



(10) **DE 11 2018 001 972 T5** 2019.12.24

(12)

## Veröffentlichung

der internationalen Anmeldung mit der  
(87) Veröffentlichungs-Nr.: **WO 2019/003029**  
in der deutschen Übersetzung (Art. III § 8 Abs. 2  
IntPatÜG)

(51) Int Cl.: **G06F 16/10** (2019.01)

(21) Deutsches Aktenzeichen: **11 2018 001 972.1**

(86) PCT-Aktenzeichen: **PCT/IB2018/054378**

(86) PCT-Anmeldetag: **14.06.2018**

(87) PCT-Veröffentlichungstag: **03.01.2019**

(43) Veröffentlichungstag der PCT Anmeldung  
in deutscher Übersetzung: **24.12.2019**

(30) Unionspriorität:

<b>15/636,770</b>	<b>29.06.2017</b>	<b>US</b>
<b>15/824,356</b>	<b>28.11.2017</b>	<b>US</b>

(74) Vertreter:

**Richardt Patentanwälte PartG mbB, 65185  
Wiesbaden, DE**

(71) Anmelder:

**International Business Machines Corporation,  
Armonk, N.Y., US**

(72) Erfinder:

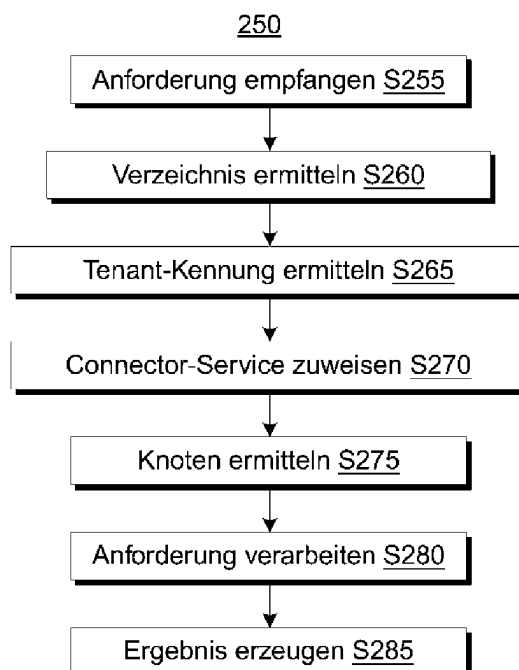
**Zheng, Yong, Beijing, CN; Yuan, Zheng Cai,  
Beijing, CN; Feng, Tian, Beijing, CN; Wang, Xin,  
Beijing, CN; Bao, Xiao Ming, Beijing, CN**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.**

(54) Bezeichnung: **MULTI-TENANT-DATEN-SERVICE IN VERTEILTEN DATEISYSTEMEN FÜR EINE BIG-DATA-ANALYSE**

(57) Zusammenfassung: Konfiguration eines verteilten Multi-Tenant-Dateisystems auf einem Knoten. Verschiedene Tenants und Tenant-Cluster sind mit einem verteilten Dateisystem korreliert, und das verteilte Dateisystem tauscht Daten durch einen Connector-Service mit verschiedenen Tenants aus. Das gesamte verteilte Dateisystem befindet sich auf einem physischen Knoten.



**Beschreibung****TECHNISCHES GEBIET**

**[0001]** Die vorliegende Erfindung bezieht sich allgemein auf das Gebiet von Speicherzugriff und -steuerung und im Besonderen auf ein Konfigurieren von Speichern.

**HINTERGRUND**

**[0002]** In einem konvergierten System stellt Virtualisierung eine Elastizität von Datenverarbeitungs-Ressourcen, Speicherplatz und/oder Anwendungsmobilität bereit. Eine konvergierte Infrastruktur fasst Komponenten einer Informationstechnologie zu einem Software-Paket zusammen. Bei einem Virtualisierungs-Container handelt es sich um ein in ein Dateisystem beinhaltendes Software-Paket zum zuverlässigen Installieren von Software auf einem Server. Ein Beispiel für einen Virtualisierungs-Container ist Docker. Einige Virtualisierungs-Container beinhalten Software-Bibliotheks-Frameworks. Ein Software-Bibliotheks-Framework ermöglicht eine verteilte Verarbeitung von großen Datensätzen mithilfe eines Programmierungsmodells. Ein Beispiel für ein solches Software-Bibliotheks-Framework ist Hadoop. Ein Portable Operating System Interface erhält eine Kompatibilität zwischen verschiedenen Betriebssystemen aufrecht. Ein Portable Operating System Interface definiert einen Satz von Anwendungsprogrammierschnittstellen. Ein Beispiel für ein Portable Operating System Interface ist POSIX.

**[0003]** Eine Big-Data-Analyse ermöglicht die Analyse von Technologie trotz des exponentiellen Wachstums und der Verfügbarkeit von Daten einschließlich sowohl strukturierter Daten als auch unstrukturierter Daten. Die Big-Data-Analyse hat sich in zwei Richtungen entwickelt: (i) exklusive Parallelverarbeitung auf Grundlage einer Relationsdatenbank; und (ii) Analyse auf Grundlage eines Software-Bibliotheks-Frameworks.

**[0004]** Es ist außerordentlich schwierig, mehrere Cluster für verschiedene Benutzer zu verwalten. Wenngleich ein Connector-Service für jede Cluster-Instanz von einem Tenant gestartet, überwacht und verwaltet werden könnte, wobei eine unabhängige Netzwerk-IP-Adresse überwacht würde, ist dieses Verfahren nicht skalierbar, da erhebliche System-Ressourcen benötigt würden.

**[0005]** Daher besteht nach dem Stand der Technik ein Bedarf, das oben genannte Problem zu lösen.

**KURZDARSTELLUNG**

**[0006]** Unter einem ersten Aspekt betrachtet, stellt die vorliegende Erfindung ein Verfahren zum Verwal-

ten einer Lese-/Schreib-Anforderung bereit, wobei das Verfahren aufweist: Ermitteln eines ersten Verzeichnisses, das einer ersten Tenant-Kennung in einem Satz von Tenant-Kennungen entspricht, wobei: das erste Verzeichnis mithilfe eines ersten Schnittstellenstandards organisiert wird und die erste Tenant-Kennung einem ersten Tenant des ersten Verzeichnisses entspricht; Zuweisen eines Connector-Service zu dem ersten Verzeichnis und der ersten Tenant-Kennung; Ermitteln eines zweiten Verzeichnisses, das dem Connector-Service entspricht, wobei: das zweite Verzeichnis mithilfe eines zweiten Schnittstellenstandards organisiert wird, ein erster Knoten einen ersten Satz von Dateien in dem zweiten Verzeichnis enthält und der erste Satz von Dateien dem ersten Tenant entspricht; Verarbeiten der ersten Lese-/Schreib-Anforderung in einem Satz von Lese-/Schreib-Anforderungen mithilfe des Connector-Service und des ersten Knotens, wobei die erste Lese-/Schreib-Anforderung von dem ersten Tenant stammt; und Erzeugen eines ersten Ergebnisses zu der ersten Lese-/Schreib-Anforderung; wobei: zumindest das Verarbeiten der ersten Lese-/Schreib-Anforderung mithilfe des Connector-Service und des ersten Knotens durch eine Computer-Software durchgeführt wird, die auf einer Computer-Hardware ausgeführt wird.

**[0007]** Unter einem weiteren Aspekt betrachtet, stellt die vorliegende Erfindung ein Computersystem zum Verwalten einer Lese-/Schreib-Anforderung bereit, wobei das System aufweist: einen Prozessorsatz; und ein computerlesbares Speichermedium; wobei: der Prozessorsatz dazu strukturiert, platziert, verbunden und/oder programmiert ist, in dem computerlesbaren Speichermedium gespeicherte Anweisungen auszuführen; und die Anweisungen beinhalten: Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein erstes Verzeichnis ermittelt, das einer ersten Tenant-Kennung in einem Satz von Tenant-Kennungen entspricht, wobei: das erste Verzeichnis mithilfe eines ersten Schnittstellenstandards organisiert wird und die erste Tenant-Kennung einem ersten Tenant des ersten Verzeichnisses entspricht; Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit einen Connector-Service dem ersten Verzeichnis und der ersten Tenant-Kennung zuweist; Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein zweites Verzeichnis ermittelt, das dem Connector-Service entspricht, wobei: das zweite Verzeichnis mithilfe eines zweiten Schnittstellenstandards organisiert wird, ein erster Knoten einen ersten Satz von Dateien in dem zweiten Verzeichnis enthält und der erste Satz von Dateien dem ersten Tenant entspricht; Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit eine erste Lese-/Schreib-Anforderung in einem Satz von Lese-/Schreib-Anforderungen mithilfe

fe des Connector-Service und des ersten Knotens verarbeitet, wobei die erste Lese-/Schreib-Anforderung von dem ersten Tenant stammt; und Programm-anweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein erstes Ergebnis zu der ersten Lese-/Schreib-Anforderung erzeugt.

**[0008]** Unter einem weiteren Aspekt betrachtet, stellt die vorliegende Erfindung ein Computerprogrammprodukt zum Verwalten einer Lese-/Schreib-Anforderung bereit, wobei das Computerprogrammprodukt ein computerlesbares Speichermedium aufweist, das durch eine Verarbeitungsschaltung gelesen werden kann und das Anweisungen zum Ausführen durch die Verarbeitungsschaltung zum Durchführen eines Verfahrens zum Durchführen der Schritte der Erfindung speichert.

**[0009]** Unter einem weiteren Aspekt betrachtet, stellt die vorliegende Erfindung ein Computerprogramm bereit, das auf einem computerlesbaren Medium gespeichert ist und in den internen Speicher eines Digitalcomputers ladbar ist, das Abschnitte eines Software-Codes aufweist, um die Schritte der Erfindung durchzuführen, wenn das Programm auf einem Computer ausgeführt wird.

**[0010]** Gemäß einem Aspekt der vorliegenden Erfindung ist ein Verfahren, ein Computerprogrammprodukt und/oder ein System vorhanden, das die folgenden Operationen (nicht zwingend in der folgenden Reihenfolge) durchführt: (i) Ermitteln eines ersten Verzeichnisses, das einer ersten Tenant-Kennung in einem Satz von Tenant-Kennungen entspricht, wobei: (a) das erste Verzeichnis mithilfe eines ersten Schnittstellenstandards organisiert wird, und (b) die erste Tenant-Kennung einem ersten Tenant des ersten Verzeichnisses entspricht; (ii) Zuweisen eines Connector-Service zu dem ersten Verzeichnis und der ersten Tenant-Kennung; (iii) Ermitteln eines zweiten Verzeichnisses, das dem Connector-Service entspricht, wobei: (a) das zweite Verzeichnis mithilfe eines zweiten Schnittstellenstandards organisiert wird, (b) ein erster Knoten einen ersten Satz von Dateien in dem zweiten Verzeichnis enthält, und (c) der erste Satz von Dateien dem ersten Tenant entspricht; (iv) Verarbeiten einer ersten Lese-/Schreib-Anforderung in einem Satz von Lese-/Schreib-Anforderungen mithilfe des Connector-Service und des ersten Knotens, wobei die erste Lese-/Schreib-Anforderung von dem ersten Tenant stammt; und (v) Erzeugen eines ersten Ergebnisses zu der ersten Lese-/Schreib-Anforderung. Zumindest das Verarbeiten der ersten Lese-/Schreib-Anforderung mithilfe des Connector-Service und des ersten Knotens wird durch eine Computer-Software durchgeführt, die auf einer Computer-Hardware ausgeführt wird.

## Figurenliste

**[0011]** Die vorliegende Erfindung wird nun lediglich beispielhaft unter Bezugnahme auf bevorzugte Ausführungsformen beschrieben, wie sie in den folgenden Figuren veranschaulicht werden:

**Fig. 1** ist eine Ansicht eines Blockschaubildes einer ersten Ausführungsform eines Systems gemäß der vorliegenden Erfindung;

**Fig. 2** ist ein Ablaufplan, der ein Verfahren einer ersten Ausführungsform darstellt, das zumindest zum Teil durch das System der ersten Ausführungsform durchgeführt wird;

**Fig. 3** ist eine Ansicht eines Blockschaubildes eines Abschnitts einer Maschinenlogik (z.B. einer Software) des Systems der ersten Ausführungsform;

**Fig. 4** ist ein Ablaufplan, der ein Verfahren einer zweiten Ausführungsform darstellt, das durch eine zweite Ausführungsform eines Systems gemäß der vorliegenden Erfindung durchgeführt wird;

**Fig. 5** ist eine Ansicht eines Blockschaubildes der zweiten Ausführungsform des Systems;

**Fig. 6** sind Referenztabelle, die durch eine dritte Ausführungsform des Systems gemäß der vorliegenden Erfindung erzeugt werden; und

**Fig. 7** ist ein Ablaufplan, der ein Verfahren einer dritten Ausführungsform darstellt, das durch eine vierte Ausführungsform eines Systems gemäß der vorliegenden Erfindung durchgeführt wird.

## AUSFÜHRLICHE BESCHREIBUNG

**[0012]** Konfiguration eines verteilten Multi-Tenant-Dateisystems auf einem Knoten. Verschiedene Tenants und Tenant-Cluster sind mit einem verteilten Dateisystem korreliert, und das verteilte Dateisystem tauscht Daten durch einen Connector-Service mit verschiedenen Tenants aus. Das gesamte verteilte Dateisystem befindet sich auf einem physischen Knoten. Dieser Abschnitt der ausführlichen Beschreibung ist in die folgenden Unterabschnitte unterteilt: (i) Hardware- und Software-Umgebung; (ii) Eine beispielhafte Ausführungsform; (iii) Weitere Anmerkungen und/oder Ausführungsformen; und (iv) Definitionen.

## HARDWARE- UND SOFTWARE-UMGEBUNG

**[0013]** Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt mit einem beliebigen Integrationsgrad technischer Details handeln. Das Computerprogrammprodukt kann (ein) durch einen Computer lesbare(n) Speichermedium (oder -medien) beinhalten, auf dem/denen durch einen Computer lesbare

Programmanweisungen gespeichert ist/sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

**[0014]** Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch ein System zur Ausführung von Anweisungen behalten und speichern kann. Bei dem durch einen Computer lesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede geeignete Kombination daraus handeln, ohne auf diese beschränkt zu sein. Zu einer nicht erschöpfenden Liste spezifischerer Beispiele des durch einen Computer lesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Nur-Lese-Speicher (ROM), ein löschbarer programmierbarer Nur-Lese-Speicher (EPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer Kompaktspeicherplatte-Nur-Lese-Speicher (CD-ROM), eine DVD (digital versatile disc), ein Speicher-Stick, eine Diskette, eine mechanisch kodierte Einheit wie zum Beispiel Lochkarten oder gehobene Strukturen in einer Rille, auf denen Anweisungen gespeichert sind, und jede geeignete Kombination daraus. Ein durch einen Computer lesbares Speichermedium soll in der Verwendung hierin nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder ein anderes Übertragungsmedium ausbreiten (z.B. durch ein Glasfaserkabel geleitete Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

**[0015]** Hierin beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs-/Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetz und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speichereinheit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Leitwegrechner, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkkartenskarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs-/Verarbeitungseinheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zur Speicherung in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs-/Verarbeitungseinheit weiter.

**[0016]** Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten, Konfigurationsdaten für integrierte Schaltungen oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o.ä. sowie prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem fernen Computer oder vollständig auf dem fernen Computer oder Server ausgeführt werden. In letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden sein, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetz (WAN), oder die Verbindung kann mit einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwendung eines Internet-Dienstansbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, im Feld programmierbare Gatter-Anordnungen (FPGA, field programmable gate arrays) oder programmierbare Logikanordnungen (PLA, programmable logic arrays) die durch einen Computer lesbaren Programmanweisungen ausführen, indem sie Zustandsinformationen der durch einen Computer lesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

**[0017]** Aspekte der vorliegenden Erfindung sind hierin unter Bezugnahme auf Ablaufpläne und/oder Blockschaltbilder bzw. Schaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es wird darauf hingewiesen, dass jeder Block der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaltbildern bzw. Schaubildern mittels durch einen Computer lesbare Programmanweisungen ausgeführt werden können.

**[0018]** Diese durch einen Computer lesbaren Programmanweisungen können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, so dass die über den Prozes-

sor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen ein Mittel zur Umsetzung der in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte erzeugen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, so dass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, ein Herstellungsprodukt aufweist, darunter Anweisungen, welche Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder der Blockschaltbilder bzw. Schaubilder angegebenen Funktion/Schritts umsetzen.

**[0019]** Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen auf einem Computer ausgeführten Prozess zu erzeugen, so dass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken der Ablaufpläne und/oder der Blockschaltbilder bzw. Schaubilder festgelegten Funktionen/Schritte umsetzen.

**[0020]** Die Ablaufpläne und die Blockschaltbilder bzw. Schaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Ausführungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in den Ablaufplänen oder Blockschaltbildern bzw. Schaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zur Ausführung der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Ausführungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt stattfinden. Zwei nacheinander gezeigte Blöcke können zum Beispiel in Wirklichkeit im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaltbilder bzw. Schaubilder und/oder der Ablaufpläne sowie Kombinationen aus Blöcken in den Blockschaltbildern bzw. Schaubildern und/oder den Ablaufplänen durch spezielle auf Hardware beruhende Systeme umgesetzt

werden können, welche die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

**[0021]** Im Folgenden wird eine Ausführungsform einer möglichen Hardware- und Software-Umgebung für Software und/oder Verfahren gemäß der vorliegenden Erfindung unter Bezugnahme auf die Figuren ausführlich beschrieben. **Fig. 1** ist ein funktionales Blockschaubild, das verschiedene Abschnitte eines Systems **100** von vernetzten Computern veranschaulicht, das beinhaltet: ein Multi-Tenant-Konfigurationsteilsystem **102**; ein Benutzerteilsystem **104**; ein Teilsystem **106** virtueller Container; ein Teilsystem **108** virtueller Container; einen Connector-Service **112**; und ein Datenübertragungs-Netzwerk **114**. Das Multi-Tenant-Konfigurationsteilsystem **102** enthält: einen Multi-Tenant-Konfigurationscomputer **200**; eine Anzeigeeinheit **212**; und externe Einheiten **214**. Der Multi-Tenant-Konfigurationscomputer **200** enthält: eine Datenübertragungseinheit **202**; einen Prozessorsatz **204**; einen Eingabe-/Ausgabe(E/A)-Schnittstellensatz **206**; eine Speichereinheit **208**; und eine persistente Speichereinheit **210**. Die Speichereinheit **208** enthält: Direktzugriffsspeicher(random access memory, RAM)-Einheiten **216**; und eine Cache-Speichereinheit **218**. Die persistente Speichereinheit **210** enthält: ein Multi-Tenant-Konfigurationsprogramm **300**. Das Teilsystem **108** virtueller Container beinhaltet: ein Software-Bibliotheks-Framework **110**.

**[0022]** Das Multi-Tenant-Konfigurationsteilsystem **102** ist in vielerlei Hinsicht repräsentativ für die verschiedenen Computerteilsysteme in der vorliegenden Erfindung. Dementsprechend werden im Folgenden mehrere Abschnitte des Multi-Tenant-Konfigurationsteilsystems **102** in den folgenden Absätzen erörtert.

**[0023]** Bei dem Multi-Tenant-Konfigurationsteilsystem **102** kann es sich um einen Laptop-Computer, einen Tablet-Computer, einen Netbook-Computer, einen Personal-Computer (PC), einen Desktop-Computer, einen persönlichen digitalen Assistenten (PDA), ein Smartphone oder eine beliebige programmierbare elektronische Einheit handeln, die in der Lage ist, Daten über das Datenübertragungs-Netzwerk **114** mit Client-Teilsystemen auszutauschen. Bei dem Multi-Tenant-Konfigurationsprogramm **300** handelt es sich um eine Sammlung von maschinenlesbaren Anweisungen und/oder Daten, die dazu verwendet werden, bestimmte Software-Funktionen zu erstellen, zu verwalten und zu steuern, die im Folgenden in dem Teilabschnitt der beispielhaften Ausführungsform dieses Abschnitts der ausführlichen Beschreibung ausführlich erörtert werden.

**[0024]** Das Multi-Tenant-Konfigurationsteilsystem **102** ist in der Lage, über das Datenübertragungs-

Netzwerk **114** Daten mit sonstigen Computerteilsystemen auszutauschen. Bei dem Datenübertragungs-Netzwerk **114** kann es sich zum Beispiel um ein lokales Netzwerk (local area network, LAN), ein Weitverkehrs-Netzwerk (wide area network, WAN) wie etwa das Internet oder um eine Kombination der beiden handeln, und es kann drahtgebundene, drahtlose oder Lichtwellenleiter-Verbindungen beinhalten. Im Allgemeinen kann es sich bei dem Datenübertragungs-Netzwerk **114** um eine beliebige Kombination von Verbindungen und Protokollen handeln, die Datenübertragungen zwischen dem Multi-Tenant-Konfigurationsteilsystem **102** und Client-Teilsystemen unterstützen.

**[0025]** Das Multi-Tenant-Konfigurationsteilsystem **102** wird als Blockschaubild mit zahlreichen Doppelpfeilen dargestellt. Diese Doppelpfeile (ohne gesonderte Bezugszeichen) stellen eine Datenübertragungsstruktur dar, die Datenübertragungen zwischen verschiedenen Komponenten des Multi-Tenant-Konfigurationsteilsystems **102** bereitstellt. Diese Datenübertragungsstruktur kann mit einer beliebigen Architektur implementiert werden, die zum Weiterleiten von Daten und/oder Steuerungsdaten zwischen Prozessoren (wie zum Beispiel Mikroprozessoren, Datenübertragungsprozessoren und/oder Netzwerkprozessoren usw.), Systemspeichern, Peripherieeinheiten und beliebigen sonstigen Hardware-Komponenten innerhalb eines Systems gestaltet ist. Beispielsweise kann die Datenübertragungsstruktur zumindest zum Teil mit einem oder mehreren Bussen implementiert werden.

**[0026]** Bei der Speichereinheit **208** und der persistenten Speichereinheit **210** handelt es sich um computerlesbare Speichermedien. Im Allgemeinen kann die Speichereinheit **208** beliebige geeignete flüchtige oder nichtflüchtige, computerlesbare Speichermedien beinhalten. Es wird des Weiteren darauf hingewiesen, dass jetzt und/oder in naher Zukunft: (i) externe Einheiten **214** in der Lage sein können, einen Teil oder den gesamten Speicher für das Multi-Tenant-Konfigurationsteilsystem **102** zu liefern; und/oder (ii) Einheiten außerhalb des Multi-Tenant-Konfigurationsteilsystems **102** in der Lage sein können, einen Speicher für das Multi-Tenant-Konfigurationsteilsystem **102** bereitzustellen.

**[0027]** Das Multi-Tenant-Konfigurationsprogramm **300** wird für einen Zugriff und/oder eine Ausführung durch einen oder mehrere Prozessoren des Prozessorsatzes **204** üblicherweise durch die Speichereinheit **208** in der persistenten Speichereinheit **210** gespeichert. Die persistente Speichereinheit **210**: (i) ist zumindest persistenter als ein Signal während einer Übertragung; (ii) speichert das Programm (einschließlich seiner Soft-Logik und/oder Daten) auf einem physischen Medium (wie zum Beispiel magnetischen oder optischen Domänen); und (iii) ist erheb-

lich weniger persistent als ein permanenter Speicher. Alternativ kann ein Datenspeicher persistenter und/oder permanenter als der Typ des durch die persistente Speichereinheit **210** bereitgestellten Speichers sein.

**[0028]** Das Multi-Tenant-Konfigurationsprogramm **300** kann substantielle Daten (das heißt, den Typ der in der Datenbank gespeicherten Daten) und/oder maschinenlesbare und -durchführbare Anweisungen beinhalten. Bei dieser bestimmten Ausführungsform (d.h., **Fig. 1**) beinhaltet die persistente Speichereinheit **210** ein Magnet-Festplattenlaufwerk. Um einige mögliche Varianten zu nennen, kann die persistente Speichereinheit **210** ein Halbleiter-Festplattenlaufwerk, eine Halbleiter-Speichereinheit, einen Festwertspeicher (read-only memory, ROM), einen löschbaren, programmierbaren Festwertspeicher (erasable programmable read-only memory, EPROM), einen Flash-Speicher oder ein beliebiges sonstiges computerlesbares Speichermedium beinhalten, das in der Lage ist, Programmanweisungen oder digitale Daten zu speichern.

**[0029]** Die durch die persistente Speichereinheit **210** verwendeten Medien können auch auswechselbar sein. Beispielsweise kann eine Wechselfestplatte als persistente Speichereinheit **210** verwendet werden. Zu sonstigen Beispielen zählen optische und Magnetplatten, USB-Speichersticks und Chip-Karten, die zum Übertragen auf ein weiteres computerlesbares Speichermedium, das ebenfalls Teil der persistenten Speichereinheit **210** ist, in ein Laufwerk eingesetzt werden.

**[0030]** Die Datenübertragungseinheit **202** stellt in diesen Beispielen einen Datenaustausch mit sonstigen Datenverarbeitungssystemen oder -einheiten außerhalb des Multi-Tenant-Konfigurationsteilsystems **102** bereit. In diesen Beispielen beinhaltet die Datenübertragungseinheit **202** eine oder mehrere Netzwerk-Schnittstellenkarten. Die Datenübertragungseinheit **202** kann eine Datenübertragung durch die Verwendung von physischen und/oder von drahtlosen Datenübertragungsverbindungen bereitstellen. Beliebige hierin erörterte Software-Module können durch eine Datenübertragungseinheit (wie zum Beispiel die Datenübertragungseinheit **202**) auf eine persistente Speichereinheit (wie zum Beispiel die persistente Speichereinheit **210**) heruntergeladen werden.

**[0031]** Der E/A-Schnittstellensatz **206** ermöglicht eine Eingabe und Ausgabe von Daten in/aus sonstige(n) Einheiten, die lokal zur Datenübertragung mit dem Multi-Tenant-Konfigurationscomputer **200** verbunden sein können. Beispielsweise stellt der E/A-Schnittstellensatz **206** eine Verbindung zu den externen Einheiten **214** bereit. Zu den externen Einheiten **214** zählen üblicherweise Einheiten wie zum Beispiel eine Tastatur, ein Tastenfeld, ein Touch-

screen und/oder eine sonstige geeignete Eingabeinheit. Zu externen Einheiten **214** können auch tragbare computerlesbare Speichermedien wie zum Beispiel USB-Speichersticks, tragbare optische oder Magnetplatten und Speicherkarten zählen. Software und Daten, die zum Anwenden von Ausführungsformen der vorliegenden Erfindung verwendet werden (z.B. das Multi-Tenant-Konfigurationsprogramm **300**), können auf solchen tragbaren computerlesbaren Speichermedien gespeichert werden. Bei diesen Ausführungsformen kann die relevante Software (ggf.) vollständig oder zum Teil über den E/A-Schnittstellensatz **206** auf die persistente Speichereinheit **210** geladen werden. Der E/A-Schnittstellensatz **206** stellt darüber hinaus eine Datenübertragungsverbindung mit der Anzeigeeinheit **212** her.

**[0032]** Die Anzeigeeinheit **212** stellt einen Mechanismus zum Anzeigen von Daten für einen Benutzer bereit und kann zum Beispiel ein Computer-Bildschirm oder ein Bildschirm eines Smartphones sein.

**[0033]** Die hierin beschriebenen Programme werden auf Grundlage der Anwendung identifiziert, für die sie in einer spezifischen Ausführungsform der Erfindung implementiert werden. Es sollte jedoch ersichtlich sein, dass jegliche besondere Programmnamen hierin lediglich der Einfachheit halber verwendet wird und die Erfindung folglich nicht auf die ausschließliche Verwendung in einer bestimmten Anwendung beschränkt sein soll, die durch eine solche Nomenklatur bezeichnet und/oder impliziert wird.

**[0034]** Die Beschreibungen der verschiedenen Ausführungsformen der vorliegenden Erfindung erfolgen zur Veranschaulichung, sind jedoch nicht erschöpfend oder auf die offenbarten Ausführungsformen beschränkt gemeint. Zahlreiche Modifizierungen und Varianten sind für Fachleute ersichtlich, ohne vom Umfang der beschriebenen Ausführungsformen abzuweichen. Die hierin verwendete Terminologie wurde gewählt, um die Grundgedanken der Ausführungsformen, die praktische Anwendung oder die technische Verbesserung gegenüber auf dem Markt erhältlichen Technologien am besten zu erläutern oder um anderen Fachleuten zu ermöglichen, die hierin offenbarten Ausführungsformen zu verstehen.

#### EINE BEISPIELHAFTE AUSFÜHRUNGSFORM

**[0035]** Fig. 2 stellt einen Ablaufplan **250** dar, der ein Verfahren gemäß der vorliegenden Erfindung darstellt. Fig. 3 stellt das Multi-Tenant-Konfigurationsprogramm **300** dar, das zumindest einige der Operationen des Verfahrens des Ablaufplans **250** durchführt. Im Folgenden werden dieses Verfahren und zugehörige Software im Laufe der folgenden Absätze unter ausführlicher Bezugnahme auf Fig. 2 (im Hinblick auf die Blöcke der Operationen des Verfahrens)

und Fig. 3 (im Hinblick auf die Software-Blöcke) erörtert.

**[0036]** Die Verarbeitung beginnt bei Operation **S255**, in der ein Anforderungsempfangsmodul („Mod“) **302** einen Satz von Anforderungen empfängt. Bei einigen Ausführungsformen der vorliegenden Erfindung empfängt das Anforderungsempfangsmod **302** einen Satz von Anforderungen von einem Satz von Anforderern. Zu Beispielen für einen Anforderer zählen ein Software-Bibliotheks-Framework, ein virtueller Container und/oder ein Benutzer, ohne darauf beschränkt zu sein. Bei einigen Ausführungsformen handelt es sich bei einem Satz von Anforderungen um einen Satz von Eingabe-/Ausgabe („E/A“-)Anforderungen. Bei weiteren Ausführungsformen handelt es sich bei einem Satz von Anforderungen um einen Satz von Lese-/Schreib-Anforderungen. Bei einigen dieser Ausführungsformen handelt es sich bei einem Satz von Anforderungen um einen Satz von E/A-Lese-/SchreibAnforderungen. Ein Beispiel für einen virtuellen Container ist Docker. Ein Beispiel für ein Software-Bibliotheks-Framework ist Hadoop. Bei weiteren Ausführungsformen empfängt das Anforderungsempfangsmod **302** einen Satz von Anforderungen von einem Satz von dynamischen Instanzierungen eines Anforderers.

**[0037]** Bei einigen Ausführungsformen handelt es sich bei einem Anforderer um ein erstes verteiltes Dateisystem. Bei einigen dieser Ausführungsformen ist ein erstes verteiltes Dateisystem nicht kompatibel mit POSIX. Bei weiteren Ausführungsformen wird ein erstes verteiltes Dateisystem mithilfe eines ersten Schnittstellenstandards organisiert. Bei einigen Ausführungsformen bezieht sich ein Satz von Anforderungen auf ein zweites verteiltes Dateisystem. Bei einigen dieser Ausführungsformen ist ein zweites verteiltes Dateisystem kompatibel mit POSIX. Bei weiteren Ausführungsformen wird ein zweites verteiltes Dateisystem mithilfe eines zweiten Schnittstellenstandards organisiert. Alternativ ist bei einigen Ausführungsformen: (i) ein erstes verteiltes Dateisystem kompatibel mit POSIX; und (ii) ein zweites verteiltes Dateisystem nicht kompatibel mit POSIX. Bei weiteren alternativen Ausführungsformen ist weder ein erstes verteiltes Dateisystem noch ein zweites verteiltes Dateisystem kompatibel mit POSIX, jedoch werden das erste verteilte Dateisystem und das zweite verteilte Dateisystem mithilfe unterschiedlicher Schnittstellenstandards organisiert.

**[0038]** Die Verarbeitung geht zu Operation **S260** über, in der ein Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen ermittelt, der einem Satz von Anforderern entspricht. Bei einigen Ausführungsformen der vorliegenden Erfindung ermittelt das Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen, der einem Satz von Anforderern entspricht. Bei einem Verzeichnis handelt es sich um ei-

ne Struktur zum Organisieren eines Satzes von Computerdateien. Ein Verzeichnis wird bisweilen auch als Pfad, Ordner und/oder Fach bezeichnet. Ein Verzeichnis kann in verschiedenen Formen ausgedrückt werden, darunter: (i) übergeordneter\_Ordner/untergeordneter\_Ordner/Datei.Erweiterung; und/oder (ii) übergeordneter Ordner > untergeordneter Ordner > Datei. Bei einigen dieser Ausführungsformen ermittelt das Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen, der einem Satz von Tenant-Kennungen entspricht. Bei sonstigen Ausführungsformen ermittelt das Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen, der einem Satz von Tenant-Kennungen entspricht, durch Zuweisen eines Verzeichnisses zu einem Satz von Anforderern. Bei weiteren Ausführungsformen ermittelt das Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen, der einem Satz von Tenant-Kennungen entspricht, durch Zuweisen eines Unterverzeichnisses zu einem Satz von Anforderern. Bei einigen Ausführungsformen entspricht ein erster Anforderer in einem ersten Satz von Anforderern einem ersten Verzeichnis. Bei sonstigen Ausführungsformen nutzt ein Satz von Anforderern ein erstes Verzeichnis gemeinsam. Bei einigen Ausführungsformen ermittelt das Verzeichnisermittlungsmod **304** einen Satz von Verzeichnissen, der einem Satz von Anforderern entspricht, aus dem das Anforderungsempfangsmod **302** in Operation **S255** einen Satz von Anforderungen empfangen hat.

**[0039]** Die Verarbeitung geht zu Operation **S265** über, in der ein Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen ermittelt, der einem Satz von Anforderungen entspricht. Bei einigen Ausführungsformen der vorliegenden Erfindung ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen, der einem Satz von Anforderungen entspricht. Bei einigen Ausführungsformen ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen für einen Satz von Anforderern, bei denen es sich um dynamische Instanzierungen handelt. Bei alternativen Ausführungsformen ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen für einen Satz von virtuellen Containern. Bei weiteren Ausführungsformen ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen für einen Satz von Software-Bibliotheks-Frameworks. Alternativ ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen für einen Satz von Benutzern. Bei einigen Ausführungsformen ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen für einen Satz von Instanzen eines Satzes von Tenants. Bei einigen Ausführungsformen ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen, der einem Satz von Anforderungen entspricht, der in Operation **S255** durch das Anforderungsempfangsmod **302** empfan-

gen worden ist. Alternativ ermittelt das Tenant-Kennungsermittlungsmod **306** einen Satz von Tenant-Kennungen, der einem Satz von Verzeichnissen entspricht, der in Operation **S260** durch das Verzeichnisermittlungsmod **304** ermittelt worden ist.

**[0040]** Die Verarbeitung geht zu Operation **S270** über, in der ein Connector-Service-Zuweisungsmod **308** einen Connector Service zuweist. Bei einigen Ausführungsformen der vorliegenden Erfindung weist das Connector-Service-Zuweisungsmod **308** einen Connector-Service zu. Bei weiteren Ausführungsformen ist ein Connector-Service ein einzelner Connector-Service auf einem Computersystem. Alternativ ist ein Connector-Service ein einzelner Connector-Service, der einem ersten verteilten Dateisystem und einem zweiten verteilten Dateisystem zugehörig ist. Bei einigen dieser Ausführungsformen leitet ein Connector-Service Anforderungen von einem Satz von Anforderern auf einem ersten verteilten Dateisystem weiter, die an ein zweites verteiltes Dateisystem gerichtet sind. Bei sonstigen Ausführungsformen weist das Connector-Service-Zuweisungsmod **308** einen Connector-Service zumindest zum Teil auf Grundlage eines Satzes von Tenant-Kennungen zu. Bei weiteren Ausführungsformen weist das Connector-Service-Zuweisungsmod **308** einen Connector-Service zumindest zum Teil auf Grundlage eines Satzes von Verzeichnissen zu. Ein Connector-Service wird bisweilen auch als Verbindungs-Server bezeichnet. Ein Connector-Service leitet einen Satz von Anforderungen durch einen Satz von geeigneten Kanälen. Ein Verbindungs-Server kann darüber hinaus Funktionen einschließlich der folgenden durchführen, ohne darauf beschränkt zu sein: (i) Prüfen eines Satzes von Benutzern auf Berechtigung; (ii) Berechtigen eines Satzes von Benutzern zu einem Satz von Ressourcen; (iii) Zuweisen eines Satzes von Paketen zu einem Satz von Ressourcen; (iv) Verwalten von lokalen und/oder entfernt angeordneten Sitzungen; (v) Herstellen eines Satzes von sicheren Verbindungen; und/oder (vi) Anwenden von Richtlinien. Bei einigen Ausführungsformen weist das Connector-Service-Zuweisungsmod **308** einen Connector-Service zumindest zum Teil auf Grundlage eines Satzes von Anforderern eines Satzes von Anforderungen zu, der in Operation **S255** durch das Anforderungsempfangsmod **302** empfangen worden ist. Bei sonstigen Ausführungsformen weist das Connector-Service-Zuweisungsmod **308** einen Connector-Service zumindest zum Teil auf Grundlage eines Satzes von Verzeichnissen zu, der in Operation **S260** durch das Verzeichnisermittlungsmod **304** ermittelt worden ist. Bei alternativen Ausführungsformen weist das Connector-Service-Zuweisungsmod **308** einen

Connector-Service zumindest zum Teil auf Grundlage eines Satzes von Tenant-Kennungen zu, der in Operation **S265** durch das Tenant-Kennungsermittlungsmod **306** ermittelt worden ist.

**[0041]** Die Verarbeitung geht zu Operation **S275** über, in der ein Knotenermittlungsmod **310** einen Knoten ermittelt, der einem Satz von Anforderern entspricht. Bei einigen Ausführungsformen der vorliegenden Erfindung ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Anforderern entspricht. Bei einigen dieser Ausführungsformen ermittelt das Knotenermittlungsmod **310**, dass ein erster Knoten jedem Anforderer in einem Satz von Anforderern entspricht. Bei einigen dieser Ausführungsformen ermittelt das Knotenermittlungsmod **310**, dass ein physischer Knoten einem Satz von Anforderern entspricht. Bei sonstigen Ausführungsformen ermittelt das Knotenermittlungsmod **310**, dass ein virtueller Knoten einem Satz von Anforderern entspricht. Bei alternativen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Anforderern entspricht, durch Zuweisen jedes Anforderers in dem Satz von Anforderern zu einem ersten Knoten. Bei einigen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Anforderungen entspricht. Bei weiteren Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Tenant-Kennungen entspricht. Bei sonstigen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten zumindest zum Teil auf Grundlage eines Connector-Service. Bei alternativen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten zumindest zum Teil auf Grundlage einer Eins-zu-Eins-Beziehung zwischen dem Knoten und einem Connector-Service. Bei sonstigen Ausführungsformen ordnet die Knotenermittlung **310** einen Pfad zwischen einem Connector-Service und einem Knoten zu. Bei einigen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Anforderern entspricht, aus dem das Anforderungsempfangsmod **302** in Operation **S255** einen Satz von Anforderungen empfangen hat. Bei sonstigen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Anforderungen entspricht, der in Operation **S255** durch das Anforderungsempfangsmod **302** empfangen worden ist. Bei weiteren Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Verzeichnissen entspricht, der in Operation **S260** durch das Verzeichnisermittlungsmod **304** ermittelt worden ist. Bei alternativen Ausführungsformen ermittelt das Knotenermittlungsmod **310** einen Knoten, der einem Satz von Tenant-Kennungen entspricht, der in Operation **S265** durch das Tenant-Kennungsermittlungsmod **306** ermittelt worden ist. Alternativ ermittelt das Knotenermittlungsmod **310** einen Knoten zumindest zum Teil auf Grundlage eines Connector-

Service, der in Operation **S270** durch das Connector-Service-Zuweisungsmod **308** zugewiesen worden ist.

**[0042]** Die Verarbeitung geht zu Operation **S280** über, in der ein Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen verarbeitet. Bei einigen Ausführungsformen der vorliegenden Erfindung verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen. Bei einigen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Satzes von Tenant-Kennungen. Bei sonstigen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Knotens. Bei weiteren Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Verzeichnisses. Bei einigen Ausführungsformen ordnet das Anforderungsverarbeitungsmod **312** ein erstes verteiltes Dateisystem über eine Mount-Operation einem zweiten verteilten Dateisystem zu. Bei alternativen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Connector-Service. Bei einer Leseanforderung liest das Anforderungsverarbeitungsmod **312** einen Satz von Daten aus einem Speicher. Bei einer Schreibanforderung modifiziert das Anforderungsverarbeitungsmod **312** einen Satz von Daten in einem Speicher. Bei einer Eingabeanforderung empfängt das Anforderungsverarbeitungsmod **312** einen Satz von Daten. Bei einer Ausgabeanforderung sendet das Anforderungsverarbeitungsmod **312** einen Satz von Daten. Bei einigen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen, der in Operation **S255** durch das Anforderungsempfangsmod **312** empfangen worden ist. Bei sonstigen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Satzes von Tenant-Kennungen, der in Operation **S265** durch das Tenant-Kennungsermittlungsmod **306** ermittelt worden ist. Bei weiteren Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Knotens, der in Operation **S275** durch das Knotenermittlungsmod **310** ermittelt worden ist. Bei sonstigen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Satzes von Verzeichnissen, der in Operation **S260** durch das Verzeichnisermittlungsmod **304** ermittelt worden ist. Bei alternativen Ausführungsformen verarbeitet das Anforderungsverarbeitungsmod **312** einen Satz von Anforderungen zumindest zum Teil auf Grundlage eines Connector-Service, der in Operation **S270** durch das Connector-Service-Ermittlungsmod **308** ermittelt worden ist.

**[0043]** Die Verarbeitung endet bei der Operation **S285**, in der ein Ergebniserzeugungsmod **314** einen Satz von Ergebnissen erzeugt. Bei einigen Ausführungsformen der vorliegenden Erfindung erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen für einen Satz von Anforderungen. Bei einigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen zu einem Satz von Leseanforderungen durch Erzeugen eines Satzes von Nachrichten, die einen Satz von Daten beinhalten. Bei einigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen zu einem Satz von Schreibanforderungen durch Erzeugen eines Satzes von neuen Dateneinträgen. Bei einigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen zu einem Satz von Eingabeanforderungen durch Speichern eines empfangenen Satzes von Daten. Bei einigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen zu einem Satz von Ausgabeanforderungen durch Erzeugen eines Satzes von Nachrichten. Bei sonstigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** Ergebnisse für ein erstes verteiltes Dateisystem, das nicht mit POSIX kompatibel ist. Bei weiteren Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen für ein erstes verteiltes Dateisystem, bei dem es sich um Hadoop handelt. Bei sonstigen Ausführungsformen beinhaltet ein Ergebnis einen neuen Dateneintrag und/oder eine Nachricht mit einem Satz von Daten, ohne darauf beschränkt zu sein. Bei einigen Ausführungsformen erzeugt das Ergebniserzeugungsmod **314** einen Satz von Ergebnissen zu einem Satz von Anforderungen, der in Operation **S255** durch das Anforderungsempfangsmod **302** empfangen worden ist.

#### WEITERE ANMERKUNGEN UND/ ODER AUSFÜHRUNGSFORMEN

**[0044]** Einige Ausführungsformen der vorliegenden Erfindung erkennen die folgenden Umstände, potentiellen Probleme und/oder potentiellen verbesserungsbedürftigen Gebiete im Hinblick auf den aktuellen Stand der Technik: (i) Ein Verwalten eines Satzes von Knoten, eines Satzes von Connector-Services und/oder eines Satzes von Verzeichnissen, die einem Satz von Tenant-Kennungen entsprechen, führt zu einem exponentiellen Anstieg der Ressourcen; (ii) verschiedene Betriebssysteme behandeln einen Satz von Knoten, einen Satz von Connector-Services und/oder einen Satz von Verzeichnissen auf vielfältige Weisen; und/oder (iii) einige verteilte Dateisysteme (distributed file systems, „DFSs“) sind nicht kompatibel mit dem Portable Operating System Interface („POSIX“); (iv) einige DFSs können nicht über eine Mount-Operation zugeordnet werden; und/oder (v) Hyperkonvergenz-Infrastrukturen versuchen, eine Ressourcen-Nutzung zu verringern. Bei herkömmli-

chen Mitteln zum Verwalten eines Satzes von Knoten, eines Satzes von Connector-Services und/oder eines Satzes von Verzeichnissen, die einem Satz von Tenant-Kennungen entsprechen, sind einzelne Knoten und einzelne Verzeichnisse erforderlich, die jeder Tenant-Kennung entsprechen.

**[0045]** Fig. 4 stellt einen Ablaufplan **400** dar, der ein Verfahren gemäß der vorliegenden Erfindung darstellt. Die Verarbeitung beginnt bei der Operation **S405**, in der ein Multi-Tenant-Konfigurationsteilsystem eine E/A-Anforderung von einer Hadoop-Container-Instanz empfängt. Die Verarbeitung geht zu Operation **S410** über, in der ein Multi-Tenant-Konfigurationsteilsystem einen Satz von Tenant-Kennungen für eine Hadoop-Container-Instanz isoliert. Die Verarbeitung geht zu Operation **S415** über, in der ein Multi-Tenant-Konfigurationsteilsystem eine Hadoop-Container-Instanz zumindest zum Teil auf Grundlage eines Satzes von Tenant-Kennungen erkennt. Die Verarbeitung geht zu Operation **S420** über, in der ein Multi-Tenant-Konfigurationsteilsystem einen Satz von Berechtigungen für eine Hadoop-Container-Instanz überprüft. Die Verarbeitung endet bei der Operation **S425**, in der ein Multi-Tenant-Konfigurationsteilsystem eine E/A-Anforderung bearbeitet.

**[0046]** Fig. 5 stellt ein funktionales Blockschaubild eines Systems **500** dar, das beinhaltet: eine Hadoop-Instanz **502**; eine Hadoop-Instanz **504**; eine Hadoop-Instanz **506**; einen Connector-Service **508**; ein verteiltes Dateisystem **510**; und einen physischen Knoten **512**. Eine Datenübertragung jeweils zwischen der Hadoop-Instanz **502**, der Hadoop-Instanz **504** und der Hadoop-Instanz **506** und dem verteilten Dateisystem **510** verläuft durch den Connector-Service **508**. Dadurch, dass es sich auf dem physischen Knoten **512** befindet, kann das verteilte Dateisystem **510** sämtliche Datenübertragungen durch den Connector-Service **508** verarbeiten.

**[0047]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Isolieren eines Satzes von DFS-Instanzdaten; (ii) Isolieren eines Satzes von Hadoop-Instanzdaten; (iii) Einführen eines Multi-Tenant-Erkennungsmoduls in einen DFS-Connector-Service; und/oder (iv) Bereitstellen einer Multi-Tenant-Fähigkeit für ein hyperkonvergentes DFS. Ein hyperkonvergentes DFS wird bisweilen auch als Multi-Tenant-DFS bezeichnet. Bei einigen Ausführungsformen der vorliegenden Erfindung enthält ein Multi-Tenant-Erkennungsmodul die Operation **S410** und die Operation **S415** von Fig. 4. Bei sonstigen Ausführungsformen führt der Connector-Service **508** in Fig. 5 die Operation **S410** und/oder die Operation **S415** von Fig. 4 durch. Bei weiteren Ausführungsformen stellt das Multi-Tenant-Konfigurationsteilsystem einen Connector-Service und einen physischen Knoten

in einer Eins-zu-Eins-Beziehung bereit. Bei alternativen Ausführungsformen konfiguriert das Multi-Tenant-Konfigurationsteilsystem einen Satz von DFS-Instanzen mit einem Satz von Adressen eines privaten Netzwerks. Alternativ konfiguriert das Multi-Tenant-Konfigurationsteilsystem einen Satz von DFS-Instanzen mit einer Adresse eines privaten Netzwerks. Bei einigen Ausführungsformen isoliert ein Multi-Tenant-Konfigurationsteilsystem eine DFS-Instanz in einem Verzeichnis. Bei weiteren Ausführungsformen isoliert ein Multi-Tenant-Konfigurationsteilsystem eine DFS-Instanz in einem Verzeichnis zumindest zum Teil auf Grundlage eines Tenants. Bei sonstigen Ausführungsformen isoliert ein Multi-Tenant-Konfigurationsteilsystem einen Satz von Operationen für eine DFS-Instanz in einem Verzeichnis.

**[0048]** Fig. 6 stellt zwei Tabellen dar. Bei der ersten Tabelle in Fig. 6 handelt es sich um eine Instanz-Container-Zuordnungsliste. Zwei Instanzen mit drei Containern werden dargestellt, was in sechs Tenant-IDs resultiert. Diese sechs Tenant-IDs werden sämtlich einem Knoten zugeordnet. Bei der zweiten Tabelle in Fig. 6 handelt es sich um eine umgekehrte Instanz-Container-Zuordnungsliste. Dieselben sechs Tenant-IDs werden dargestellt. Die zweite Tabelle ist jedoch sortiert, um eine entsprechende Instanz zu ermitteln.

**[0049]** Fig. 7 stellt einen Ablaufplan 700 dar, der ein Verfahren gemäß der vorliegenden Erfindung darstellt. Die Verarbeitung beginnt bei der Operation S705, in der ein Multi-Tenant-Konfigurationsteilsystem eine E/A-Lese-/Schreib-Anforderung von einem Hadoop-Vorgang in einem Container empfängt. Die Verarbeitung geht zu Operation S710 über, in der ein Multi-Tenant-Konfigurationsteilsystem eine Container-IP-Adresse aus einer E/A-Anforderung abrufen. Die Verarbeitung geht zu Operation S715 über, in der ein Multi-Tenant-Konfigurationsteilsystem eine IP-Adresse eines physischen Knotens abrufen. Die Verarbeitung geht zu Operation S720 über, in der ein Multi-Tenant-Konfigurationsteilsystem eine Instanz-Container-Zuordnungsliste aufgrund einer Container-IP und einer Knoten-IP abfragt. Die Verarbeitung geht zu Operation S725 über, in der ein Multi-Tenant-Konfigurationsteilsystem eine Instanzen-ID abrufen. Die Verarbeitung geht zu Operation S730 über, in der ein Multi-Tenant-Konfigurationsteilsystem ein Instanzenverzeichnis abrufen. Die Verarbeitung geht zu Operation S735 über, in der ein Multi-Tenant-Konfigurationsteilsystem einen Satz von E/A-Pfaden umwandelt. Die Verarbeitung endet bei der Operation S470, in der ein Multi-Tenant-Konfigurationsteilsystem einen Satz von E/A-Anforderungen bearbeitet.

**[0050]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Ein DFS ermöglicht einen Zugriff auf einen Satz von Dateien von einer Vielfalt von Hosts;

(ii) ein DFS ermöglicht einem Satz von Benutzern, einen Satz von Dateien über einen Satz von Einheiten hinweg gemeinsam zu nutzen; und/oder (iii) bei einem DFS handelt es sich um ein gängiges Speichersystem. Zu Beispielen für DFSs zählen: IBM General Parallel File System („GPFS™“) File Placement Optimizer („FPO“), Red Hat Linux®, GlusterFS, Lustre, Ceph und Apache Hadoop Distributed File System („HDFS“). IBM und GPFS sind in vielen Ländern weltweit eingetragene Marken von International Business Machines Corporation. Linux ist eine eingetragene Marke von Linus Torvalds in den USA, anderen Ländern oder beidem.

**[0051]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Zuordnen eines DFS über eine Mount-Operation; (ii) Lesen von Daten von einem DFS; (iii) Schreiben von Daten in ein DFS; (iv) Lesen von Daten von einem DFS mithilfe einer POSIX-Anwendung; (v) Schreiben von Daten in ein DFS mithilfe einer POSIX-Anwendung; (vi) Lesen von Daten von einem DFS mithilfe einer POSIX-Anwendung in dem DFS-Ökosystem; und/oder (vii) Schreiben von Daten in ein DFS mithilfe einer POSIX-Anwendung in dem DFS-Ökosystem. Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Ermitteln eines Satzes von Berechtigungen zumindest zum Teil auf Grundlage einer Benutzer-ID; (ii) Ermitteln eines Satzes von Berechtigungen zumindest zum Teil auf Grundlage einer Gruppen-ID; (iii) Ermitteln eines Satzes von Berechtigungen für eine Betriebsumgebung; und/oder (iv) Ermitteln eines Satzes von Berechtigungen für ein Betriebssystem.

**[0052]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Ausführen eines DFS mithilfe einer POSIX-Anwendung; (ii) Übertragen eines Satzes von Dateien über einen einzelnen Connector-Service; (iii) Übertragen eines Satzes von Dateien über einen einzelnen Connector-Service in einem DFS mithilfe einer POSIX-Anwendung; und/oder (iv) Ausführen eines hyperkonvergenten DFS mithilfe einer POSIX-Anwendung. Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Ausführen eines DFS mithilfe einer anderen als einer POSIX-Anwendung; (ii) Übertragen eines Satzes von Dateien über einen einzelnen Connector-Service; (iii) Übertragen eines Satzes von Dateien über einen einzelnen Connector-Service in einem DFS mithilfe einer anderen als einer POSIX-Anwendung; und/oder (iv) Ausführen eines hyperkonvergenten DFS mithilfe einer anderen als einer POSIX-Anwendung. Einige Ausführungs-

formen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Erstellen eines Satzes von Clustern eines Satzes von DFS-Instanzen; (ii) Erstellen eines Satzes von Clustern eines Satzes von DFS-Instanzen für einen Satz von Benutzern; (iii) Zuweisen eines Satzes von Netzwerkadressen zu einem Satz von Clustern; (iv) Zuweisen eines Satzes von Tenant-Kennungen zu einem Satz von Clustern; (v) Zuweisen eines Satzes von Netzwerkadressen zu einem Satz von Clustern, wobei der Satz von Netzwerkadressen nicht mit einem DFS in Zusammenhang steht; und/oder (vi) Zuweisen eines Satzes von Tenant-Kennungen zu einem Satz von Clustern, wobei der Satz von Netzwerkadressen nicht mit einem DFS in Zusammenhang steht.

**[0053]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Verringern einer Anzahl von Connector-Services; (ii) Verwenden eines einzelnen Connector-Service; (iii) Verringern einer Anzahl von Connector-Services, die dazu erforderlich ist, eine Multi-Tenant-Konfiguration zu verwalten; (iv) Verringern einer Anzahl von Connector-Services, die dazu erforderlich ist, eine Multi-Tenant-Konfiguration auf einer exponentiellen Ebene zu verwalten; (v) Verringern einer Anzahl von Tenant-Kennungen, die einer Anzahl von Clients in einem DFS entspricht; und/oder (vi) Verringern einer Anzahl von IP-Adressen, die einer Anzahl von Clients in einem DFS entspricht.

**[0054]** Bei einigen Ausführungsformen der vorliegenden Erfindung erzeugt ein Multi-Tenant-Konfigurationsteilsystem einen DFS-Cluster für einen Tenant. Bei weiteren Ausführungsformen erzeugt ein Multi-Tenant-Konfigurationsteilsystem eine Tenant-ID, die einem DFS-Cluster entspricht. Ein DFS-Cluster wird bisweilen auch als erstes verteiltes Dateisystem mit mehreren Anforderern und/oder mehreren Tenants bezeichnet. Bei einigen Ausführungsformen weist ein Multi-Tenant-Konfigurationsteilsystem eine Tenant-ID einem Knoten zu.

**[0055]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Konfigurieren eines Satzes von Verzeichnissen in einem DFS; (ii) Konfigurieren eines Satzes von Verzeichnissen in einem DFS und Neustarten eines Connector-Service; (iii) Erstellen eines Satzes von Software-Bibliotheks-Framework-Instanzen für eine DFS-Instanz; (iv) Speichern eines Satzes von Tenant-Daten in einem Verzeichnis in einem hyperkonvergenten DFS; (v) Erkennen eines DFS-Verzeichnisses ohne Neustarten; (vi) Neustarten eines DFS, ohne eine neue DFS-Instanz zu erstellen; (vii) Bereitstellen eines DFS-Clusters für einen Tenant; (viii) Verwalten eines DFS-Clusters für einen Tenant;

und/oder (ix) Isolieren eines DFS zumindest zum Teil auf Grundlage eines Satzes von Hardware-Ressourcen. Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Erzeugen einer Benutzer-ID beim Erstellen eines Software-Bibliotheks-Frameworks; (ii) Erzeugen einer Benutzer-ID beim Kompilieren eines Software-Bibliotheks-Frameworks; (iii) Erzeugen einer Gruppen-ID beim Erstellen eines Software-Bibliotheks-Frameworks; und/oder (iv) Erzeugen einer Gruppen-ID beim Kompilieren eines Software-Bibliotheks-Frameworks.

**[0056]** Einige Ausführungsformen der vorliegenden Erfindung können ein(e,en) oder mehrere der folgenden Merkmale, Eigenschaften und/oder Vorteile beinhalten: (i) Verwalten eines hyperkonvergenten Big-Data-DFS; (ii) Verwalten eines Multi-Tenant-Big-Data-DFS; (iii) Verwalten eines hyperkonvergenten DFS in einem Cloud-System; und/oder (iv) Verwalten eines hyperkonvergenten DFS in einem virtuellen System.

## DEFINITIONEN

**[0057]** „Vorliegende Erfindung“ erzeugt keine absolute Angabe und/oder Implikation, dass der beschriebene Gegenstand durch den ursprünglichen Anspruchssatz, wie er eingereicht wurde, durch einen geänderten Anspruchssatz, der während des Anmeldeverfahrens erstellt wurde, und/oder durch den endgültigen Anspruchssatz, der durch das Patentanmeldeverfahren zugelassen wurde und in dem erteilten Patent beinhaltet ist, abgedeckt wird. Der Begriff „vorliegende Erfindung“ wird verwendet, um einen Teil oder mehrere Teile der Offenbarung anzugeben, die möglicherweise eine Weiterentwicklung oder mehrere Weiterentwicklungen gegenüber dem Stand der Technik beinhalten könnten. Dieses Verständnis des Begriffs „vorliegende Erfindung“ und seiner Angaben und/oder Implikationen ist vorläufig und provisorisch und kann sich im Verlaufe des Patentanmeldeverfahrens ändern, wenn relevante Informationen entwickelt werden und die Ansprüche möglicherweise geändert werden.

**[0058]** Zu „Ausführungsform“ wird auf die Definition für die „vorliegende Erfindung“ verwiesen.

**[0059]** „Und/oder“ ist eine einschließende Oder-Verknüpfung, die auch als logische Oder-Verknüpfung bekannt ist und allgemein als „einschließendes Oder“ bekannt ist. Beispielsweise bedeutet der Ausdruck „A, B und/oder C“, dass zumindest eines von A oder B oder C wahr ist; und „A, B und/oder C“ ist nur falsch, wenn jedes von A und B und C falsch ist.

**[0060]** Ein „Satz von“ Elementen bedeutet, dass ein oder mehrere Elemente vorhanden sind; es muss zu-

mindest ein Element vorhanden sein, es können jedoch auch zwei, drei oder mehr Elemente vorhanden sein. Ein „Teilsatz von“ Elementen bedeutet, dass ein oder mehrere Elemente innerhalb einer Gruppierung von Elementen vorhanden sind, die eine gemeinsame Eigenschaft enthalten.

**[0061]** Eine „Mehrzahl von“ Elementen bedeutet, dass mehr als ein Element vorhanden ist; es müssen zumindest zwei Elemente vorhanden sein, es können jedoch auch drei, vier oder mehr Elemente vorhanden sein.

**[0062]** „Beinhaltet“ und jegliche Varianten (z.B. beinhalten, beinhalten usw.) bedeutet, sofern nicht ausdrücklich etwas anderes angegeben wird, „beinhaltet“, ist jedoch nicht zwingend darauf beschränkt“.

**[0063]** Ein „Benutzer“ oder ein „Abonnent“ beinhaltet, ohne zwingend darauf beschränkt zu sein: (i) eine einzelne Person; (ii) eine Einheit künstlicher Intelligenz mit ausreichender Intelligenz, um anstelle einer einzelnen Person oder mehr als einer Person zu handeln; (iii) eine Geschäftseinheit, für die Maßnahmen durch eine einzelne Person oder mehr als eine Person ergriffen werden; und/oder (iv) eine Kombination eines oder mehrerer beliebiger verwandter „Benutzer“ oder „Abonnenten“, die als einzelner „Benutzer“ oder „Abonnent“ handeln.

**[0064]** Die Begriffe „empfangen“, „bereitstellen“, „senden“, „eingeben“, „ausgeben“ und „melden“ sollten, sofern nicht ausdrücklich etwas anderes angegeben wird, nicht so verstanden werden, dass sie angeben oder implizieren: (i) einen bestimmten Grad an Direktheit im Hinblick auf die Beziehung zwischen einem Objekt und einem Subjekt; und/oder (ii) ein Vorhandensein oder ein Nichtvorhandensein eines Satzes von zwischenliegenden Komponenten, zwischenliegenden Vorgängen und/oder Dingen, die zwischen ein Objekt und ein Subjekt gefügt werden.

**[0065]** Bei einem „Modul“ handelt es sich um einen beliebigen Satz aus Hardware, Firmware und/oder Software, der funktionell arbeitet, um eine Funktion auszuüben, ohne zu berücksichtigen, ob das Modul: (i) sich in einer einzelnen lokalen Umgebung befindet; (ii) über ein großes Gebiet verteilt ist; (iii) sich in einer einzelnen lokalen Umgebung innerhalb eines größeren Teils eines Software-Codes befindet; (iv) sich innerhalb eines einzelnen Teils eines Software-Codes befindet; (v) sich innerhalb einer einzelnen Speichereinheit, eines Speichers oder eines Mediums befindet; (vi) mechanisch verbunden ist; (vii) elektrisch verbunden ist; und/oder (viii) zur Datenübertragung verbunden ist. Ein „Untermodule“ ist ein „Modul“ innerhalb eines „Moduls“.

**[0066]** Ein „Computer“ ist eine beliebige Einheit mit signifikanten Fähigkeiten zur Datenverarbeitung und/oder zum Lesen von maschinenlesbaren Anweisungen, darunter, ohne zwingend darauf beschränkt zu sein: Desktop-Computer, Großrechner, Laptop-Computer, Einheiten auf Grundlage von feldprogrammierbaren Gate-Arrays (FPGA); Smartphones; persönliche digitale Assistenten (PDAs); Einbau- oder Einschubcomputer; integrierte Computer; und/oder Einheiten auf Grundlage von anwendungsspezifischen integrierten Schaltungen (application specific integrated circuits, ASIC).

**[0067]** „Elektrisch verbunden“ bedeutet entweder indirekt elektrisch verbunden, sodass zwischenliegende Elemente vorhanden sind, oder direkt elektrisch verbunden. Eine „elektrische Verbindung“ kann Elemente wie zum Beispiel Kondensatoren, Induktoren, Transformatoren, Vakuumröhren und dergleichen beinhalten, muss jedoch nicht darauf beschränkt sein.

**[0068]** „Mechanisch verbunden“ bedeutet entweder indirekte mechanische Verbindungen, die durch zwischenliegende Komponenten hergestellt werden, oder direkte mechanische Verbindungen. „Mechanisch verbunden“ beinhaltet starre mechanische Verbindungen wie auch eine mechanische Verbindung, die eine relative Bewegung zwischen den mechanisch verbundenen Komponenten zulässt. „Mechanisch verbunden“ beinhaltet, ohne darauf beschränkt zu sein: geschweißte Verbindungen; Lotverbindungen; Verbindungen durch Befestigungselemente (z.B. Nägel, Bolzen, Schrauben, Muttern, Klettverschlüsse, Knoten, Nieten, Schnellspannverbindungen, Riegel und/oder magnetische Verbindungen); kraftschlüssige Verbindungen; reibschlüssige Verbindungen; Verbindungen, die durch einen durch Schwerkkräfte verursachten Eingriff gesichert werden; Schwenk- oder Drehverbindungen; und/oder verschiebbare mechanische Verbindungen.

**[0069]** Eine „Datenübertragung“ beinhaltet, ohne zwingend darauf beschränkt zu sein, jede Art von heute bekanntem oder künftig zu entwickelndem Datenübertragungsschema. „Datenübertragungen“ beinhalten, sind jedoch nicht zwingend beschränkt auf: eine drahtlose Datenübertragung; eine drahtgebundene Datenübertragung; und/oder Datenübertragungswege mit drahtlosen und drahtgebundenen Abschnitten. Eine „Datenübertragung“ ist nicht zwingend beschränkt auf: (i) eine direkte Datenübertragung; (ii) eine indirekte Datenübertragung; und/oder (iii) eine Datenübertragung, bei der das Format, der Paketierungsstatus, das Medium, der Verschlüsselungsstatus, und/oder das Protokoll über den gesamten Verlauf der Datenübertragung konstant bleiben.

**[0070]** Der Ausdruck „ohne wesentliches menschliches Eingreifen“ bedeutet einen Prozess, der auto-

matisch, mit wenig oder ohne menschliches Eingreifen, auftritt (häufig durch den Einsatz von Maschinenlogik wie zum Beispiel Software). Zu einigen Beispielen, die „ohne wesentliches menschliches Eingreifen“ einhergehen, zählen: (i) Ein Computer führt eine komplexe Verarbeitung durch, und eine Person schaltet den Computer aufgrund eines Ausfalls des Netzstroms zu einer alternativen Stromversorgung um, sodass die Verarbeitung ununterbrochen fortgesetzt wird; (ii) ein Computer ist im Begriff, eine Ressourcen-intensive Verarbeitung durchzuführen, und eine Person bestätigt, dass die Ressourcen-intensive Verarbeitung tatsächlich durchgeführt werden soll (in diesem Fall geschieht der Prozess der Bestätigung isoliert betrachtet mit wesentlichem menschlichen Eingreifen, die Ressourcen-intensive Verarbeitung beinhaltet jedoch trotz der einfachen Bestätigung im Ja-Nein-Stil, die durch eine Person erfolgen muss, kein wesentliches menschliches Eingreifen); und (iii) ein Computer hat mithilfe einer Maschinenlogik eine schwerwiegende Entscheidung getroffen (zum Beispiel eine Entscheidung, in der Erwartung schlechten Wetters sämtlichen Flugzeugen Startverbot zu erteilen), jedoch muss der Computer vor der Umsetzung der schwerwiegenden Entscheidung von einer menschlichen Quelle eine einfache Bestätigung im Ja-Nein-Stil einholen.

**[0071]** „Automatisch“ bedeutet „ohne jegliches menschliches Eingreifen“.

**[0072]** Der Begriff „Echtzeit“ (und der Ausdruck „in Echtzeit“) beinhaltet einen beliebigen Zeitrahmen von ausreichend kurzer Dauer, um eine angemessene Reaktionszeit für eine Datenverarbeitung, wie beschrieben, bereitzustellen. Darüber hinaus beinhaltet der Begriff „Echtzeit“ (und der Ausdruck „in Echtzeit“), was üblicherweise als „echtzeitnah“ bezeichnet wird, im Allgemeinen einen beliebigen Zeitrahmen von ausreichend kurzer Dauer, um eine angemessene Reaktionszeit für eine Datenverarbeitung auf Anforderung, wie beschrieben, bereitzustellen (z.B. innerhalb eines Bruchteils einer Sekunde oder innerhalb weniger Sekunden). Es ist zwar schwierig, diese Begriffe genau zu definieren, für Fachleute sind sie jedoch gut verständlich.

### Patentansprüche

1. Verfahren zum Verwalten einer Lese-/Schreib-Anforderung, wobei das Verfahren aufweist: Ermitteln eines ersten Verzeichnisses, das einer ersten Tenant-Kennung in einem Satz von Tenant-Kennungen entspricht, wobei: das erste Verzeichnis mithilfe eines ersten Schnittstellenstandards organisiert wird und die erste Tenant-Kennung einem ersten Tenant des ersten Verzeichnisses entspricht; Zuweisen eines Connector-Service zu dem ersten Verzeichnis und der ersten Tenant-Kennung;

Ermitteln eines zweiten Verzeichnisses, das dem Connector-Service entspricht, wobei: das zweite Verzeichnis mithilfe eines zweiten Schnittstellenstandards organisiert wird, ein erster Knoten einen ersten Satz von Dateien in dem zweiten Verzeichnis enthält, und der erste Satz von Dateien dem ersten Tenant entspricht; Verarbeiten der ersten Lese-/Schreib-Anforderung in einem Satz von Lese-/Schreib-Anforderungen mithilfe des Connector-Service und des ersten Knotens, wobei die erste Lese-/Schreib-Anforderung von dem ersten Tenant stammt; und Erzeugen eines ersten Ergebnisses zu der ersten Lese-/Schreib-Anforderung; wobei: zumindest das Verarbeiten der ersten Lese-/Schreib-Anforderung mithilfe des Connector-Service und des ersten Knotens durch eine Computer-Software durchgeführt wird, die auf einer Computer-Hardware ausgeführt wird.

2. Verfahren nach Anspruch 1, das des Weiteren aufweist:

Ermitteln eines dritten Verzeichnisses, das einer zweiten Tenant-Kennung in dem Satz von Tenant-Kennungen entspricht, wobei:

die zweite Tenant-Kennung einer zweiten Lese-/Schreib-Anforderung in dem Satz von Lese-/Schreib-Anforderungen entspricht und das dritte Verzeichnis mithilfe des ersten Schnittstellenstandards organisiert wird;

Zuweisen des Connector-Service zu dem dritten Verzeichnis und der zweiten Tenant-Kennung;

Verarbeiten der zweiten Lese-/Schreib-Anforderung mithilfe des Connector-Service und eines zweiten Knotens; wobei:

ein zweiter Knoten einen zweiten Satz von Dateien in dem zweiten Verzeichnis enthält, und der zweite Satz von Dateien dem zweiten Tenant entspricht; und

Erzeugen eines zweiten Ergebnisses zu der zweiten Lese-/Schreib-Anforderung.

3. Verfahren nach Anspruch 2, wobei es sich bei dem zweiten Knoten um den ersten Knoten handelt.

4. Verfahren nach einem der vorhergehenden Ansprüche, wobei das erste Ergebnis aus einer Gruppe ausgewählt wird, die besteht aus:

einem neuen Dateneintrag, und einer Nachricht mit einem Satz von Daten.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei der erste Schnittstellenstandard nicht kompatibel mit POSIX ist.

6. Verfahren nach einem der vorhergehenden Ansprüche, wobei der zweite Schnittstellenstandard kompatibel mit POSIX ist.

7. Verfahren nach einem der vorhergehenden Ansprüche, wobei das erste Verzeichnis mithilfe eines Apache Hadoop Distributed File System („HDFS“) organisiert wird.

8. Computerprogrammprodukt zum Verwalten einer Lese-/Schreib-Anforderung, wobei das Computerprogrammprodukt aufweist:  
ein computerlesbares Speichermedium, das durch eine Verarbeitungsschaltung lesbar ist und das Befehle zur Ausführung durch die Verarbeitungsschaltung zum Durchführen eines Verfahrens nach einem der Ansprüche 1 bis 7 speichert.

9. In einem computerlesbaren Medium gespeichertes und in den internen Speicher eines Digitalcomputers ladbares Computerprogramm, das Abschnitte eines Software-Codes aufweist, um das Verfahren nach einem der Ansprüche 1 bis 7 durchzuführen, wenn das Programm auf einem Computer ausgeführt wird.

10. Computersystem zum Verwalten einer Lese-/Schreib-Anforderung, wobei das System aufweist:  
einen Prozessorsatz; und  
ein computerlesbares Speichermedium;  
wobei:  
der Prozessorsatz dazu strukturiert, platziert, verbunden und/oder programmiert ist, in dem computerlesbaren Speichermedium gespeicherte Anweisungen auszuführen; und  
die Anweisungen beinhalten:  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein erstes Verzeichnis ermittelt, das einer ersten Tenant-Kennung in einem Satz von Tenant-Kennungen entspricht, wobei:  
das erste Verzeichnis mithilfe eines ersten Schnittstellenstandards organisiert wird und die erste Tenant-Kennung einem ersten Tenant des ersten Verzeichnisses entspricht;  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit einen Connector-Service dem ersten Verzeichnis und der ersten Tenant-Kennung zuweist;  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein zweites Verzeichnis ermittelt, das dem Connector-Service entspricht, wobei:  
das zweite Verzeichnis mithilfe eines zweiten Schnittstellenstandards organisiert wird,  
ein erster Knoten einen ersten Satz von Dateien in dem zweiten Verzeichnis enthält, und  
der erste Satz von Dateien dem ersten Tenant entspricht;  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit eine erste Lese-/Schreib-Anforderung in einem Satz von Lese-/Schreib-Anforderungen mithilfe des Connector-Service und des ersten Knotens verarbeitet,

wobei die erste Lese-/Schreib-Anforderung von dem ersten Tenant stammt; und  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein erstes Ergebnis zu der ersten Lese-/Schreib-Anforderung erzeugt.

11. Computersystem nach Anspruch 10, das des Weiteren aufweist:  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein drittes Verzeichnis ermittelt, das einer zweiten Tenant-Kennung in dem Satz von Tenant-Kennungen entspricht, wobei:  
die zweite Tenant-Kennung einer zweiten Lese-/Schreib-Anforderung in dem Satz von Lese-/Schreib-Anforderungen entspricht und das dritte Verzeichnis mithilfe des ersten Schnittstellenstandards organisiert wird;  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit den Connector-Service dem dritten Verzeichnis und der zweiten Tenant-Kennung zuweist;  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit die zweite Lese-/Schreib-Anforderung mithilfe des Connector-Service und eines zweiten Knotens verarbeitet, wobei:  
ein zweiter Knoten einen zweiten Satz von Dateien in dem zweiten Verzeichnis enthält, und der zweite Satz von Dateien dem zweiten Tenant entspricht; und  
Programmanweisungen, die durch eine Einheit dazu ausführbar sind zu bewirken, dass die Einheit ein zweites Ergebnis zu der zweiten Lese-/Schreib-Anforderung erzeugt.

12. Computersystem nach Anspruch 10 oder 11, wobei das erste Ergebnis aus einer Gruppe ausgewählt wird, die besteht aus:  
einem neuen Dateneintrag, und  
einer Nachricht mit einem Satz von Daten.

13. Computersystem nach einem der Ansprüche 10 bis 12, wobei der erste Schnittstellenstandard nicht kompatibel mit POSIX ist.

14. Computersystem nach einem der Ansprüche 10 bis 13, wobei der zweite Schnittstellenstandard kompatibel mit POSIX ist.

15. Computersystem nach einem der Ansprüche 10 bis 14, wobei das erste Verzeichnis mithilfe eines Apache Hadoop Distributed File System („HDFS“) organisiert wird.

Es folgen 4 Seiten Zeichnungen

## Anhängende Zeichnungen

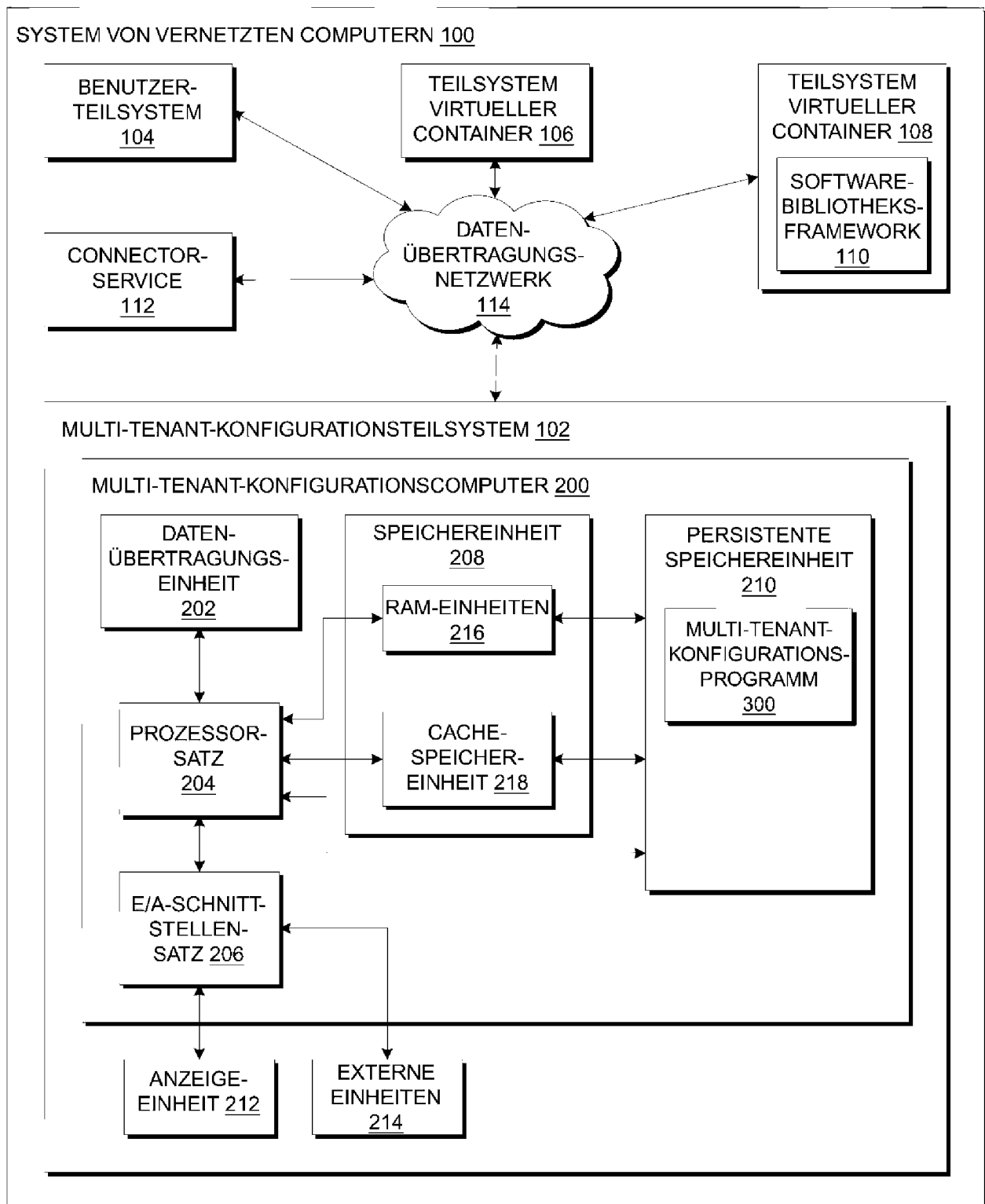


Fig. 1

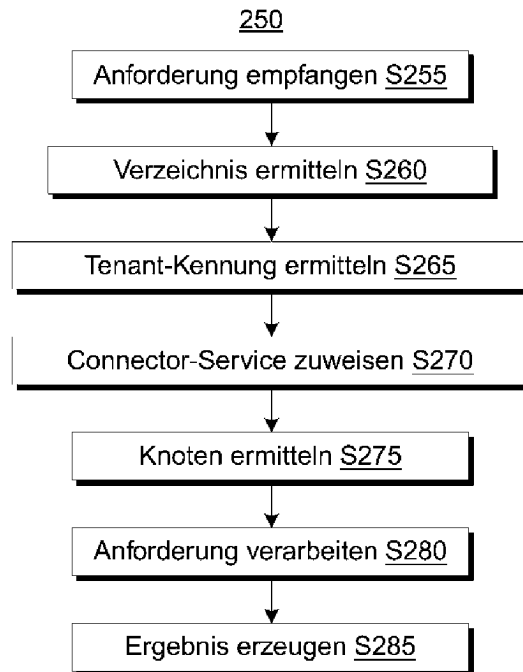


Fig. 2

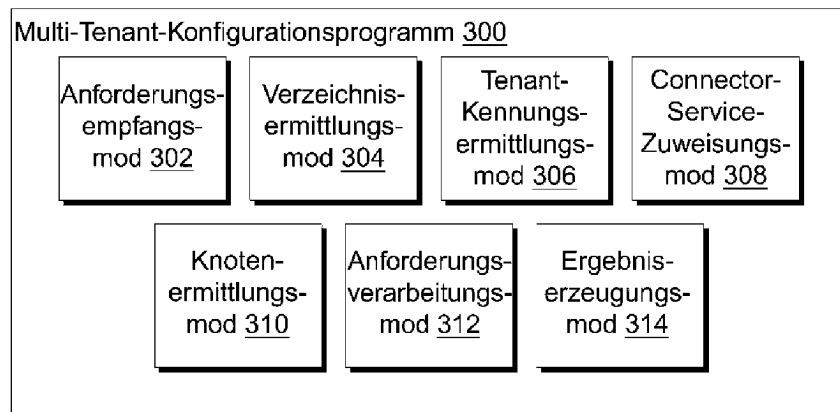


Fig. 3

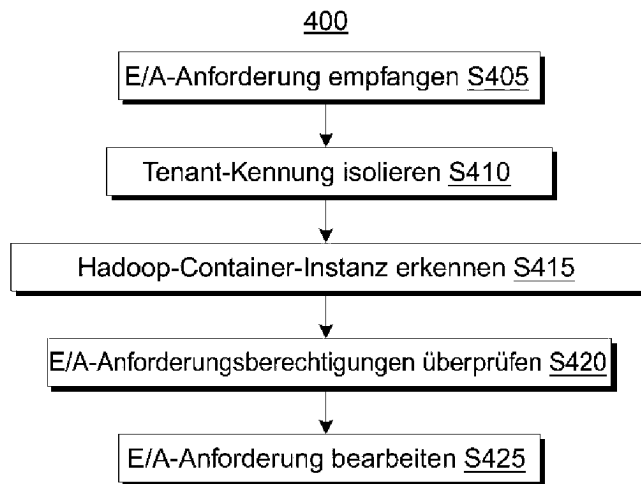


Fig. 4

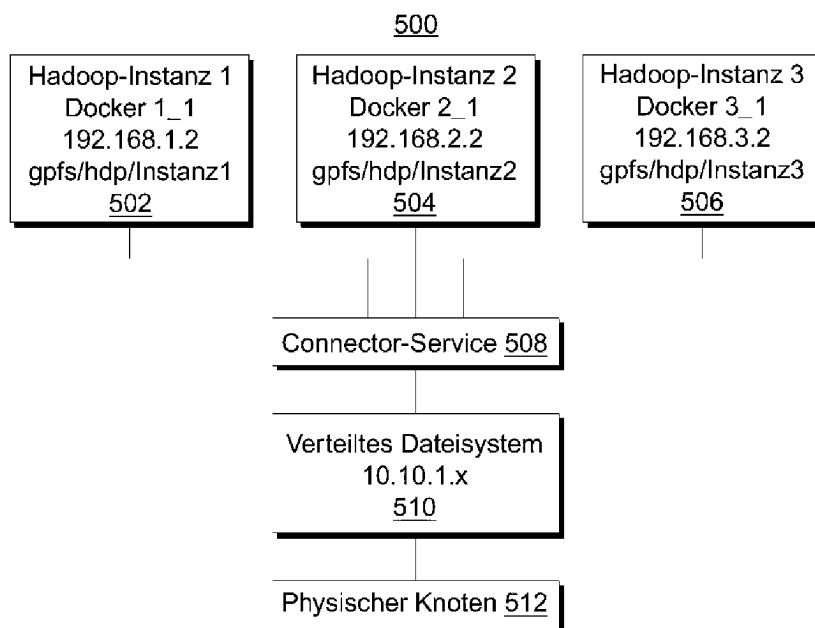


Fig. 5

<b>Instanz</b>	<b>Container</b>	<b>Tenant-ID</b>	<b>Knoten</b>
Instanz 1	Container 1_1	192.168.1.2	Knoten1
Instanz 1	Container 2_1	192.168.1.3	Knoten1
Instanz 1	Container 3_1	192.168.1.4	Knoten1
Instanz 2	Container 1_2	192.168.2.2	Knoten1
Instanz 2	Container 2_2	192.168.2.3	Knoten1
Instanz 2	Container 3_2	192.168.2.4	Knoten1

<b>Knoten</b>	<b>Container</b>	<b>Tenant-ID</b>	<b>Instanz</b>
Knoten1	Container 1_1	192.168.1.2	Instanz 1
Knoten1	Container 2_1	192.168.1.3	Instanz 1
Knoten1	Container 3_1	192.168.1.4	Instanz 1
Knoten1	Container 1_2	192.168.2.2	Instanz 2
Knoten1	Container 2_2	192.168.2.3	Instanz 2
Knoten1	Container 3_2	192.168.2.4	Instanz 2

Fig. 6

700

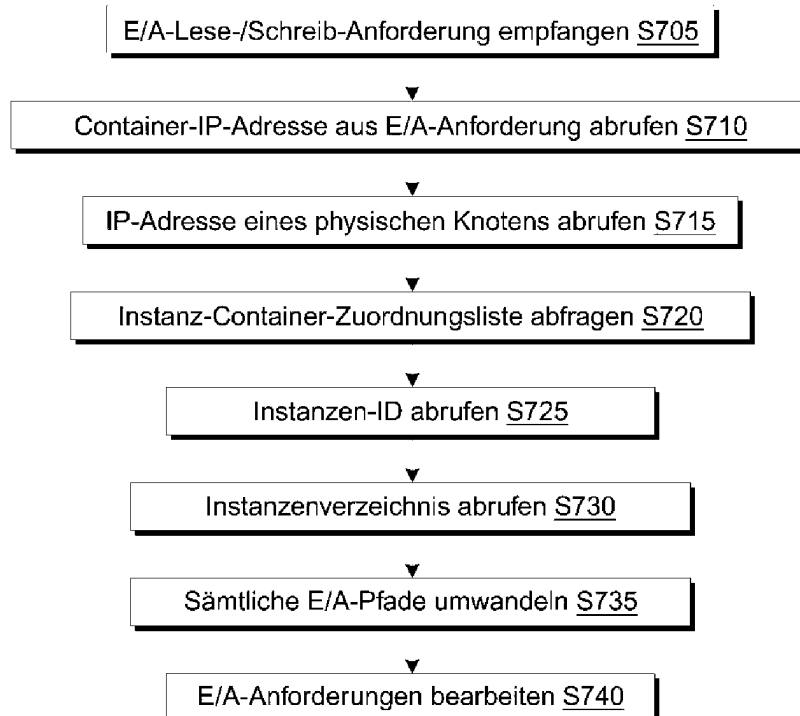


Fig. 7