlich und können im Speicher gespeichert und verschlüsselt sein. In einer Ausführungsform wird die Liste der Metadaten-Dateien, die zu einem speziellen Punkt in der Transaktionszeit mit einer Tabelle verbunden sind, in einem Metadaten-Speicher aufrechterhalten, der von den Datenbankdaten getrennt ist.

[0081] In einer Ausführungsform ist der Anfangspunkt zum Erzeugen einer Momentaufnahme der Datenbank das DatabaseDPO („Datenbankdaten-Beständigkeitsobjekt“), das in den Metadaten gespeichert ist. Das DatabaseDPO ist eine Datenstruktur zum Wechselwirken mit beständigen Kataloginformationen, die in den Metadaten gespeichert sind. Das DatabaseDPO selbst ist effektiv die Wurzel eines Baumes, der alle Objekte innerhalb der Datenbank enthält, d. h., alle für die Momentaufnahme benötigten Objekte. Jedes Objekt im Baum, der an der gewünschten DatbaseDPO verwurzelt ist, kann in die Momentaufnahme serialisiert werden. Die serialisierte Darstellung eines Objekts kann alles einkapseln, was notwendig ist, um eine exakte Kopie des Objekts am entfernten Ort (der Zieldatenbank) wiederherzustellen.

[0082] Für eine Tabelle kann es ein zusätzliche Frage geben, wie die Tabelleninhalte zu serialisieren sind. In einer Ausführungsform kann das Lesen der gesamten Tabelle und das Serialisieren der Inhalte umfangreiche Rechenbetriebsmittel erfordern und zu sehr großen Momentaufnahmegrößen führen. In einer derartigen Ausführungsform kann es ausreichend sein, die Liste der Metadaten-Dateien für die Tabelle zu serialisieren. Wenn der Momentaufnahme als solche in der Zieldatenbank verbraucht wird, können die Metadaten-Dateien auf das Ziel kopiert werden, am Ziel gelesen werden, um die Liste der Mikropartitionen mit allen Tupeln abzuleiten, und können diese Mikropartitionen ebenfalls auf das Ziel kopiert werden. Sowohl die Metadaten-Dateien als auch die Mikropartitionen können in der Momentaufnahme verschlüsselt sein und können Informationen enthalten, die es dem Ziel erlauben, die geeigneten Schlüssel zu erhalten, um die Dateien zu entschlüsseln. Die Dateien im Ziel müssen vielleicht mit neuen Schlüsseln, die durch das Ziel gemanagt werden, neu verschlüsselt werden. In einer Ausführungsform enthält das Momentaufnahmebild einer Tabelle eine Metadaten-Dateiliste als einen Vertreter der Tabelleninhalte. In einer weiteren Ausführungsform enthält die Momentaufnahme einige Teile der Informationen, um es dem Ziel zu ermöglichen, einen oder mehrere Schlüssel zu erhalten, um eine Kopie der Metadaten-Dateien und der Mikropartitionen herzustellen.

Erzeugen von Transaktionsprotokolldatensätzen

[0083] In einer Ausführungsform stellt ein Transaktionsprotokolldatensatz sicher, dass die Protokolldatensätze selbst genügend Informationen enthalten, um die Transaktionsänderung am Ziel richtig und eindeutig zu reproduzieren. Dies kann zufriedenstellend sein, weil die durch das Transaktionsprotokoll vorgenommenen Änderungen zum Zeitpunkt der Übergabe bekannt sind und das Verfahren das Erfassen und Serialisieren der durch die Transaktion vorgenommenen Metadatenänderungen enthalten kann. In einer Ausführungsform ist der Transaktionsprotokolldatensatz ungeachtet der Bereitstellung, des Gebiets oder des zugrundeliegenden Cloud-Anbieters für alle Zieldatenbanken zugänglich. Der Transaktionsprotokolldatensatz kann in einen entfernten Speicher geschrieben werden.

[0084] In einer Ausführungsform wird eine primäre Bereitstellung nicht verfügbar, wobei alle Datenbankoperationen zu einer sekundären Bereitstellung verlagert werden. Während der Zeit, wenn die primäre Bereitstellung nicht verfügbar ist, können alle Aktualisierungen an den Datenbankdaten in der sekundären Bereitstellung ausgeführt werden. Für alle Aktualisierungen, die in der sekundären Bereitstellung ausgeführt werden, kann ein

Transaktionsprotokolldatensatz erzeugt werden, wobei der Transaktionsprotokolldatensatz verwendet werden kann, diese Aktualisierungen zur primären Bereitstellung zu übertragen, wenn die primäre Bereitstellung nicht länger nicht verfügbar ist. In einer derartigen Ausführungsform kann die Verwendung des Transaktionsprotokolldatensatzes sicherstellen, dass nur jene neuen Aktualisierungen (die in der sekundären Bereitstellung ausgeführt wurden) in der primären Bereitstellung ausgeführt werden und dass keine veralteten Daten oder zuvor aufgenommene Daten zu der primären Bereitstellung übertragen werden.

[0085] In einer Ausführungsform sind die hier offenbarten Systeme, Verfahren und Vorrichtungen hinsichtlich dessen, wann der Transaktionsprotokolldatensatz erzeugt wird, konfiguriert, um sicherzustellten, dass das Schreiben des Transaktionsprotokolldatensatzes effektiv Teil der Transaktion selbst ist. Der Transaktionsprotokolldatensatz kann nur in einen entfernten Speicher geschrieben werden, falls die Transaktion übergeben wird, wobei die Transaktion ferner nur übergeben wird, falls der Transaktionsprotokolldatensatz in den entfernten Speicher geschrieben wird. Eine Abweichung von einer derartigen Prozedur kann zu einer transaktionalen Inkonsistenz zwischen der Quelldatenbank und der Zieldatenbank führen.

[0086] Fig. 6 ist eine schematische graphische Darstellung, die einen Prozessablauf **600** zum Erzeugen von Transaktionsprotokollen zum Replizieren einer Datenbank veranschaulicht. Der Prozessablauf **600** veranschaulicht eine von links nach rechts verlaufende Zeitachse. In **Fig. 6** sind die Transaktionen, die im internen Transaktionszustand stattfinden, über der Zeitlinie veranschaulicht, während die Maßnahmen, um die Gleichzeitigkeitssteuerung und die Transaktionsverarbeitung zu unterstützen, unter der Zeitlinie veranschaulicht sind. Zum Zeitpunkt t_0 ist die Transaktion bei **602** offen und im aktiven Zustand, wobei zu diesem Zeitpunkt keine Maßnahmen der Datenmanipulationssprache (DML) ausgeführt worden sind. Zum Zeitpunkt t_1 ist die Verarbeitung einer DML-Anweisung im Gange. Während der Verarbeitung einer DML-Anweisung enthält der Prozessablauf **600** das Erhalten einer Dateisperre bei **604** an den betroffenen Tabellen, um mehrere gleichzeitige DML-Operationen zu unterstützen, die auf dieselbe Tabelle abzielen. Es sollte erkannt werden, dass das Erhalten einer Dateisperre bei **604** beliebig oft vorkommen kann und in einer Transaktion mit mehreren Anweisungen mehrmals vorkommt. Zum Zeitpunkt t_2 beginnt die Übergabeverarbeitung, wobei der Beginn der Übergabeverarbeitung durch das Übergehen zu dem Vorübergabezustand bei **606** aufgezeichnet wird. Zum Zeitpunkt t_3 werden die Tabellensperren an allen in der Transaktion modifizierten Tabellen bei **608** erhalten. Nachdem alle Tabellensperren bei **608** erfasst worden sind, wird ein Lamport-Takt auf Kontenebene verwendet, um eine neue und eindeutige Transaktionskennung bei **610** zum Zeitpunkt t_4 zu erzeugen. In einer Ausführungsform stellt das Erhalten der Transaktionskennung bei **610** nach dem Erfassen aller Tabellensperren bei **608** eine eindeutige Übergabereihenfolge (basierend auf dem Lamport-Taktwert) zwischen zwei potentiell widersprüchlichen Transaktionen sicher. Nach dem Erhalten einer Transaktionskennung bei **610** kann die Transaktion zum Zeitpunkt t_5 bei **612** in den Übergabezustand bei **612** übergehen. In einer Ausführungsform kann der Übergang in den Übergabezustand bei **612** einen „Punkt, an dem es kein Zurück mehr gibt“ **620** für die Transaktion darstellen. Vor diesem Übergang könnte die Transaktionsübergabe selbst aufgrund einer Anwenderhandlung, eines Verarbeitungsfehlers, eines Systemausfalls usw. noch storniert oder abgebrochen werden. Sobald der Übergang in den Übergabezustand **612** stattgefunden hat, wird jedoch die Transaktion aus der Systemperspektive effektiv übergeben. Zum Zeitpunkt t_5 werden die Wirkungen der Transaktion auf das System angewendet, wobei sie nicht länger rückgängig gemacht werden können. Es wird außerdem angegeben, dass zum Zeitpunkt t_5 (unterhalb der Zeitachse) die neue Tabellenversion jetzt bei **614** durch andere gleichzeitig laufende Transaktionen lesbar ist. Zum Zeitpunkt t_6 werden alle durch die Transaktion gehaltenen Sperren bei **616** freigegeben. Das Freigeben von Sperren ermöglicht es irgendwelchen potentiell widersprüchlichen Transaktionen, die auf diesen Sperren warten (bei **604** oder **608**), die Sperren zu erfassen, die erforderlich sind, um durch das Übergabeprotokoll weiterzugehen. Zum Zeitpunkt t_7 geht die Transaktion bei **618** in den übergebenen Zustand über, wobei sie die gesamte Verarbeitung abgeschlossen hat. In einer Ausführungsform führen alle Störungen nach dem Zeitpunkt t_5 nicht zu einem Rückgängigmachen der Transaktion. Falls z. B. der Knoten, der die Transaktion verarbeitet, unmittelbar nach dem Zeitpunkt t_5 ausfällt, nimmt ein neuer Knoten die Verarbeitung dort wieder auf, wo sie aufgehört hat, wobei er die Sperren freigibt und die Transaktion bis zum Abschluss prolongiert.

[0087] In einer Ausführungsform wird der Transaktionsprotokolldatensatz zum Zeitpunkt t_5 beim Übergang in den Übergabezustand bei **612** in den entfernten Speicher geschrieben. In bestimmten Ausführungsformen kann es problematisch sein, den Transaktionsprotokolldatensatz vor dem Zeitpunkt t_5 in den entfernten Speicher zu schreiben, weil es immer noch möglich sein kann, dass der Prozessablauf **600** vor dem Zeitpunkt t_5 abbricht. Ferner kann das Schreiben des Transaktionsprotokolldatensatzes als Teil der Nachübergabeverarbeitung nach dem Zeitpunkt t_5 die Probleme fehlgeleiteter Transaktionen vermeiden.

[0088] In einer Ausführungsform kann das Schreiben des Transaktionsprotokolldatensatzes als Teil der Nachübergabeverarbeitung nach dem Zeitpunkt ts, wie in **Fig. 6** veranschaulicht ist, das Problem der fehlgeleiteten Transaktionen vermeiden, wobei es aber die Möglichkeit einer Störung zwischen dem Zeitpunkt t_5 und dem Schreiben des Transaktionsprotokolls in einen entfernten Speicher eröffnen kann, was zu einer Transaktion führen kann, in der Quelldatenbank, aber nicht in der Zieldatenbank übergeben wird. Weil das Schreiben in den entfernten Speicher ein Teil der Nachtransaktionsverarbeitung sein kann, kann vernünftig angenommen werden, dass es zu einem Punkt stattfindet, nachdem die Quelle wieder betriebsbereit ist und die Transaktionsbereinigung bis zu Abschluss weitergeht. Nachdem dies stattgefunden hat, kann das Ziel die Änderung aufnehmen, wobei es nicht länger eine fehlende Transaktion gibt. Es kann sich jedoch ein Problemszenario ergeben, bei dem es eine Ausfallsicherung **720** gibt, wie in **Fig. 7** veranschaulicht wird. Falls es eine Ausfallsicherung **720** von der Quelle zum Ziel gibt, die in dem Fenster zwischen dem Überschreiten des Punkts **620**, an dem es kein Zurück mehr gibt, und dem Schreiben in den entfernten Speicher auftritt. In diesem Fall kann eine Transaktion an der Quelle übergeben worden sein, wobei sie an dem Ziel nicht vorhanden sein würde. Wenn in einer Ausführungsform das Schreiben in den entfernten Speicher zwischen dem Zeitpunkt t_5 und dem Zeitpunkt t_6 positioniert ist, wo alle Sperren bei **616** freigegeben sind, dann kann alles das, was verloren wird, das letzte Schreiben in eine oder mehrere Tabellen sein, wobei weiterhin niemals eine explizite Quittung der Transaktionsübergabe des verbunden Schreibens an einen Endanwender zurückgeschickt wird.

[0089] **Fig. 7** ist eine schematische graphische Darstellung, die einen Prozessablauf **700** für zwei verschiedene Transaktionen zum Replizieren einer Datenbank veranschaulicht. Der Prozessablauf **700** veranschaulicht einen Fall, der zu einer verlorenen Transaktion beim Vorhandensein einer zeitlich schlecht abgestimmten Ausfallsicherung **720** führen kann. **Fig. 7** veranschaulicht, wo die Wechselwirkung mit den Metadaten zum Zeitpunkt t_5 nicht nur die Transaktion in den Übergabezustand überführt, sondern außerdem die neue Version irgendwelcher modifizierter Tabellen dem Rest des Systems in einer atomaren Operation aufdeckt. Im Ergebnis kann die Folge von Ereignissen, wie sie im Prozessablauf **700** veranschaulicht ist, möglich sein.

[0090] Der Prozessablauf **700** in **Fig. 7** veranschaulicht eine hypothetische Zeitachse von zwei verschiedenen Transaktionen, der Transaktion T1 und der Transaktion T2. Die durch die Transaktion T1 ergriffenen Maßnahmen erscheinen oberhalb der Zeitachse, während die durch die Transaktion T2 ergriffenen Maßnahmen unterhalb der Zeitachse erscheinen. Zum Zeitpunkt t_0 führt die Transaktion T1 ein Schreiben gegen die Tabelle T bei **702** aus. Zum Zeitpunkt t_1 erhält die Transaktion T1 als Teil der Vorübergabeverarbeitung eine Sperre für die Tabelle T bei 704. Zum Zeitpunkt t_2 überquert die Transaktion T1 den „Punkt, an dem es kein Zurück mehr gibt," indem sie atomar in den Übergabezustand übergeht und die neue Version von Tabelle T mit dem durch die Transaktion T1 ausgeführten Schreiben durch das Übergeben der neue lesbare Version bei **706** dem Rest des Systems aufdeckt. In dem in **Fig. 7** veranschaulichen Prozessablauf **700** können verschiedene Gründe verursachen, dass das Schreiben des Transaktionsprotokolldatensatzes in den entfernten Speicher verzögert ist. Die Transaktion T1 beleibt als solche in der Vorübergabeverarbeitung, wobei sie den Transaktionsprotokolldatensatz noch nicht in den entfernten Speicher geschrieben hat, die Sperre an der Tabelle T noch nicht freigegeben hat und den Erfolg noch nicht an einen Endanwender zurückgemeldet hat. Zum Zeitpunkt t_3 liest die Transaktion T2 die neueste Version der Tabelle T (einschließlich des durch die Transaktion T1 ausgeführten Schreibens) als Teil einer CTAS-Operation, um die Tabelle X bei **708** zu erzeugen/zu befüllen. Zum Zeitpunkt t_4 überschreitet diese neue Transaktion den „Punkt, an dem es kein Zurück mehr gibt," wobei sie bei **710** übergeben wird. In bestimmten Implementierungen ist ein derartiger Übergang möglich, weil die Transaktion T2 keine Sperre an der Tabelle T benötigt, weil sie nur aus Tabelle T lesen kann, wobei sie deshalb nicht durch Transaktion T1 blockiert wird, die noch eine Sperre an der Tabelle T hält. Zum Zeitpunkt t_5 schreibt diese neue Transaktion ihren Protokolldatensatz bei **712** in den entfernten Speicher. Der Protokolldatensatz für Transaktion T1 ist immer noch nicht in den entfernten Speicher geschrieben worden. Der Prozessablauf **700** enthält ferner eine Ausfallsicherung **720** für eine Kopie. In einer Ausführungsform, wie in **Fig. 7** veranschaulicht ist, ist die Kopie nicht transaktional konsistent mit irgendeiner Version der Quelldatenbank. Die Kopie enthält die Ergebnisse der Transaktion T2, enthält aber nicht die Ergebnisse der Transaktion T1, ungeachtet dessen, dass die Transaktion T2 die Version der Tabelle T gelesen hat, die durch die Transaktion T1 erzeugt wurde. Die Transaktion T1 ist effektiv verloren worden, wobei dennoch die Transaktion T2, die von Transaktion T1 abhängt, nicht verloren wurde. In der in **Fig. 7** veranschaulichten Ausführungsform ist die Datenbank folglich inkonsistent. Schließlich wird der Transaktionsprotokolldatensatz für die Transaktion T1 bei **714** in den entfernten Speicher geschrieben, wobei aber, wie in **Fig. 7** veranschaulicht ist, zu irgendeinem Zeitpunkt einschließlich vor dem erfolgreichen Schreiben in den entfernten Speicher bei **714** eine Ausfallsicherung **720** stattfinden kann, die zu einer potentiell verlorenen Transaktion führt.

[0091] Die potentielle Aufdeckung eines Problem eines des verlorenen Schreibens oder einer verlorenen Transaktion, wie in **Fig. 7** veranschaulicht ist, kann von der neuen Version einer Tabelle stammen, die für den

Rest des Systems verfügbar ist, bevor der Transaktionsprotokolldatensatz mit der neuen Version verbunden wird, die in die Datenbank geschrieben wird. Die in **Fig. 7** veranschaulichte Ausführungsform einer verlorenen Transaktion kann vermieden werden, indem eine neue Version der Tabelle für den Rest des Systems erst sichtbar gemacht wird, bis das Schreiben in den entfernten Speicher stattgefunden hat. Hier ist eine Herangehensweise offenbart, um das Aufdecken der neuen Tabellenversion zu verschieben, bis das Schreiben in den entfernten Speicher stattgefunden hat, ohne die bestehende Nachübergabelogik zu unterbrechen. Die Herangehensweise enthält das Aufnehmen einer Vorbereitungsphase für die Transaktion, wie in **Fig. 8** veranschaulicht ist.

[0092] **Fig. 8** ist eine schematische graphische Darstellung, die einen Prozessablauf **800** veranschaulicht, der eine Vorbereitungsphase für die Replikation einer Datenbank enthält. Die Vorbereitungsphase wird eingeführt, um das Aufdecken der neuen Tabellenversion zu verschieben, bis das Schreiben in einen entfernten Speicher stattgefunden hat, ohne die bestehende Nachübergabelogik zu unterbrechen. Die Vorbereitungsphase kann nach der Erfassung von Tabellensperren und vor dem Übergang in den Übergabezustand stattfinden. In einer Ausführungsform ist es der Zweck der Vorbereitungsphase, das System selbst im Fall einer Störung vor der tatsächlichen Übergabe in einen Zustand zu überführen, in dem die Transaktion während der Bereinigung oder Wiederherstellung übergeben werden kann. Die Vorbereitungsphase schreibt den Transaktionsprotokolldatensatz in einen entfernten Speicher. Nur nach einem bestätigten erfolgreichen Schreiben in den entfernten Speicher würde die Transaktion von da an weiter wie in den in den **Fig. 6-7** veranschaulichten Protokollen mit derselben Semantik (z. B. die neue Version irgendwelcher modifizierter Objekte durch den Rest des Systems lesbar machen) in die Übergabephase übergehen.

[0093] Der Prozessablauf **800** enthält einen offiziellen Übergabepunkt, an dem der Transaktionsprotokolldatensatz bei **814** in die Datenbank geschrieben wird, wobei dies als „harter Punkt, an dem es kein Zurück mehr gibt," **826** bezeichnet werden kann. Der Prozessablauf **800** enthält ferner einen vorbereiteten Zustand bei **812** zum Zeitpunkt t_5. Der Zustandsübergang in den vorbereiteten Zustand bei **812** kann als ein „weicher Punkt, an dem es kein Zurück mehr gibt," **824** bezeichnet werden. Zum Zeitpunkt t_0 ist die Transaktion offen und im aktiven Zustand bei **802**, wobei an diesem Punkt sind keine DML-Aktionen ausgeführt worden sind. Zum Zeitpunkt t_1 ist die Verarbeitung einer DML-Anweisung im Gange. Während der Verarbeitung einer DML-Anweisung enthält der Prozessablauf **800** das Erhalten einer Dateisperre bei 804 an den betroffenen Tabellen, um mehrere gleichzeitige DML-Operationen zu unterstützen, die auf dieselbe Tabelle abzielen. Zum Zeitpunkt t_2 beginnt die Übergabeverarbeitung, wobei der Beginn der Übergabeverarbeitung wird durch das Übergehen in den Vorübergabezustand bei **806** aufgezeichnet wird. Zum Zeitpunkt t_3 werden die Tabellensperren an allen in der Transaktion modifizierten Tabellen bei **808** erhalten. Nachdem alle Tabellensperren bei **808** erfasst worden sind, wird ein Lamport-Takt auf Kontenebene verwendet, um eine neue und eindeutige Transaktionskennung bei **810** zum Zeitpunkt t_4 zu erzeugen. In einer Ausführungsform stellt das Erhalten der Transaktionskennung bei **810** nach dem Erfassen aller Tabellensperren bei **808** eine eindeutige Übergabereihenfolge (basierend auf dem Lamport-Taktwert) zwischen irgendwelchen zwei potentiell widersprüchlichen Transaktionen sicher. Zum Zeitpunkt t_5 tritt die Transaktion bei **812** in den vorbereiteten Zustand ein, was als ein weicher Punkt, an dem es kein Zurück mehr gibt, **824** betrachtet werden kann. Die Transaktion schreibt dann den Transaktionsprotokolldatensatz bei **814** in eine Datenbank, was als harter Punkt, an dem es kein Zurück mehr gibt, **826** betrachtet werden kann. Zum Zeitpunkt t_7 wird bei **816** in den Übergabezustand eingetreten, wobei eine neue Version bei **818** lesbar ist. Zum Zeitpunkt t_8 werden alle Sperren bei **820** freigegeben. Zum Zeitpunkt t_9 tritt die Transaktion bei **822** in den Übergabezustand ein.

[0094] Der Übergang in den vorbereiteten Zustand bei **812** kann aufgrund des verzögerten Schreibens des Transaktionsprotokolldatensatzes in die Datenbank, wie bezüglich **Fig. 7** erörtert worden ist, als ein weicher Punkt, an dem es kein Zurück mehr gibt, **824** bezeichnet werden. Kurz gesagt, es ist möglich, dass ein Schreiben in die Datenbank eine Zeitüberschreitung verursacht oder abbricht, dann aber den Eintrag enthält, der geschrieben wurde, um später zu erscheinen. Das ursprüngliche Schreiben kann die durch das erneute Schreiben erzeugte Version des Eintrags effektiv überschreiben. Es kann eine nichttriviale Verzögerung zwischen dem Zeitpunkt, zu dem eine Schreibanforderung an die Datenbank ergebnislos endet, und dem Zeitpunkt, zu dem dieses Schreiben tatsächlich in der Datenbank erscheint, geben. Folglich gibt es ein Zeitfenster nach dem ergebnislosen Abbruch einer Schreibanforderung, aber vor dem Punkt, an dem das Schreiben in der Datenbank erscheint, während dessen eine Prüfung in der Datenbank, um zu bestimmen, ob ein Eintrag vorhanden ist oder nicht, ein „Falsch Negativ" zurückgeben kann - d. h., die Prüfung sieht den Eintrag nicht, daher annimmt sie an, dass das Schreiben nicht stattgefunden hat und nicht stattfinden wird, wobei aber in Wirklichkeit das Schreiben stattgefunden hat oder stattfinden wird und der Eintrag noch nicht sichtbar ist. Falls die Prüfung angibt, dass der Eintrag nicht vorhanden ist, ist es nicht sicher, anzunehmen, dass der Datensatz nicht geschrieben wurde, und die Transaktion rückgängig zu machen - wenn der Eintrag nach der Prüfung erscheinen würde, dann wür-

de die Transaktion nicht an der Quelle angewendet worden sein, sondern würde potentiell an dem Ziel (den Zielen) angewendet werden, wobei dies eine Inkonsistenz erzeugen kann. Im Fall einer Störung, während sich die Transaktion im vorbereiteten Zustand befindet, kann eine sichere Vorgehensweise sein, die Transaktion durch das Sicherstellen, dass der Transaktionsprotokolldatensatz neu in die Datenbank geschrieben wird, das Übergehen in den Übergabezustand und das Ausführen der Nachübergabemaßnahmen, um die Transaktion abzuschließen, bis zum Abschluss zu prolongieren. Folglich kann der Übergang in den vorbereiteten Zustand bei **812** als ein weicher Rückkehrpunkt **824** charakterisiert werden, weil die Transaktion nicht tatsächlich übergeben wurde, aber der einzige verfügbare Endzustand für die Transaktion eine erfolgreiche Übergabe ist.

[0095] In einer Ausführungsform, in der eine Ausfallsicherung stattfindet, kann die Semantik des Prozessablaufs **800** modifiziert sein. Falls eine Ausfallsicherung stattfindet, wenn sich eine Transaktion im aktiven Zustand bei **802** oder im Zustand vor der Übergabe bei **806** befindet, erscheinen die Ergebnisse der Transaktion nicht im Ziel, weil der Transaktionsprotokolldatensatz noch nicht bei **814** in die Datenbank geschrieben worden ist und daher nicht durch das Ziel aufgenommen werden kann. Falls eine Ausfallsicherung stattfindet, wenn sich eine Transaktion im Übergabezustand bei **816** oder im übergebenen Zustand bei **822** befindet, erscheinen die Ergebnisse der Transaktion im Ziel, weil der Transaktionsprotokolldatensatz zum Zeitpunkt der Ausfallsicherung bereits bei **814** in die Datenbank geschrieben worden, so dass er durch das Ziel aufgenommen werden kann.

[0096] Falls eine Ausfallsicherung auftritt, wenn sich eine Transaktion bei **812** im vorbereiteten Zustand befindet, können die Ergebnisse der Transaktion abhängig davon, ob der Transaktionsprotokolldatensatz seit dem Zeitpunkt der Ausfallsicherung in die Datenbank geschrieben wurde, im Ziel erscheinen oder nicht. Falls das Ziel die Transaktion aufnimmt, weil der Transaktionsprotokolldatensatz zum Zeitpunkt der Ausfallsicherung in der Datenbank sichtbar war, kann er weiterhin im Ziel angewendet werden, weil die Quelle bis zum Abschluss prolongiert worden wäre und das zugehörige Kundenkonto noch nicht über den Erfolg oder Misserfolg informiert wurde. Falls das Ziel die Transaktion nicht aufnimmt, weil die Transaktionsprotokolldatensätze zum Zeitpunkt der Ausfallsicherung in der Datenbank nicht sichtbar waren, kann weiterhin eine Inkonsistenz vermieden werden, weil die Quelle die Ergebnisse des Schreibens nicht aufdeckt und den Erfolg nicht an das zugehörige Kundenkonto zurückgemeldet haben würde.

[0097] Gemäß dem in **Fig. 8** veranschaulichten Prozessablauf **800** können bestimmte Beziehungen gehalten werden, wie hier erörtert ist. Falls zwei Transaktionen ein widersprüchliches Schreiben aufweisen, kann die Transaktionskennung für das erste übergebene Schreiben kleiner als die Transaktionskennung für das zweite übergebene Schreiben sein. Diese Reihenfolge ist so, dass die Schreibvorgänge mit einer Tabellensperre synchronisiert sind und die Transaktionskennung bei **810** erhalten wird, nachdem die Sperre bei **808** erhalten wurde und bevor die Sperre bei **814** freigegeben wird. Falls zwei Transaktionen ein widersprüchliches Schreiben aufweisen, kann weiterhin der Transaktionsprotokolldatensatz für das erste übergebene Schreiben vor dem Transaktionsprotokolldatensatz für das zweite übergebene Schreiben erscheinen. Diese Reihenfolge kann garantiert werden, weil die Transaktionskennungsreihenfolge garantiert ist, weil das Schreiben des Transaktionsprotokolldatensatzes stattfindet, während die Transaktion die Tabellensperre hält. Falls weiterhin eine Transaktion (der Leser) eine durch eine weitere Transaktion (den Schreiber) erzeugte Tabellenversion liest, kann der Transaktionsprotokolldatensatz für den Schreiber erscheinen, bevor die neue Tabellenversion durch den Leser gelesen wurde. Diese Reihenfolge kann garantiert werden, weil das Schreiben in die Datenbank bei **814** stattfindet, bevor die neue Tabellenversion bei **818** lesbar wird.

[0098] In einer Ausführungsform ist es möglich, zu garantieren, dass die Übergabereihenfolge widersprüchlicher Schreibvorgänge sowohl für die Quell- als auch für die Zieldatenbank die gleiche ist. Die Logik zum Anwenden der Transaktionsprotokolldatensätze auf das Ziel muss die geeignete Reihenfolge erzwingen, dies kann aber möglich sein, weil garantiert ist, dass das Auftreten der Transaktionsprotokolldatensätze in der Datenbank (bezüglich der Konflikte) das gleiche wie die Reihenfolge ist, in der die Konflikte an der Quelle gelöst wurden. In einer Ausführungsform kann es an der Replikationsinfrastruktur liegen, sicherzustellen, dass die geeignete Reihenfolge der Anwendung der Transaktionsprotokolldatensätze durchgesetzt wird.

[0099] In einer Ausführungsform ist es möglich, dass zwei nicht miteinander in Beziehung stehende Transaktionen in der Datenbank in einer anderen Reihenfolge als ihre Transaktionskennungsreihenfolge in der Quelle erscheinen. Der Protokolldatensatz einer Transaktion mit der Kennung abc könnte z. B. in der Datenbank vor dem für die Transaktion mit der Kennung xyz erscheinen, vorausgesetzt, die beiden Transaktionen weisen keine Konflikte miteinander und keine Abhängigkeiten voneinander auf. Dies könnte z. B. geschehen, falls die Transaktion mit der ID abc bei **812** in den vorbereiteten Zustand eintritt und dann steckenbleibt, wonach die Transaktion mit der Kennung xyz bei **812** in den vorbereiteten Zustand eintritt und ihren Protokolldatensatz erfolgreich in der Datenbank aufzeichnet. Im Allgemeinen wirft dies kein transaktionales Konsistenzproblem

auf, weil die Reihenfolge zwischen zwei nicht in Beziehung stehenden Transaktionen undefiniert ist. Falls aus irgendeinem Grund es eine Anforderung wird, sicherzustellen, dass Transaktionsprotokolldatensätze in der Datenbank in der Transaktionskennungsreihenfolge erscheinen, kann die Transaktionsprotokolllogik z. B. über des Einführen einer Transaktionsprotokollsperre und/oder das Stehenbleiben von Schreibvorgängen, bis alle vorbereiteten Transaktionen mit einer niedrigeren Transaktionskennung in die Datenbank entleert worden sind, erweitert werden, um diese Reihenfolge durchzusetzen.

[0100] In einer Ausführungsform kann, wenn eine Momentaufnahme der Inhalte einer Tabelle (z. B. der Metadaten-Dateiliste und der implizierten Mikropartitionsliste) erzeugt wird, die Semantik einer gemeinsamen Zeit durch das Extrahieren der Tabellenversion erreicht werden, die der beginnenden Momentaufnahmezeit entspricht. Eine derartige Ausführungsform kann möglich gemacht werden, weil alle DMLs über geeignete Sperren synchronisiert sind, alle DMLs über den internen Lamport-Takt geordnet sind und alle früheren Versionen bis zur Zeitreisegrenze beibehalten werden.

[0101] **Fig. 9** ist eine schematische graphische Darstellung, die eine Auffrischungsanforderung **900** zum Replizieren einer Datenbank veranschaulicht. In einer Ausführungsform umfasst das Synchronisieren des Zustands einer Tabelle der Zielbereitstellung mit dem Zustand der Tabelle der Quellbereitstellung (a) das Senden einer Auffrischungsanforderung von der Zielbereitstellung an die Quellbereitstellung; (b) das Senden einer Momentaufnahmeantwort von der Quellbereitstellung an die Zielbereitstellung; und (c) das Importieren der Momentaufnahmeantwort in die Zielbereitstellung. Die in **Fig. 9** veranschaulichte Auffrischungsanforderung **900** berechnet ein Inventar der aktiven Mikropartitionen der Tabelle bei der Zielbereitstellung. Das Inventar enthält die aktuelle Tabellenversion und den Satz der globalen Mikropartitionsverweise, die in der Tabellenversion aktiv sind. In einer Ausführungsform, wie in **Fig. 9** veranschaulicht ist, werden keine anderen Metadaten oder Kurznamen lokaler Datei-Mikropartitionen gesendet.

[0102] **Fig. 9** veranschaulicht das Senden einer Auffrischungsanforderung von einer Zielbereitstellung dep2 an eine Quellbereitstellung dep1. Die Quellbereitstellung dep1 enthält eine Auflistung der aktiven Dateien der Tabelle T. Die Zielbereitstellung d2 enthält außerdem eine Auflistung der aktiven Dateien der Tabelle T. Wie in dem Inventarkasten dargestellt, ist die aktuelle Tabellenversion zur Veranschaulichung Nr. 342. Das Inventar enthält eine Auflistung der relevanten globalen Dateiverweise. Die Zielbereitstellung d2 setzt alle aktiven Dateien in der Tabellenversion Nr. 342 in eine Liste der globalen Dateiverweise gemäß dem Inventar um. Die lokal hinzugefügten Mikropartitionen fdn27 und fdn28 werden in globale Dateiverweise (dep2, fdn27) bzw. (dep2, fdn28) umgesetzt. Die Namensgebungskonvention „fdn“, gefolgt von einer Nummer, wie sie hier verwendet wird, kann sich auf eine bestimmte Mikropartition in einer Tabelle der Datenbank beziehen. Wie in **Fig. 9** veranschaulicht ist, werden nur globale Dateiverweise als Teil des Tabelleninventars gesendet und werden nur aktive Dateien gesendet.

[0103] **Fig. 10** ist eine schematische graphische Darstellung, die eine Momentaufnahmeantwort **1000** zum Replizieren einer Datenbank veranschaulicht. Die Momentaufnahmeantwort **1000** wird durch die Quellbereitstellung dep1 in Reaktion auf die Auffrischungsanforderung **900** erzeugt. Die Momentaufnahmeantwort **1000** enthält eines oder mehrere des Folgenden: (a) alle Mikropartitions-Metadaten, die zu der Tabelle hinzuzufügen sind; (b) die tatsächlichen Mikropartitionen in einem neu verschlüsselten Zustand; (c) alle globalen Mikropartitionsverweise, die aus der Tabelle zu entfernen sind; (d) die von der Zielbereitstellung dep2 gesendete Tabellenversion; und (e) den Replikations-Masterschlüssel, von dem aus die Mikropartitionen neu verschlüsselt wurden. In einer Ausführungsform wird die Momentaufnahmeantwort **1000** in die Momentaufnahmeantwortnachricht, die Metadaten-Dateien und die Mikropartitionen partitioniert. Die Momentaufnahmeantwortnachricht **1000** kann Zeiger auf die Metadaten-Dateien enthalten. Die Metadaten-Dateien können die hinzugefügten Mikropartitions-Metadaten und die gelöschten globalen Dateiverweise enthalten. Die Metadaten-Dateien und Mikropartitionen können in den Eingangsdatenträger der Zielbereitstellung d2 kopiert werden.

[0104] **Fig. 10** veranschaulicht die Quellbereitstellung dep1, die die Momentaufnahmeantwort **1000** zu der Zielbereitstellung dep2 überträgt. Sowohl die Quellbereitstellung dep1 als auch die Zielbereitstellung dep2 enthalten eine Auflistung der aktiven Dateien der Tabelle T. Die Momentaufnahmeantwort **1000** stellt zu Veranschaulichungszwecken die Tabellenversion Nr. 342 dar und gibt die hinzuzufügenden und zu löschenden Dateien und Metadaten an. In der in **Fig. 10** veranschaulichten Ausführungsform gibt die Momentaufnahmeantwort **1000** an, dass (fdn15 und ihre zugehörigen Metadaten) zusammen mit (fdn16_g (dep0, fdn6) und ihren zugehörigen Metadaten) hinzugefügt werden sollten. Die Momentaufnahmeantwort **1000** gibt an, dass (dep1, fdn12) und (dep0, fdn4) und (dep2, fdn27) gelöscht werden sollten.

[0105] **Fig. 10** veranschaulicht, dass die Tabellenversion Nr. 342 der Zielbereitstellung dep2 an die Zielbereitstellung dep2 zurückgeschickt wird. Wie in der Diskrepanz zwischen der Quellbereitstellung dep1 und der Zielbereitstellung dep2 veranschaulicht ist und wie in der Momentaufnahmeantwort **1000** dargestellt ist, müssen die Mikropartitionen mit den Kurznamen fdn15 und fdn16_g zu der Tabelle T in der Zielbereitstellung dep2 hinzugefügt werden. Ferner müssen die Mikropartitionen mit den globalen Dateiverweisen (dep1, fdn12), (dep0, fdn4) und (dep2, fdn27) aus Tabelle T entfernt werden. Die Mikropartitionen fdn15 und fdn16_g werden neu verschlüsselt und auf den Eingangsdatenträger der Zielbereitstellung dep2 hochgeladen. Der Replikations-Masterschlüssel ist ein Teil der (in **Fig. 10** nicht veranschaulichten) Momentaufnahmeantwort.

[0106] **Fig. 11** ist eine schematische graphische Darstellung, die den Import **1100** einer Momentaufnahmeantwort zum Replizieren einer Datenbank veranschaulicht. In einer Ausführungsform wird, wenn eine Momentaufnahmeantwort importiert wird, die Tabelle in der Zielbereitstellung dep2 bei Bedarf zu der gesendeten Tabellenversion rückgängig gemacht. Die hinzugefügten Dateien der Momentaufnahmeantwort können einen lokalen Kurznamen empfangen, der auf der Job-ID der DML basiert, und können ein Postfix oder eine andere geeignete Kennung enthalten (das Postfix „_g“ ist in den **Fig. 9-11** dargestellt). Die ursprünglichen globalen Dateiverweise können als Teil der Metadaten gespeichert sein. Die globalen Dateiverweise, die gelöscht werden müssen, können an der Zielbereitstellung dep2 unter Verwendung eines Index im Speicher in lokale Kurznamen umgesetzt werden. In einer Ausführungsform werden die lokalen Kurznamen zu den Metadaten-Dateien, die den DML-Befehl betreffen, als Teil des Abschnitts für gelöschte Kurznamen hinzugefügt.

[0107] Der Import **1100** der Momentaufnahmeantwort, wie in **Fig. 11** veranschaulicht ist, veranschaulicht, dass die Tabelle T bei Bedarf zur Tabellenversion Nr. 342 rückgängig gemacht wird. Wie in der Ausführungsform nach **Fig. 11** veranschaulicht ist, werden die hinzugefügten Dateien unter Verwendung eines lokalen Kurznamens mit einem angehängten „_g“ zu der Tabelle hinzugefügt, z. B. fdn25_g und fdn26_g. Die ursprünglichen globalen Dateiverweise werden bewahrt, einschließlich (dep1, fdn15) und (dep0, fdn6). Zusätzlich werden die gelöschten globalen Dateiverweise in lokale Kurznamen umgesetzt, einschließlich (dep1, fdn12), (dep0, fdn4) und (dep2, fdn27), die zu fdn22_g, fdn24_g und fdn27 umgesetzt werden. Zusätzlich werden, wie in **Fig. 11** veranschaulicht ist, die lokal gelöschten Kurznamen zu einem gelöschten Abschnitt der Metadaten-Dateien hinzugefügt, die den DML-Befehl betreffen. Die Tabelle kann mit einem Kompaktierer gekürzt werden, wobei beide Tabellen denselben Zustand enthalten können.

[0108] **Fig. 12** ist eine schematische graphische Darstellung, die eine Bereitstellungarchitektur **1200** zum Replizieren einer Datenbank veranschaulicht. Die Bereitstellungarchitektur **1200** enthält eine Bereitstellung D1, eine Bereitstellung D2 und eine Bereitstellung D3. Die Bereitstellung D1 enthält einen D1-Replikationsbehälter **1204**, in dem sie die Nachrichten von anderen Bereitstellungen empfängt. Ähnlich enthält die Bereitstellung D2 einen D2-Replikationsbehälter **1210** und die enthält Bereitstellung D3 einen D3-Replikationsbehälter **1216**. Jeder der Replikationsbehälter **1204**, **1210**, **1216** ist in Unterbehälter aufgeteilt, einschließlich eines Unterbehälters pro Bereitstellung. Jeder der Unterbehälter der Replikationsbehälter **1204**, **1210**, **1216** kann unabhängig mit Berechtigungen und Zugangsberechtigungsnachweisen konfiguriert sein. Die Bereitstellung D1 enthält einen D1-EP/Mikropartitionsbehälter **1206**, die Bereitstellung D2 enthält einen D2-EP/Mikropartitionsbehälter **121** und die Bereitstellung D3 enthält einen D2-EP/Mikropartitionsbehälter **1218**.

[0109] In einer Ausführungsform werden alle Stufen für die Replikation unter einer dedizierten Datenbank erzeugt, so dass auf die Datenbank mit einem Kurznamen verwiesen werden kann, wenn die Bereitstellungen erzeugt werden, wobei die Bereitstellungen logisch gruppiert werden können. In einer Ausführungsform wird ein DeploymentDPO verwendet, um die Bereitstellunginformationen zu speichern, die durch einen Nachrichtenübermittlungsdienst und andere Teile der Infrastruktur verwendet werden. Das DeploymentDPO ist eine reguläre Wörterbuchentität in einer Ausführungsform, wobei der Zugriff auf es über Erzeuge-, Zeige- und Fallenlassanweisungen eingeschränkt ist. Das DeploymentDPO ist eine Datenstruktur, die Informationen (d. h., Metadaten) über eine spezielle Bereitstellung enthält. Das DeploymentDPO kann für Operationen verwendet werden, die diese spezielle Bereitstellung einbeziehen.

[0110] Jeder der Replikationsbehälter **1204**, **1210**, **1216** kann die server-seitige Verschlüsselung aktiviert aufweisen. Zusätzlich können alle Dateien, die Kundendaten enthalten, auf der Client-Seite verschlüsselt sein. Eine Bereitstellung kann vollen Zugriff auf ihren eigenen Replikationsbehälter haben, wobei eine Bereitstellung Schreibzugriff nur auf ihren Unterbehälter des Replikationsbehälters einer weiteren Bereitstellung haben kann, in den sie Nachrichten schreibt.

[0111] In einer Ausführungsform wird, wenn eine neue Bereitstellung erzeugt wird, ein neuer Replikationsbehälter für diese Bereitstellung erzeugt, einschließlich aller Unterbehälter für alle Bereitstellungen, so dass

andere Bereitstellungen Nachrichten an die neue Bereitstellung senden können. Zusätzlich kann zu den Replikationsbehältern aller anderen Bereitstellungen ein neuer Unterbehälter für die Bereitstellung hinzugefügt werden, so dass die neue Bereitstellung Nachrichten an die vorhandenen Bereitstellungen senden kann.

[0112] Die in **Fig. 12** veranschaulichte Nachrichtenübermittlungsinfrastruktur stellt eine Infrastruktur bereit, die es den Bereitstellungen ermöglicht, generische Nachrichten durch das Austauschen von Dateien in den Behältern auszutauschen. Die Nachrichten können über einen Cloud-Speicher ausgetauscht werden und können für ein zugehöriges Kundenkonto transparent sein. Für ein zugeordnetes Kundenkonto kann es den Anschein haben, dass das Konto nur mit einem regulären Datenbeständigkeitsobjekt (DPO) in den lokalen Metadaten wechselwirkt. Eine Nachrichtendienstschicht kann einkapseln, wie die Nachrichten-DPOs serialisiert und ausgetauscht werden.

[0113] **Fig. 13** ist eine schematische graphische Darstellung, die einen Prozessablauf **1300** für das Senden von Nachrichten veranschaulicht, wenn eine Datenbank repliziert wird. Der Prozessablauf **1300** beginnt, wobei bei **1302** eine neue Nachricht eingeleitet und gespeichert wird. Die Nachricht wird bei **1304** zum Senden vorbereitet, wobei ein Kundenkonto bei **1306** aufruft und angibt, dass die Nachricht gesendet werden sollte. Nachdem alle Dateien hochgeladen sind, ist die Nachricht bei **1308** sendebereit, wobei ein Nachrichtendienst bei **1310** einen Stapel von Nachrichten extrahiert. Der Nachrichtendienst extrahiert periodisch einen Stapel von Nachrichten, die sendebereit sind, und verschiebt sie bei **1312** zu der Sendescheibe, wobei ein Nachrichtendienst die Nachrichten bei **1314** in einen Behälter schreibt. Nachdem die Nachrichtendatei erzeugt worden ist und die Nachricht gesendet worden ist, wird die Nachricht bei **1316** durch die Empfangsseite gelöscht und aus dem Speicher entfernt. Falls bei **1322** während dem Vorbereiten ein Fehler auftritt oder ist die Nachricht in der Vorbereitungsscheibe überaltert ist, werden die hochgeladenen Dateien bei **1320** aus dem Cloud-Speicher gelöscht. Nachdem die Bereinigungsnachricht bei **1318** an die empfangende Bereitstellung gesendet worden ist, kann die Nachricht aus dem Speicher entfernt werden. Der Nachrichtendienst extrahiert bei **1312** periodisch einen Stapel versandfertiger Nachrichten und verschiebt sie zu der Sendescheibe. In einer Ausführungsform berechnet und cachet der Nachrichtensendedienst den abgeleiteten Schlüssel, der verwendet wird, um den Datenverschlüsselungsschlüssel einzuwickeln. Der Nachrichtendienst muss ferner vielleicht den aktuellen und den nächsten Diversifizierer und den aktuellen und den nächsten Nachrichtendateinamen im Speicher bestehen lassen.

[0114] **Fig. 14** ist eine schematische graphische Darstellung, die einen Prozessablauf **1400** zum Empfangen von Nachrichten veranschaulicht, wenn eine Datenbank repliziert wird. Der Prozessablauf **1400** enthält einen Nachrichtendienst, der eine aus einem Replikationsbehälter abgelegte Nachricht heruntergeladen hat und die Nachrichtendatei bei **1402** deserialisiert. Die Nachricht wird deserialisiert und bei **1404** im Speicher bestehenbleibend gemacht. Die Nachricht wird bei **1406** empfangen, wobei bei **1408** ein Stapel empfangener Nachrichten extrahiert wird. Der Nachrichtenverarbeitungsdienst bildet einen Stapel empfangener Nachrichten und verschiebt sie bei **1410** zu einer verarbeitungsbereiten Scheibe. Nachdem bei **1412** ein Arbeiter-Thread die Nachricht verarbeitet hat, ist die Nachricht bei **1414** löschbereit. Eine Löscheinrichtung entfernt bei **1416** die Nachrichtendateien aus dem Cloud-Speicher, wobei dann bei **1418** die Nachricht aus den Metadaten entfernt werden kann.

[0115] In einer Ausführungsform erfolgt das Bereinigen der Nachricht auf der Empfangsseite, weil der Sender nur Schreibzugriff auf die Replikationsbehälter hat. Nachdem eine Nachricht verarbeitet worden ist, kann der Bereinigungsdienst alle in Beziehung stehenden Dateien im Cloud-Speicher löschen und die Nachricht aus dem Speicher entfernen. In einer Ausführungsform wird für alle Nachrichten, die während der Vorbereitung fehlerhaft entfernt worden sind, eine Bereinigungsanforderung an die empfangende Bereitstellung gesendet.

[0116] **Fig. 15** ist eine schematische graphische Darstellung, die eine globale Bereitstellungsgruppe **1500** veranschaulicht, die drei Bereitstellungen zum Replizieren einer Datenbank enthält. Während der Datenbankreplikation, wie sie hier offenbart wird, werden Metadaten beständig gehalten und innerhalb der Bereitstellungsreplikationsgruppen (die als eine Bereitstellungsgruppe bezeichnet werden können) ausgetauscht. Die Bereitstellungsgruppen werden erzeugt, um die Replikation zwischen den einzelnen Bereitstellungsgruppen zu ermöglichen. In einer Ausführungsform erhält jede Bereitstellung eine Liste aller anderen Bereitstellungen in der Gruppe einschließlich sich selbst aufrecht. In einer Ausführungsform wird die Liste innerhalb jeder Bereitstellung unter Verwendung einer „Erzeuge-Bereitstellung“-Da-tendefinitionssprache (DDL) manuell aufrechterhalten, die verwendet wird, um neue Bereitstellungen zu der Gruppe hinzuzufügen. Diese DDL kann in jeder vorhandenen Bereitstellung ausgeführt werden. Innerhalb einer Bereitstellung kann ein Konto global (gegen lokal) gemacht werden, um eine neue Kontenreplikationsgruppe zu bilden oder einer vorhandenen Kontenreplikationsgruppe beizutreten. Nur die Konten, die ein Teil derselben Kontenreplikationsgruppe sind, können die

Daten inmitten der Gruppe replizieren. In einer Ausführungsform wird das Bilden einer neuen Kontoreplikationsgruppe anfangs in Reaktion auf die Kundenkontoanforderung ausgeführt, zwei oder mehr der Konten des Kunden miteinander zu verknüpfen. Die neuen Konten können automatisch in derselben Replikationsgruppe wie das Konto, von dem die Erzeugungsanweisung ausgegeben wurde, angeordnet werden.

[0117] In einer Ausführungsform können die Konten innerhalb einer einzigen Kontengruppe lokale Objekte fördern, so dass sie global sind, oder direkt globale Objekte erzeugen. In verschiedenen Ausführungsformen kann ein Objekt eine Datenbank, einen Anwender, eine Rolle, ein Lager, eine globale Verbindung, eine Organisation usw. enthalten. Sobald ein Objekt global ist, kann es innerhalb irgendeines Kontos in der globalen Kontengruppe repliziert werden. Die Replikation eines globalen Objekts wird erreicht, indem zuerst eine lokale Kopie für dieses globale Objekt in allen Konten erzeugt wird, wo das Objekt repliziert werden soll, und dann diese Kopien explizit, planmäßig oder kontinuierlich aufgefrischt werden. In einer Ausführungsform können nur Datenbanken durch einen Kontomanager global gemacht werden, wobei die Kopien nur durch den Eigentümer der Datenbank explizit aufgefrischt werden können.

[0118] In einer Ausführungsform gibt es drei Klassen von Metadaten, um Datenbankdaten zu managen und zu replizieren. Eine Klasse von Metadaten ist auf Bereitstellungen gerichtet, einschließlich der Metadaten über jede Bereitstellung einer Bereitstellungsgruppe, die manuell durch Replikation erzeugt und repliziert wird. Eine Klasse von Metadaten ist auf globale Konten gerichtet, wobei alle globalen Konten einer Bereitstellung eine Replikation zu allen anderen Bereitstellungen innerhalb der Bereitstellungsgruppe, zu der sie gehört, sein können. Eine Klasse von Metadaten enthält globale Datenbanken, einschließlich aller globalen Datenbanken in einem Konto, das außerdem innerhalb derselben Kontengruppe repliziert werden kann. In einer Ausführungsform sind nur die Informationen über alle Kopien einer globalen Datenbank eine Replikation in der Kontengruppe zu der Teilmenge von Bereitstellungen, wo die Kontengruppe vorhanden ist.

[0119] **Fig. 15** veranschaulicht ein Beispiel unter Verwendung einer globalen Bereitstellungsgruppe, die drei Bereitstellungen, die Bereitstellung D1, die Bereitstellung D2 und die Bereitstellung D3, enthält. Wie in **Fig. 15** veranschaulicht ist, enthält der Bereitstellung D1 fünf Konten, einschließlich D1.A1, D1.A2, D1.A3, D1.A4 und D1.A5. Die Bereitstellung D2 enthält vier Konten, einschließlich D2.A1, D2.A2, D2.A3 und D2.A4. Die Bereitstellung D3 enthält vier Konten, einschließlich D3.A1, D3.A2, D3.A3 und D3.A4. In der in **Fig. 15** veranschaulichten Ausführungsform gibt es vier lokale Konten, die kein Teil irgendeiner Gruppe sind und keine globalen Objekte aufweisen können. Die vier lokalen Konten enthalten D1.A3, D2.A2, D3.A3 und D3.A4 und sind mit einer punktierten Linie veranschaulicht. Nur globale Konten (d. h., die mit einer durchgezogenen Linie veranschaulichten und schattierten Konten ohne Füllung, mit hellgrauer oder dunkelgrauer Füllung) können globale Datenbanken erzeugen oder replizieren. In dem in **Fig. 15** veranschaulichten Beispiel gibt es vier globale Datenbanken, einschließlich DB1, DB2, DB3 und DB4. Dieselbe globale Datenbank kann nur innerhalb derselben Kontengruppe vorhanden sein oder repliziert werden. In dem in **Fig. 15** veranschaulichten Beispiel sind DB1 und DB2 globale Datenbanken, die nur innerhalb der Kontengruppe repliziert werden können, die D1.A1, D1.A4, D2.A4 und D3.A2 enthält. Ferner kann DB3 nur innerhalb der Kontengruppe repliziert werden, die D1.A2 und D2.A1 enthält. Ferner kann DB4 nur innerhalb der Kontengruppe repliziert werden, die D1.A5 und D2.A3 enthält. Zusätzlich werden, wie in **Fig. 15** veranschaulicht ist, die globalen Datenbanken nicht notwendigerweise durch alle Konten innerhalb einer globalen Kontengruppe repliziert. Ein Kunde, der Eigentümer der dunkel schattierten Kontengruppe (der die DB1 und die DB2 zugeordnet sind) ist, repliziert die DB2 nicht mit dem D1.A4-Konto.

[0120] In einer Ausführungsform werden die Metadaten über alle Kopien eines globalen Objekts zu allen Konten in der Kontengruppe repliziert. In bestimmten Ausführungsformen kann dies einem Manager eines lokalen Kontos (d. h., jener, die mit einer punktierten Linie veranschaulicht sind) erlauben, alle lokalen oder entfernten Kopien irgendeines globalen Objekts in der Gruppe aufzulisten. Dies kann es einem Kundenkontomanager ermöglichen, neue Kopien eines globalen Objekts in anderen Konten in der Kontengruppe (z. B. der veranschaulichten Kontengruppe ohne Füllung, mit hellgrauer Füllung oder dunkelgrauer Füllung) durch das Spezifizieren zu erzeugen, dass das neue Objekt, das erzeugt wird, eine Kopie dieses globalen Objekts ist.

[0121] Als ein Beispiel möchte das Kundenkonto für das Konto D2.A4 (das der Kontengruppe mit dunkelgrauer Füllung zugeordnet ist) die globale Datenbank DB2 zu diesem Konto replizieren. In diesem Konto kann das Kundenkonto einen Befehl ausführen, die globale Datenbanken zu zeigen. Der Befehl listet die Kopien aller globalen Datenbanken in der Kontengruppe auf. Basierend auf diesem Beispiel zeigt der Befehl fünf Beispiele, wie in der Tabelle 1 im Folgenden veranschaulicht ist.

Tabelle 1

Gebiet	Konto	Replikationsgruppe	Name
D1	A1	b4a193a3-77cc-49dc-a9c8-2a2ee1ae9b1e	DB1
D1	A4	b4a193a3-77cc-49dc-a9c8-2a2ee1ae9b1e	DB1
D3	A2	b4a193a3-77cc-49dc-a9c8-2a2ee1ae9b1e	DB1
D1	A1	0400d847-4199-4f79-9a74-381761bc0cc9	DB2
D3	A2	0400d847-4199-4f79-9a74-381761bc0cc9	DB2

[0122] Wie in der Tabelle 1 veranschaulicht ist, stellt die Spalte „Replikationsgruppe“ die gleichen Werte für alle Kopien derselben Datenbank dar. Die Datenbankkopien sind wie ein Konto in der Kontengruppe miteinander verknüpft. Diese Datenbanken bilden ferner eine Replikationsgruppe mit einer Identifikationsnummer, die gleich der Nummer der Replikationsgruppe ist. Weiter im obenerwähnten Beispiel kann das Kundenkonto von D2.A4 eine neue Kopie in der Datenbank-Replikationsgruppe mit dem Namen „0400d847-4199-4f79-9a74-381761bc0cc9“ erzeugen, indem es einen Befehl ausgibt, dies auszuführen. Es sollte erkannt werden, dass der lokale Name der Kopie irgendetwas sein kann, wobei das Spezifizieren der Identifikationsnummer der Replikationsgruppe die Datenbank zu einem Teil derselben Replikationsgruppe wie die anderen Datenbanken, die in dieser Gruppe waren, macht. Nach dem Erzeugen einer neuen Datenbankkopie kann das Kundenkonto für D2.A4 dann einen Befehl ausgeben, um alle Datenbankkopien zu zeigen, wobei es dann eine Liste mit der Kopie empfängt, die soeben erzeugt wurde, wie in der Tabelle 2 im Folgenden veranschaulicht ist.

Tabelle 2

Gebiet	Konto	Replikationsgruppe	Name
D1	A1	b4a193a3-77cc-49dc-a9c8-2a2ee1 ae9b1e	DB1
D1	A4	b4a193a3-77cc-49dc-a9c8-2a2ee1 ae9b1e	DB1
D3	A2	b4a193a3-77cc-49dc-a9c8-2a2ee1 ae9b1e	DB1
D1	A1	0400d847-4199-4f79-9a74-381761bc0cc9	DB2
D3	A2	0400d847-4199-4f79-9a74-381761bc0cc9	DB2
D2	A4	0400d847-4199-4f79-9a74-381761bc0cc9	DB5

[0123] Weiter im obenerwähnten Beispiel erzeugt derselbe Befehl, der von irgendeinem Konto in dieser Gruppe (d. h., D1.A1 oder D1.A4) ausgegeben wird, genau dieselbe Liste. Die Übertragung der replizierten Metadaten kann einen Zeitraum brauchen, sie kann z. B. mehrere Sekunden brauchen, wobei nach diesem Zeitraum jede andere Bereitstellung über die neue Kopie Bescheid weiß.

[0124] Ähnlich dem „Zeige-globale-Datenbanken“-Befehl kann ein „Zeige-globale-Konten“-Befehl ausgegeben werden, um eine Liste der Menge von Konten in der Gruppe zu erzeugen. Falls das Kundenkonto für D3.A2 den „Zeige-globale-Konten“-Befehl das obenerwähnte Beispiel fortsetzend ausgibt, gibt er eine Liste wie in der Tabelle 3 im Folgenden zurück.

Tabelle 3

Bereich	Konto
D1	A1
D1	A4
D2	A4
D3	A2

[0125] Wie in der Tabelle 3 veranschaulicht ist, wird die Identifikationsnummer der Kontenreplikationsgruppe nicht aufgedeckt, weil es für einen gegebenen Kunden nur eine Kontenreplikationsgruppe gibt. Wenn derselbe Befehl von irgendeinem Kundenkonto in der Bereitstellungsgruppe ausgeführt wird, erzeugt der Befehl eine

Liste, die alle Kontengruppen zeigt, wobei in diesem Fall eine Spalte hinzugefügt werden kann, die die Identifikationsnummer der Replikationsgruppe zeigt.

Speichern von Metadaten über globale Entitäten

[0126] Jede Bereitstellung in einer Bereitstellungsgruppe kann Metadaten über alle globalen Konten in der Gruppe aufrechterhalten. Abermals unter Verwendung des obererwähnten Beispiels kann jede Bereitstellung die Liste aller globalen Konten, d. h., D1.A1, D1.A2, D1.A4, D1.A5, D2.A1, D2.A2, D3.A1 und D3.A3, aufrechterhalten. Die Liste aller globalen Konten kann vollständig repliziert werden. Zusätzlich erhält jede Bereitstellung die Metadaten über alle globalen Objekte in der Teilmenge der Kontengruppen, die in dieser Bereitstellung vorhanden sind, aufrecht. Weiterhin unter Verwendung des Beispiels erhält die Bereitstellung D1 die Metadaten über alle globalen Objekte aufrecht, die sich im Besitz der nicht gefüllten, hellgrauen und dunkelgrauen Untergruppen befinden. Weil die Bereitstellung D2 nur Konten aus der dunkelgrauen und der nicht gefüllten Kontengruppe hostet, muss sie nur die Metadaten über die Datenbanken aufrechterhalten, die zu diesen beiden Kontengruppen gehören. Ferner muss die Bereitstellung D3 nur die Informationen über die globalen Datenbanken in den hellgrauen und nicht ausgefüllten Kontengruppen aufrechterhalten.

[0127] In jeder Bereitstellung kann ein einzelnes DPO verwendet werden, das als GlobalEntitiesDPO bezeichnet werden kann. Das GlobalEntitesDPO ist eine Datenstruktur, die die Informationen und/oder Metadaten über Entitäten, die repliziert werden, wie z. B. Konten, Datenbanken, Organisationen und/oder Verbindungen, aufrechterhält. Das einzelne DPO kann die Metadaten über alle globalen Objektkopien einschließlich der globalen Konten speichern. Die Konten können in einer Kontengruppe als die Kopien desselben globalen Kontos modelliert werden. Folglich werden die Informationen über die globalen Konten und die auszugleichenden Kontenentitäten, wie z. B. die Datenbanken, die Anwender, die Rollen und die Lager, vereinheitlicht. Ferner kann bei jeder Bereitstellung das GlobalEntitiesDPO die Informationen über irgendeine Kopie einer globalen Entität speichern, die die Bereitstellung kennen muss, d. h., die Informationen über alle globalen Konten und Datenbankkopien, über die die Bereitstellung Bescheid wissen muss (z. B. irgendwelche Kopien in irgendeiner Kontengruppe, die in der Bereitstellung vorhanden sind).

[0128] Zusätzlich zu dem GlobalEntitiesDPO, dessen Inhalt zwischen den Bereitstellungen repliziert wird, kann eine Bereitstellung alle Entitäten identifizieren, die in der Bereitstellung global sind. Dafür ist kein neues DPO erforderlich, kann aber das vorhandene BaseDictionaryDPO verbessern. Das BaseDictionaryDPO ist eine den DPO-Datenstrukturen zugrundeliegende Abstraktion, die verwendet werden kann, um die Informationen zu managen, die in einem Katalog zugänglich sind. Es kann ein Feld für die globale Identifikationsnummer hinzugefügt werden, die, falls sie nicht null ist, angibt, dass die Wörterbuchentität global ist. Ferner können alle globalen Wörterbuchentitäten indexiert werden, indem ein neues, „global“ genanntes Objekt hinzugefügt wird, um irgendeine globale Einheit in Anbetracht der globalen Identifikationsnummer zu finden. In einer Ausführungsform kann dies den Prozess des Findens aller globalen Entitäten eines bestimmten Typs in einer speziellen Bereitstellung oder in einem speziellen Konto vereinfachen.

[0129] In einer Ausführungsform enthält das Erzeugen einer globalen Datenbank das Erzeugen der ersten Master-Kopie in einer Replikationsgruppe globaler Datenbanken. Wenn diese erste Master-Kopie erzeugt wird, kann für sie automatisch eine Replikationsgruppe globaler Datenbanken erzeugt werden. Andere Kopien in der Gruppe können unter Verwendung eines „Replikationsgruppen“-Befehls erzeugt werden.

[0130] In einer Ausführungsform kann ein globales Objekt zurück in ein lokales Objekt umgesetzt werden. Einem Kunden- oder Managerkonto kann ein Befehl, um das Konto zu ändern, bereitgestellt werden, um ein vorhandenes globales Konto in ein lokales Konto zu überführen. Als eine Nebenwirkung dieses Befehls können alle globalen Objekte innerhalb des Kontos lokal gemacht werden. Ferner kann eine einzelne globale Datenbank unter Verwendung eines ähnlichen Befehls zurück in eine reguläre lokale Datenbank geändert werden.

[0131] In einer Ausführungsform wird irgendeine an einer Kopie vorgenommene Änderung zu allen anderen an der Änderung interessierten Bereitstellungen repliziert. Eine Änderung kann ein Erzeugen, Fallenlassen, Aktualisieren oder eine andere Einstellung enthalten. Die Replikation der Änderung geschieht so schnell wie möglich und kann in weniger als fünf Sekunden geschehen. Ferner wird eine Replikation aller in der Bereitstellung erzeugten Kopien in einem regelmäßigen Zeitraum, z. B. einmal pro Stunde, ausgeführt, auch wenn sich nichts geändert hat. Die kann sicherstellen, dass, falls etwas ausfällt, es dennoch etwas Abdeckung gibt.

[0132] Zusätzlich kann die Replikation von Kopie-Metadaten im Hintergrund stattfinden. Die Metadaten für eine Kopie können durch ein Kundenkonto oder einen Manager, dem die Kopie gehört, geändert werden, wobei

die Transaktion, die die Änderung vornimmt, außerdem die Benachrichtigung verwenden kann, um zu melden, dass eine Änderung vorgenommen wurde. In einer Ausführungsform sind die Nutzdaten der Benachrichtigung nur die Domäne, die die Änderungen verbraucht. Sobald eine Änderung vorgenommen wird, repliziert der Thread die Änderung für alle relevanten Implementierungen. Für eine Kontoänderung können dies alle Bereitstellungen in der Bereitstellungsgruppe sein, wobei für eine Datenbankänderung sie nur um eine Teilmenge der Bereitstellungen sein können, wo das Konto eine Replikation ist.

[0133] In einer Ausführungsform verwendet das Replizieren einer Änderung den globale Nachrichtenübermittlungsrahmen. Die Änderung kann unter Verwendung einer globalen Nachricht pro Bereitstellung geschoben werden. Dieselben Informationen können mehr als einmal repliziert werden, so dass die Änderungsbenachrichtigung nur aus dem Speicher entfernt werden kann, wenn alle globalen Nachrichten für diese Änderung in einer Warteschlange angeordnet worden sind.

[0134] Fig. 16 ist eine schematische graphische Darstellung, die ein Verschlüsselungssystem **1600** zum Replizieren einer Datenbank veranschaulicht. In einer Ausführungsform wird die Verschlüsselung durch das Verschlüsseln jeder Datei mit einem anderen Schlüssel und das Begrenzen der Anzahl der Zugriffe auf das HSM (Hardware-Sicherheitsmodul) ausgeführt. Zusätzlich kann die Verschlüsselung sicherstellen, dass es keinen bereitstellungsübergreifenden Zugriff auf das KMS gibt. In einer Ausführungsform enthält eine Nachrichtendatei eine Liste serialisierter GlobalMessageDPOs, den Namen der nächsten Nachrichtendatei und den Diversifizierer, der für die nächste Nachrichtendatei zu verwenden ist. Ein GlobalMessageDPO kann auf eine oder mehrere Nachrichtenkörperdateien zeigen. Eine Nachrichtendatei kann mit einem zufälligen Datenverschlüsselungsschlüssel (DEK) verschlüsselt sein, der in einer Ausführungsform durch Java erzeugt werden kann. Die Nachrichtenkörperdatei kann außerdem mit einem zufälligen DEK verschlüsselt sein. Jeder DEK kann mit einem Schlüssel eingewickelt sein, der von einem globalen Master-Schlüssel (GMK) und dem Diversifizierer abgeleitet ist, der in der vorhergehenden Nachrichtendatei spezifiziert wurde. Die Schlüsselableitung kann in dem HSM unter Verwendung eines HMAC-Algorithmus ausgeführt werden. Der abgeleitete Schlüssel kann in globalen Diensten gecachet werden, so dass er wiederverwendet werden kann, um den nächsten DEK einzuwickeln, wenn sich der Diversifizierer nicht geändert hat. Der eingewickelte DEK kann im Kopf der Nachrichtendatei im Cloud-Storage gespeichert werden. In einer Ausführungsform ist der Diversifizierer ein Zeitstempelwert, der in irgendeinem Zeitraum geändert werden kann und z. B. stündlich geändert werden kann. Aufgrund des Zeitstempels kann das Ziel Diversifizierer ablehnen, die älter als z. B. ein Tag oder ein anderer geeigneter Zeitraum sind. In dieser Weise kann das Ziel einen Satz von weniger granulären Eigenschaften am Diversifizierer durchsetzen.

[0135] Fig. 16 veranschaulicht ein Verschlüsselungssystem **1600** zur Verwendung bei der Verschlüsselung replizierter Daten. Das Verschlüsselungssystem **1600** enthält einen globalen Master-Schlüssel des HSM (Hardware-Sicherheitsmodul) (HSMGMK) **1602**. Der GSMGMK **1602** ist für mehrere Bereitstellungen vorgesehen. Der DEK1 wird z. B. bei **1618** mit dem HMAC eingewickelt, der DEK2 wird bei **1620** mit einem gecacheten HMAC eingewickelt und der DEK3 wird bei **1622** mit einem neuen HMAC eingewickelt. Die nächste DEK1-Datei **1604** wird der nächsten DEK2-Datei **1606** zugeführt, die ferner der nächsten DEK3-Datei **1608** zugeführt wird. Die Nachrichtendatei enthält eine Liste der GlobalMessageDPOs (**1614a**, **1614b**, **1614c**), den Namen und den Diversifizierer (**1610**, **1612a**, **1612b**) für die nächste Nachrichtendatei. Die GlobalMessageDPOs **1614a**, **1614b**, **1614c** zeigen in einer Ausführungsform auf null oder mehr Nachrichtenkörperdateien **1616a-1616e**. In einer Ausführungsform wird jede Nachrichtenkörperdatei **1616a-1616e** mit einem zufälligen DEK verschlüsselt und wird jeder DEK mit einem von dem HMSGMK und einem Diversifizierer einer vorherigen Nachrichtendatei abgeleiteten Schlüssel eingewickelt. Wie in **Fig. 16** veranschaulicht ist, ist die Nachrichtenkörperdatei **1616a** mit dem DEK4 verschlüsselt, ist die Nachrichtenkörperdatei **1616b** mit dem DEK5 verschlüsselt, ist die Nachrichtenkörperdatei **1616c** mit dem DEK6 verschlüsselt, ist die Nachrichtenkörperdatei **1616d** mit dem DEK7 verschlüsselt und ist die Nachrichtenkörperdatei **1616e** mit dem DEK8 verschlüsselt. Der eingewickelte DEK kann im Kopf der Nachrichtendatei im Cloud-Speicher gespeichert sein. Der abgeleitete Schlüssel kann gecachet und wiederverwendet werden. Ferner kann der nächste Diversifizierer **1610**, **1612a**, **1612b** in irgendeinem geeigneten Zeitintervall, z. B. stündlich, geändert werden.

[0136] Fig. 17 ist eine schematische graphische Darstellung, die eine Verschlüsselung **1700** von Dateien unter Verwendung eines client-gemanagten Schlüssels zum Replizieren einer Datenbank veranschaulicht. Wenn Daten repliziert werden, werden die Metadaten-Dateien und Mikropartitionen von der Quellbereitstellung zu der Zielbereitstellung kopiert. In einer Ausführungsform beinhaltet dies zwei Kopieroperationen. Zuerst werden die Dateien vom Mikropartitionsdatenträger der Quellbereitstellung in die Eingangsstufe der Zielbereitstellung kopiert. Zweitens werden die Dateien von der Eingangsstufe der Zielbereitstellung zum Mikropartitionsdatenträger der Zielbereitstellung kopiert. Die doppelte Kopie ist in einer Ausführungsform erforderlich, in der auf

die Mikropartitionsdatenträger über die Bereitstellungen nicht direkt zugegriffen werden kann und folglich die Eingangsstufe in die Zielbereitstellung umgeleitet wird.

[0137] In einer Ausführungsform wird die Datenreplikation durch eine Auffrischungsanforderung **900** von der Zielbereitstellung zur Quellbereitstellung ausgelöst. Die Auffrischungsanforderung **900** wird durch eine Momentaufnahmeantwort **1000** beantwortet, die durch die Quellbereitstellung erzeugt wird. Die Momentaufnahmeantwort **1000** enthält in der Nachrichtenkörperdatei eine Momentaufnahme der Wörterbuch-Metadaten einschließlich z. B. der Schema-DPOs, der Tabellen-DPOs usw. zusammen mit einer Liste von Metadaten-Dateinamen und einen zufälligen Replikations-Master-Schlüssel (RepMK). Der RepMK wird durch den HSM des Quellbereitstellung erzeugt. In einer Ausführungsform enthält jede Kopieroperation die Neuverschlüsselung aller Dateien. Wenn die Dateien in den Eingangsdatenträger kopiert werden, werden die Dateien mit einem individuellen Schlüssel, der aus dem RepMK und dem Dateinamen jeder Datei abgeleitet wird, neu verschlüsselt. Wenn die Dateien in den Ziel-Mikropartitionsdatenträger kopiert werden, werden die Dateien mit den jeweiligen Master-Schlüsseln der Metadaten-Datei und Tabellen-Master-Schlüsseln des Kundenkontos in der Zielbereitstellung neu verschlüsselt. In einer Ausführungsform wird der RepMK unter Verwendung des HSMGMK eingewickelt, bevor er als Teil der Momentaufnahmeantwort gesendet wird.

[0138] In einer Ausführungsform enthält die Auffrischungsanforderung **900** bei der Replikation einer Datei eines Kundenkontos, das vom Kunden gemanagte Schlüssel verwendet, einen öffentlichen Schlüssel. Der öffentliche Schlüssel ist ein Teil eines öffentlich-geheimen Schlüsselpaares, das unter Verwendung des KMS-Schlüssels des Kunden in der Zielbereitstellung erzeugt wird. Der eingewickelte RepMK in der Momentaufnahmeantwort **1000** wird zusätzlich durch den öffentlichen Schlüssel eingewickelt, bevor er gesendet wird. In einer Ausführungsform wird der RepMK folglich zweimal eingewickelt: erstens durch den HSMGMK und zweitens durch den öffentlichen Schlüssel. Während der zweiten Kopie von der Eingangsstufe zum Ziel-Mikropartitionsdatenträger wird das RepMK zuerst unter Verwendung des geheimen Schlüssels ausgewickelt und dann unter Verwendung des HSMGMK ausgewickelt.

[0139] Wie in **Fig. 17** veranschaulicht ist, enthält die Quellbereitstellung **1702** ein KMS **1706a** und ein HSM **1708a**. Das KMS **1706a** und das HSM **1708a** werden verwendet, um den AMK **1710a** und den TMK **1712a** zu erzeugen. Die Dateien werden bei 1714 mit den von dem Quell-EPFM/TMK abgeleiteten Schlüsseln verschlüsselt. Die Zielbereitstellung **1704** enthält ihre eigenen KMS **1706b** und HSM **1708b**, die verwendet werden, um den AMK **1710b** und den TMK **1712b** zu erzeugen. Die Dateien werden mit den vom RepMK abgeleiteten Schlüsseln bei **1716** in der Eingangsstufe verschlüsselt, wobei die Dateien bei **1718** mit den vom Ziel-EP/TMK abgeleiteten Schlüsseln im Mikropartitionsdatenträger verschlüsselt werden. Die Auffrischungsanforderung enthält bei **1720** einen öffentlichen Schlüssel von KMS, wenn sie von der Zielbereitstellung **1704** an die Quellbereitstellung **1702** gesendet wird. Die Momentaufnahmeantwort enthält bei **1722** den mit dem HSMGMK eingewickelten RepMK und einen durch die Quellbereitstellung **1702** erzeugten und an die Zielbereitstellung **1704** gesendeten öffentlichen Schlüssel. Wie veranschaulicht ist, enthält die Aktualisierungsanforderung einen von einem Kunden-KMS-Schlüssel in der Zielbereitstellung abgeleiteten öffentlichen Schlüssel, während die Momentaufnahmeantwort ein mit dem HSMGMK und dem öffentlichen Schlüssel doppelt eingewickelten zufälligen RepMK enthält. Die Dateien in der Eingangsstufe werden mit einem aus dem RepMK und dem Dateinamen jeder Datei abgeleiteten individuellen Schlüssel verschlüsselt. Ferner werden in einer weiteren Ausführungsform die Dateien im Mikropartitionsdatenträger (der Quelle oder dem Ziel) wie üblich unter Verwendung eines Master-Schlüssels der Metadaten-Datei und eines Master-Schlüssel der Tabelle verschlüsselt.

[0140] **Fig. 18** ist ein schematischer Ablaufplan eines Verfahrens **1800** zur Ausfallsicherung einer Datenbank zwischen einer primären Bereitstellung und einer sekundären Bereitstellung. Das Verfahren **1800** kann durch ein oder mehrere Rechenbetriebsmittel, wie z. B. einen Betriebsmittelmanager **102**, eine Ausführungsplattform **112** und/oder einen Replikations- und Ausfallsicherungsmanager **228**, ausgeführt werden, wie hier offenbart wird.

[0141] Das Verfahren **1800** beginnt, wobei ein Rechenbetriebsmittel bei **1802** die in einer primären Bereitstellung gespeicherten Datenbankdaten repliziert, so dass die Datenbankdaten weiter in einer sekundären Bereitstellung gespeichert sind. Das Verfahren **1800** wird fortgesetzt, wobei ein Computerbetriebsmittel bei **1804** bestimmt, dass die primäre Bereitstellung nicht verfügbar ist. Die primäre Bereitstellung kann z. B. aufgrund eines Leistungsausfalls, eines Fehlers, der zu einer ungeeigneten Modifikation oder Löschung von Datenbankdaten in der primären Bereitstellung führt, eines Ausfalls des Datenzentrums, eines Ausfalls des Cloud-Anbieters, eines Fehlers, einer geplanten Ausfallzeit usw. nicht verfügbar sein. Das Verfahren **1800** wird fortgesetzt, wobei ein Computerbetriebsmittel bei **1806** in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, eine oder mehrere Transaktionen an den Datenbankdaten in der sekundären Bereitstellung ausführt.

Die eine oder die mehreren Transaktionen können eine Anweisung einer Datenmanipulationssprache (DML) enthalten, wie z. B. einen Einfüge-, Lösch-, Aktualisierungs- und/oder Verschmelzungsbefehl, eine Abfrage, die an den Datenbankdaten ausgeführt wird, usw. enthalten. Das Verfahren **1800** wird fortgesetzt, wobei ein Computerbetriebsmittel bei **1808** bestimmt, dass die primäre Bereitstellung nicht länger nicht verfügbar ist und in einen verfügbaren Zustand zurückgekehrt ist. Das Verfahren **1800** wird fortgesetzt, wobei ein Rechenbetriebsmittel bei **1810** in Reaktion auf das Bestimmen, dass die primäre Bereitstellung wieder verfügbar ist, die eine oder die mehreren Transaktionen an den Datenbankdaten zur primären Bereitstellung überträgt. In einer Ausführungsform werden die eine oder die mehreren Transaktionen über eine hybride Replikationsherangehensweise, wie sie hier offenbart wird, zu der primären Bereitstellung übertragen. In einer Ausführungsform werden die eine oder die mehreren Transaktionen, die in der sekundären Bereitstellung ausgeführt wurden, über ein Transaktionsprotokoll bestimmt, das in die Datenbank geschrieben wird, wie z. B. in den **Fig. 6-8** offenbart ist. In einer Ausführungsform wird die primäre Bereitstellung gemäß den in den **Fig. 9-11** bereitgestellten Offenbarungen aufgefrischt. Das Verfahren **1800** ist so, dass ein Computerbetriebsmittel, wie z. B. der Betriebsmittelmanager **102** und/oder die Ausführungsplattform **112**, die Abfragen an die Datenbankdaten in der primären Bereitstellung ausführt, wenn die primäre Bereitstellung verfügbar ist (siehe **1812**).

[0142] **Fig. 19** ist ein Blockschaltplan, der eine beispielhafte Rechenvorrichtung **1900** darstellt. In einigen Ausführungsformen wird die Rechenvorrichtung **1900** verwendet, um eines oder mehrere der hier erörterten Systeme und eine oder mehrere der hier erörterten Komponenten zu implementieren. Die Rechenvorrichtung **1900** kann es z. B. einem Anwender oder Manager ermöglichen, auf den Betriebsmittelmanager **1902** zuzugreifen. Ferner kann die Rechenvorrichtung **1900** mit irgendeinem der hier beschriebenen Systeme und irgendeiner der hier beschriebenen Komponenten wechselwirken. Entsprechend kann die Rechenvorrichtung **1900** verwendet werden, um verschiedene Prozeduren und Aufgaben, wie z. B. jene, die hier erörtert sind, auszuführen. Die Rechenvorrichtung **1900** kann als ein Server, ein Client oder irgendeine andere Rechenentität arbeiten. Die Rechenvorrichtung **1900** kann irgendeine einer umfassenden Vielfalt von Rechenvorrichtungen, wie z. B. ein Desktop-Computer, ein Notebook-Computer, ein Server-Computer, ein Handheld-Computer, ein Tablet-PC und dergleichen, sein.

[0143] Die Rechenvorrichtung **1900** enthält einen oder mehrere Prozessor(en) **1902**, eine oder mehrere Speichervorrichtung(en) **1904**, eine oder mehrere Schnittstelle(n) **1906**, eine oder mehrere Massenspeichervorrichtung(en) **1908** und eine oder mehrere Eingabe-/Ausgabe- (E/A-) Vorrichtung(en) **1910**, von denen alle an einen Bus **1912** gekoppelt sind. Der (die) Prozessor(en) **1902** enthält (enthalten) einen oder mehrere Prozessoren oder Controller, die die in der (den) Speichervorrichtung(en) **1904** und/oder der (den) Massenspeichervorrichtung(en) **1908** gespeicherten Anweisungen ausführen. Der (die) Prozessor(en) **1902** kann (können) außerdem verschiedene Typen von computerlesbaren Medien, wie z. B. einen Cache-Speicher, enthalten.

[0144] Die Speichervorrichtung(en) **1904** enthält (enthalten) verschiedene computerlesbare Medien, wie z. B. flüchtigen Speicher (z. B. Schreib-Lese-Speicher (RAM)) und/oder nichtflüchtigen Speicher (z. B. Festwertspeicher (ROM)). Die Speichervorrichtung(en) **1904** kann (können) außerdem wiederbeschreibbaren ROM, wie z. B. Flash-Speicher, enthalten.

[0145] Die Massenspeichervorrichtung(en) **1908** enthält (enthalten) verschiedene computerlesbare Medien, wie z. B. Magnetbänder, Magnetplatten, optische Platten, Festkörperspeicher (z. B. Flash-Speicher) usw. In der (den) Massenspeichervorrichtung(en) **1908** können außerdem verschiedene Laufwerke enthalten sein, um das Lesen von den und/oder das Schreiben in die verschiedenen computerlesbaren Medien zu ermöglichen. Die Massenspeichervorrichtung(en) **1908** enthält (enthalten) abnehmbare Medien und/oder nicht abnehmbare Medien.

[0146] Die E/A-Vorrichtung(en) **1910** enthält (enthalten) verschiedene Vorrichtungen, die es ermöglichen, dass Daten und/oder andere Informationen in die Computervorrichtung **1900** eingegeben oder aus der Computervorrichtung **1900** wiedergewonnen werden. Eine beispielhafte E/A-Vorrichtung(en) **1910** enthält (enthalten) Cursor-Steuervorrichtungen, Tastaturen, Tastenfelder, Mikrophone, Monitore oder andere Anzeigevorrichtungen, Lautsprecher, Drucker, Netzschnittstellenkarten, Modems, Objektive, CCDs oder andere Bildaufnahmevorrichtungen und dergleichen.

[0147] Die Schnittstelle(n) **1906** enthält (enthalten) verschiedene Schnittstellen, die es der Rechenvorrichtung **1900** ermöglichen, mit anderen Systemen, Vorrichtungen oder Rechenumgebungen zu wechselwirken. Eine beispielhafte Schnittstelle(n) **1906** enthält (enthalten) irgendeine Anzahl verschiedener Netzschnittstellen, wie z. B. Schnittstellen zu lokalen Netzen (LANs), Weitbereichsnetzen (WANs), drahtlosen Netzen und dem Internet.

[0148] Der Bus **1912** ermöglicht es dem (den) Prozessor(en) **1902**, der (den) Speichervorrichtung(en) **1904**, der (den) Schnittstelle(n) **1906**, der (den) Massenspeichervorrichtung(en) **1908** und der (den) E/A-Vorrichtung (en) **1910**, sowohl miteinander als auch mit anderen Vorrichtungen oder Komponenten, die an den Bus **1912** gekoppelt sind, zu kommunizieren. Der Bus **1912** repräsentiert einen oder mehrere von mehreren Typen von Busstrukturen, wie z. B. einen Systembus, einen PCI-Bus, einen IEEE 1394-Bus, einen USB-Bus usw.

[0149] Für Veranschaulichungszwecke sind Programme und andere ausführbare Programmkomponenten hier als diskrete Blöcke gezeigt, obwohl erkannt wird, dass sich derartige Programme und Komponenten zu verschiedenen Zeiten in verschiedenen Speicherkomponenten der Rechenvorrichtung **1900** befinden können und durch den (die) Prozessor(en) **1902** ausgeführt werden. Alternativ können die hier beschriebenen Systeme und Prozeduren in Hardware oder einer Kombination aus Hardware, Software und/oder Firmware implementiert sein. Es können z. B. eine oder mehrere anwendungsspezifische integrierte Schaltungen (ASICs) programmiert sein, um eines oder mehrere der Systeme und eine oder mehrere der Prozeduren, die hier beschrieben sind, auszuführen. Es ist vorgesehen, dass der Begriff „Modul“, wie er hier verwendet wird, die Implementierungsvorrichtung zum Ausführen eines Prozesses, wie z. B. durch Hardware oder eine Kombination aus Hardware, Software und/oder Firmware, für die Zwecke des Ausführens aller oder Teile der Abfrageoperationen übermittelt.

Beispiele

[0150] Die folgenden Beispiele betreffen weitere Ausführungsformen.

[0151] Das Beispiel 1 ist ein System. Das System enthält Mittel zum Replizieren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind. Das System enthält Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist. Das System enthält Mittel zum Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist. Das System enthält Mittel zum Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist. Das System enthält Mittel zum Übertragen der einen oder der mehrerer Transaktionen an den Datenbankdaten zu der primären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist. Das System enthält Mittel zum Ausführen von Abfragen an den Datenbankdaten in der primären Bereitstellung, wenn die primäre Bereitstellung verfügbar ist.

[0152] Das Beispiel 2 ist ein System wie im Beispiel 1, das ferner Mittel zum Ausführen neuer Transaktionen an den Datenbankdaten in der primären Bereitstellung und der sekundären Bereitstellung umfasst, wenn sowohl die primäre als auch die sekundäre Bereitstellung verfügbar sind.

[0153] Das Beispiel 3 ist ein System wie in einem der Beispiele 1-2, das ferner Mittel zum Verlagern des Ausführens von Abfragen an den Datenbankdaten von der primären Bereitstellung zu der sekundären Bereitstellung für eine Zeitdauer, in der die primäre Bereitstellung nicht verfügbar ist, umfasst.

[0154] Das Beispiel 4 ist ein System wie in einem der Beispiele 1-3, wobei sich die primäre Bereitstellung und die sekundäre Bereitstellung an verschiedenen geographischen Orten befinden.

[0155] Das Beispiel 5 ist ein System wie in einem der Beispiele 1-4, wobei die primäre Bereitstellung und die sekundäre Bereitstellung durch verschiedene cloud-basierte Speicheranbieter bereitgestellt werden.

[0156] Das Beispiel 6 ist ein System wie in einem der Beispiele 1-5, das ferner Mittel zum Bereitstellen einer Benachrichtigung an ein den Datenbankdaten zugeordnetes Konto umfasst, wenn sich ein Verfügbarkeitsstatus entweder der primären Bereitstellung oder der sekundären Bereitstellung geändert hat.

[0157] Das Beispiel 7 ist ein System wie in einem der Beispiele 1-6, das ferner Mittel zum Anhängen an eine anwenderdefinierte maximal akzeptable Zeitdauer, damit die zweite Bereitstellung für das Ausführen von Abfragen an den Datenbankdaten verfügbar wird, nachdem bestimmt worden ist, dass die primäre Bereitstellung nicht verfügbar ist, umfasst.

[0158] Das Beispiel 8 ist ein System wie in einem der Beispiele 1-7, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen eines oder mehrerer des Folgenden umfasst: in der primären Bereitstellung ist ein Leistungsausfall aufgetreten, es ist ein Fehler aufgetreten, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung geführt hat, in

der primären Bereitstellung ist ein Datenzentrumsausfall aufgetreten, ein Cloud-Anbieter der primären Bereitstellung hat einen Ausfall erfahren, in der primären Bereitstellung ist ein Fehler aufgetreten, oder die primäre Bereitstellung wird einer geplanten Ausfallzeit unterzogen.

[0159] Das Beispiel 9 ist ein System wie in einem der Beispiele 1-8, das ferner Mittel zum Anhängen an eine anwenderdefinierte maximale Anzahl von Datenbanktransaktionen umfasst, deren Verlust eine Anwendung tolerieren kann, wenn Datenbankoperationen in Reaktion darauf, dass die primäre Bereitstellung nicht verfügbar wird, von der primären Bereitstellung zu der sekundären Bereitstellung verlagert werden.

[0160] Das Beispiel 10 ist ein System wie in einem der Beispiele 1-9, wobei die Mittel zum Replizieren der in der primären Bereitstellung gespeicherten Datenbankdaten konfiguriert sind, in Reaktion darauf, dass die primäre Bereitstellung nicht verfügbar wird, zu replizieren.

[0161] Das Beispiel 11 ist ein System wie in einem der Beispiele 1-10, das ferner Mittel zum Verlagern einer Kundenkontoverbindung in Reaktion darauf, dass die primäre Bereitstellung nicht verfügbar wird, von der primären Bereitstellung zu der sekundären Bereitstellung umfasst.

[0162] Das Beispiel 12 ist ein System wie in einem der Beispiele 1-11, wobei die Mittel zum Übertragen der einen oder der mehreren Transaktionen zur primären Bereitstellung konfiguriert sind, nur die eine oder die mehreren Transaktionen zu übertragen und keine Daten zu replizieren, die bereits in der primären Bereitstellung vorhanden waren, bevor die primäre Bereitstellung nicht verfügbar wurde.

[0163] Das Beispiel 13 ist ein System wie in einem der Beispiele 1-12, wobei die Mittel zum Übertragen der einen oder der mehreren Transaktionen zu der primären Bereitstellung konfiguriert sind, die eine oder die mehreren Transaktionen basierend auf einer globalen Dateikennung zu bestimmen, die angibt, welche Dateien in den Datenbankdaten aktualisiert worden sind, seit die primäre Bereitstellung nicht verfügbar wurde.

[0164] Das Beispiel 14 ist ein Verfahren. Das Verfahren enthält das Replizieren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind. Das Verfahren enthält in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, das Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung. Das Verfahren enthält in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist, das Übertragen der einen oder der mehreren Transaktionen an den Datenbankdaten zu der primären Bereitstellung. Das Verfahren ist so, dass die Abfragen an den Datenbankdaten in der primären Bereitstellung ausgeführt werden, wenn die primäre Bereitstellung verfügbar ist.

[0165] Das Beispiel 15 ist ein Verfahren wie im Beispiel 14, das ferner, wenn sowohl die primäre Bereitstellung als auch die sekundäre Bereitstellung verfügbar sind, das Ausführen neuer Transaktionen an den Datenbankdaten in der primären Bereitstellung und der sekundären Bereitstellung umfasst.

[0166] Das Beispiel 16 ist ein Verfahren wie in einem der Beispiele 14-15, das ferner in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, das Verlagern der Ausführung von Abfragen an den Datenbankdaten von der primären Bereitstellung zu der sekundären Bereitstellung für eine Zeitdauer, während der die primäre Bereitstellung nicht verfügbar ist, umfasst.

[0167] Das Beispiel 17 ist ein Verfahren wie in einem der Beispiele 14-16, wobei das Verlagern der Ausführung von Abfragen von der primären Bereitstellung zu der sekundären Bereitstellung innerhalb einer anwenderdefinierten maximal akzeptablen Zeitdauer erfolgt, damit die sekundäre Bereitstellung für das Ausführen von Abfragen verfügbar wird, nachdem bestimmt worden ist, dass die primäre Bereitstellung nicht verfügbar ist.

[0168] Das Beispiel 18 ist ein Verfahren wie in einem der Beispiele 14-17, das ferner das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, durch das Bestimmen eines oder mehrerer des Folgenden umfasst: in der primären Bereitstellung ist ein Leistungsausfall aufgetreten, es ist ein Fehler aufgetreten, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung geführt hat, in der primären Bereitstellung ist ein Datenzentrumsausfall aufgetreten, ein Cloud-Anbieter der primären Bereitstellung hat einen Ausfall erfahren, in der primären Bereitstellung ist ein Fehler aufgetreten, oder die primäre Bereitstellung wird einer geplanten Ausfallzeit unterzogen.

[0169] Das Beispiel 19 ist ein Prozessor, der programmierbar ist, Anweisungen auszuführen, die in nicht transitorischen computerlesbaren Speichermedien gespeichert sind, wobei die Anweisungen umfassen: Replizie-

ren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind; in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung; in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist, Übertragen der einen oder der mehreren Transaktionen an den Datenbankdaten zu der primären Bereitstellung; und während die primäre Bereitstellung verfügbar ist, Ausführen von Abfragen an den Datenbankdaten in der primären Bereitstellung.

[0170] Das Beispiel 20 ist ein Prozessor wie im Beispiel 19, wobei die Anweisungen ferner das Ausführen neuer Transaktionen an den Datenbankdaten in der primären Bereitstellung und der sekundären Bereitstellung umfassen, wenn sowohl die primäre Bereitstellung als auch die sekundäre Bereitstellung verfügbar sind.

[0171] Das Beispiel 21 ist ein Prozessor wie in einem der Beispiele 19-20, wobei die Anweisungen ferner in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, das Verlagern der Ausführung von Abfragen an den Datenbankdaten von der primären Bereitstellung zu der sekundären Bereitstellung für eine Zeitdauer, während der die primäre Bereitstellung nicht verfügbar ist, umfassen.

[0172] Das Beispiel 22 ist ein Prozessor wie in einem der Beispiele 19-21, wobei die Anweisungen ferner das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, durch das Bestimmen eines oder mehrerer des Folgenden umfassen: in der primären Bereitstellung ist ein Leistungsausfall aufgetreten, es ist ein Fehler aufgetreten, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung geführt hat, in der primären Bereitstellung ist ein Datenzentrumsausfall aufgetreten, ein Cloud-Anbieter der primären Bereitstellung hat einen Ausfall erfahren, in der primären Bereitstellung ist ein Fehler aufgetreten, oder die primäre Bereitstellung wird einer geplanten Ausfallzeit unterzogen.

[0173] Die hier beschriebenen Systeme und Verfahren ermöglichen, dass als ein Dienst Daten gespeichert werden und auf Daten zugegriffen wird, der von den Rechen- (oder Verarbeitungs-) Betriebsmitteln getrennt ist. Selbst wenn von der Ausführungsplattform keine Rechenbetriebsmittel zugewiesen worden sind, sind die Daten für ein virtuelles Lager verfügbar, ohne ein erneutes Laden der Daten von einer entfernten Datenquelle zu erfordern. Folglich sind die Daten unabhängig von der Zuweisung der den Daten zugeordneten Rechenbetriebsmittel verfügbar. Die beschriebenen Systeme und Verfahren sind mit jedem Typ von Daten nützlich. In speziellen Ausführungsformen sind die Daten in einem strukturierten, optimierten Format gespeichert. Das Entkoppeln des Datenspeicher-/-zugriffsdienstes von den Rechendiensten vereinfacht außerdem das gemeinsame Benutzen von Daten zwischen verschiedenen Anwendern und Gruppen. Wie hier erörtert ist, kann jedes virtuelle Lager auf irgendwelche Daten, für die es Zugriffsberechtigungen aufweist, sogar zum gleichen Zeitpunkt, zu dem andere virtuelle Lager auf die gleichen Daten zugreifen, zugreifen. Diese Architektur unterstützt das Ausführen von Abfragen, ohne dass echte Daten im lokalen Cache gespeichert werden. Die hier beschriebenen Systeme und Verfahren können eine transparente dynamische Datenbewegung ausführen, die bei Bedarf Daten von einer entfernten Speichervorrichtung in einer für den Anwender des Systems transparente Weise zu einem lokalen Cache bewegt. Ferner unterstützt diese Architektur das gemeinsame Benutzen von Daten ohne eine vorherige Datenbewegung, weil irgendein virtuelles Lager aufgrund des Entkoppelns des Datenspeicherdienstes vom Rechendienst auf irgendwelche Daten zugreifen kann.

[0174] Obwohl die vorliegende Offenbarung hinsichtlich bestimmter bevorzugter Ausführungsformen beschrieben worden ist, sind angesichts des Vorteils dieser Offenbarung andere Ausführungsformen für Durchschnittsfachleute auf dem Gebiet offensichtlich, einschließlich der Ausführungsformen, die nicht alle der hier dargelegten Vorteile und Merkmale bereitstellen, die sich außerdem innerhalb des Schutzumfangs dieser Offenbarung befinden. Es soll erkannte werden, dass andere Ausführungsformen verwendet werden können, ohne vom Schutzumfang der vorliegenden Offenbarung abzuweichen.

Schutzansprüche

1. System, das umfasst:
einen Speicher, um Datenbankdaten zu speichern; und
einen Prozessor, der betriebstechnisch mit dem Speicher gekoppelt ist, um:
die in einer primären Bereitstellung gespeicherten Datenbankdaten zu replizieren, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind;
zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist; und
eine oder mehrere Transaktionen an den Datenbankdaten in der sekundären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, auszuführen.

2. System nach Anspruch 1, wobei der Prozessor ferner:
zu bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist;
die eine oder die mehreren Transaktionen an den Datenbankdaten in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist, zu der primären Bereitstellung zu übertragen; und
Abfragen an den Datenbankdaten in der primären Bereitstellung auszuführen.

3. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass in der primären Bereitstellung ein Leistungsausfall aufgetreten ist.

4. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass ein Fehler, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung führt, aufgetreten ist.

5. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass in der primären Bereitstellung ein Datenzentrumsausfall aufgetreten ist.

6. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass ein Cloud-Anbieter der primären Bereitstellung einen Ausfall erfahren hat.

7. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass in der primären Bereitstellung ein Fehler aufgetreten ist.

8. System nach Anspruch 1, wobei der Prozessor, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, ferner bestimmt, dass die primäre Bereitstellung einer geplanten Ausfallzeit unterzogen wird.

9. Vorrichtung, die umfasst:
Mittel zum Replizieren von Datenbankdaten, die in einer primären Bereitstellung gespeichert sind, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind;
Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist; und
Mittel zum Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist.

10. Vorrichtung nach Anspruch 9, die ferner umfasst:
Mittel zum Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist;
Mittel zum Übertragen der einen oder der mehreren Transaktionen an den Datenbankdaten zu der primären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist; und
Mittel zum Ausführen von Abfragen an den Datenbankdaten in der primären Bereitstellung.

11. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass in der primären Bereitstellung ein Leistungsausfall aufgetreten ist.

12. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass ein Fehler, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung führt, aufgetreten ist.

13. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass in der primären Bereitstellung ein Datenzentrumsausfall aufgetreten ist.

14. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass ein Cloud-Anbieter der primären Bereitstellung einen Ausfall erfahren hat.

15. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass in der primären Bereitstellung ein Fehler aufgetreten ist.

16. Vorrichtung nach Anspruch 9, wobei die Mittel zum Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Mittel zum Bestimmen umfassen, dass die primäre Bereitstellung einer geplanten Ausfallzeit unterzogen wird.

17. Nicht transitorische computerlesbare Speichermedien, die Anweisungen umfassen, die, wenn sie durch einen Prozessor ausgeführt werden, den Prozessor veranlassen, um:
die in einer primären Bereitstellung gespeicherten Datenbankdaten zu replizieren, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind;
zu bestimmen, durch den Prozessor, dass die primäre Bereitstellung nicht verfügbar ist; und
eine oder mehrere Transaktionen an den Datenbankdaten in der sekundären Bereitstellung in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, auszuführen.

18. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei die Anweisungen den Prozessor ferner veranlassen, um:
zu bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist;
die eine oder die mehreren Transaktionen an den Datenbankdaten in Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist, zu der primären Bereitstellung zu übertragen; und
Abfragen an den Datenbankdaten in der primären Bereitstellung auszuführen.

19. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass in der primären Bereitstellung ein Leistungsausfall aufgetreten ist.

20. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass ein Fehler, der zu einer ungeeigneten Modifikation oder Löschung der Datenbankdaten in der primären Bereitstellung führt, aufgetreten ist.

21. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass in der primären Bereitstellung ein Datenzentrumsausfall aufgetreten ist.

22. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass ein Cloud-Anbieter der primären Bereitstellung einen Ausfall erfahren hat.

23. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass in der primären Bereitstellung ein Fehler aufgetreten ist.

24. Nicht transitorische computerlesbare Speichermedien nach Anspruch 17, wobei, um zu bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, die Anweisungen den Prozessor ferner veranlassen, zu bestimmen, dass die primäre Bereitstellung einer geplanten Ausfallzeit unterzogen wird.

Es folgen 19 Seiten Zeichnungen

Anhängende Zeichnungen

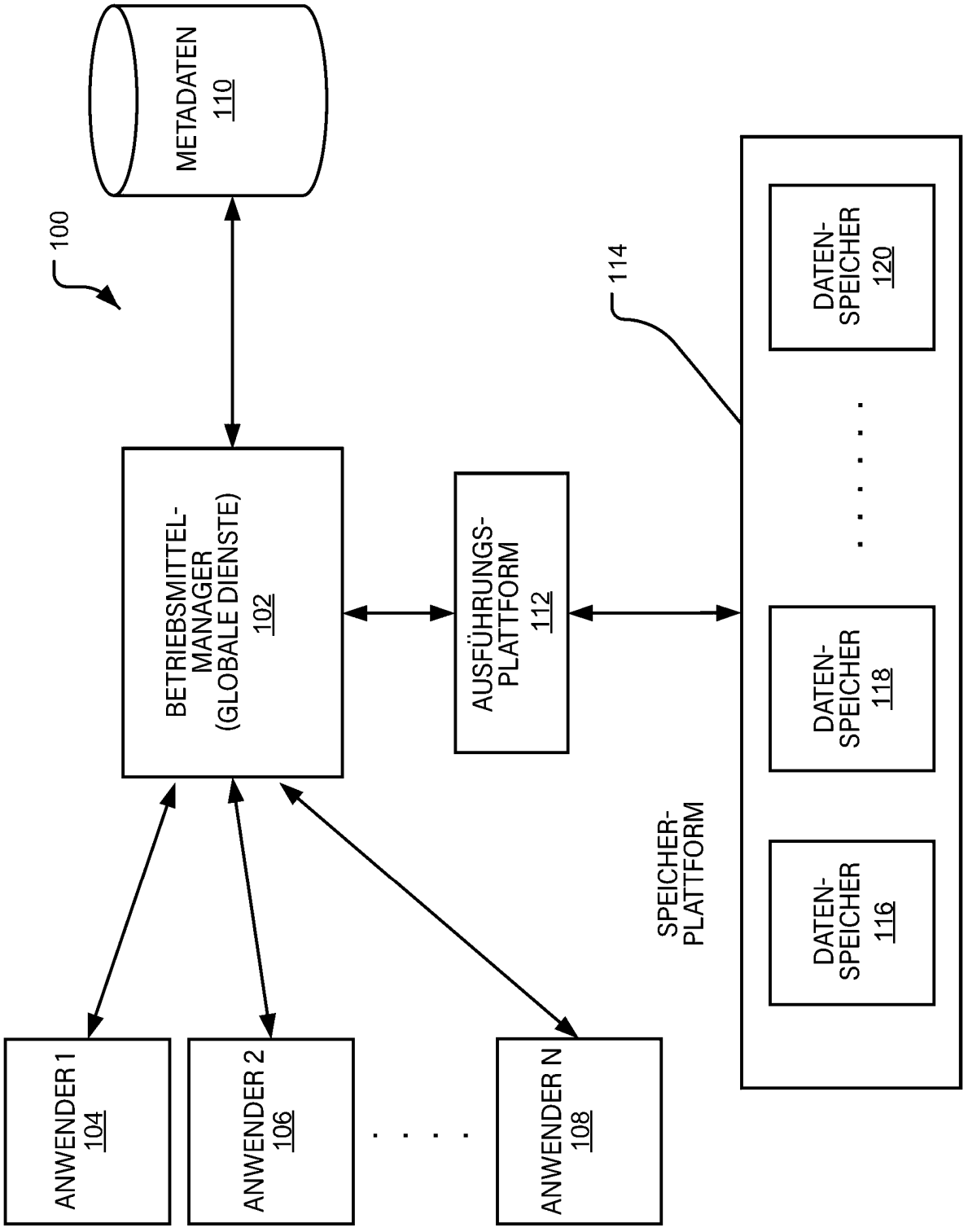


FIG. 1

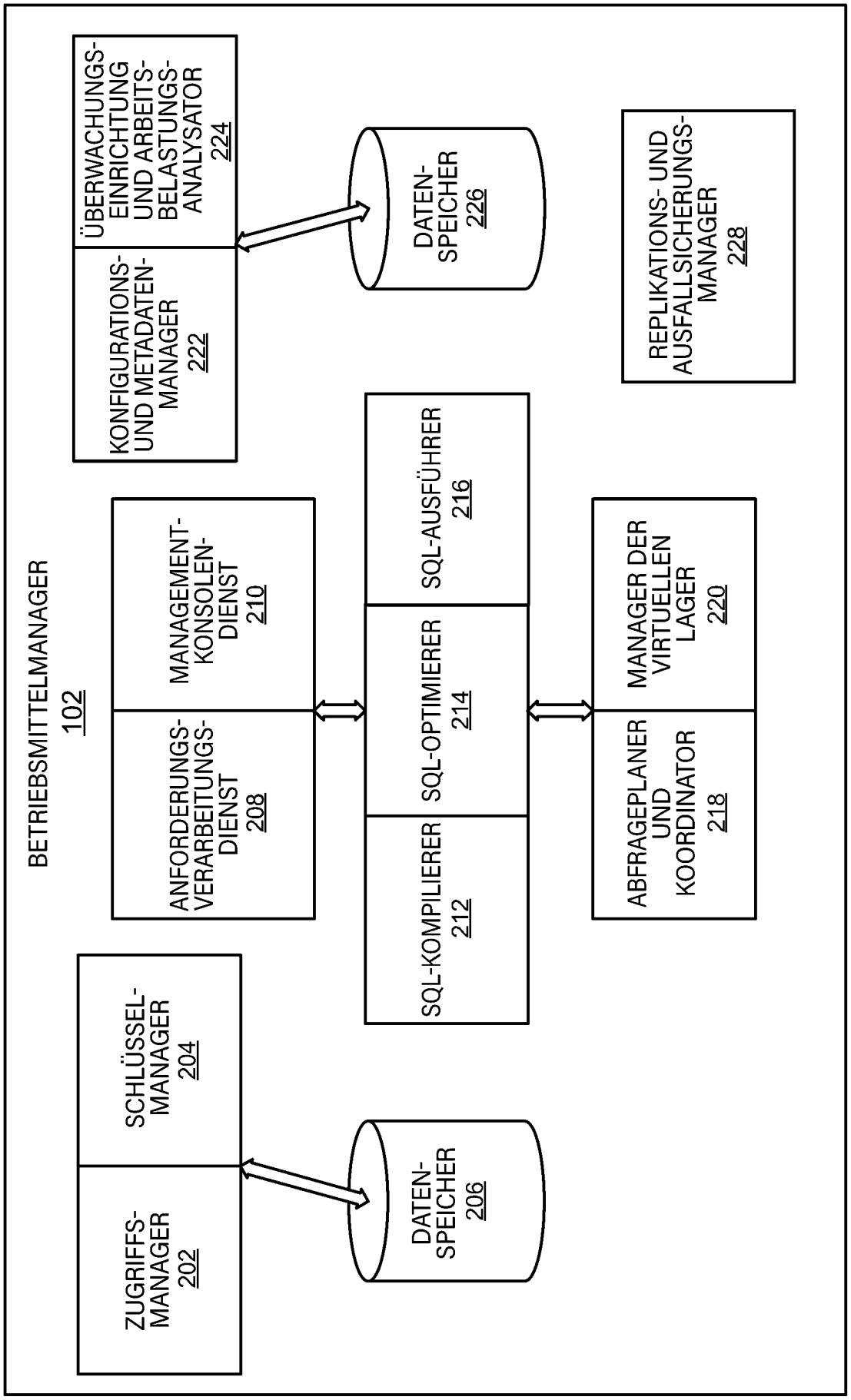

BETRIEBSMITTELMANAGER
102
ZUGRIFFS-MANAGER 202
SCHLÜSSEL-MANAGER 204
DATEN-SPEICHER 206
ANFORDERUNGS-VERARBEITUNGS-DIENST 208
MANAGEMENT-KONSOLEN-DIENST 210
SQL-KOMPILIERER 212
SQL-OPTIMIERER 214
SQL-AUSFÜHRER 216
ABFRAGEPLANER UND KOORDINATOR 218
MANAGER DER VIRTUELLEN LAGER 220
KONFIGURATIONS- UND METADATEN-MANAGER 222
ÜBERWACHUNGS-EINRICHTUNG UND ARBEITS-BELASTUNGS-ANALYSATOR 224
DATEN-SPEICHER 226
REPLIKATIONS- UND AUSFALLSICHERUNGS-MANAGER 228

FIG. 2

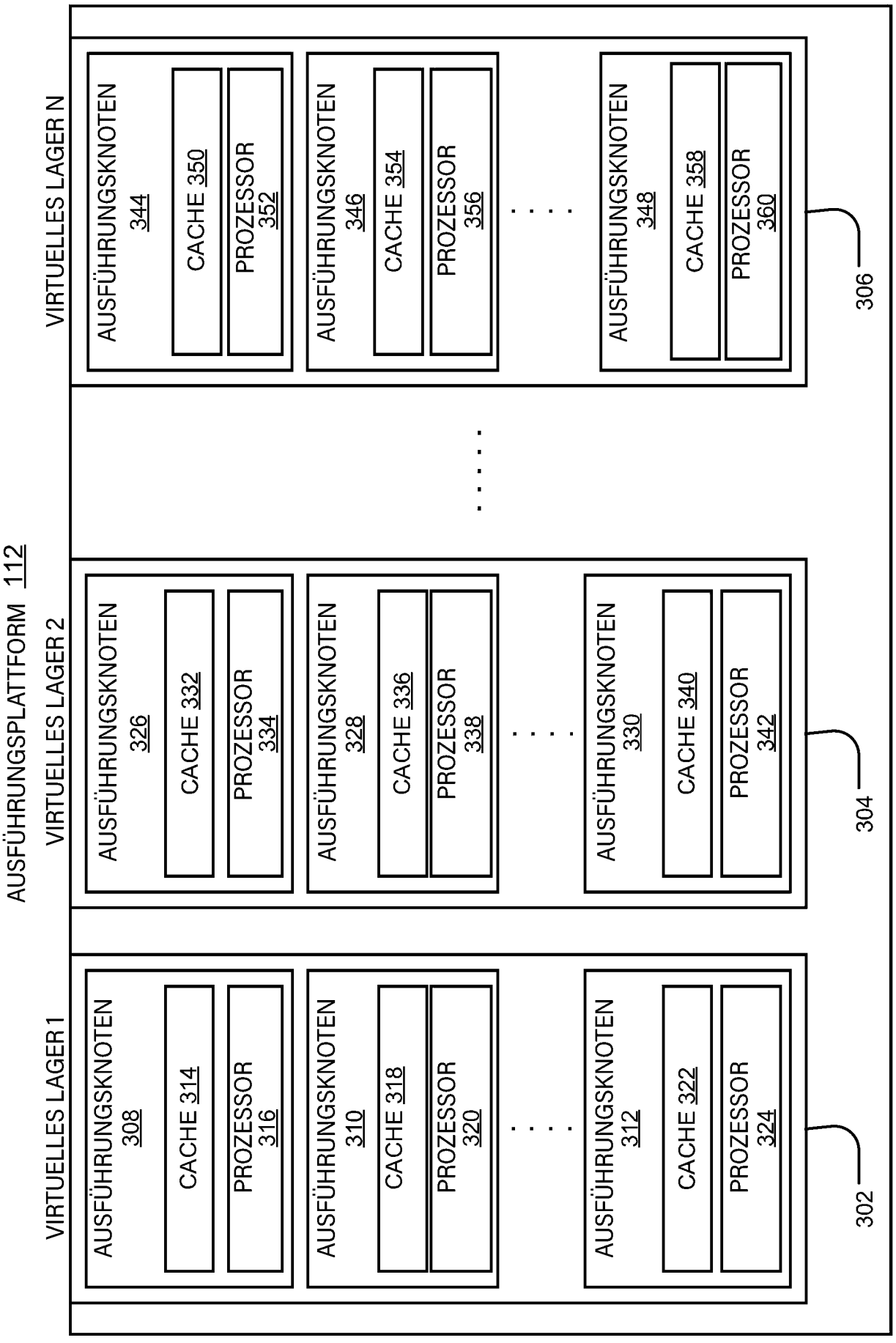

AUSFÜHRUNGSPLATTFORM 112
VIRTUELLES LAGER 1
VIRTUELLES LAGER 2
VIRTUELLES LAGER N
AUSFÜHRUNGSKNOTEN 308
CACHE 314
PROZESSOR 316
AUSFÜHRUNGSKNOTEN 310
CACHE 318
PROZESSOR 320
AUSFÜHRUNGSKNOTEN 312
CACHE 322
PROZESSOR 324
302
AUSFÜHRUNGSKNOTEN 326
CACHE 332
PROZESSOR 334
AUSFÜHRUNGSKNOTEN 328
CACHE 336
PROZESSOR 338
AUSFÜHRUNGSKNOTEN 330
CACHE 340
PROZESSOR 342
304
AUSFÜHRUNGSKNOTEN 344
CACHE 350
PROZESSOR 352
AUSFÜHRUNGSKNOTEN 346
CACHE 354
PROZESSOR 356
AUSFÜHRUNGSKNOTEN 348
CACHE 358
PROZESSOR 360
306

FIG. 3

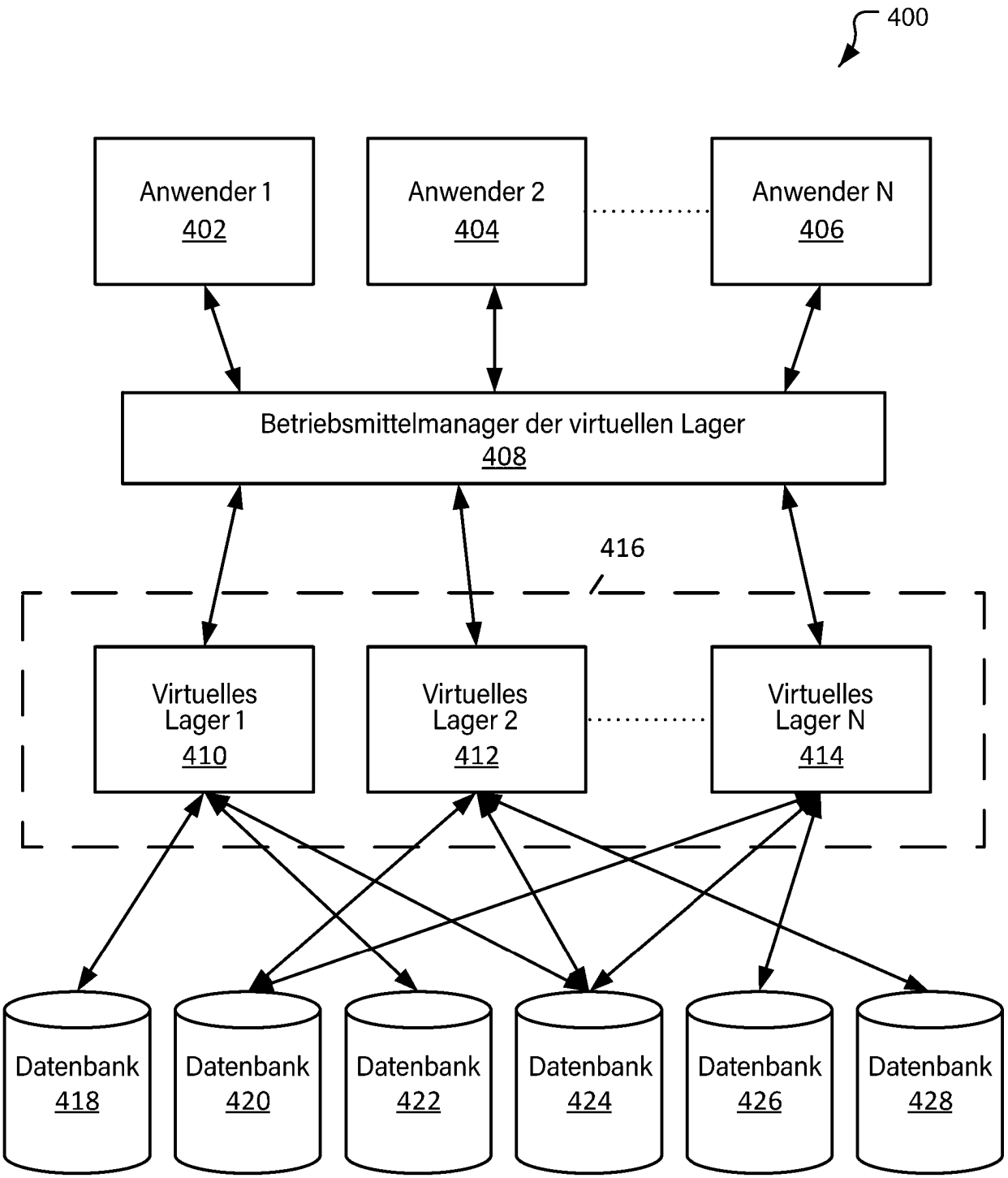

400
Anwender 1
402
Anwender 2
404
Anwender N
406
Betriebsmittelmanager der virtuellen Lager
408
416
Virtuelles
Lager 1
410
Virtuelles
Lager 2
412
Virtuelles
Lager N
414
Datenbank
418
Datenbank
420
Datenbank
422
Datenbank
424
Datenbank
426
Datenbank
428

FIG. 4

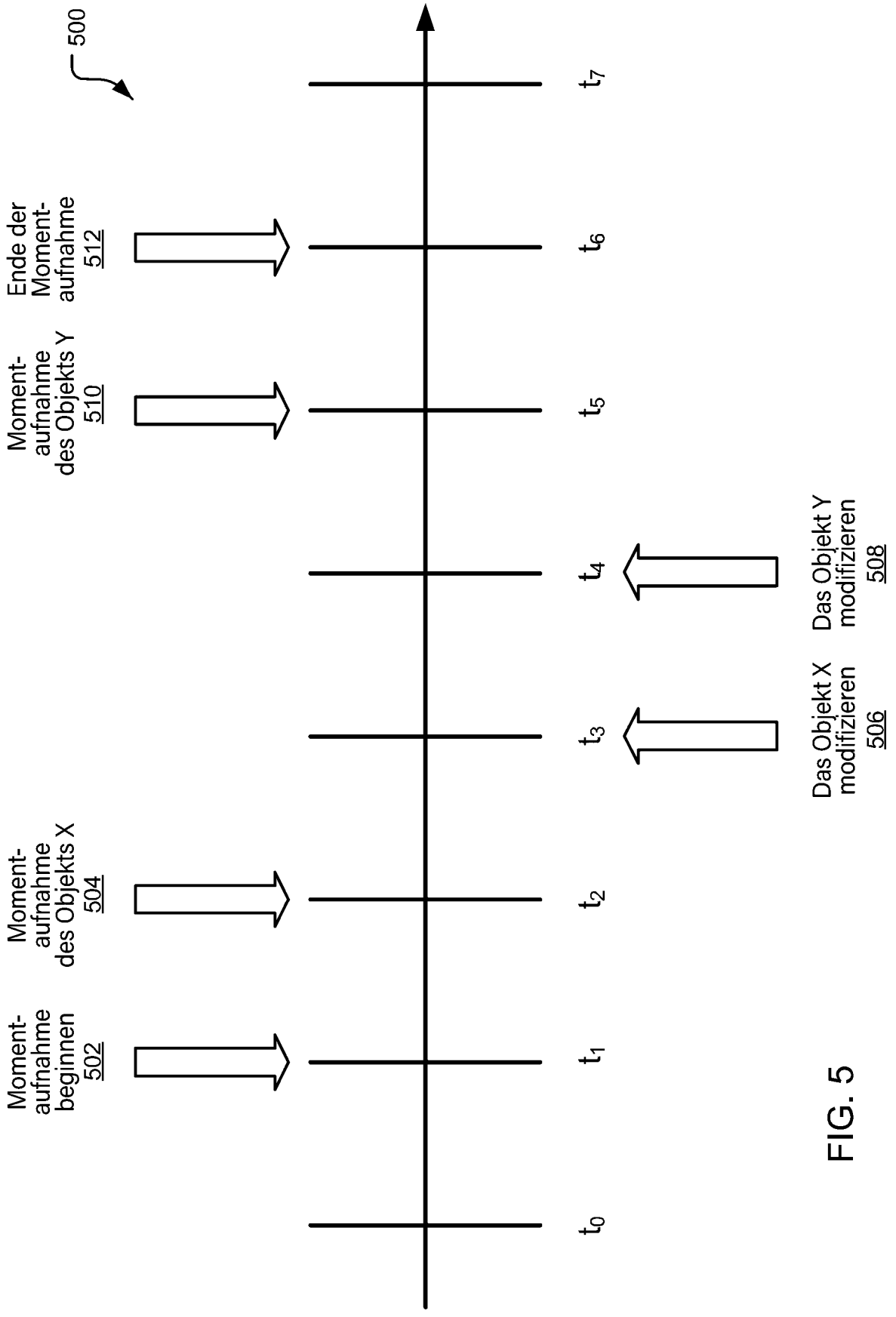

500
Moment-aufnahme beginnen 502
Moment-aufnahme des Objekts X 504
Moment-aufnahme des Objekts Y 510
Ende der Moment-aufnahme 512
Das Objekt X modifizieren 506
Das Objekt Y modifizieren 508
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_7

FIG. 5

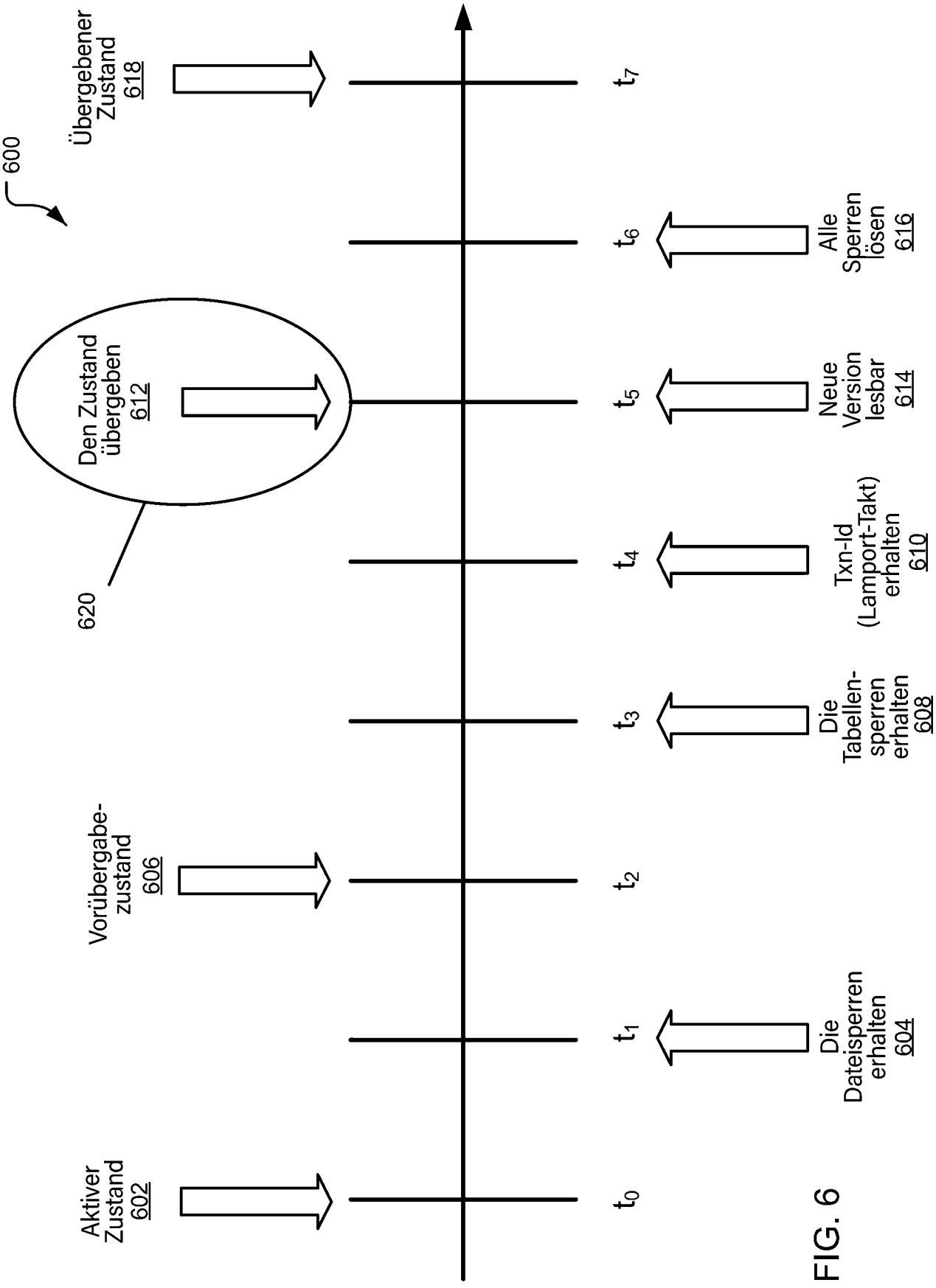

600
Übergebener Zustand 618
Den Zustand übergeben 612
620
Vorübergabe-zustand 606
Aktiver Zustand 602
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_7
Die Dateisperren erhalten 604
Die Tabellen-sperren erhalten 608
Txn-Id (Lamport-Takt) erhalten 610
Neue Version lesbar 614
Alle Sperren lösen 616

FIG. 6

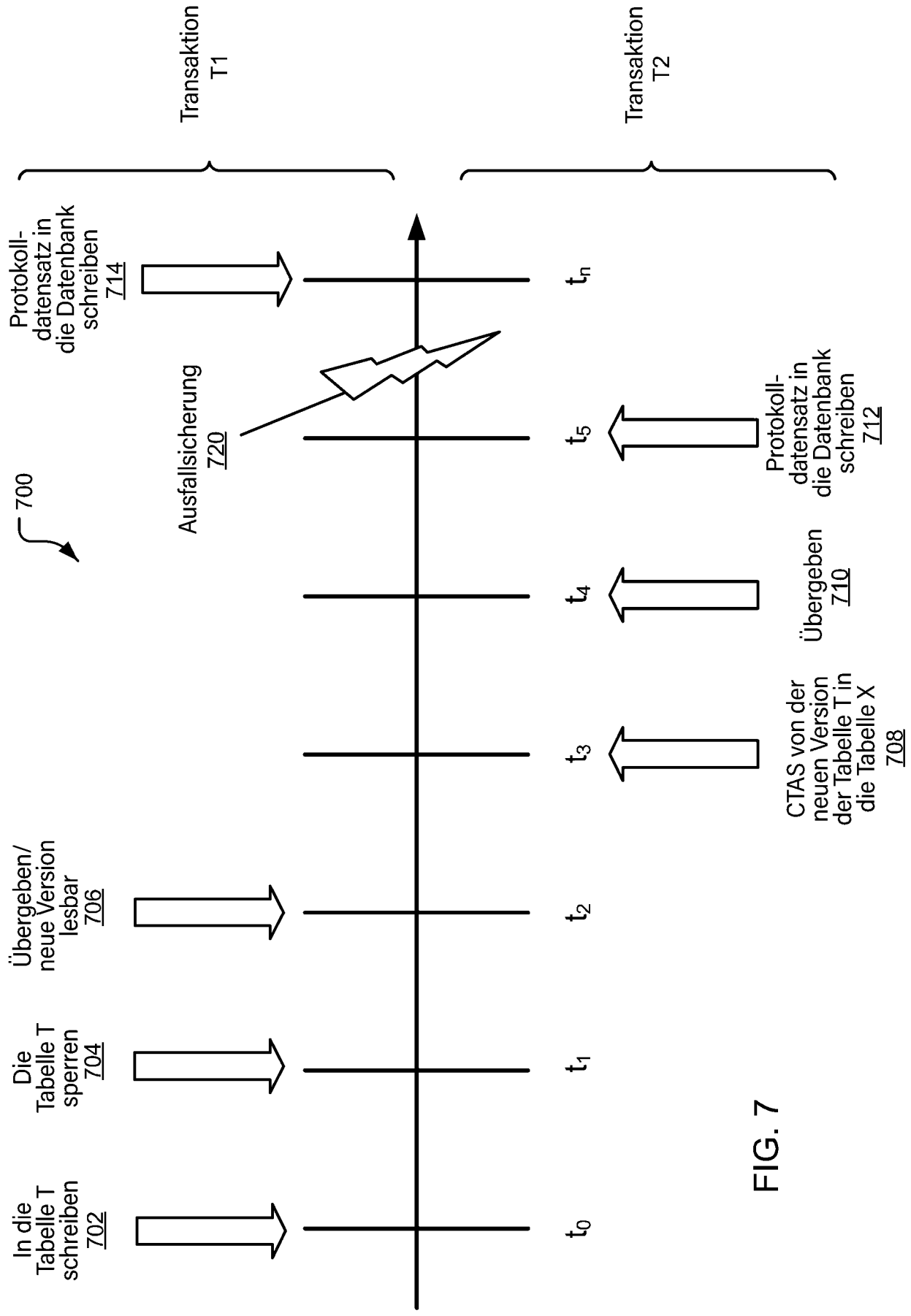

Transaktion T1
Transaktion T2
Protokoll-datensatz in die Datenbank schreiben 714
Ausfallsicherung 720
700
Übergeben/ neue Version lesbar 706
Die Tabelle T sperren 704
In die Tabelle T schreiben 702
t_0
t_1
t_2
t_3
t_4
t_5
t_n
CTAS von der neuen Version der Tabelle T in die Tabelle X 708
Übergeben 710
Protokoll-datensatz in die Datenbank schreiben 712

FIG. 7

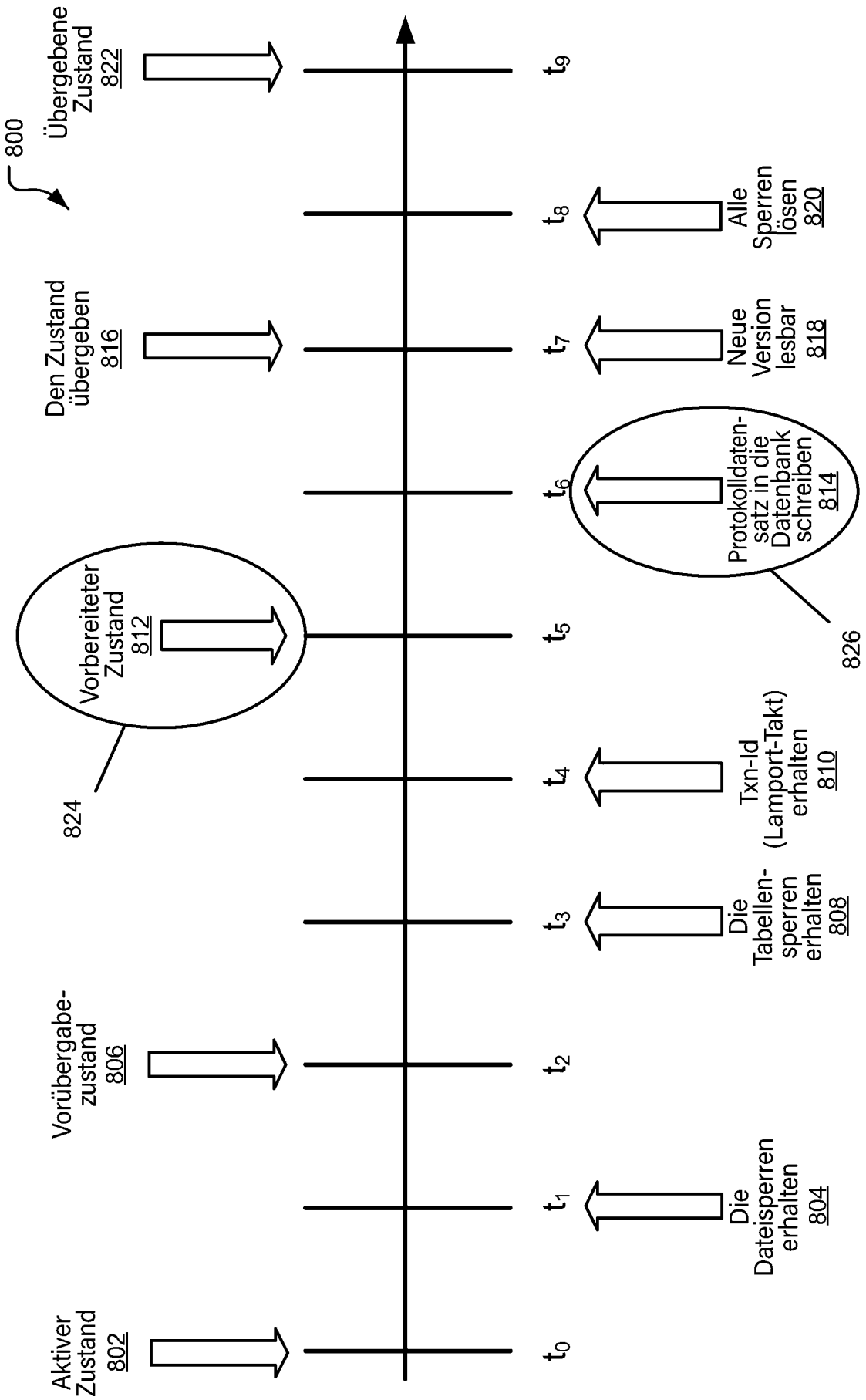

800
Aktiver Zustand 802
Vorübergabe-zustand 806
Vorbereiteter Zustand 812
824
Den Zustand übergeben 816
Übergebene Zustand 822
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_7
t_8
t_9
Die Dateisperren erhalten 804
Die Tabellen-sperren erhalten 808
Txn-Id (Lamport-Takt) erhalten 810
Protokolldaten-satz in die Datenbank schreiben 814
826
Neue Version lesbar 818
Alle Sperren lösen 820

FIG. 8

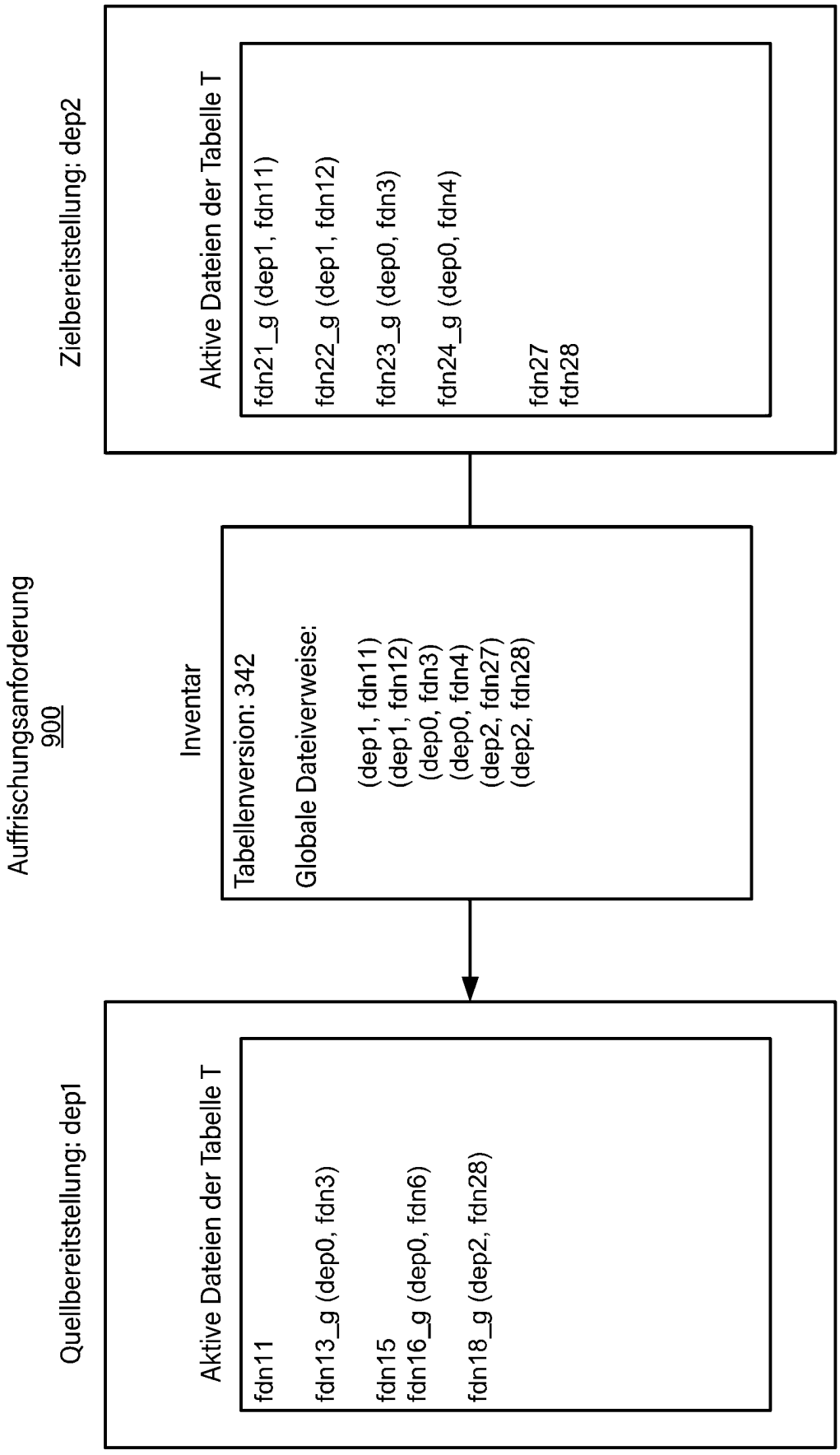

Auffrischungsanforderung
900
Quellbereitstellung: dep1
Aktive Dateien der Tabelle T
fdn11
fdn13_g (dep0, fdn3)
fdn15
fdn16_g (dep0, fdn6)
fdn18_g (dep2, fdn28)
Inventar
Tabellenversion: 342
Globale Dateiverweise:
(dep1, fdn11)
(dep1, fdn12)
(dep0, fdn3)
(dep0, fdn4)
(dep2, fdn27)
(dep2, fdn28)
Zielbereitstellung: dep2
Aktive Dateien der Tabelle T
fdn21_g (dep1, fdn11)
fdn22_g (dep1, fdn12)
fdn23_g (dep0, fdn3)
fdn24_g (dep0, fdn4)
fdn27
fdn28

FIG. 9

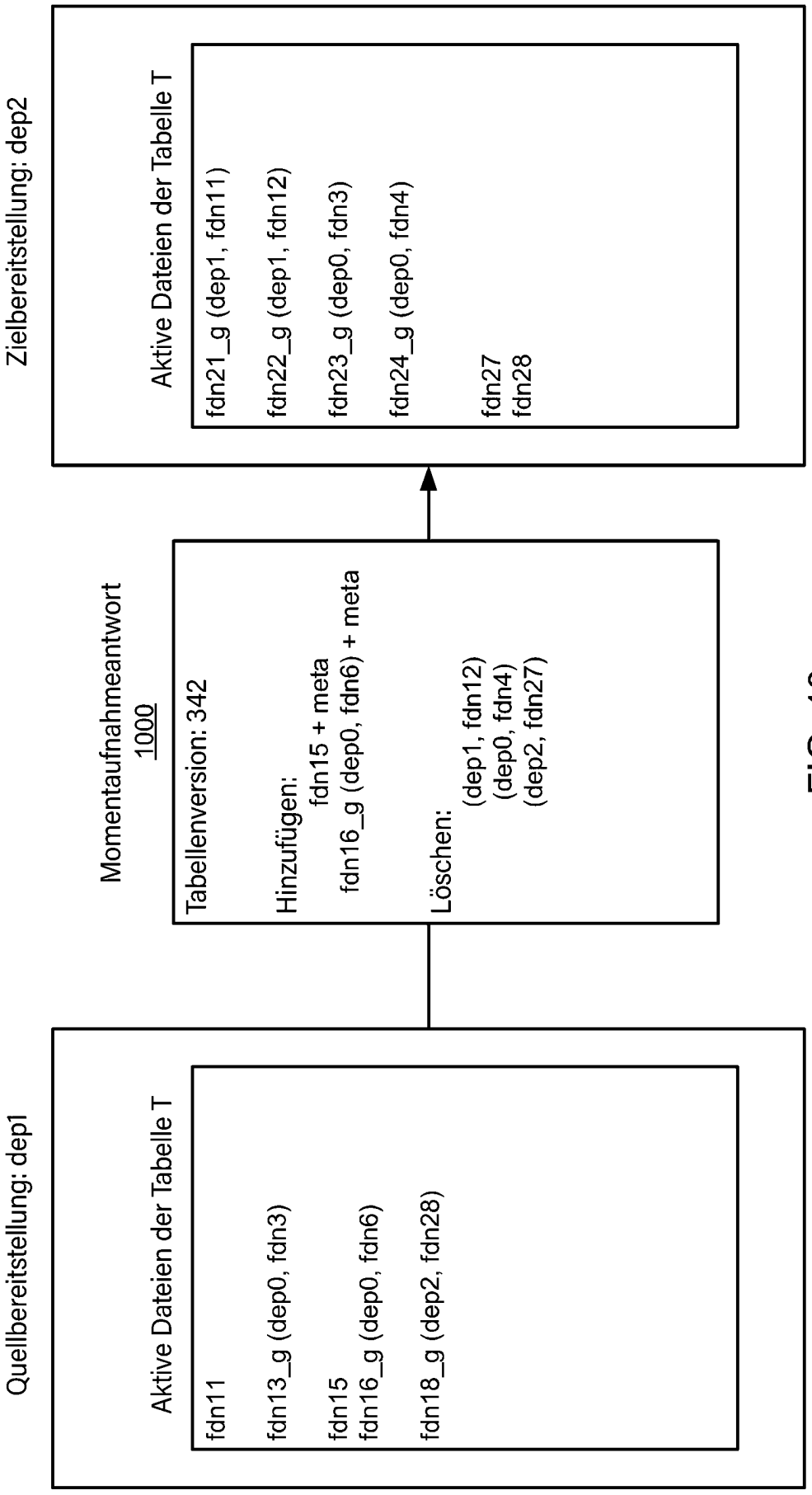

Quellbereitstellung: dep1
Aktive Dateien der Tabelle T
fdn11
fdn13_g (dep0, fdn3)
fdn15
fdn16_g (dep0, fdn6)
fdn18_g (dep2, fdn28)
Momentaufnahmeantwort
1000
Tabellenversion: 342
Hinzufügen:
fdn15 + meta
fdn16_g (dep0, fdn6) + meta
Löschen:
(dep1, fdn12)
(dep0, fdn4)
(dep2, fdn27)
Zielbereitstellung: dep2
Aktive Dateien der Tabelle T
fdn21_g (dep1, fdn11)
fdn22_g (dep1, fdn12)
fdn23_g (dep0, fdn3)
fdn24_g (dep0, fdn4)
fdn27
fdn28

FIG. 10

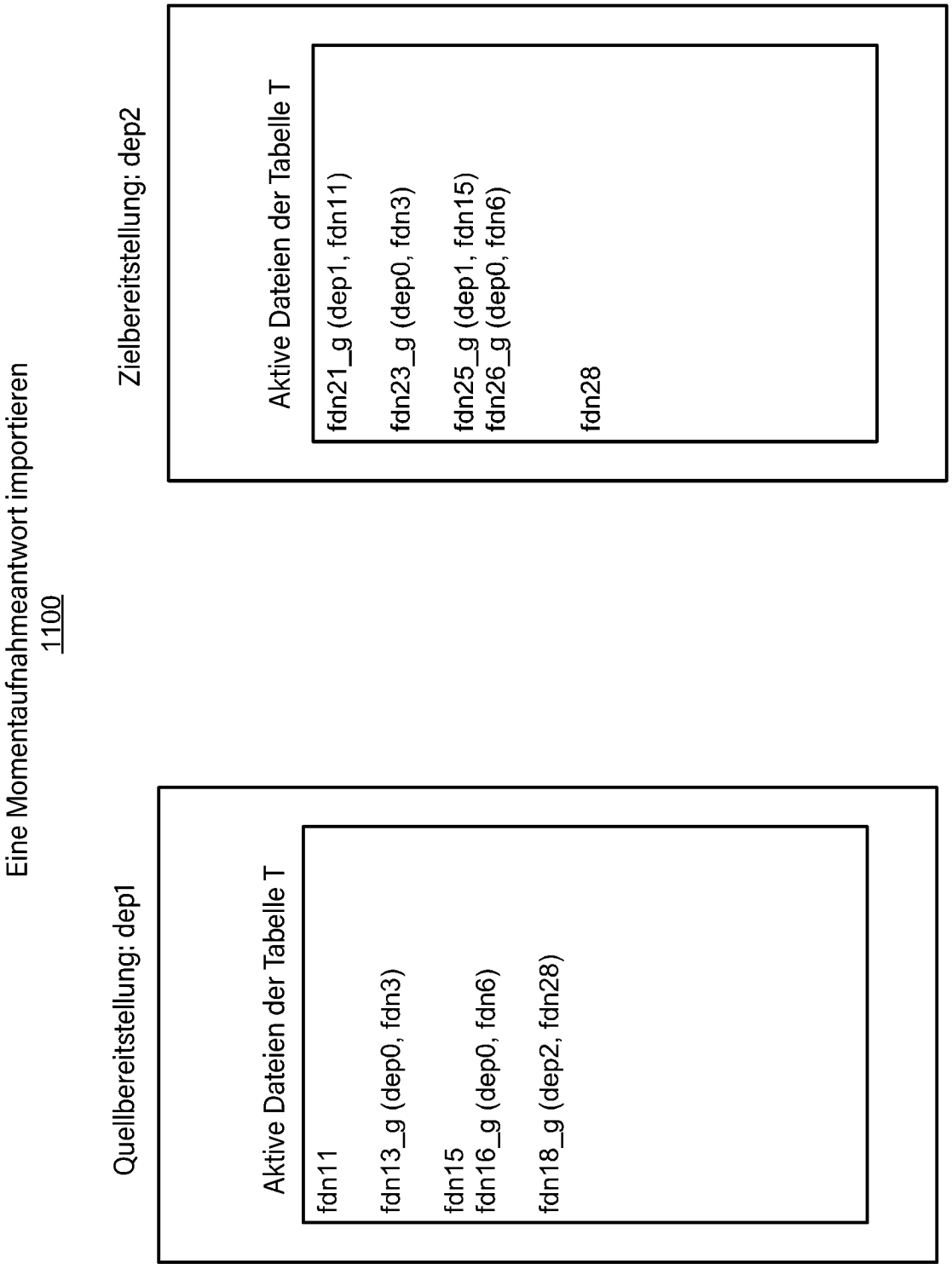

Eine Momentaufnahmeantwort importieren
1100
Quellbereitstellung: dep1
Aktive Dateien der Tabelle T
fdn11
fdn13_g (dep0, fdn3)
fdn15
fdn16_g (dep0, fdn6)
fdn18_g (dep2, fdn28)
Zielbereitstellung: dep2
Aktive Dateien der Tabelle T
fdn21_g (dep1, fdn11)
fdn23_g (dep0, fdn3)
fdn25_g (dep1, fdn15)
fdn26_g (dep0, fdn6)
fdn28

FIG. 11

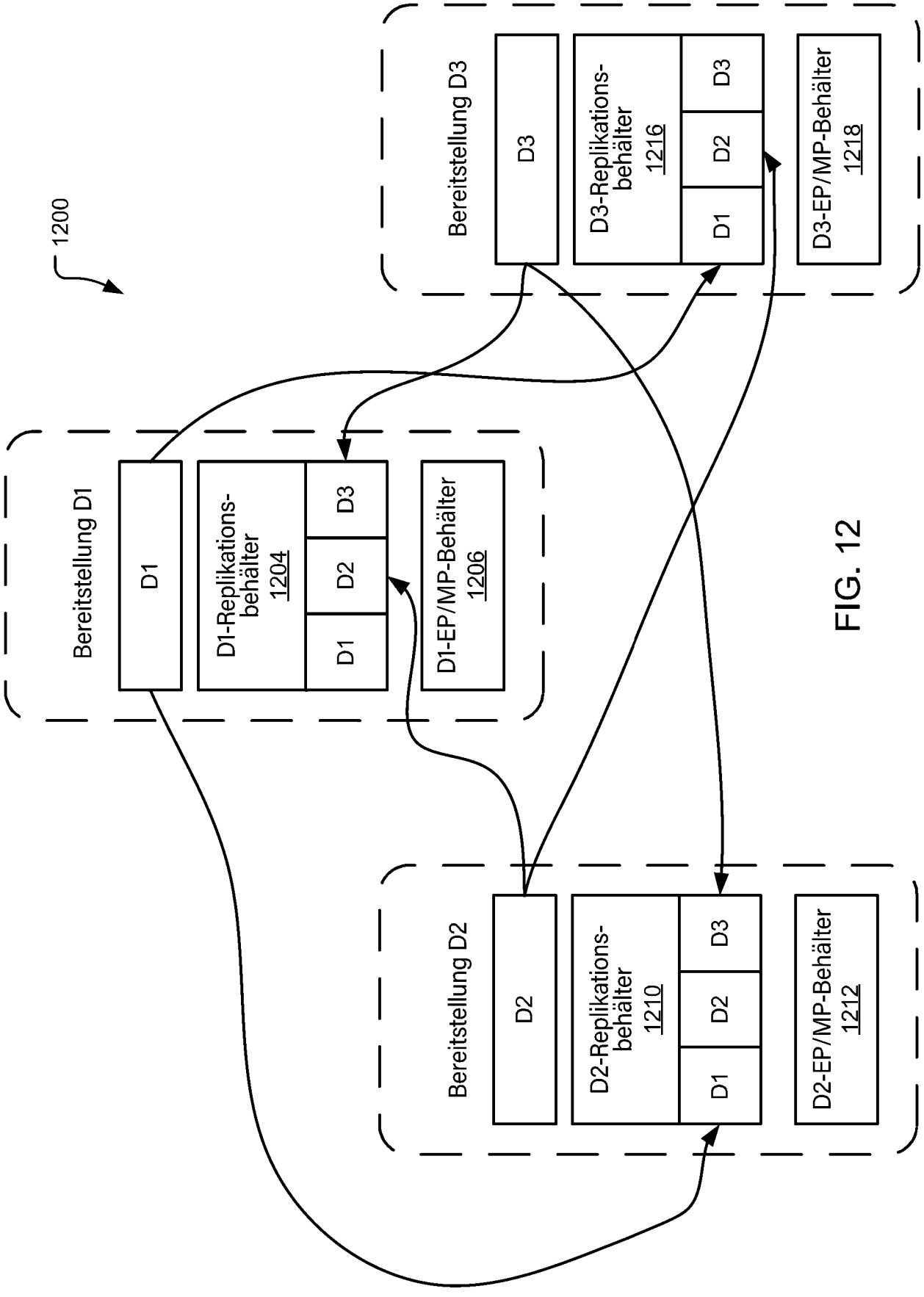

1200
Bereitstellung D1
D1
D1-Replikations-behälter 1204
D3
D2
D1
D1-EP/MP-Behälter 1206
Bereitstellung D3
D3
D3-Replikations-behälter 1216
D3
D2
D1
D3-EP/MP-Behälter 1218
Bereitstellung D2
D2
D2-Replikations-behälter 1210
D3
D2
D1
D2-EP/MP-Behälter 1212

FIG. 12

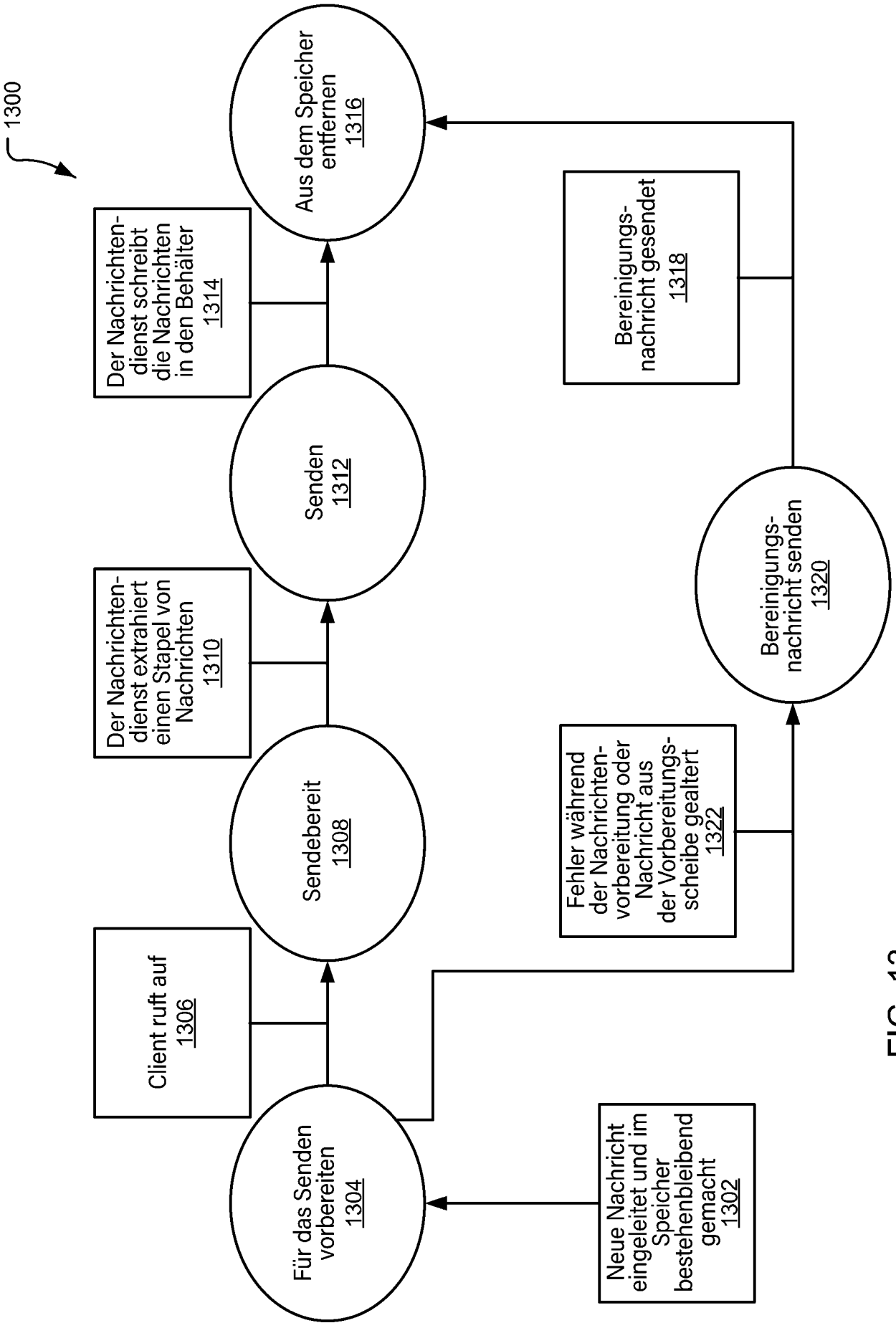

1300
Neue Nachricht eingeleitet und im Speicher bestehenbleibend gemacht 1302
Für das Senden vorbereiten 1304
Client ruft auf 1306
Sendebereit 1308
Der Nachrichten-dienst extrahiert einen Stapel von Nachrichten 1310
Senden 1312
Der Nachrichten-dienst schreibt die Nachrichten in den Behälter 1314
Aus dem Speicher entfernen 1316
Bereinigungs-nachricht gesendet 1318
Bereinigungs-nachricht senden 1320
Fehler während der Nachrichten-vorbereitung oder Nachricht aus der Vorbereitungs-scheibe gealtert 1322

FIG. 13

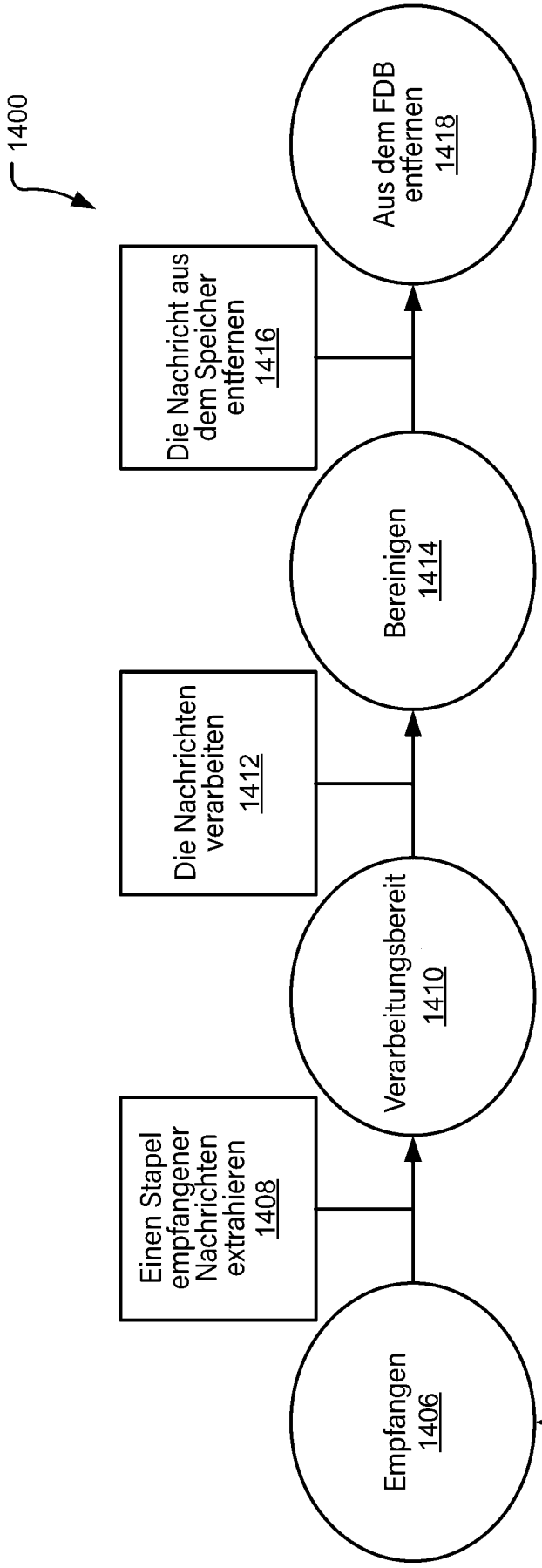

1400
Nachricht vom Replikationsbehälter heruntergeladen und deserialisiert 1402
Die Nachricht im FDB bestehen-bleibend machen 1404
Empfangen 1406
Einen Stapel empfangener Nachrichten extrahieren 1408
Verarbeitungsbereit 1410
Die Nachrichten verarbeiten 1412
Bereinigen 1414
Die Nachricht aus dem Speicher entfernen 1416
Aus dem FDB entfernen 1418

FIG. 14

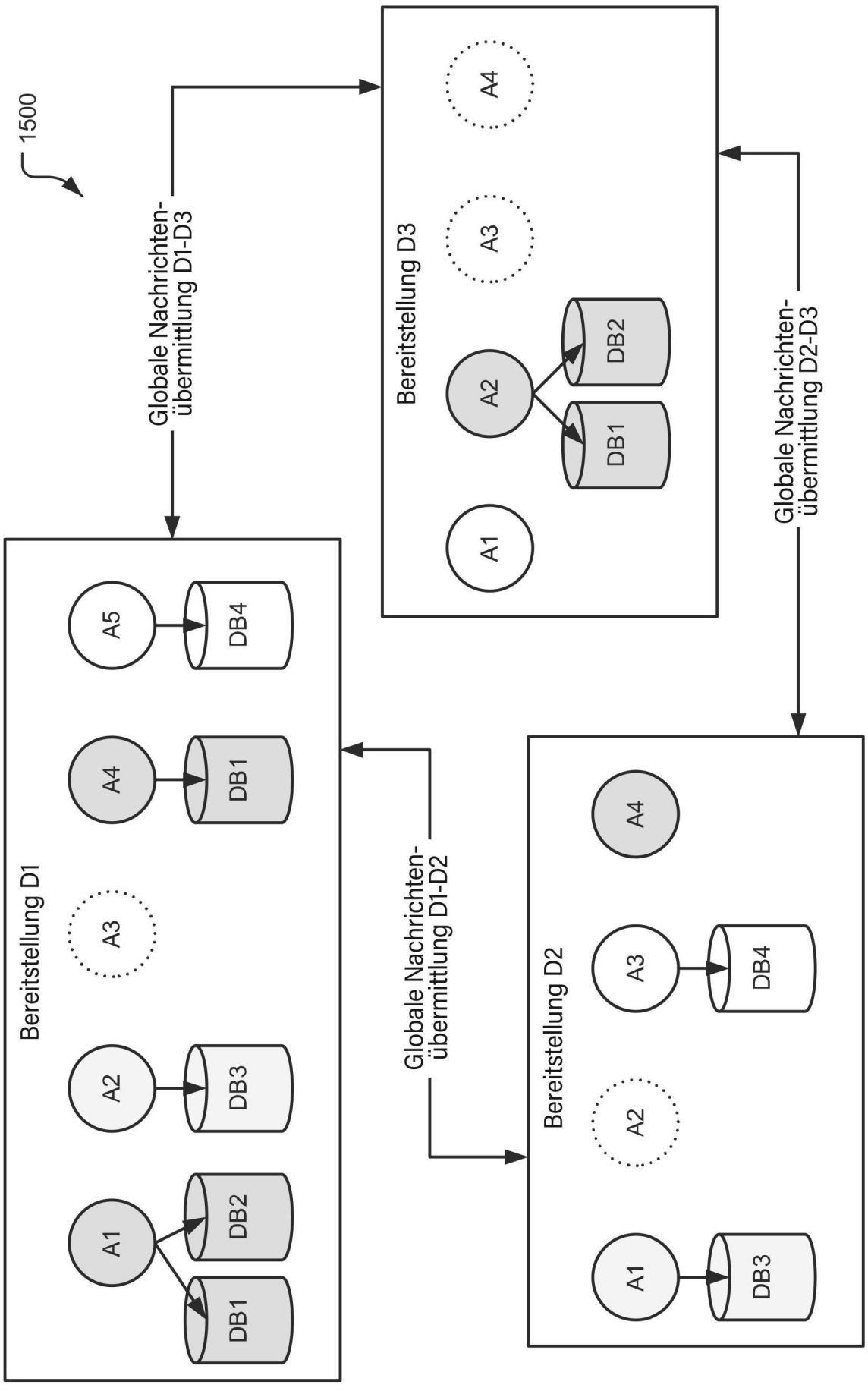

1500
Bereitstellung D1
A1
DB2
DB1
A2
DB3
A3
A4
DB1
A5
DB4
Bereitstellung D2
A1
DB3
A2
A3
DB4
A4
Bereitstellung D3
A1
A2
DB1
DB2
A3
A4
Globale Nachrichten-übermittlung D1-D2
Globale Nachrichten-übermittlung D1-D3
Globale Nachrichten-übermittlung D2-D3

FIG. 15

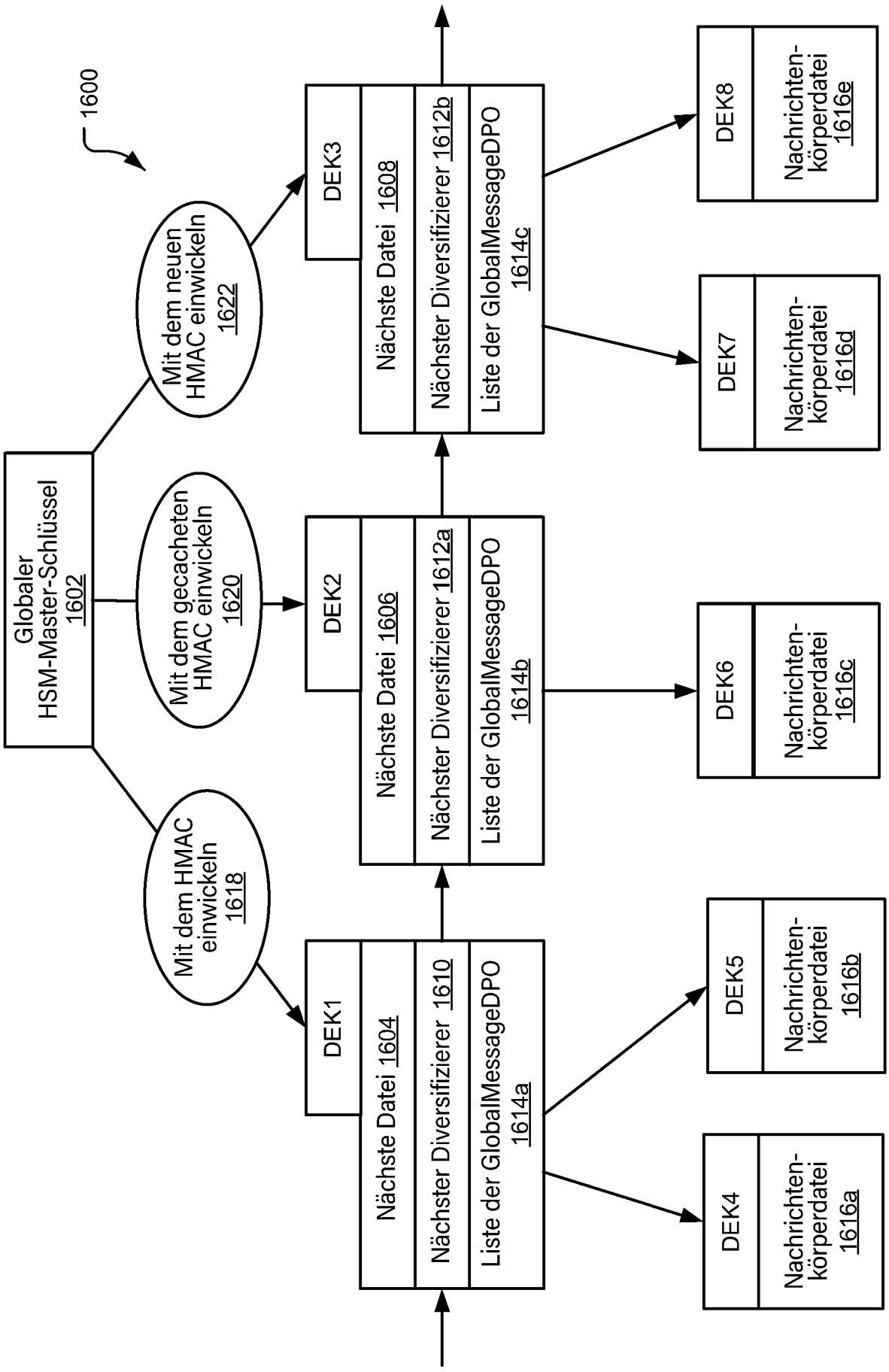

1600
Globaler HSM-Master-Schlüssel 1602
Mit dem HMAC einwickeln 1618
Mit dem gecacheten HMAC einwickeln 1620
Mit dem neuen HMAC einwickeln 1622
DEK1
Nächste Datei 1604
Nächster Diversifizierer 1610
Liste der GlobalMessageDPO 1614a
DEK2
Nächste Datei 1606
Nächster Diversifizierer 1612a
Liste der GlobalMessageDPO 1614b
DEK3
Nächste Datei 1608
Nächster Diversifizierer 1612b
Liste der GlobalMessageDPO 1614c
DEK4
Nachrichten-körperdatei 1616a
DEK5
Nachrichten-körperdatei 1616b
DEK6
Nachrichten-körperdatei 1616c
DEK7
Nachrichten-körperdatei 1616d
DEK8
Nachrichten-körperdatei 1616e

FIG. 16

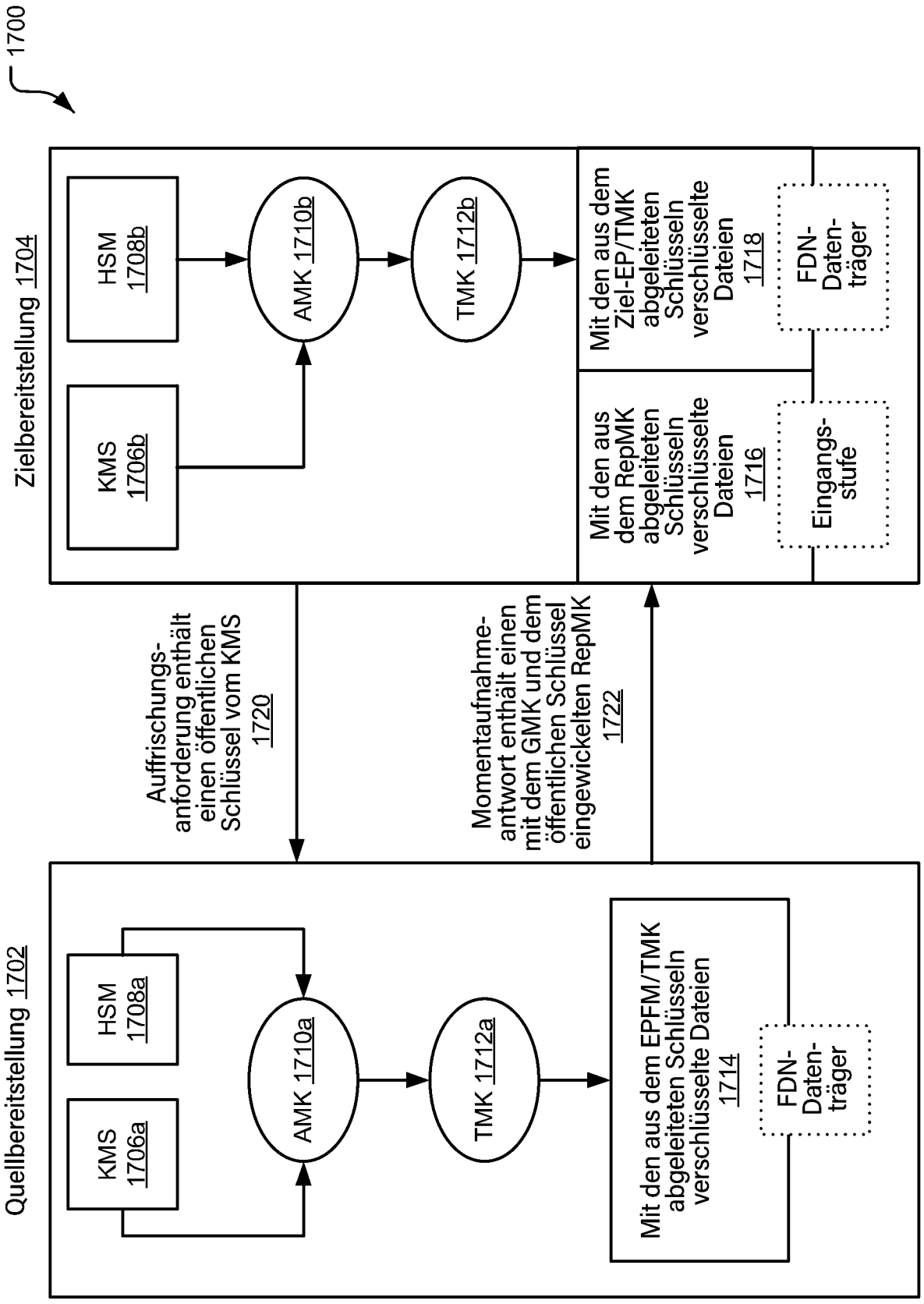

1700
Quellbereitstellung 1702
KMS 1706a
HSM 1708a
AMK 1710a
TMK 1712a
Mit den aus dem EPFM/TMK abgeleiteten Schlüsseln verschlüsselte Dateien 1714
FDN-Daten-träger
Auffrischungs-anforderung enthält einen öffentlichen Schlüssel vom KMS 1720
Momentaufnahme-antwort enthält einen mit dem GMK und dem öffentlichen Schlüssel eingewickelten RepMK 1722
Zielbereitstellung 1704
KMS 1706b
HSM 1708b
AMK 1710b
TMK 1712b
Mit den aus dem RepMK abgeleiteten Schlüsseln verschlüsselte Dateien 1716
Eingangs-stufe
Mit den aus dem Ziel-EP/TMK abgeleiteten Schlüsseln verschlüsselte Dateien 1718
FDN-Daten-träger

FIG. 17

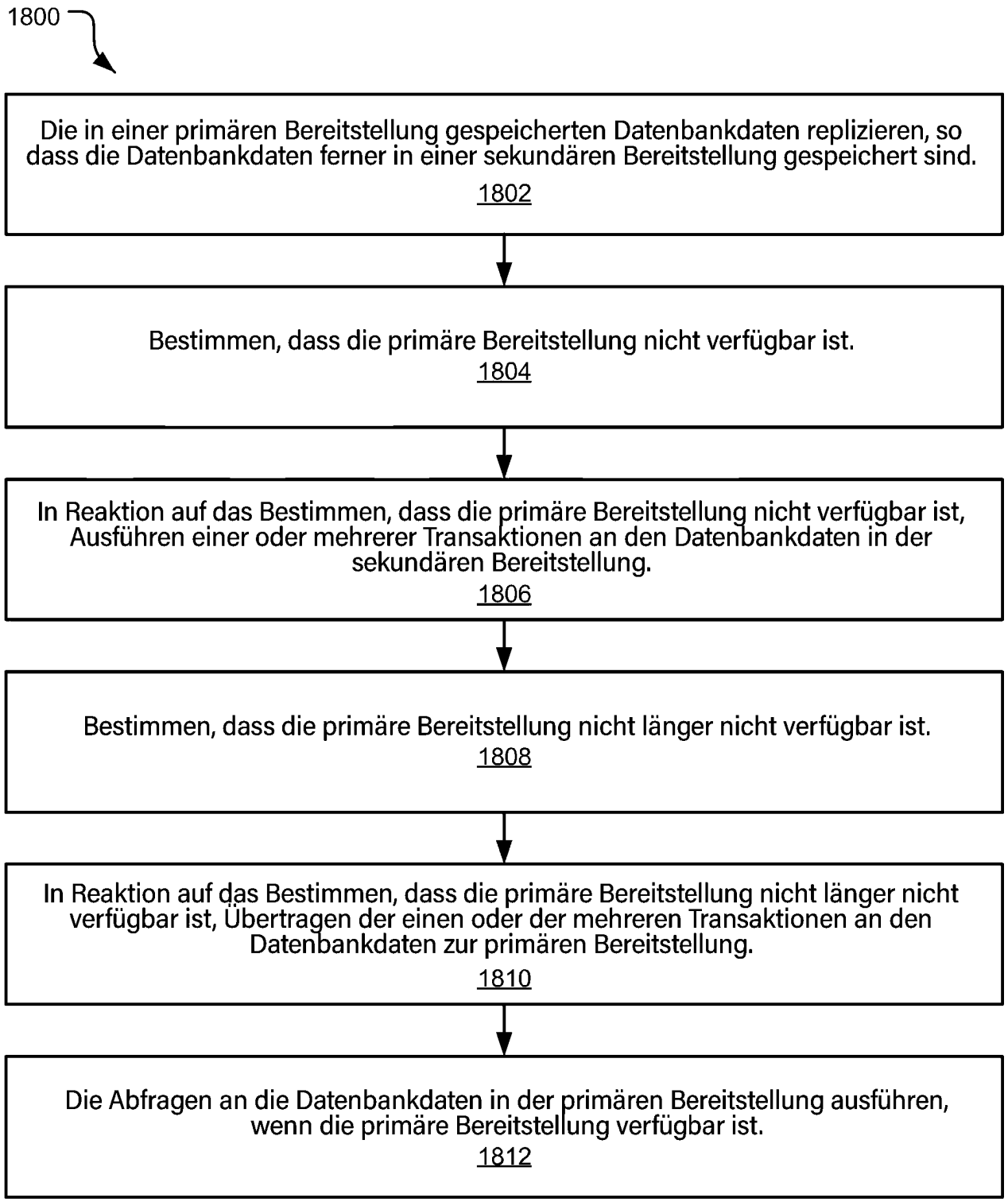

1800
Die in einer primären Bereitstellung gespeicherten Datenbankdaten replizieren, so dass die Datenbankdaten ferner in einer sekundären Bereitstellung gespeichert sind.
1802
Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist.
1804
In Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht verfügbar ist, Ausführen einer oder mehrerer Transaktionen an den Datenbankdaten in der sekundären Bereitstellung.
1806
Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist.
1808
In Reaktion auf das Bestimmen, dass die primäre Bereitstellung nicht länger nicht verfügbar ist, Übertragen der einen oder der mehreren Transaktionen an den Datenbankdaten zur primären Bereitstellung.
1810
Die Abfragen an die Datenbankdaten in der primären Bereitstellung ausführen, wenn die primäre Bereitstellung verfügbar ist.
1812

FIG. 18

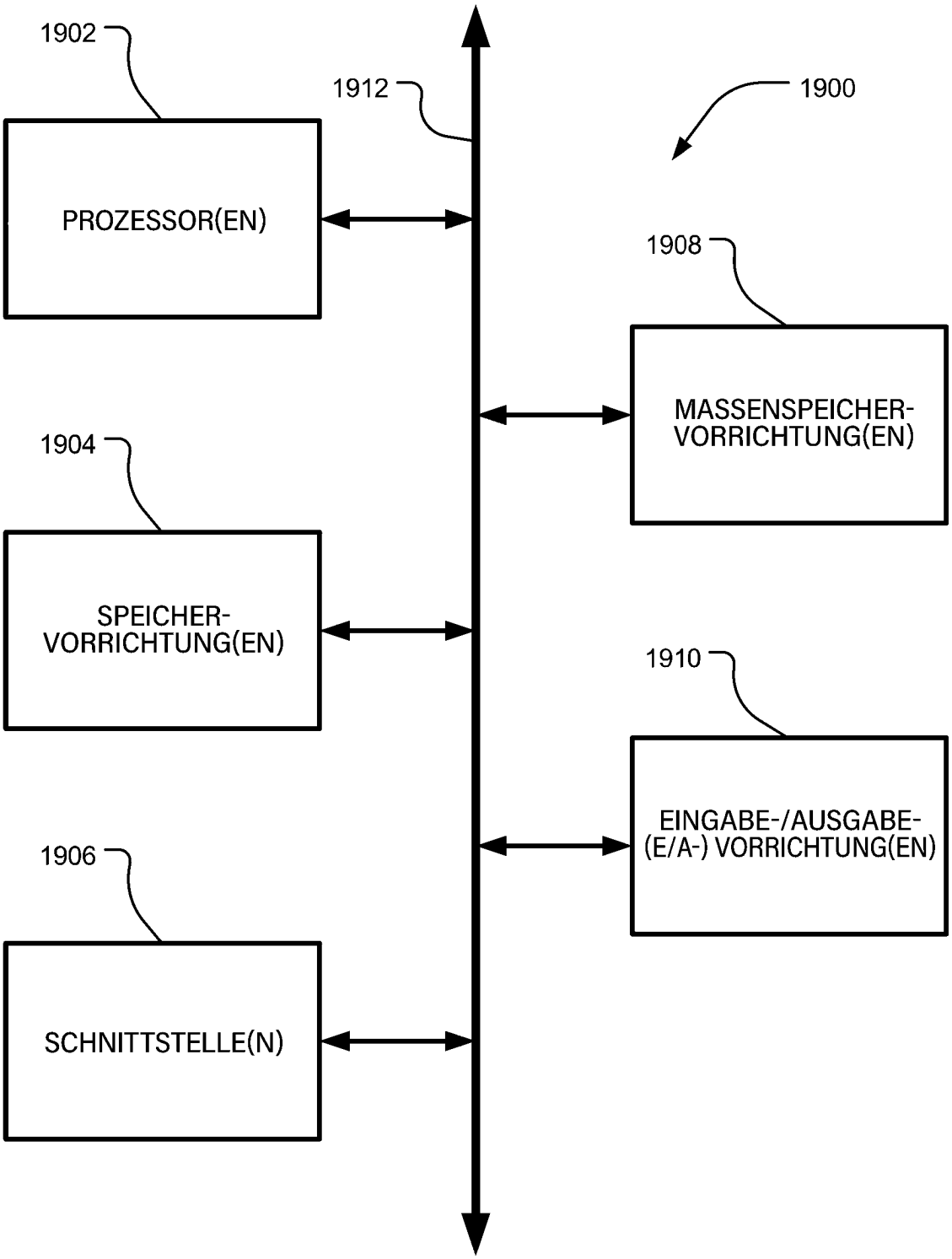


FIG. 19