



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 36 818 T2** 2007.12.27

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 957 616 B1**

(21) Deutsches Aktenzeichen: **699 36 818.9**

(96) Europäisches Aktenzeichen: **99 303 685.4**

(96) Europäischer Anmeldetag: **12.05.1999**

(97) Erstveröffentlichung durch das EPA: **17.11.1999**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **15.08.2007**

(47) Veröffentlichungstag im Patentblatt: **27.12.2007**

(51) Int Cl.⁸: **H04L 12/24** (2006.01)
H04L 29/06 (2006.01)

(30) Unionspriorität:

79499 14.05.1998 US

(84) Benannte Vertragsstaaten:

DE, GB

(73) Patentinhaber:

Sun Microsystems, Inc., Palo Alto, Calif., US

(72) Erfinder:

**Schmidt, Jeffrey A., Boulder Creek, CA 95006, US;
Saulpaugh, Thomas, San Jose, CA 95120, US;
Traversat, Bernard A., San Francisco, CA 94109,
US; Slaughter, Gregory L., Palo Alto, CA 94306, US**

(74) Vertreter:

HOFFMANN & EITLE, 81925 München

(54) Bezeichnung: **Protokoll zum Austausch von Konfigurationsdaten in einem Computernetzwerk**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung**HINTERGRUND DER ERFINDUNG****1. GEBIET DER ERFINDUNG**

[0001] Die vorliegende Erfindung bezieht sich allgemein auf Computersoftware und Computernetz-anwendungen. Genauer gesagt, dieselbe bezieht sich auf Client-Server-Anwendungen und den Austausch von Konfigurationsdaten zwischen Komponenten in einem Computernetz.

2. ERÖRTERUNG DER VERWANDTEN TECHNIK

[0002] Ein Typ eines herkömmlichen Computernetzes betrifft ein Verbinden einer Reihe von Personalcomputern, auf die als Clients Bezug genommen ist (zum Beispiel Sun SPARC-Workstations oder IBM-PCs), mit einem oder mehreren Server-Computern. Die Client-Computer sind allgemein unabhängig und enthalten in dem eigenen Speicher derselben viele der Informationen, die benötigt werden, um Benutzeranwendungen laufen zu lassen und Netzoperationen durchzuführen. Das heißt, dass dieselben Informationen bezüglich der eigenen Konfiguration derselben im Hinblick auf sowohl Software- als auch Hardwarefähigkeiten und -erfordernisse enthalten. Die Client-Computer greifen typischerweise aus einer Vielfalt von Gründen, wie beispielsweise ein Zugreifen auf Netzsoftwareanwendungen, ein Senden und Empfangen von E-Mails, ein Wiedergewinnen und Speichern von Informationen in einer Netzdatenbank, auf die Netz-Server zu. Informationen, die für einen speziellen Client-Computer spezifisch sind, befinden sich jedoch allgemein in dem Client-Computer. Diese Informationen können beispielsweise die Menge an Speicher oder Datenbustypen, Hardware-Spezifikationen, wie einen Datenbustyp oder zusätzliche Prozessoren, umfassen. Da die Client-Computer relativ unabhängig sind und die eigenen Konfigurationsinformationen derselben speichern (und diese daher in dem Server-Computer nicht verfügbar sind), wurde die Aufgabe einer Daten- und Anwendungsverwaltung in einem Client-Computer immer beschwerlicher.

[0003] Obwohl es möglich ist, geringfügige Änderungen oder Reparaturen von Anwendungen, die sich in einem Server in dem Netz befinden, zu den Client-Computern zu verbreiten, erfordert jede wesentliche Erweiterung oder Reparatur, oder eine Installation einer neuen Anwendung, die jeden Client beeinflusst, dass auf jeden Client-Computer durch einen Netzverwalter einzeln zugegriffen und derselbe aktualisiert wird. Da eine zunehmende Zahl von Computern, die in manchen Unternehmen in die Zehntausende reicht, in Netzen verbunden wird, wurde die Belastung, größere Bearbeitungen oder Erweiterungen an Anwendungssoftware oder an alle-

meiner Konfigurationssoftware zu installieren, aufwendig, ineffizient und zeitaufwendig. Da die meisten Client-Computer unabhängig sind, ist es für Benutzer, die unterschiedliche Client-Computer an unterschiedlichen Orten verwenden müssen, zusätzlich schwer, persönliche Vorlieben im Hinblick auf die Anwendungen und Konfigurationsdaten aufrechtzuerhalten. Selbst wenn ein Benutzer persönliche Vorlieben als Vorgaben in dem normalerweise verwendeten Client-Computer desselben installieren kann, ist es möglich, diese Vorgaben in andere Client-Computer zu kopieren, ohne Vorgaben in diesen Computern zu ändern.

[0004] Ein anderer Typ einer Computernetzkonfiguration betrifft die Verwendung von nichtintelligenten Endgeräten oder „Thin“-Clients, die typischerweise mit einem Mainframe-Computer verbunden sind. Bei diesem Netztyp befinden sich das gesamte Verarbeiten und Daten in dem Mainframe-Computer. Der „Thin“-Client führt keine von diesen Tätigkeiten durch. Bei diesem Konfigurationstyp stehen alle Informationen hinsichtlich des Clients unter einer Steuerung des Mainframe-Computers. Wenn die Verbindung zwischen dem Client und dem Mainframe beendet wird, hört alles Verarbeiten auf, und der Client kann keine Tätigkeit ausführen.

[0005] Wie im Vorhergehenden beschrieben ist, ist das Verfahren des Installierens einer neuen Software oder neuer Anwendungen bei der herkömmlichen Netzkonfiguration ein statisches Verfahren. Bei einer solchen Konfiguration sind die Konfigurationsinformationen für jeden PC in jedem Client-Gerät definiert. Der Netzverwalter muss daher jede Konfiguration auf jedem PC statisch definieren. Bei einer herkömmlichen Computernetzkonfiguration sind Konfigurationsinformationen für jedes spezielle Untersystem oder jeden Client in dem speziellen Client hartkodiert. Weiterhin erfordert eine Anwendungswartung, wie ein Installieren neuer Versionen oder größerer Erweiterungen an einer Software, bei der die Erweiterung eine Kenntnis oder einen Zugriff auf eine Untersystemkonfiguration erfordert, bei herkömmlichen Netzkonfigurationen, die unabhängige Clients, die mit Netz-Servern verbunden sind, verwenden, normalerweise, dass die Anwendung oder das Netz „heruntergefahren“ wird, um die Wartung vorzunehmen.

[0006] Bei herkömmlichen Computernetzen, die mehrere Clients und einen Server haben, wobei der Server Informationen enthält, die aus verschiedenen Gründen durch einen Client benötigt werden, werden in vielen Fällen alle Daten in dem Server, die durch einen Client benötigt werden oder für denselben relevant sind, von dem Server zu dem Client bewegt. Dies kann typischerweise ein Bewegen großer Mengen an Daten, von welchen viele möglicherweise nicht verwendet werden oder für den Client, um in Betrieb zu sein, nicht notwendig sind, betreffen. Ein

Übertragen aller Daten zu dem Client ist ineffizient und lässt einen Netzverkehr zunehmen. Zusätzlich muss der Client ausreichend Arbeitsspeicher und Speicher haben, um alle Informationen, die sich auf diesen speziellen Client beziehen, von dem Server zu speichern. Vorrichtungen wie PDA und intelligente Karten (engl.: smart cards), die keine großen Mengen an Speicher haben, können beispielsweise in dem Arbeitsspeicher derselben nicht alle Informationen, die Konfigurationsinformationen umfassen, die für diese spezielle Vorrichtung relevant sein könnten, enthalten.

[0007] TRAVERSAT B: „Java System Database Server“, verfügbar unter URL: <http://www.open-group.org/ncmg/members/meetings/apr98/jsds-pres.pdf>, offenbart eine Client/Server-Architektur für eine Java-System-Datenbank, bei der der Client Konfigurationsinformationen, die für die Hardware-Plattform und den aktuellen Benutzer spezifisch sind, speichert und der Server Konfigurationsinformationen für alle unterstützte Hardware und alle Benutzer speichert. Das Dokument nimmt auf ein „JSD Client Server Schema Mapping“, bei dem Client-Konfigurationsinformationen mit Server-Konfigurationsinformationen vereinigt sind, Bezug.

[0008] Die GB-A-2311389 offenbart ein Verfahren und ein System zum automatischen Installieren von Betriebssystemsoftware in einem Client-Gerät, wenn das Client-Gerät gestartet wird. Die Konfiguration eines Clients kann vor einer Installation des Betriebssystems durch den Server überprüft werden. Der Server, der einen neuen Client in dem Netz erfasst, lädt einen Scan-Code auf den Client herunter. Der Client lässt dann den Scan-Code laufen und sendet Hardwarekonfigurationsdaten zu dem Server zurück. Basierend auf diesen Informationen weist der Server dem Client ein Softwareprofil zu und lädt ein Installationsprogramm und ein Betriebssystem auf den Client herunter. Der Client lässt dann das Installationsprogramm laufen, um die Betriebssystemsoftware auf die Client-Harddatei (engl.: hardfile) zu laden.

[0009] Daher wäre es wünschenswert, ein System zu haben, das eine verteilte Verwaltung von Client-Konfigurationen durch Speichern solcher Konfigurationsinformationen an einem zentralen Verwahrungsort unterstützt. Dies würde einem Netzverwalter ermöglichen, Untersystemkonfigurationen von dem Server zu verwalten und alle Typen von Änderungen an Anwendungen von einem Server zu verbreiten. Es wäre ferner wünschenswert, einem Netzbutzer zu ermöglichen, sich bei unterschiedlichen Clients anzumelden und immer noch fähig zu sein, auf jedem Client in dem Netz auf persönliche Vorlieben und Profile dieses Benutzers zuzugreifen.

[0010] Weiterhin wäre es wünschenswert, ein Verfahren zu haben, das die Übertragung von Daten von

einem Server zu einem Client auf eine effiziente und voll betriebsbereite Art und Weise ermöglicht. Ferner wäre es wünschenswert, Daten zu dem Client zu senden, die eine verdichtete und kompakte Form haben, in dem Sinne, dass der Client keine sich wiederholenden Daten empfängt.

[0011] Um das im Vorhergehenden Beschriebene zu erreichen, und gemäß dem Zweck der vorliegenden Erfindung wird ein Verfahren nach Anspruch 1 geschaffen.

[0012] Gemäß einem Aspekt der Erfindung wird ein Verfahren zum Konfigurieren eines Client-Computers in einem Computernetz, das einen Server-Computer umfasst, geschaffen, wobei das Verfahren folgende Schritte aufweist: Einrichten einer Verbindung zwischen dem Client-Computer, der ein Profil hat, und dem Server-Computer; Austauschen von Versionsinformationen zwischen dem Client-Computer und dem Server-Computer und Bestimmen einer Version einer Software, die verwendet werden soll, um Daten zwischen dem Client-Computer und dem Server-Computer auszutauschen; Senden des Profils zu dem Server; Wiedergewinnen von Konfigurationsinformationen, die sich auf den Client-Computer beziehen, wobei die Konfigurationsinformationen unter einer Steuerung des Server-Computers gespeichert werden und gemäß dem Profil abgeleitet werden; und Konfigurieren des Client-Computers unter Verwendung der Konfigurationsinformationen und dadurch Erleichtern einer zentralisierten Verwaltung der Konfigurationsinformationen durch den Server-Computer.

[0013] Diese Anwendung beschreibt ein Datenschema, das eine n-astige Baumtyp-Struktur mit einer Wurzelknotenschicht, Zwischenknotenschichten und einer Datenschicht zum Speichern von Konfigurationsdaten hat. Die Zwischenknotenschichten enthalten eine Vielzahl von Knoten, die kategorische Informationen, die sich auf Komponenten und verschiedene Aspekte des Computernetzes beziehen, enthalten. Einer Baumstruktur folgend hat jeder Zwischenknoten und Wurzelknoten Äste, die zu darunter liegenden Knoten ausstrahlen. Auf diese untergeordneten Knoten wird als Kindknoten Bezug genommen. Die Datenknotenschicht befindet sich in dem untersten Teil des Baums und enthält tatsächliche spezifische Konfigurationsdaten, die sich auf Komponenten und andere Aspekte des Computernetzes beziehen. Bestimmte Abschnitte der Zwischenknoten und Datenknoten bilden Dauerspeicherbereiche, in denen die tatsächlichen spezifischen Konfigurationsdaten in den Datenknoten entweder auf einem Client- oder in einem Server-Computer modifiziert und in dem Server-Computer gespeichert werden und dadurch die zugeordneten spezifischen Informationen nichtflüchtig und für eine Vielzahl von Client-Computern zugänglich gemacht werden.

[0014] Bei einem Ausführungsbeispiel der vorliegenden Erfindung hat das Datenschema oder -rahmenwerk eine Client-Schema-Unterkomponente, die sich in einem Client-Computer befindet, und eine Server-Schema-Unterkomponente, die sich in einem oder mehreren Server-Computern befindet. Bei noch einem anderen Ausführungsbeispiel sind Benutzer-vorlieben und -profile in dem Server-Schema gespeichert und ermöglichen dadurch einem Benutzer in dem Netz, jeden beliebigen der Client-Computer in dem Netz zu verwenden und auf die Vorlieben des Benutzers zuzugreifen. Ferner sind eine Client-Computer-Plattform und Profilinformationen in dem Server-Schema gespeichert. Bei noch einem anderen Ausführungsbeispiel enthalten die Dauerspeicherbereiche eine Vielzahl von Dateneinträgen, wobei jeder Dateneintrag einen Namen, eine Liste von verwandten Knoten, Eigenschaftsnamen und zugeordnete Eigenschaftswerte hat.

[0015] Diese Anwendung beschreibt ferner ein Datenschema zum Anordnen und Speichern von Informationen, die sich auf einen Client-Computer in einem Computernetz, das einen Server-Computer hat, beziehen. Das Datenschema umfasst einen Wurzelknoten und eine Vielzahl von Zwischenknotenebenen, die Zwischenknoten enthalten. Jeder Zwischenknoten stellt kategorische Informationen, die sich auf das Computernetz und den Client-Computer beziehen, dar oder speichert dieselben. Das Schema enthält ferner tatsächliche Konfigurationsinformationen, die den kategorischen Informationen in den Zwischenknoten entsprechen. Die kategorischen und Konfigurations-Informationen befinden sich in dem Client-Computer und ermöglichen dem Client-Computer dadurch, sich an verschiedene Profile und Vorlieben, die bestimmt werden, wenn der Client-Computer mit Strom versorgt und mit dem Netz verbunden ist, anzupassen.

[0016] Bei einem Ausführungsbeispiel umfasst die Vielzahl von Zwischenknotenebenen eine oberste Zwischenknotenebene, die oberste Zwischenknoten hat. Jeder oberste Zwischenknoten definiert einen Datenraum, der eine Vielzahl von Dateneinträgen hat. Bei einem anderen Ausführungsbeispiel ist einer der Datenräume, auf den als der Software-Datenraum Bezug genommen ist, ein Dauerdatenraum, der ermöglicht, dass Dateneinträge in dem Datenraum abgespeichert werden, wenn der Client-Computer abgeschaltet oder deaktiviert wird.

[0017] Diese Anwendung beschreibt ferner ein Datenschema zum Anordnen und Speichern von Konfigurationsinformationen; die sich auf eine Vielzahl von Client-Computern in einem Computernetz, das einen Server-Computer hat, der das Datenschema speichert, beziehen. Das Datenschema umfasst einen Wurzelknoten und eine Vielzahl von Zwischenknotenebenen, die Zwischenknoten enthalten. Jeder Zwi-

schenknoten stellt kategorische Informationen, die sich auf das Computernetz und Client-Computer in dem Netz beziehen, dar oder speichert dieselben. Das Schema enthält ferner tatsächliche Konfigurationsinformationen entsprechend den kategorischen Informationen in den Zwischenknoten, die in einer Datenebene des Schemas angeordnet sind. Die kategorischen und Konfigurationsinformationen befinden sich in dem Server-Computer und ermöglichen dem Server-Computer dadurch, Client-Computer-Profilen und -Vorlieben zu verbreiten, wenn dies durch einen Client-Computer angefragt wird.

[0018] Bei einem Ausführungsbeispiel umfasst die Vielzahl von Zwischenknotenebenen eine oberste Zwischenknotenebene, die oberste Zwischenknoten hat. Jeder oberste Zwischenknoten definiert einen Datenraum, der eine Vielzahl von Dateneinträgen hat. Bei einem anderen Ausführungsbeispiel gibt es zwei Datenräume, einen Gerätedatenraum und einen Benutzerdatenraum, die beide Dauerdatenräume sind, was Dateneinträgen in dem Datenraum ermöglicht, abgespeichert zu werden, wenn der Client-Computer abgeschaltet oder deaktiviert wird. Bei noch einem anderen Ausführungsbeispiel speichert der Gerätedatenraum Computerkonfigurationsdaten, die sich auf die Vielzahl von Computertypen in dem Netz beziehen, und der Benutzerdatenraum speichert Benutzerkonfigurationsdaten, die sich auf die Vielzahl von Benutzern, die registriert sind, um auf das Computernetz zuzugreifen, beziehen.

[0019] Diese Anwendung beschreibt ferner Verfahren, Vorrichtungen und ein computerlesbares Medium zum Vorbereiten von Daten für eine Sendung über ein Computernetz, die vorgesehen sind. Ein Datensatz, der Werte, die sich auf das Profil und die Plattform eines Client-Computers beziehen, enthält, wird in einem Server-Computer wiedergewonnen. Ein anderer Datensatz, der Werte enthält, die sich auf eine Benutzervorliebe und eine Benutzergruppe, die einem Benutzer, der den Client-Computer verwendet, zugeordnet sind, beziehen, wird ebenfalls in dem Server-Computer wiedergewonnen. Die zwei Datensätze sind insofern verwandt, dass dieselben einem Benutzer entsprechen, der sich bei einem speziellen Client-Computer anmeldet. Eine Datenelemententsprechung wird durch Zuordnen von Datenelementen in den zwei Datensätzen auf einer Eigenschaft-für-Eigenschaft-Basis erzeugt. Ein Satz von Regeln, der eine Liste von Prioritäten unter Datenkategorien enthält, wird auf die Datenelemententsprechung angewendet. Ein endgültiger Datensatz wird aus einem Anwenden des Regelsatzes auf die Datenelemententsprechung abgeleitet und wird über das Netz zu einem Zielcomputer gesendet. Durch Senden des endgültigen Datensatzes über das Netz wird die Menge von Daten, die gesendet würden, wenn die ersten zwei Datensätze in vollem Umfang gesendet worden wären, reduziert.

[0020] Bei einem Ausführungsbeispiel wird zwischen dem Client-Computer und dem Server-Computer eine Verbindung eingerichtet. Ein Client-Profil wird von dem Client-Computer zu dem Server-Computer gesendet. Bei einem anderen Ausführungsbeispiel umfasst der Satz von Regeln eine Reihenfolge von Vorlieben zum Bestimmen, welche Datenelemente von den ersten zwei Datensätzen in dem endgültigen Datensatz umfasst sind. Bei noch einem anderen Ausführungsbeispiel ist der Client-Computer ein Netzcomputer.

[0021] Diese Anwendung beschreibt ferner Verfahren, Vorrichtungen und ein computerlesbares Medium zum Konfigurieren eines Computers in einem Netz, das einen Server-Computer hat, die vorgesehen sind. Eine Verbindung zwischen dem Computer, der ein Profil hat, und einem Server-Computer wird eingerichtet. Das Profil des Computers wird zu dem Server gesendet. Der Server gewinnt dann Konfigurationsinformationen, die unter einer Steuerung des Servers gemäß dem Profil gespeichert werden, wieder. Der Computer wird gemäß den Konfigurationsinformationen konfiguriert, die in dem Server wiedergewonnen und zu dem Computer gesendet werden, wodurch eine zentralisierte Verwaltung von Konfigurationsinformationen durch den Server-Computer erleichtert wird.

[0022] Bei einem Ausführungsbeispiel umfasst das Profil Hardwareinformationen, die sich auf den Client-Computer beziehen, und Informationen, die sich auf einen Benutzer des Client-Computers beziehen. Bei einem anderen Ausführungsbeispiel tauschen der Client-Computer und der Server-Computer Versionsinformationen aus, die einer Software zugeordnet sind, die verwendet werden soll, um Daten zwischen den zwei Computern auszutauschen. Die exakte Version der zu verwendenden Software wird durch den Server-Computer bestimmt. Bei noch einem anderen Ausführungsbeispiel ist der Client-Computer ein Netzcomputer. Bei noch einem anderen Ausführungsbeispiel speichert der Server-Computer Konfigurationsinformationen, die sich auf eine Vielzahl von Client-Computern in dem Netz und auf Benutzer, die registriert sind, um den Computer zu verwenden, beziehen.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0023] Die Erfindung ist zusammen mit weiteren Vorteilen derselben am besten unter Bezugnahme auf die folgende Beschreibung in Verbindung mit den beigefügten Zeichnungen zu verstehen, in denen:

[0024] [Fig. 1](#) ein Blockdiagramm ist, das Komponenten einer Computernetzkonfiguration zeigt, die ein systemweites Datenschema gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt.

[0025] [Fig. 2](#) eine Darstellung einer n-astigen Baumstruktur ist, die eine Client-Schema-Hierarchie gemäß einem Ausführungsbeispiel der vorliegenden Erfindung darstellt.

[0026] [Fig. 3](#) ein Blockdiagramm ist, das eine Struktur eines JSD-Server-Schemas gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt.

[0027] [Fig. 4](#) ein Blockdiagramm ist, das eine Baumstruktur eines Namenraums Gerät in einem Server-Schema zeigt.

[0028] [Fig. 6a-Fig. 6c](#) Flussdiagramme sind, die ein Verfahren zum Einrichten einer Verbindung zwischen einem Client und einem Server und Austauschen von Konfigurationsinformationen gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigen.

[0029] [Fig. 7](#) ein Flussdiagramm ist, das ein Verfahren zum Verhandeln einer Kommunikationsverbindung zwischen einem Client und einem Server gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt.

[0030] [Fig. 8a](#) ein Flussdiagramm ist, das die Schritte **617** und **627** von [Fig. 6](#) detaillierter zeigt.

[0031] [Fig. 8b](#) ein Blockdiagramm ist, das die Hierarchie, die verwendet wird, um Daten gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zu vereinigen, zeigt.

[0032] [Fig. 9a](#) und [Fig. 9b](#) Flussdiagramme sind, die ein Verfahren zum periodischen Senden von Signalen zwischen einem Server und einem Client, um anzuzeigen, dass die Verbindung zwischen denselben immer noch lebt, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigen.

[0033] [Fig. 10](#) ein Flussdiagramm eines Verfahrens einer Ereignisbenachrichtigung in dem Client/Server-Protokoll gemäß einem Ausführungsbeispiel der vorliegenden Erfindung ist.

[0034] [Fig. 11](#) ein Blockdiagramm eines Allzweck-Computersystems ist, das zum Implementieren des beschriebenen Ausführungsbeispiels der vorliegenden Erfindung geeignet ist.

[0035] Es wird nun auf ein bevorzugtes Ausführungsbeispiel der Erfindung detailliert Bezug genommen. Ein Beispiel des bevorzugten Ausführungsbeispiels ist in den beigefügten Zeichnungen dargestellt. Obwohl die Erfindung in Verbindung mit einem bevorzugten Ausführungsbeispiel beschrieben ist, ist verständlich, dass nicht beabsichtigt ist, die Erfindung auf ein bevorzugtes Ausführungsbeispiel zu begrenzen. Es ist im Gegenteil beabsichtigt, Alternativen, Mo-

difikationen und Äquivalente, die in dem Schutzbereich der Erfindung, wie durch die angefügten Ansprüche definiert ist, umfasst sein können, abzudecken.

[0036] Ein Datenrahmenwerk oder -schema und ein zugeordnetes Protokoll zum Austauschen von Daten, die sich in dem Schema zwischen Computern in einem Computernetz befinden, sind in den verschiedenen Zeichnungen beschrieben. Die vorliegende Erfindung offenbart eine Hierarchie oder ein Datenschema zum Darstellen und Speichern von Konfigurationsinformationen und verwandten Informationen in einer Systemdatenbank. Zum Zweck des Darstellens eines Ausführungsbeispiels der vorliegenden Erfindung wird eine Java-System-Datenbank (JSD) untersucht. Bei anderen bevorzugten Ausführungsbeispielen kann die Systemdatenbank auf anderen Typen von Plattformen in Betrieb sein. Die JSD des beschriebenen Ausführungsbeispiels ist ein einzelnes Untersystem, das mindestens zwei größere Unterkomponenten, oder Unterschemas, das Client-Schema und das Server-Schema, umfasst. Bei dem beschriebenen Ausführungsbeispiel sind Daten, die sich auf einen Client beziehen, in einem Client-Schema, das sich in dem Client-Speicher befindet, gespeichert. Konfigurationsdaten für jeden der Clients sind in einem Server-Schema, das sich in einem Netz-Server befindet, gespeichert. Konfigurationsdaten werden zwischen den zwei Schemas oder Hierarchien durch ein Client/Server-Protokoll, das sicherstellt, dass die korrekten Informationen aus dem Server-Schema abgeleitet und zu dem Client-Schema in dem Client-Gerät transportiert oder in dasselbe eingegeben werden und umgekehrt, ausgetauscht. Die Konfigurationsinformationen für jeden Client, auf den ferner als ein Untersystem Bezug genommen ist, sind in dem Server-Schema gespeichert. Dies steht im Gegensatz zu herkömmlichen Netzen, bei denen Konfigurationsinformationen hinsichtlich eines Clients hartkodiert oder in dem Client-Gerät gespeichert sind. Das Datenschema der vorliegenden Erfindung ermöglicht einem Netzverwalter, Konfigurationsinformationen für jeden der Computer in dem Netz von einem zentralen Verwahrungsort, wie einem einzelnen Server, aus zu verwalten. Softwareaktualisierungen, Versionserweiterungen oder eine Installation von neuen Anwendungen, die eine Kenntnis von und einen Zugriff auf eine Untersystemkonfiguration erfordern, können daher von dem zentralen Verwahrungsort aus implementiert und zu den einzelnen Clients verbreitet werden. Benutzer an den Client-Geräten werden Anwendungen nicht verlassen müssen, und außerdem braucht das Netz zur Wartung nicht heruntergefahren zu werden, um die neue Erweiterung oder Version der Anwendung zu installieren oder zu verbreiten.

[0037] [Fig. 1](#) ist ein Blockdiagramm, das Komponenten einer Computernetzkonfiguration zeigt, die

ein systemweites Datenschema gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt. Bei dem beschriebenen Ausführungsbeispiel ist das systemweite Datenschema als eine Java-System-Datenbank JSB (**101**) dargestellt, die aus einem Client-Schema **103** besteht, das sich in einem Client-Gerät **105** als ein Teil eines Netzes **107** befindet. Ein Server-Schema **111** befindet sich in einem Server-Computer **109**, der ein Teil des Netzes **107** ist.

[0038] [Fig. 2](#) ist eine Darstellung einer n-astigen Baumstruktur, die eine Client-Schema-Hierarchie **103** gemäß einem Ausführungsbeispiel der vorliegenden Erfindung darstellt. Die Hierarchie des Client-Schemas **103**, ebenso wie die Hierarchie eines Server-Schemas **111**, ist unter Verwendung eines n-astigen Baums aufgezeigt. An der Wurzel des Baums ist ein Wurzeleintrag **201**, der keine Daten enthält und das einzige Knotenrahmenwerk in der Hierarchie ist, das auf sich selbst Bezug nimmt. Eine erste Ebene von Knoten **203** in dem Client-Schema **103** definiert gemeinsam einzelne Namenräume in dem allgemeinen Client-Schema. Die erste Ebene **203** in der Hierarchie ist unter dem Wurzeleintrag **201** und enthält die Namenraumeinträge.

[0039] Bei dem beschriebenen Ausführungsbeispiel gibt es in dem allgemeinen Client-Schema **103** sechs Namenräume. Bei anderen bevorzugten Ausführungsbeispielen können abhängig von den Bedürfnissen einer speziellen Netzkonfiguration mehr oder weniger Namenräume vorhanden sein. Bei dem beschriebenen Ausführungsbeispiel lauten die Standardnamenräume der obersten Ebene für den JSD-Client SOFTWARE, VORRICHTUNG, SCHNITTSTELLE, SOFTWAREKONFIGURATION, ALIAS, und TEMP. Der Namenraum SOFTWARE beginnt beispielsweise bei einem Knoten **205** und umfasst alle Knoten und Daten, die von dem Knoten **205** abzweigen. Die spezifischen Einträge in der Schicht **203** der Hierarchie sind Wurzeln von Unterbäumen, die die eindeutigen Namenräume definieren. Alle Einträge in einem bestimmten Namenraum, wie SOFTWARE, sind Einträge, die sich auf Konfigurationsdaten hinsichtlich von Softwareanwendungen für den Client **105** beziehen. Einträge in dem Datenschema der vorliegenden Erfindung sind aus einem eindeutigen Namen, einer Liste von Kindern (Einträge unter dem gegebenen Eintrag) und einem Satz von Tupeln gebildet. Jedes Tupel enthält einen Eigenschaftsnamen und einen zugeordneten Eigenschaftswert. Bei einem Textverarbeitungsprogramm kann beispielsweise ein Eigenschaftsname „Schriftart“ sein, und der Eigenschaftswert kann Palentino sein. Ähnlich sind alle Einträge unter einem Namenraum VORRICHTUNG **207** Einträge, die sich auf Konfigurationsinformationen eines Client-Geräts **105**, in dem sich das Client-Schema **103** befindet, beziehen. Jeder Eintrag in der Hierarchie kann sowohl als ein Eintrag in einem Unterbaum als auch als die

Wurzel eines Unterbaums, der Abkömmlingeinträge oder Kindknoten hat, wirken. Jeder Namenraum in der Schicht **203** ist in der vorläufigen US-Anmeldung mit dem Titel „JAVA SYSTEM DATABASE“, die hiermit an dem gleichen Tag eingereicht und gemeinschaftlich übertragen wurde, beschrieben.

[0040] Wie in [Fig. 2](#) gezeigt ist, hat jeder Eintrag in dem Baum einen einzelnen Elternteil und kann mehrere Kindknoten haben. Ein Namenraum wie ein Namenraum SOFTWARE **209** ist ein besonders bestimmter Unterbaum, der Einträge umfasst, die sich auf Konfigurationsdaten über eine Software für einen speziellen Client, wie den Client **105**, beziehen. Wie in [Fig. 2](#) bei dem beschriebenen Ausführungsbeispiel gezeigt ist, sind Namenräume immer direkte Abkömmlinge eines Wurzeleintrags **201**, auf den auch als eine Superwurzel Bezug genommen ist. Bei anderen bevorzugten Ausführungsbeispielen können Namenräume auf anderen Ebenen in der Hierarchie definiert sein und müssen nicht direkt von der Wurzel **201** abstammen. Die Standardnamenräume des JSD-Schema-Client werden während des Hochstartens oder der Startprozedur des Client-Computers erzeugt. Jeder der Namenräume bei dem beschriebenen Ausführungsbeispiel ist in allen Implementierungen der Java-Plattform verfügbar. Die sechs Namenräume sind gut bekannte Namenräume, die durch die Java-Plattform initialisiert werden. Andere dynamisch aufgebaute Namenräume können nach der Initialisierung zu den Standarddatenbanknamenräumen hinzugefügt werden.

[0041] Jeder Namenraum wird durch einen Vorgabenamenraumverwalter verwaltet. Der Namenraumverwalter steuert, wie die Einträge gespeichert werden und wie auf dieselben innerhalb des Namenraums zugegriffen wird. Der Verwalter implementiert eine Standardschnittstelle, die Sicherheits-, Speicher- und Eigentumsattribute eines jeden Eintrags in dem Namenraum exportiert.

[0042] Der Namenraum SOFTWARE **209** beispielsweise enthält eine Liste von installierten und/oder verfügbaren Systemdiensten, wie Gerätetreiber, Benutzeranwendungen und Benutzerkonfigurationsinformationen. Der Namenraum SOFTWARE ist der einzige Namenraum in dem Client-Schema, der dahingehend dauerhaft ist, dass ein Server einen Sicherungsspeicher für alle Einträge in diesem Namenraum vorsieht. Ein dauerhafter Namenraum oder Eintrag im Gegensatz zu einem vorübergehenden Namenraum ist ein Eintrag, der an einem Dauerspeicherort abgespeichert werden muss. Ein Beispiel von dauerhaften Einträgen sind Konfigurationsinformationen, die sich auf Benutzerumgebungen beziehen und in einem Dauerspeicher gespeichert sein müssen. Wenn sich ein Benutzer oder eine Benutzerin anmeldet, muss seine oder ihre zuletzt abgespeicherte Umgebung wiedergewonnen werden, so dass

derselbe oder dieselbe die Umgebung nicht neu einstellen muss. Dauerhafte Einträge sind Einträge, die von einem permanenten Speicherort aus abgespeichert und wiedergewonnen werden können. Ein Beispiel von dauerhaften Einträgen sind Konfigurationsinformationen, die sich auf Benutzerumgebungen beziehen und in einem Dauerspeicher gespeichert sein müssen. Wenn sich ein Benutzer oder eine Benutzerin anmeldet, muss seine oder ihre zuletzt abgespeicherte Umgebung wiedergewonnen werden, so dass derselbe oder dieselbe die Umgebung nicht neu einstellen muss. Dauerhafte Einträge sind Einträge, die von einem permanenten Speicherort aus abgespeichert und wiedergewonnen werden können. Dauerhafte und vorübergehende Namenräume sind statisch getrennt, wenn Namenräume erzeugt werden. Einem dauerhaften Eintrag ist es nicht ermöglicht, sich in einem vorübergehenden Namenraum zu befinden, und/oder einem vorübergehenden Eintrag ist es nicht ermöglicht, sich in einem dauerhaften Namenraum zu befinden. Bei dem beschriebenen Ausführungsbeispiel sind dauerhafte Einträge auf einem entfernten JSD-Server gespeichert. Bei dem beschriebenen Ausführungsbeispiel gibt es unter dem Namenraum SOFTWARE vier Kategorien: Anwendung, System, Dienst und öffentlich. Bei dem beschriebenen Ausführungsbeispiel sind unter Verwendung der Java-Plattform einige Einträge in dem Namenraum SOFTWARE unter Verwendung von Java-eindeutigen Namensgebungskonventionen und anderen nicht-Java-verwandten Einträgen, die Namensgebungskonventionen basierend auf spezifischen Anwendungen haben, angeordnet. Bei dem beschriebenen Ausführungsbeispiel erhalten Firmennamen wie IBM, Sun oder Lotus Namen wie com.IBM, com.Sun und com.Lotus. Diese Firmennamen unterscheiden firmenspezifische Informationen. Eintragsnamen unter dem Firmeneintrag sind firmenspezifisch.

[0043] Wie beschrieben ist, ist der Namenraum SOFTWARE **209** der einzige Namenraum von den sechs, der bei dem beschriebenen Ausführungsbeispiel einen Dauerspeicher hat. Die anderen Namenräume, wie der Namenraum VORRICHTUNG **207**, haben einen vorübergehenden Speicher. Einträge in diesen Namenräumen gehen verloren, wenn der Client-Computer abgeschaltet wird. Dies gilt für das beschriebene Ausführungsbeispiel, da die fünf vorübergehenden Namenräume Daten speichern, die sich spezifisch auf einen Client-Computer beziehen. Bei dem beschriebenen Ausführungsbeispiel enthält der Namenraum SOFTWARE Anwendungskonfigurationsinformationen, die abgespeichert werden müssen, nachdem der Computer abgeschaltet wird.

[0044] Unter dem Namenraum SOFTWARE existieren vier Kategorien: Anwendung, System, Dienst und öffentlich. Unter Verwendung der Anwendungskategorie als ein Beispiel enthält ein Eintrag com.Netsca-

pe **213** den firmeneindeutigen Namen (zum Beispiel Netscape), und unter einem com.Netscape-Eintrag **215** ist ein Eintrag für ein Netscape-Erzeugnis, Netscape Navigator. Unter dem Eintrag **215** sind firmenspezifische Informationen **227**, die sich auf den Netscape Navigator beziehen.

[0045] Die Einträge **219**, **221** und **223** sind Einträge für andere Verkäufer, die ebenfalls Einträge, die dem Eintrag **215** ähnlich sind, haben werden. Bei dem beschriebenen Ausführungsbeispiel spiegelt die Struktur eines Vorrichtungsnamenraums **225** einige oder alle von dem Eingabe/Ausgabe-Bus und den Vorrichtungen, die in dem Client vorhanden sind, wider. Mit anderen Worten, die physikalische Vernetzungsmöglichkeit von Bussen und Vorrichtungen ist als ein Baum von Einträgen dargestellt, bei dem ein spezieller Bus die Eltern sind und Blatteinträge Konfigurationsdaten über die Vorrichtungen enthalten.

[0046] In dem Softwareramenraum enthält die Blattknotenebene der Hierarchie Daten **227**, die konfigurationsspezifisch sind und je nachdem, wie die Anwendung, beispielsweise Netscape Navigator, die spezifischen Daten in der Blattknotenebene ordnen will, angeordnet sind. Für eine Textverarbeitungsanwendung würden die Blattknoteneinträge spezifische Informationen wie Schriftart, Wörterbuchdefinitionen und andere Konfigurationsdaten eines Textverarbeitungstyps enthalten.

[0047] Die Namenräume in der Server-Schema-Komponente der JSD sind Dauerspeicherräume; das heißt, dieselben bleiben erhalten, nachdem der Client-Computer abgeschaltet wird. Bei dem beschriebenen Ausführungsbeispiel gibt es zwei Namenräume in dem Server-Schema: Gerät und Benutzer. **Fig. 3** ist ein Blockdiagramm, das eine Struktur eines JSD-Server-Schemas gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt. Dasselbe zeigt den Server-Computer **109** und das Server-Schema **111** von **Fig. 1** detaillierter. In dem obersten Teil des n-astigen Baums ist ein Wurzeleintrag **301**, der bei dem beschriebenen Ausführungsbeispiel ferner einen Namenraum KONFIG darstellt. Wie erwähnt ist, gibt es in dem Server-Schema zwei Namenräume. Ein Bereich **303** stellt den Namenraum Gerät, der einen Geräteknoten **305** hat, dar. Ein Bereich **307** stellt den Namenraum Benutzer, der einen Benutzerknoten **309** hat, dar.

[0048] Der Namenraum Gerät **303** ist bei dem beschriebenen Ausführungsbeispiel aus drei Kategorien gebildet. Bei anderen bevorzugten Ausführungsbeispielen kann der Namenraum Gerät abhängig von der Plattform und den Erfordernissen des Netzes mehr oder weniger Unterkategorien haben. Die drei Kategorien oder Unterbereiche sind eine Plattform **311**, ein Bezeichner **313** und ein Profil **315**. Unter der Plattform **311** befindet sich eine Zahl von Einträgen,

die sich auf spezifische Computerhersteller, wie Sun Microsystems und IBM Corporation, beziehen. Dies ist in **Fig. 4** detaillierter gezeigt.

[0049] **Fig. 4** ist ein Blockdiagramm, das eine Baumstruktur des Namenraums Gerät **303** in dem Server-Schema **111** zeigt. Wie erwähnt ist, existieren unter der Kategorie Plattform **311** herstellerspezifische Unterbereiche **401** und **403**. Die Zahl von Einträgen auf dieser Ebene hängt von der Zahl von unterschiedlichen Computerherstellern, die in dem Netz verwendet sind, ab. Unter einem speziellen Hersteller wie com.Sun besteht eine Zahl von Einträgen, die als Einträge **405** und **407** gezeigt sind, wobei sich jeder Eintrag auf ein spezielles Computermodell oder einen speziellen Computertyp, der von diesem Hersteller hergestellt wird, bezieht. Unter com.Sun gibt es beispielsweise den Computertyp JDM1, und unter com.IBM gibt es Computertypen NS1000 und NS2000. Unter jedem Computertyp oder -modell sind Blattknoten **409**, die Anwendungskonfigurationen für diesen speziellen Computertyp spezifizieren. Im Hinblick auf die Unterkategorie Typ enthalten die Anwendungskonfigurationsinformationen in den Blatteinträgen oder Blattknoten alle möglichen Konfigurationen, die auf einen speziellen Computer, das heißt den Computer, der in dem Elterneintrag angezeigt ist, anwendbar sind.

[0050] Unter der Unterkategorie Bezeichner, die den Bezeichnerwurzeleintrag **313** hat, befinden sich Einträge, die eine eindeutige Kennung **411** für jeden Computer in dem Netz **107** enthalten. Bei dem beschriebenen Ausführungsbeispiel ist eine MAC-Adresse für jeden Computer als die eindeutige Kennung verwendet. Die Daten **413** unter einer speziellen Client-Kennung **411** sind Anwendungskonfigurationsinformationen, die für diesen speziellen Computer spezifisch sind. Die Konfigurationsdaten **413** sind von den Konfigurationsdaten **409** unter der Kategorie Plattform dahingehend unterscheidbar, dass sich die Daten **413** unter Kennung auf einen spezifischen Computer beziehen, wie durch einen speziellen Benutzer konfiguriert ist. Bei dem beschriebenen Ausführungsbeispiel gibt es Querverweise (nicht gezeigt) zwischen den eindeutigen Kennungen **411** unter der Kategorie Kennung und den Einträgen unter der Kategorie Plattform. Das heißt, dass es von einer spezifischen Kennung eine Bezugnahme auf einen speziellen Typ von Computer gibt. Dies ermöglicht dem Server zu bestimmen, auf welche Plattform oder welchen Typ von Computer sich eine spezielle eindeutige Kennung bezieht.

[0051] Unter der Kategorie Profil, die einen Profilwurzeleintrag **315** hat, sind Einträge, die spezielle Kategorien oder Verwendungen von Computern in dem Netz beschreiben. Die Konfigurationsinformationen für die speziellen Profile, die beispielsweise auch Abteilungen einer Firma beschreiben können, sind

unter der Unterkategorie des Profils enthalten. Beispiele sind bei den Knoten **415** und **417**, die Profile von Finanzwesen und Vertreib anzeigen, gezeigt. Unter dem Finanzwesenknoten **415** sind anwendungsspezifische Daten **419**, die Daten, die sich auf das Finanzwesenprofil beziehen, enthalten. Ähnlich zu Bezugnahmen von den eindeutigen Kennungen auf die Platfformeinträge gibt es ferner eine Bezugnahme von spezifischen Kennungen auf einen Profileintrag, wenn anwendbar. Das heißt, wenn ein spezieller Computer, wie ein Computer, der in der Buchhaltungsabteilung verwendet wird oder ein Computer, der strikt als ein Rezeptionsdamenendgerät verwendet wird, ein bestimmtes Profil hat, gibt es eine Bezugnahme von der Kennung dieses speziellen Computers auf den entsprechenden Profileintrag.

[0052] [Fig. 5](#) ist ein Blockdiagramm, das einen Namenraum Benutzer gemäß einem Ausführungsbeispiel der vorliegenden Erfindung darstellt. Bei dem beschriebenen Ausführungsbeispiel hat der Namenraum Benutzer **307** zwei Kategorien: Benutzer und Gruppen. Der Benutzerknoten **317** stellt Namen der einzelnen Benutzer in dem Computernetz dar, wie bei den Knoten **501**, **503** und **505** gezeigt ist. Unter dem einzelnen Knoten eines jeden Benutzers stehen spezifische Konfigurationsdaten, die die persönlichen Vorlieben dieses einzelnen Benutzers enthalten, wie bei **507** gezeigt ist. In einer Textverarbeitungsanwendung könnte beispielsweise eine spezielle Benutzervorliebe eine Vorgabeschriftart und -größe für Dokumente sein. Diese Kategorie ermöglicht einem einzelnen Benutzer, jeden Computer in dem Netz **107** zu verwenden und die persönliche Konfiguration dieses Benutzers auf diesem Computer verfügbar zu haben. Wenn der Benutzer beispielsweise ein Textverarbeitungsprogramm hochfährt, werden die Vorlieben des Benutzers die Vorgabe anstatt der Vorgabe des normalen Benutzers dieses Computers sein. Die andere Kategorie in dem Namenraum Benutzer ist die Kategorie Gruppen, die einen Gruppenknoten **319** hat. Diese Kategorie enthält Einträge, die sich auf spezielle Gruppen von Benutzern beziehen. Gruppen können eine Vielzahl von Kategorien umfassen, wie Abteilungen innerhalb einer Firma oder Kategorien, die Angestellte in einer Firma von anderen Angestellten unterscheiden, wie bei den Knoten **509** und **511** gezeigt ist. Bei dem beschriebenen Ausführungsbeispiel gibt es Bezugszeiger, die von den einzelnen Benutzern **503** und **505** unter der Kategorie Benutzer **317** zu einer oder mehreren speziellen Gruppen gehen, wo geeignet.

[0053] Bei der vorliegenden Erfindung werden Daten zwischen einem Client und einem Server in dem Computernetz gemäß einem Client/Server-Protokoll, das in den verschiedenen Figuren beschrieben ist, ausgetauscht. Dieses einzelne Protokoll, das durch einen Pfeil **108** in [Fig. 1](#) zwischen dem Client **105** und dem Server **109** dargestellt ist, wie in [Fig. 1](#) ge-

zeigt ist, ist ein Satz von Regeln und Prozeduren, durch welche zwischen einem Client und einem Server Verbindungen eingerichtet und Daten ausgetauscht werden. Der Server ist bei dem beschriebenen Ausführungsbeispiel fähig, unter Verwendung von vielen unterschiedlichen Protokollen Daten auszutauschen und mit äußeren Diensten und Speichermedien zu kommunizieren. Die Protokolle, die durch den JSD-Server unterstützt werden, sind verschiedenartige Protokolle eines dauerhaften Typs an dem hinteren Ende des Servers. Das Client/Server-Protokoll zwischen dem JSD-Client und dem JSD-Server ist andererseits ausreichend allgemein und breit, um die vielfältige Gruppe von Protokollen zu unterstützen, die gewöhnlich durch den JSD-Server unterstützt werden können, um Daten mit äußeren Einheiten, wie Dauerspeichern, Datenregistern, Datei-Servern und Datenverzeichnissen auszutauschen. Das Client/Server-Protokoll ist in diesem Zusammenhang entworfen, um eine kleine „Platzbedarf“ zu haben. Durch Platzieren der Komplexität des Verwaltens der mehreren Protokolle, die notwendig sind, um mit äußeren Einheiten, die mit dem Server verbunden sind, zu kommunizieren, wird die Belastung, diese Protokolle verwalten zu müssen, nicht auf den einzelnen Clients platziert. Es ist daher ein Zwei-Niveau-Modell beschrieben, bei dem das erste Niveau das Client/Server-Protokoll der vorliegenden Erfindung zwischen dem JSD-Client und dem JSD-Server ist und das zweite Niveau die mehreren Protokolle, die durch den JSD-Server unterstützt werden, und die verschiedenen äußeren Dienste und Datenspeichereinheiten ist.

[0054] [Fig. 6a–Fig. 6c](#) sind Flussdiagramme, die ein Verfahren zum Einrichten einer Verbindung zwischen einem Client und einem Server und Austausch von Konfigurationsinformationen gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigen. Bei einem Schritt **601** von [Fig. 6a](#) versucht ein Client-Computer in dem Computernetz, eine Kommunikationsverbindung mit dem Netz zu eröffnen. Bei diesem Schritt sendet der Client zu dem Netz ein Entdeckungspaket mit der Absicht eines Verbindens mit einem Server aus. Bei dem beschriebenen Ausführungsbeispiel ist das Datenpaket ein DHCP-Entdeckungspaket, das Informationen über den Client enthält. Bei einem Schritt **603** empfängt ein Netzserver das Entdeckungspaket. Bei dem beschriebenen Ausführungsbeispiel enthält der Client-Computer das Client-Schema **103** und der Server-Computer enthält das Server-Schema **111**, die gemeinsam die JSD **101** definieren, wie in [Fig. 1](#) beschrieben ist. Bei einem Schritt **605** liest der Server-Computer die IP-Adresse des Senders (ein Beispiel einer eindeutigen Kennung) des Entdeckungspakets. Bei dem beschriebenen Ausführungsbeispiel wird die IP-Adresse als eine eindeutige ID für den Client, der das Datenpaket sendet, verwendet. Bei anderen bevorzugten Ausführungsbeispielen können

andere Kennungen, wie ein MAC-ID, als eine eindeutige Kennung verwendet sein.

[0055] Bei einem Schritt **607** sendet der Server ein Paket, das angibt, dass der Client nun eine Verbindung mit diesem speziellen Server eingerichtet hat, zu dem Client. Der Server sagt dem Client im Wesentlichen, dass der Client nun mit dem Server kommunizieren wird. Bei einem Schritt **609** sendet der Client ein Verhandlungsdatenpaket zu dem Server. Dieser Schritt ist in [Fig. 7](#) genauer beschrieben. Der Zweck des Verhandlungspakets ist es, die Version der Software, die für eine Kommunikation zwischen dem Client und dem Server verwendet werden wird, einzurichten. Wurde die Version der Software eingerichtet, sendet der Client bei einem Schritt **611** die spezifischen Profilinformationen desselben zu dem Server. In vielen Fällen sind diese spezifischen Profilinformationen in dem Client-PROM typischerweise hartkodiert. Die spezifischen Profilinformationen für einen Client würden beispielsweise den Computertyp, wie eine Sun-JDM1-Workstation oder ein IBM-NS1000-Computer, umfassen. Sobald der Server die Profilinformationen von dem Client empfängt, versucht der Server bei einem Schritt **613** von [Fig. 6b](#), das Profil mit einem spezifischen Profil in dem Namenraum Gerät **303** desselben unter der Unterkategorie Plattform **311** in Übereinstimmung zu bringen. Der Server-Schema-Namenraum Gerät **303** enthält die Kategorie Plattformen **311**, die spezifische Profilinformationen über spezielle Typen von Computern, die von Computerherstellern hergestellt werden, speichert. Bei dem Schritt **613** von [Fig. 6b](#) sucht der Server das Profil des Client in dem Namenraum Gerät. Bei einem Schritt **615** verwendet der Server den eindeutigen ID von dem Client, um zu bestimmen, ob der Client ein spezifisches Profil hat. Nicht zu vergessen ist, dass die Kategorie Kennung in dem Namenraum Gerät eindeutige IDs aller Clients in dem Netz enthält. Wenn der Client ein spezifisches Profil hat, wie Rezeption oder Vertrieb, gibt es einen Bezugszeiger von der eindeutigen Kennung zu dem entsprechenden Profil.

[0056] Bei einem Schritt **617** zieht der Server die Profildaten (wenn vorhanden) und die Plattformdaten zusammen oder kombiniert dieselben und sendet den zusammengezogenen oder vereinigten Dateneintrag zu dem Namenraum SOFTWARE des Client. Ein Verfahren zum Vereinigen der Profildaten und der Plattformdaten ist in [Fig. 8a](#) und [Fig. 8b](#) detaillierter beschrieben. Bei dem beschriebenen Ausführungsbeispiel werden Daten hinsichtlich der Plattform des Computers durch Daten in dem computerspezifischen Profil, wenn es eines für diesen Computer gibt, aufgehoben. Bei einem Schritt **619** besetzt der Client den Namenraum SOFTWARE **209** desselben innerhalb des Client-Schemas **103** desselben mit dem vereinigten Dateneintrag von dem Server. An diesem Punkt ist die Startstufe für den Client-Computer be-

endet.

[0057] Bei einem Schritt **621** meldet sich ein Benutzer bei dem Client-Computer an und startet dadurch die Anmeldephase. An diesem Punkt wird der Name des Benutzers zu dem Server gesendet. Bei einem Schritt **623** nimmt der Server den Benutzernamen, um eine Konfiguration in dem Namenraum Benutzer **307** in dem Server-Schema wiederzugewinnen. Man erinnere sich daran, dass das Server-Schema **111** bei dem beschriebenen Ausführungsbeispiel zwei Namenräume hat: den Namenraum Geräte **303** und den Namenraum Benutzer **307**. Unter dem Namenraum Benutzer gibt es zwei Kategorien: die Benutzer **317** und die Benutzergruppen **319**. Bei einem Schritt **623** sucht der Server unter Verwendung des Namens des Benutzers eine Benutzerkonfiguration. Bei einem Schritt **625** von [Fig. 6c](#) bestimmt der Server, indem derselbe nach Bezugszeigern von dem Benutzereintrag zu einem Eintrag in der Kategorie Gruppe sucht, ob der Benutzer zu einer Gruppe, wie durch den Gruppenknoten **319** dargestellt ist, gehört. Bei einem Schritt **627** werden der Benutzerkonfigurationseintrag und der Gruppenkonfigurationseintrag in einen einzelnen Benutzereintrag vereinigt und zu dem Client gesendet. Bei dem beschriebenen Ausführungsbeispiel heben Daten in dem Benutzerkonfigurationseintrag entsprechende Daten in dem Benutzergruppenkonfigurationseintrag auf. Das Verfahren eines Vereinigens der Daten von dem Benutzereintrag und dem Gruppeneintrag ist in [Fig. 8a](#) und [Fig. 8b](#) detaillierter beschrieben. Bei einem Schritt **629** empfängt der Client den Benutzereintrag und hebt, wenn notwendig, die Daten, die vorher in dem Softwarenamenraum in dem Client-Schema besetzt waren, durch den Plattform/Profil-Eintrag auf. Bei dem beschriebenen Ausführungsbeispiel hebt der Benutzerkonfigurationseintrag, der bei dem Schritt **627** zu dem Client gesendet wird, entsprechende Daten, die bei dem Schritt **619** gesendet wurden, auf. Es sei bemerkt, dass bei dem beschriebenen Ausführungsbeispiel lediglich der Namenraum SOFTWARE in dem Client-Schema durch Daten von dem Server besetzt ist, da derselbe der einzige Namenraum SOFTWARE ist, der dauerhafte Daten, das heißt, Daten, die sich in einem Dauerspeicher oder einem nichtflüchtigen Speicher in dem Server-Computer befinden, enthält. Bei anderen bevorzugten Ausführungsbeispielen können andere Namenräume dauerhafte Daten abhängig von den Erfordernissen des Computernetzes enthalten. Bei einem Schritt **631** kann der Benutzer des Client-Computers eine spezielle Anwendung starten. Auf dieser Stufe kann der Client-Computer eine spezielle Anwendung starten, indem derselbe in den Namenraum SOFTWARE des Client-Schemas derselben geht, um die geeigneten Konfigurationsdaten für die Anwendung zu bekommen. Auf dieser Stufe wurden alle Konfigurationsdaten, die notwendig sind, um Anwendungen in dem Client-Computer laufen zu lassen, von dem Server zu dem Client übertra-

gen. Ferner werden lediglich Konfigurationsdaten, die von dem Client-Computer verwendet werden, als ein Resultat des Vereinigens, das in dem Server durchgeführt wird, von dem Server zu dem Client übertragen.

[0058] [Fig. 7](#) ist ein Flussdiagramm, das ein Verfahren zum Verhandeln einer Kommunikationsverbindung zwischen einem Client und einem Server gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zeigt. Dasselbe zeigt detaillierter den Schritt **609** von [Fig. 6a](#). Bei einem Schritt **701** informiert ein Client einen Server über Versionsinformationen hinsichtlich der Software des Clients. Die Versionsinformationen können ferner Hardwarekomponenten des Client beschreiben. Diese Daten werden in Datenpaketen über TCP/IP gesendet. Bei diesem Schritt informiert der Client den Server einfach, welche Version der Betriebssoftware der Client verwendet. Bei einem Schritt **703** untersucht der Server diese Versionsinformationen und informiert den Client, welche Version einer Betriebssoftware der Server unterstützen kann. Indem derselbe dies tut, diktiert der Server, welche Version der Betriebssoftware verwendet werden wird, um Konfigurationsdaten zwischen dem Client und dem Server auszutauschen.

[0059] [Fig. 8a](#) ist ein Flussdiagramm, das die Schritte **617** und **627** von [Fig. 6](#) detaillierter zeigt. Dasselbe zeigt ein Verfahren eines Vereinigens von Konfigurationsdaten, die sich bei dem beschriebenen Ausführungsbeispiel auf einen speziellen Client in dem Server beziehen. Die Daten sind gemäß einer in [Fig. 8b](#) gezeigten Hierarchie vereinigt, dahingehend, dass bestimmte Datenelemente in einem Dateneintrag durch entsprechende Datenelemente von anderen Namenräumen oder Kategorien aufgehoben werden. Bei einem Schritt **801** gewinnt der Server einen Platformeintrag von dem Plattform-Unterbaum in dem Namenraum Gerät in dem Server wieder. Der Platformeintrag enthält spezifische Informationen zu dem Client-Computer-Typ. Bei einem Schritt **803** bestimmt der Server, ob der Client einen entsprechenden Profileintrag hat, indem derselbe überprüft, ob es einen Bezugszeiger von dem Client-Geräteindeutigen ID zu einem speziellen Profil in der Kategorie Profil gibt. Wenn der Client einen Profileintrag hat, gewinnt der Server den Profileintrag bei einem Schritt **805** wieder. Bei dem Schritt **805** werden die Werte in dem Profileintrag und dem Platformeintrag vereinigt. Werte für Eigenschaften, die in beiden Einträgen enthalten sind, werden gemäß einer in [Fig. 8b](#) gezeigten Hierarchie auf einer Eigenschaft-für-Eigenschaft-Basis vereinigt. Bei dem beschriebenen Ausführungsbeispiel werden durch die Profileintragswerte Werte in dem Profileintrag, die einen übereinstimmenden Eigenschaftswert in dem Platformeintrag haben, aufgehoben. Das Vereinigen von Einträgen erfolgt auf einer Eigenschaft-für-Eigenschaft-Basis. Wenn der Client keinen Profileintrag, wie bei dem Schritt **803**

bestimmt wird, hat, geht die Steuerung zu einem Schritt **807**.

[0060] Bei dem Schritt **807** gewinnt der Server einen Benutzereintrag von dem Namenraum Benutzer in dem Server-Schema wieder. Bei einem Schritt **809** bestimmt der Server, ob der Benutzer, der sich angemeldet hat, zu einer der Gruppen **319** gehört. Bei einem Schritt **811** gewinnt der Server einen Gruppeneintrag wieder, hebt jedoch, wieder auf einer Eigenschaft-für-Eigenschaft-Basis, Werte in dem Benutzereintrag durch Werte in dem Gruppeneintrag auf. Bei einem Schritt **813** hebt der Server Wert in dem Profil/Plattform-Eintrag von dem Schritt **805** durch Werte in dem vereinigten Benutzereintrag auf. Wenn der Server bestimmt, dass der Benutzer nicht zu einer Gruppe in **809** gehört, schreitet derselbe mit dem Schritt **813** fort. Bei einem Schritt **815** wird der vereinigte Dateneintrag, der sich auf den Benutzer, der sich bei dem Client anmeldet, bezieht, zu dem Client-Computer gesendet und das Verfahren wird beendet.

[0061] [Fig. 8b](#) ist ein Blockdiagramm, das die Hierarchie, die verwendet wird, um Daten gemäß einem Ausführungsbeispiel der vorliegenden Erfindung zu vereinigen, zeigt. Der am höchsten eingestufte Eintrag und daher der Eintrag der gegenüber allen anderen Einträgen Vorrang hat, ist der Benutzereintrag, der bei einem Block **815** gezeigt ist. Darunter ist der am nächsthöchsten eingestufte Eintrag, der Benutzergruppeneintrag, der bei einem Block **817** gezeigt ist. Werte in Eigenschaften, die in dem Gruppeneintrag enthalten sind, werden durch Werte übereinstimmender Eigenschaften in dem Benutzereintrag aufgehoben. Ein Block **819** ist der Clientprofileintrag. Werte von Eigenschaften in dem Gruppeneintrag heben übereinstimmende Eigenschaftswerte in dem Clientprofileintrag auf. In dem untersten Teil der Hierarchie steht bei dem beschriebenen Ausführungsbeispiel der Clientplatformeintrag, der bei einem Block **821** gezeigt ist. Werte von Eigenschaften in diesem Eintrag werden durch Werte übereinstimmender Eigenschaften in einer der anderen drei Kategorien aufgehoben. Bei anderen bevorzugten Ausführungsbeispielen kann die Vorrangreihenfolge unterschiedlich angeordnet sein und kann ferner zusätzliche oder weniger Einträge in der Hierarchie umfassen.

[0062] Auf ein Merkmal des Client/Server-Protokolls ist als eine Herzschlageinrichtung, die dem Client oder dem Server ermöglicht, informiert zu werden, wenn die Verbindung zwischen den beiden unterbrochen wird, Bezug genommen. [Fig. 9a](#) und [Fig. 9b](#) sind Flussdiagramme, die ein Verfahren zum periodischen Senden von Signalen zwischen einem Server und einem Client, um anzuzeigen, dass die Verbindung zwischen denselben immer noch lebt, gemäß einem Ausführungsbeispiel der vorliegenden Erfin-

dung zeigen. Die Herzschlageinrichtung beginnt gewöhnlich bald nach der Verhandlungsphase (Schritt **609** und [Fig. 7](#)) und setzt sich die ganze Zeit, in der die Verbindung zwischen dem Client und dem Server aufrechterhalten wird, fort. Es ist ein wichtiges Merkmal, dass entweder der Client oder der Server informiert werden, wenn die Verbindung zwischen den beiden getrennt oder unterbrochen wird. Bei einem Schritt **901** sendet der Server ein Datenpaket zu dem Client. Zum ersten Mal tritt dies gewöhnlich gleich nach der Verhandlungsphase auf, und es wird mit regelmäßigen Intervallen gesendet, während die Verbindung steht. Bei dem beschriebenen Ausführungsbeispiel läuft ein Herzschlag-Daemon sowohl in dem Client als auch in dem Server und ist im Wesentlichen ein einziger Programmfaden (engl.: thread) in dem Client/Server-Protokoll. Bei einem anderen bevorzugten Ausführungsbeispiel kann der Daemon entweder in dem Server oder in dem Client ablaufen und kann ein Verfahren mit mehreren Programmfäden sein. Der Server erhält Zustandsinformationen über die Verbindung aufrecht und umfasst Daten, wie die Zeit, zu der Datenpakete zu dem Client gesendet wurden. Bei einem Schritt **903** bestimmt der Server, ob es ein Rücksendedatenpaket von dem Client, das heißt einen Herzschlag, gibt. Wenn es einen Herzschlag gibt, wird bei einem Schritt **905** das Datenpaket von dem Client zu dem Server gesendet. Bei **901** sendet der Server nach einem eingestellten Zeitintervall, wie 10 Minuten, wieder ein Datenpaket zu dem Client. Bei anderen bevorzugten Ausführungsbeispielen kann die Zeit länger oder kürzer sein.

[0063] Wenn bei dem Schritt **903** bestimmt wird, dass der Client kein Datenpaket zu dem Server zurücksendet hat, das heißt, dass es keinen Herzschlag gibt, verwendet der Server die Zustandsinformationen und die Struktur (einen Verzeichnis-Cache), um anzuzeigen, dass es kein Rücksendedatenpaket gab. Dies erfolgt bei einem Schritt **907**. Ein verschmutztes Bit wird in die Struktur platziert, um anzuzeigen, dass die Verbindung unterbrochen wurde. Bei einem Schritt **909** versuchen der Server und der Client, sich erneut zu verbinden, und zu dieser Zeit überprüft der Server das verschmutzte Bit in der Zustandsstruktur desselben, um zu sehen, ob in dem Client eine Aktualisierung notwendig ist. Dies kann entstehen, wenn an den Konfigurationsinformationen in dem Server, die sich auf den Client beziehen, Änderungen vorgenommen werden. Bei einem Schritt **911** aktualisiert der Server die Client-Konfiguration. Ähnlich der Verzeichnis-Cache-Zustandsstruktur in dem Server hat der Client ferner eine ähnliche Struktur, die die Zustandsinformationen anzeigt, die ein verschmutztes Bit enthalten können, um anzuzeigen, ob von dem Client eine Aktualisierung zu dem Server gesendet werden soll. Bei einem Schritt **913** überprüft der Client daher das verschmutzte Bit desselben, um zu sehen, ob eine Aktualisierung zu dem Server gesendet werden soll. Bei einem Schritt **915**

aktualisiert der Client das Server-Schema. An diesem Punkt ist die erneute Verbindung zwischen dem Client und dem Server wieder eingerichtet und die Herzschlageinrichtung nimmt den Betrieb wieder auf, nachdem die Verhandlungsphase beendet ist.

[0064] Ein anderes Merkmal des Client/Server-Protokolls der vorliegenden Erfindung ist die Fähigkeit, Clients, die eine spezielle Anwendung verwenden, über Aktualisierungen dieser Anwendung, die in dem Server vorgenommen werden, zu benachrichtigen. [Fig. 10](#) ist ein Flussdiagramm eines Verfahrens einer Ereignisbenachrichtigung in dem Client/Server-Protokoll gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. Bei einem Schritt **1301** wird das Server-Schema mit einer Anwendungsreparatur oder -änderung aktualisiert oder mit einer allgemeinen Konfigurationsaktualisierung modifiziert. Bei einem Schritt **1003** untersucht der Server das Cache-Verzeichnis desselben, um zu sehen, welche Clients die Konfigurationsdaten, die geändert wurden, verwenden, oder welche Clients über die allgemeine Konfigurationsänderung, die in dem Server-Schema vorgenommen wurde, informiert werden wollen. Das Server-Cache-Verzeichnis ist eine Liste, die von Clients verwendet wird, die mit dem Server verbunden sind. Bei einem Schritt **1005** breitet der Server die Aktualisierung oder Änderung zu den geeigneten Clients aus. Auf der Clientseite bestimmt der Client bei einem Schritt **1007**, welche Anwendungen durch die Aktualisierung beeinflusst werden könnten oder ob die allgemeine Konfigurationsaktualisierung für diesen speziellen Client relevant ist. Bei einem Schritt **1009** bestimmt der Client, ob die Anwendung, die durch die Aktualisierung beeinflusst wird, registriert ist, um von Änderungen der Anwendung benachrichtigt zu werden. Wenn die Anwendung registriert ist, um benachrichtigt zu werden, geht die Steuerung zu einem Schritt **1011**, wobei die Anwendung bei dem Schritt **1011** über die Aktualisierung benachrichtigt wird. Auf dieser Stufe ist das Verfahren beendet. Wenn die Anwendung sich nicht registriert hat, um von der Änderung benachrichtigt zu werden, ignoriert die Anwendung die Änderung oder die Aktualisierung und fährt fort, ohne benachrichtigt zu werden.

[0065] Ein anderes Merkmal des Client/Server-Protokolls der vorliegenden Erfindung ist die Fähigkeit zu garantieren, dass eine spezielle Operation entweder vorgenommen wurde oder nicht. Auf dieses Merkmal wird als eine Wertigkeit des Client/Server-Protokolls Bezug genommen. Das Client/Server-Protokoll implementiert eine zweiphasige Übergabeprozedur zum Durchführen von Aktualisierungen in dem Client- und Server-Schema. In der ersten Phase wird die Wurzel des Unterbaums des Schemas verriegelt, damit an einem der Einträge in diesem speziellen Unterbaum eine Aktualisierung vorgenommen wird. In der zweiten Phase wird die Aktualisierung in einem der Einträge in dem Unterbaum durchgeführt und die Wurzel

dieses Unterbaums wird dann entriegelt. Indem die Wurzel des Unterbaums verriegelt wird, kann auf alle Einträge und Knoten unter der Wurzel in diesem Unterbaum nicht durch andere Clients oder Anwendungen zugegriffen werden. Eine andere Client-Anwendung, die versucht, einen Eintrag in dem Unterbereich, der verriegelt wurde, zu aktualisieren, wird entweder eine Sperr- oder eine Entsperr-Transaktion haben. Wenn die Client-Anwendung warten möchte, bis die Entriegelung auftritt, und dann die eigene Aktualisierung derselben vornehmen möchte, wird diese Transaktion als eine Sperrtransaktion bezeichnet. Bei diesem Szenario wird die Aktualisierung in eine Warteschlange platziert. Wenn die Anwendung nicht warten möchte, bis die Verriegelung entfernt wird, wird auf die Transaktion als eine Entsperrtransaktion Bezug genommen und dieselbe zu der Anwendung zurückgesendet. Durch diese Einrichtung kann das Client/Server-Protokoll einer Client-Anwendung garantieren, dass entweder die angefragte Operation oder Aktualisierung durchgeführt wird, das heißt, dass die Transaktion gesperrt ist, oder dass die Operation nicht durchgeführt wurde, das heißt, die Transaktion war nicht gesperrt.

[0066] Die vorliegende Erfindung verwendet verschiedene computerimplementierte Operationen, die Daten betreffen, die in Computersystemen gespeichert sind. Diese Operationen umfassen solche, die eine physische Handhabung von physikalischen Größen erfordern, sind jedoch nicht auf dieselben begrenzt. Üblicherweise, wenn auch nicht notwendigerweise, nehmen diese Größen die Form von elektrischen oder magnetischen Signalen an, die fähig sind, gespeichert, übertragen, kombiniert, verglichen und anderweitig gehandhabt zu werden. Die hierin beschriebenen Operationen, die einen Teil der Erfindung bilden, sind nützliche Geräteoperationen. Auf die durchgeführten Handhabungen wird häufig mit Begriffen wie Erzeugen, Identifizieren, Laufenlassen, Bestimmen, Vergleichen, Ausführen, Herunterladen oder Erfassen Bezug genommen. Manchmal ist es günstig, hauptsächlich aus Gründen des allgemeinen Sprachgebrauchs, auf diese elektrischen oder magnetischen Signale als Bits, Werte, Elemente, Variablen, Zeichen, Daten oder dergleichen Bezug zu nehmen. Man sollte sich jedoch erinnern, dass all diese und ähnliche Begriffe den geeigneten physikalischen Größen zuzuordnen sind und lediglich günstige Bezeichnungen sind, die auf diese Größen angewendet werden.

[0067] Die vorliegende Erfindung bezieht sich ferner auf eine Vorrichtung, ein System oder eine Vorrichtung zum Durchführen der im Vorhergehenden erwähnten Operationen. Das System kann speziell für die erforderlichen Zwecke aufgebaut sein, oder dasselbe kann ein Allzweckcomputer sein, der durch ein Computerprogramm, das in dem Computer gespeichert ist, selektiv aktiviert oder konfiguriert wird. Die

im Vorhergehenden dargestellten Verfahren beziehen sich nicht inhärent auf einen speziellen Computer oder eine spezielle andere Rechenvorrichtung. Insbesondere können verschiedene Allzweckrechner mit Programmen, die gemäß den hierin enthaltenen Lehren geschrieben sind, verwendet sein, oder es kann alternativ günstiger sein, ein spezialisierteres Computersystem aufzubauen, um die erforderlichen Operationen durchzuführen.

[0068] [Fig. 11](#) ist ein Blockdiagramm eines Allzweckcomputersystems **1100**, das geeignet ist, um das Verarbeiten gemäß einem Ausführungsbeispiel der vorliegenden Erfindung auszuführen. [Fig. 11](#) stellt ein Ausführungsbeispiel eines Allzweckcomputersystems dar. Andere Computersystemarchitekturen und -konfigurationen können verwendet sein, um das Verarbeiten der vorliegenden Erfindung auszuführen. Das Computersystem **1100**, das aus verschiedenen Untersystemen, die im Folgenden beschrieben sind, gebildet ist, umfasst mindestens ein Mikroprozessor-Untersystem (auch als eine Zentralverarbeitungseinheit (engl.: Central Processing Unit) oder CPU Bezug genommen) **1102**. Das heißt, dass die CPU **1102** durch einen Einzelchip-Prozessor oder durch Mehrfachprozessoren implementiert sein kann. Die CPU **1102** ist ein Allzweckdigitalprozessor, der den Betrieb des Computersystems **1100** steuert. Die CPU **1102** steuert unter Verwendung von Anweisungen, die aus einem Speicher wiedergewonnen werden, den Empfang und die Handhabung von Eingabedaten und die Ausgabe und Anzeige von Daten auf Ausgabevorrichtungen.

[0069] Die CPU **1102** ist über einen Speicherbus **1108** bidirektional mit einem ersten Hauptspeicher **1104**, typischerweise einem Direktzugriffs-Arbeitspeicher (engl.: Random Access Memory; RAM), und unidirektional mit einem zweiten Hauptspeicherbereich **1106**, typischerweise einem Nur-Lese-Speicher (engl.: Read-Only Memory; ROM) über einen Speicherbus **1108** gekoppelt. Wie in der Technik allgemein bekannt ist, kann der Hauptspeicher **1104** als ein Allzweckspeicherbereich und als ein Notizblockspeicher verwendet werden und kann ferner verwendet werden, um Eingabedaten und verarbeitete Daten zu speichern. Derselbe kann ferner Programmieranweisungen und Daten in der Form eines Nachrichtenspeichers zusätzlich zu anderen Daten und Anweisungen für Verfahren, die auf der CPU **1102** in Betrieb sind, speichern und wird typischerweise für eine schnelle Übertragung von Daten und Anweisungen auf eine bidirektionale Art und Weise über den Speicherbus **1108** verwendet. Wie ferner in der Technik allgemein bekannt ist, umfasst der Hauptspeicher **1106** gewöhnlich grundlegende Betriebsanweisungen, einen Programmcode, Daten und Objekte, die die CPU **1102** verwendet, um die Funktionen derselben durchzuführen. Die Hauptspeichervorrichtungen **1104** und **1106** können abhängig davon, ob beispiels-

weise ein Datenzugriff bidirektional oder unidirektional sein muss, geeignete computerlesbare Speichermedien, die im Folgenden beschrieben sind, umfassen. Die CPU **1102** kann ferner direkt und sehr schnell häufig benötigte Daten in einem Cache-Speicher **1110** wiedergewinnen und speichern.

[0070] Eine auswechselbare Massenspeichervorrichtung **1112** sieht eine zusätzliche Datenspeicherkapazität für das Computersystem **1100** vor und ist über einen Peripheriebus **1114** entweder bidirektional oder unidirektional mit der CPU **1102** gekoppelt. Eine spezifische entfernbare Massenspeichervorrichtung, die allgemein als eine CD-ROM bekannt ist, lässt beispielsweise typischerweise Daten unidirektional zu der CPU **1102** durch, während eine Floppy-Diskette Daten bidirektional zu der CPU **1102** durchlassen kann. Der Speicher **1112** kann ferner computerlesbare Medien, wie ein Magnetband, einen Flash-Speicher, Signale, die in einer Trägerwelle enthalten sind, PC-Karten, tragbare Massenspeichervorrichtungen, holografische Speichervorrichtungen und andere Speichervorrichtungen umfassen. Ein fester Massenspeicher **1116** sieht ferner eine zusätzliche Datenspeicherkapazität vor und ist über den Peripheriebus **1114** bidirektional mit der CPU **1102** gekoppelt. Das üblichste Beispiel des Massenspeichers **1116** ist ein Festplattenlaufwerk. Allgemein ist ein Zugriff auf diese Medien langsamer als ein Zugriff auf die Hauptspeicher **1104** und **1106**. Die Massenspeicher **1112** und **1116** speichern allgemein zusätzliche Programmanweisungen, Daten und dergleichen, die gewöhnlich nicht in einer aktiven Verwendung durch die CPU **1102** stehen. Es ist offensichtlich, dass die Informationen, die innerhalb der Massenspeicher **1112** und **1116** gehalten sind, wenn benötigt, in einer Standardweise als ein Teil des Hauptspeichers **1104** (zum Beispiel ein RAM) als ein virtueller Speicher aufgenommen werden können.

[0071] Zusätzlich zu einem Vorsehen eines Zugriffs der CPU **1102** auf Speicherundersysteme wird der Peripheriebus **1114** verwendet, um ebenso einen Zugriff auf andere Undersysteme und Vorrichtungen vorzusehen. Bei dem beschriebenen Ausführungsbeispiel umfassen dieselben einen Anzeigebildschirm **1118** und einen Adapter **1120**, eine Druckervorrichtung **1122**, eine Netzchnittstelle **1124**, eine zusätzliche Eingabe-Ausgabe-Vorrichtungsschnittstelle **1126**, eine Soundkarte **1128** und Lautsprecher **1130** und, wie benötigt, andere Undersysteme.

[0072] Die Netzchnittstelle **1124** ermöglicht der CPU **1102**, mit einem anderen Computer, Computernetz oder Telekommunikationsnetz unter Verwendung einer Netzverbindung, wie gezeigt, gekoppelt zu werden. Durch die Netzchnittstelle **1124** wird erwogen, dass die CPU **1102** beim Durchführen der im Vorhergehenden beschriebenen Verfahrensschritte Informationen, zum Beispiel Datenobjekte oder Pro-

grammanweisungen, von einem anderen Netz empfangen könnte oder Informationen zu einem anderen Netz ausgeben könnte. Informationen, die häufig als eine Folge von auf einer CPU auszuführenden Anweisungen dargestellt sind, können von einem anderen Netz empfangen und zu einem anderen Netz ausgegeben werden, beispielsweise in der Form eines Computerdatensignals, das in einer Trägerwelle verkörpert ist. Eine Schnittstellenkarte oder eine ähnliche Vorrichtung und eine geeignete Software, die durch die CPU **1102** implementiert ist, können verwendet sein, um das Computersystem **1100** mit einem äußeren Netz zu verbinden und Daten gemäß Standardprotokollen zu übertragen. Das heißt, Verfahrensausführungsbeispiele der vorliegenden Erfindung können allein in der CPU **1102** ablaufen oder können in Verbindung mit einer entfernten CPU, die einen Anteil des Verarbeitens übernimmt, über ein Netz, wie das Internet, Intranet-Netze oder Nahbereichsnetze, durchgeführt werden. Zusätzliche Massenspeichervorrichtungen (nicht gezeigt) können ferner mit der CPU **1102** durch die Netzchnittstelle **1124** verbunden sein.

[0073] Eine zusätzliche E/A-Vorrichtungsschnittstelle **1126** stellt allgemeine und kundenspezifische Schnittstellen dar, die der CPU **1102** ermöglichen, Daten zu senden und, typischerweise, von anderen Vorrichtungen, wie Mikrofonen, berührungsempfindlichen Bildschirmen, Wandlerkartenlesern, Lochbandlesern, Sprach- oder Handschrifterkennern, Biometriklesern, Kameras, tragbaren Massenspeichervorrichtungen und anderen Computern zu empfangen.

[0074] Ferner ist mit der CPU **1102** über einen lokalen Bus **1134** eine Tastatursteuerung **1132** zum Empfangen einer Eingabe von einer Tastatur **1136** oder einer Zeigervorrichtung **1138** und Senden von dekodierten Symbolen von der Tastatur **1136** oder der Zeigervorrichtung **1138** zu der CPU **1102** gekoppelt. Die Zeigervorrichtung kann eine Maus, ein Eingabestift, ein Trackball oder ein Tablett sein und ist zum Wechselwirken mit einer grafischen Benutzeroberfläche nützlich.

[0075] Zusätzlich beziehen sich die Ausführungsbeispiele der vorliegenden Erfindung ferner auf Computerspeichererzeugnisse mit einem computerlesbaren Medium, die einen Programmcode zum Durchführen verschiedener computerimplementierter Operationen enthalten. Das computerlesbare Medium ist eine Datenspeichervorrichtung, die Daten, die danach durch ein Computersystem gelesen werden können, speichern kann. Das Medium und der Programmcode können solche sein, die speziell für die Zwecke der vorliegenden Erfindung entworfen und aufgebaut sind, oder dieselben können von der Art sein, die Fachleuten in der Computersoftwaretechnik allgemein bekannt ist. Beispiele von computerlesbaren Medien umfassen, sind jedoch nicht begrenzt auf,

alle im Vorhergehenden erwähnten Medien: magnetische Medien, wie Festplatten, Floppy-Disketten und ein Magnetband; optische Medien, wie CD-ROMs; magneto-optische Medien, wie optische Disketten; und speziell konfigurierte Hardwarevorrichtungen, wie anwendungsspezifische integrierte Schaltungen (engl.: Application-Specific Integrated Circuits; ASICs), programmierbare Logikbausteine (engl.: Programmable Logic Devices; PLDs) und ROM- und RAM-Vorrichtungen. Das computerlesbare Medium kann ferner als ein Datensignal, das in einer Trägerwelle verkörpert ist, über ein Netz von gekoppelten Computersystemen verteilt werden, so dass der computerlesbare Code gespeichert und auf eine verteilte Weise ausgeführt wird. Beispiele eines Programmkodes umfassen sowohl einen Geräte-Code, wie beispielsweise derselbe, der durch einen Compiler erzeugt wird, als auch Dateien, die einen höheren Code, der unter Verwendung eines interpretierenden Programms ausgeführt werden kann, enthalten.

[0076] Es ist für Fachleute offensichtlich, dass die im Vorhergehenden beschriebenen Hardware- und Softwareelemente einen Standardentwurf und -aufbau aufweisen. Andere Computersysteme, die für eine Verwendung mit der Erfindung geeignet sind, können zusätzliche oder weniger Untersysteme umfassen. Ferner stellen der Speicherbus **1108**, der Peripheriebus **1114** und der lokale Bus **1134** ein allgemeines Verbindungsschema, das dazu dient, die Untersysteme zu verbinden, dar. Ein lokaler Bus könnte beispielsweise verwendet sein, um die CPU mit einem festen Massenspeicher **1116** und einem Anzeigeadapter **1120** zu verbinden. Das Computersystem, das in [Fig. 11](#) gezeigt ist, ist lediglich ein Beispiel eines Computersystems, das für eine Verwendung mit der Erfindung geeignet ist. Andere Computerarchitekturen, die unterschiedliche Konfigurationen von Untersystemen haben, können ebenfalls benutzt sein, um den Client-Computer oder den Server-Computer der vorliegenden Erfindung zu implementieren. Bei einem anderen bevorzugten Ausführungsbeispiel der vorliegenden Erfindung ist der Client-Computer ein Netzcomputer oder NC, der hinsichtlich einer Funktionsfähigkeit und einer Speicherfähigkeit zwischen einem völlig unabhängigen „Fat“-Client-Computer, der als ein Einzelplatzcomputer funktionieren könnte, und einem nichtintelligenten Client, der fast vollständig von einem Server- oder Mainframe-Computer abhängig ist, angeordnet ist. Bei noch anderen bevorzugten Ausführungsbeispielen kann sich das Client-Schema zusätzlich zu Computern mit einem begrenzten Arbeitsspeicher, wie persönliche digitale Assistenten (engl.: Personal Digital Assistant; PDA) und tragbare Computer, in Nicht-Computer-Vorrichtungen, wie Smart Cards und anderen intelligenten Vorrichtungen, die die Java-Plattform laufen lassen, befinden.

[0077] Obwohl die vorhergehende Erfindung für

Zwecke einer Klarheit des Verständnisses in einer gewissen Ausführlichkeit beschrieben wurde, ist offensichtlich, dass innerhalb des Schutzbereichs der beigefügten Ansprüche bestimmte Änderungen und Modifikationen praktiziert werden können. Weiterhin sei bemerkt, dass es alternative Weisen eines Implementierens sowohl des Verfahrens als auch der Vorrichtung der vorliegenden Erfindung gibt. Demgemäß sind die vorliegenden Ausführungsbeispiele als darstellend und nicht beschränkend zu betrachten, und die Erfindung soll nicht auf die hierin angegebenen Details begrenzt sein, sondern kann innerhalb des Schutzbereichs und der Äquivalente der beigefügten Ansprüche modifiziert werden.

Patentansprüche

1. Verfahren zum Konfigurieren eines Client-Computers (**105**) in einem Computernetz (**107**), das einen Server-Computer (**109**) umfasst, mit folgenden Schritten:

Einrichten einer Verbindung zwischen dem Client-Computer, der ein Profil hat, und dem Server-Computer (**601–607**);

Austauschen von Versionsinformationen zwischen dem Client-Computer und dem Server-Computer (**701, 703**) und Bestimmen einer Version einer Software, die verwendet werden soll, um Daten zwischen dem Client-Computer und dem Server-Computer (**609**) auszutauschen;

Senden des Profils zu dem Server (**611**);

Wiedergewinnen von Konfigurationsinformationen, die sich auf den Client-Computer (**619**) beziehen, wobei die Konfigurationsinformationen unter einer Steuerung des Server-Computers gespeichert werden und gemäß dem Profil abgeleitet werden; und

Konfigurieren des Client-Computers unter Verwendung der Konfigurationsinformationen (**619**) und dadurch Erleichtern einer zentralisierten Verwaltung von Konfigurationsinformationen durch den Server-Computer.

2. Verfahren nach Anspruch 1, bei dem ein Senden des Profils zu dem Server (**109**) ferner ein Anordnen der Profilinformationen aufweist, um Hardwareinformationen, die sich auf den Client-Computer (**105**) beziehen, und Informationen, die sich auf einen Benutzer des Client-Computers beziehen, zu umfassen.

3. Verfahren nach Anspruch 1 oder 2, bei dem der Client-Computer (**105**) ein Netzcomputer ist.

4. Verfahren nach Anspruch 1, 2 oder 3, bei dem der Server-Computer (**105**) Konfigurationsinformationen, die sich auf eine Mehrzahl von Client-Computern beziehen, speichert.

5. Verfahren nach Anspruch 1, bei dem das Profil Informationen, die sich auf einen Benutzer des Cli-

ent-Computers (**105**) beziehen, umfasst.

6. Verfahren nach Anspruch 1, bei dem die Konfigurationsinformationen in einem Datenschema unter einer Steuerung des Server-Computers (**109**) gespeichert werden.

7. Verfahren nach Anspruch 1, bei dem die Konfigurationsinformationen in einer Datenschicht eines Datenschemas mit einer Baumtyp-Struktur gespeichert werden.

8. Verfahren nach Anspruch 7, bei dem die Baumtyp-Struktur des Datenschemas ferner eine Wurzelknotenschicht umfasst.

9. Verfahren nach Anspruch 8, bei dem die Baumtyp-Struktur des Datenschemas ferner eine Zwischenknotenschicht zeigt.

Es folgen 15 Blatt Zeichnungen

Anhängende Zeichnungen

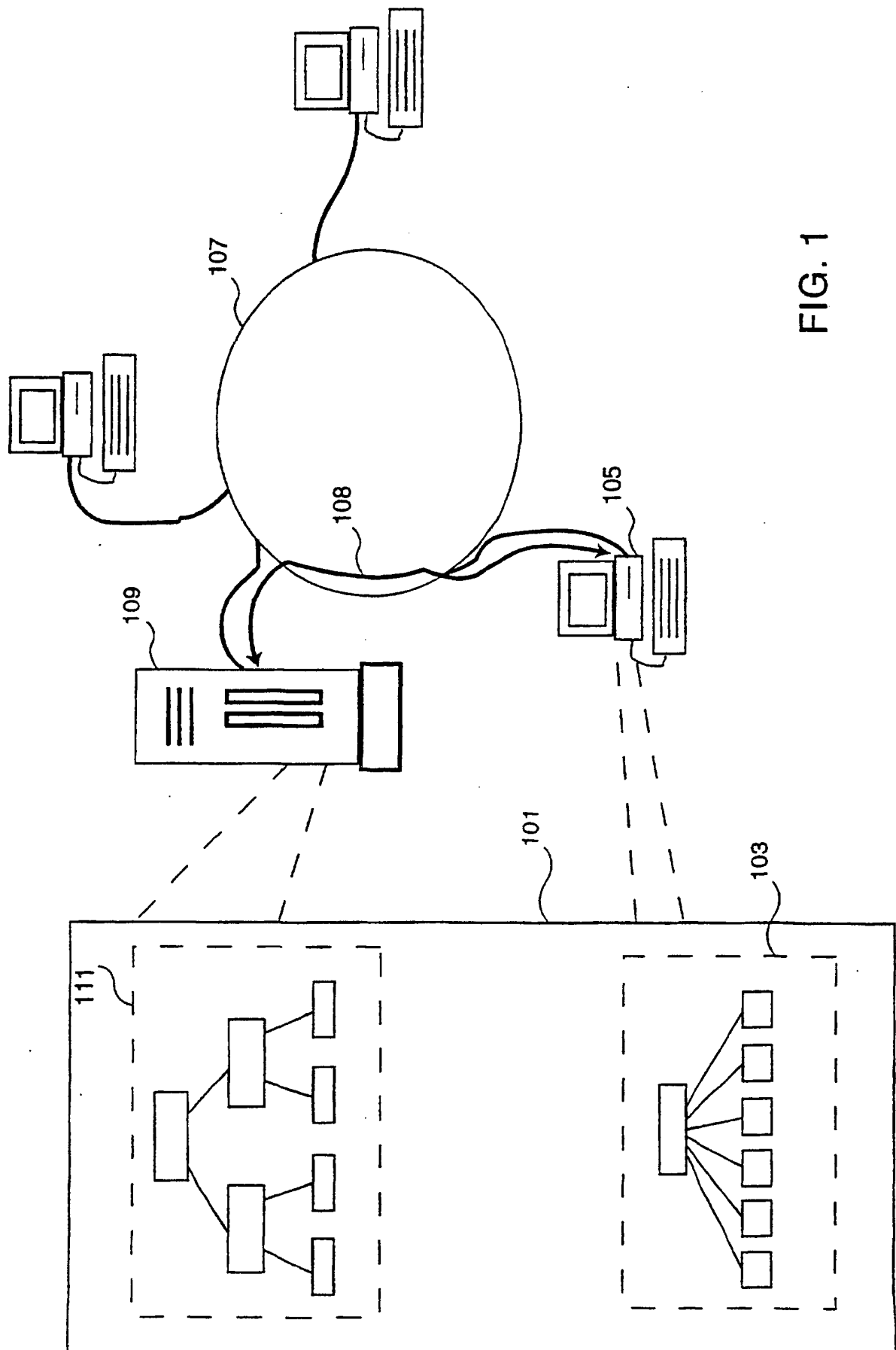


FIG. 1

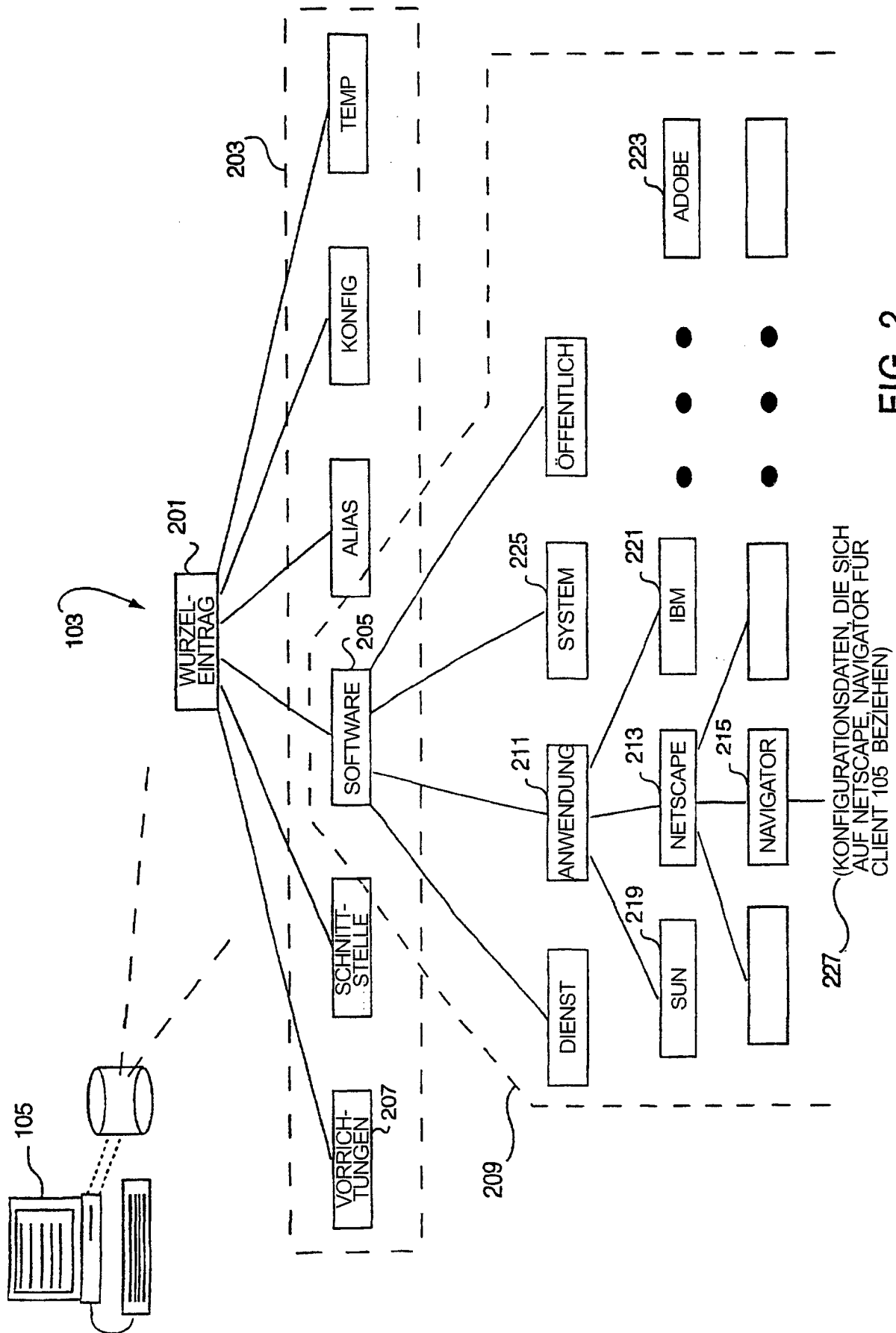


FIG. 2

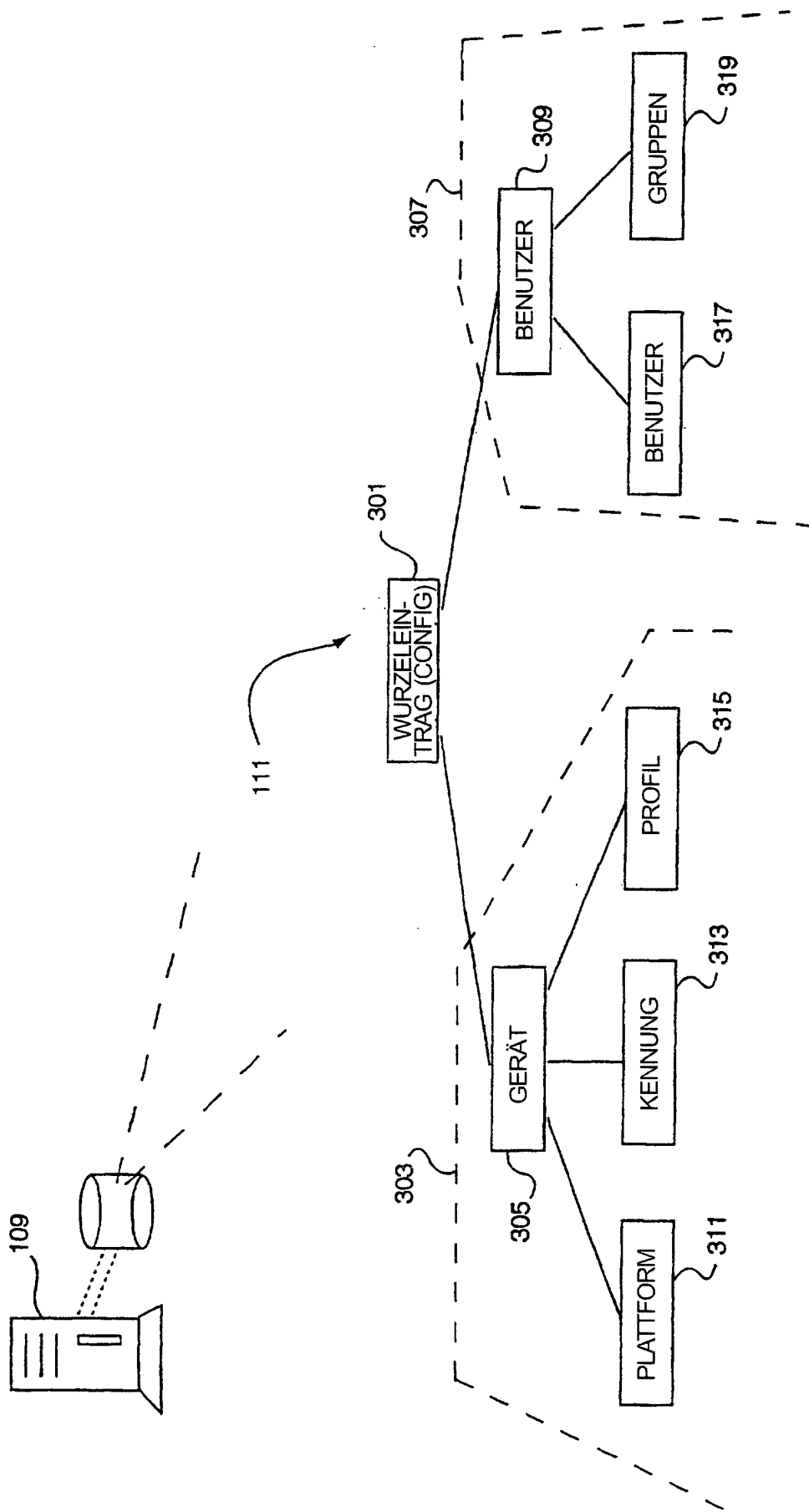


FIG. 3

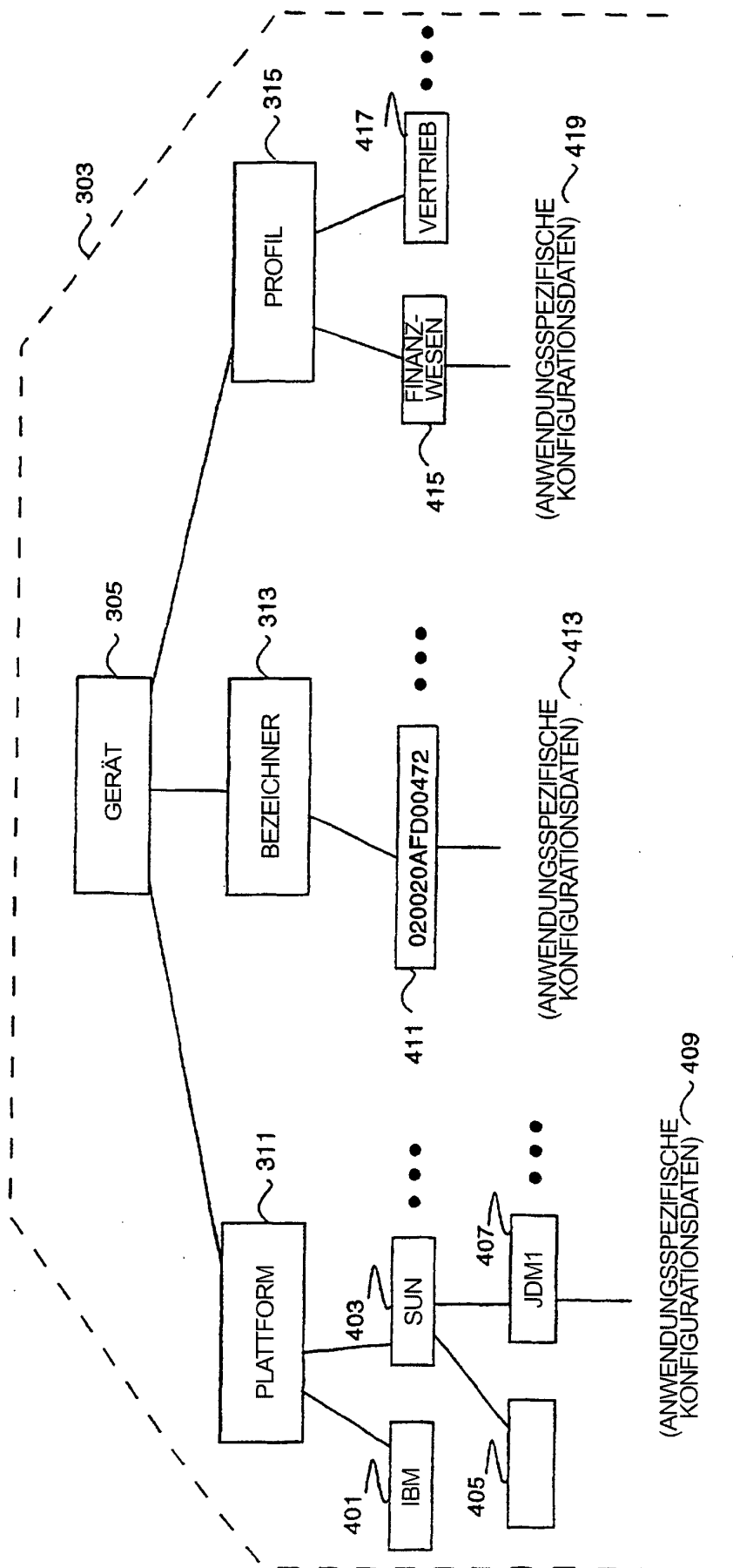


FIG. 4

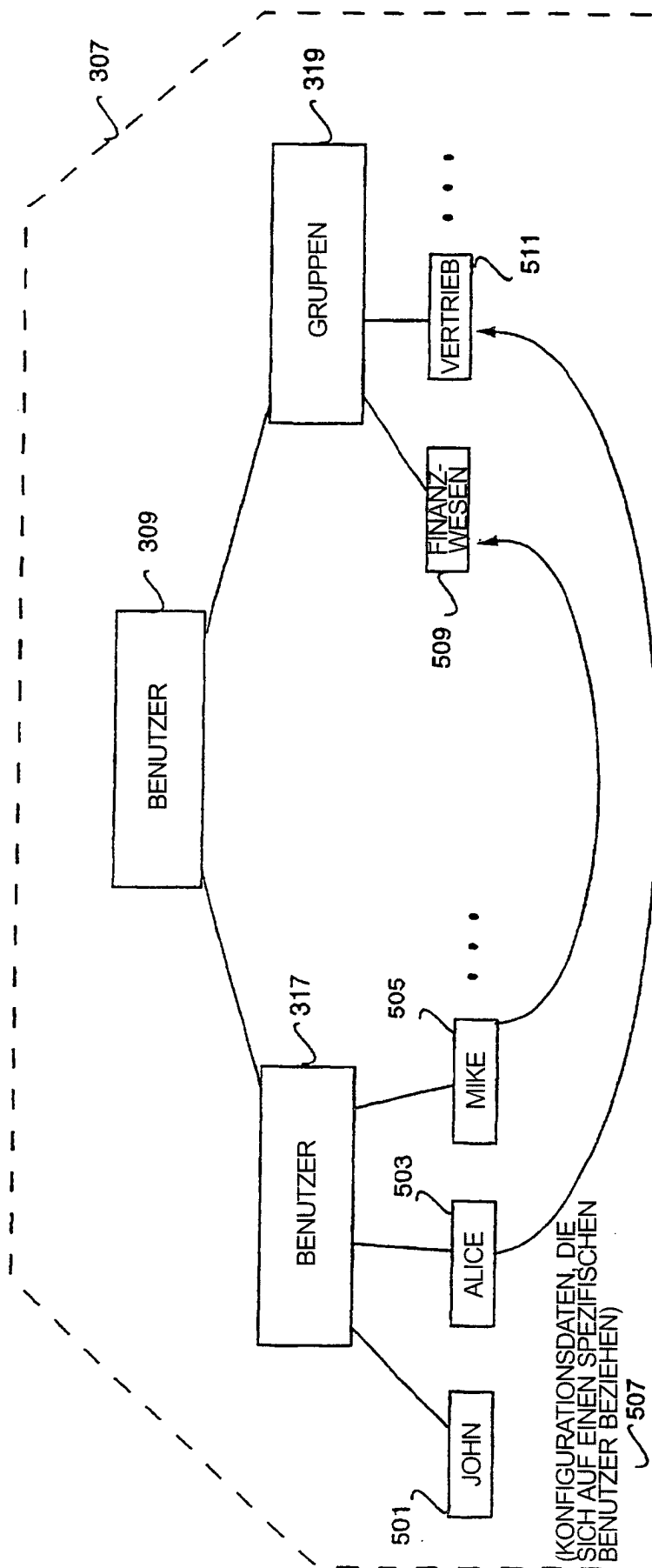


FIG. 5

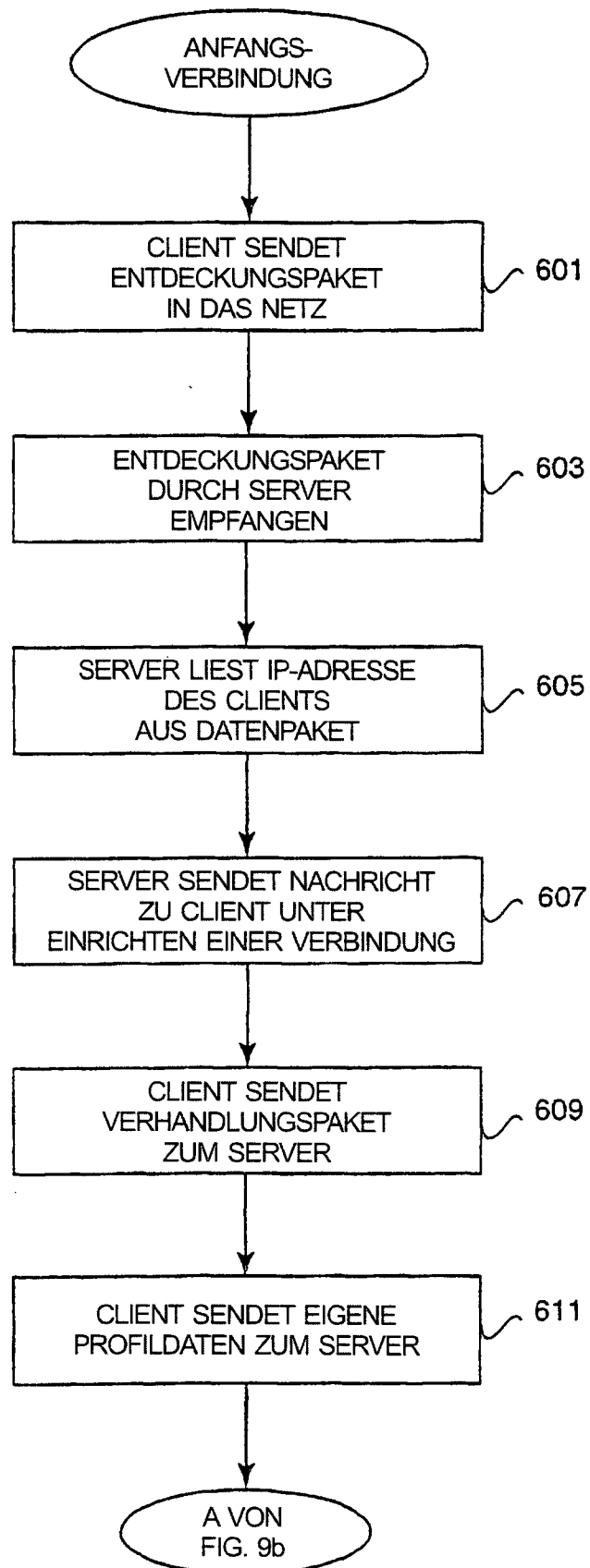


FIG. 6a

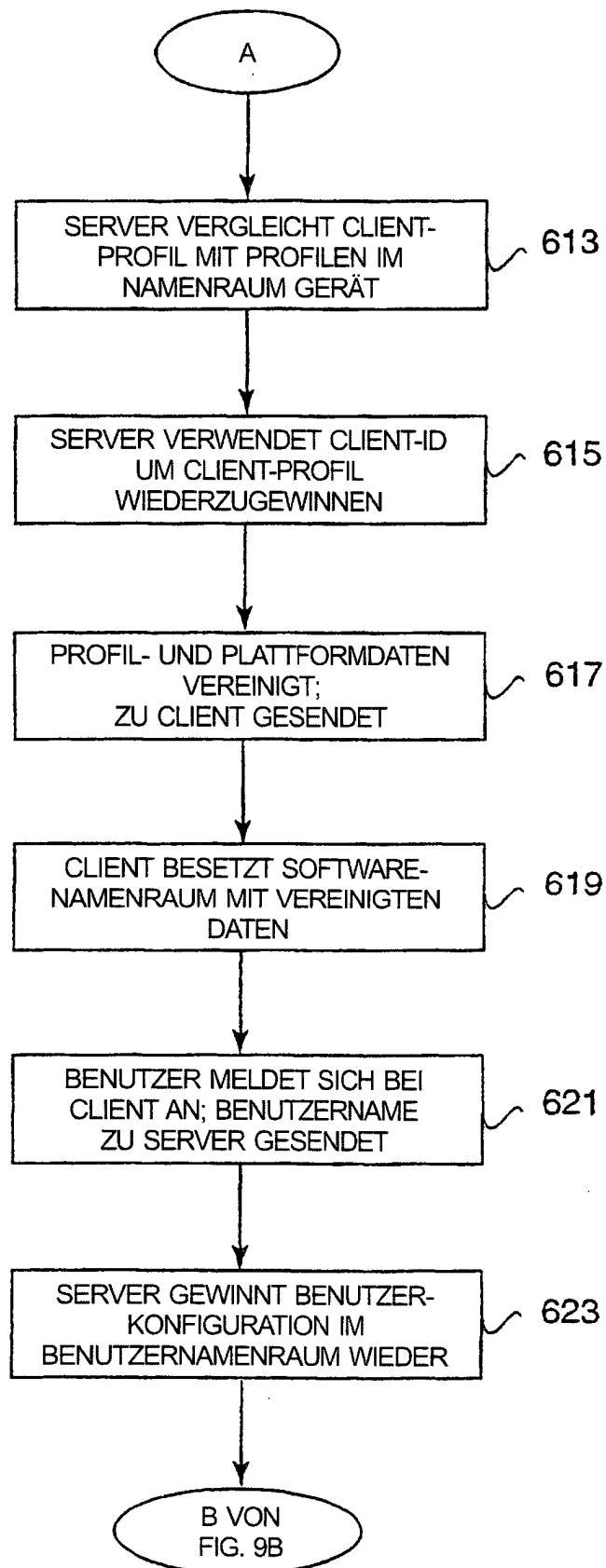


FIG. 6b

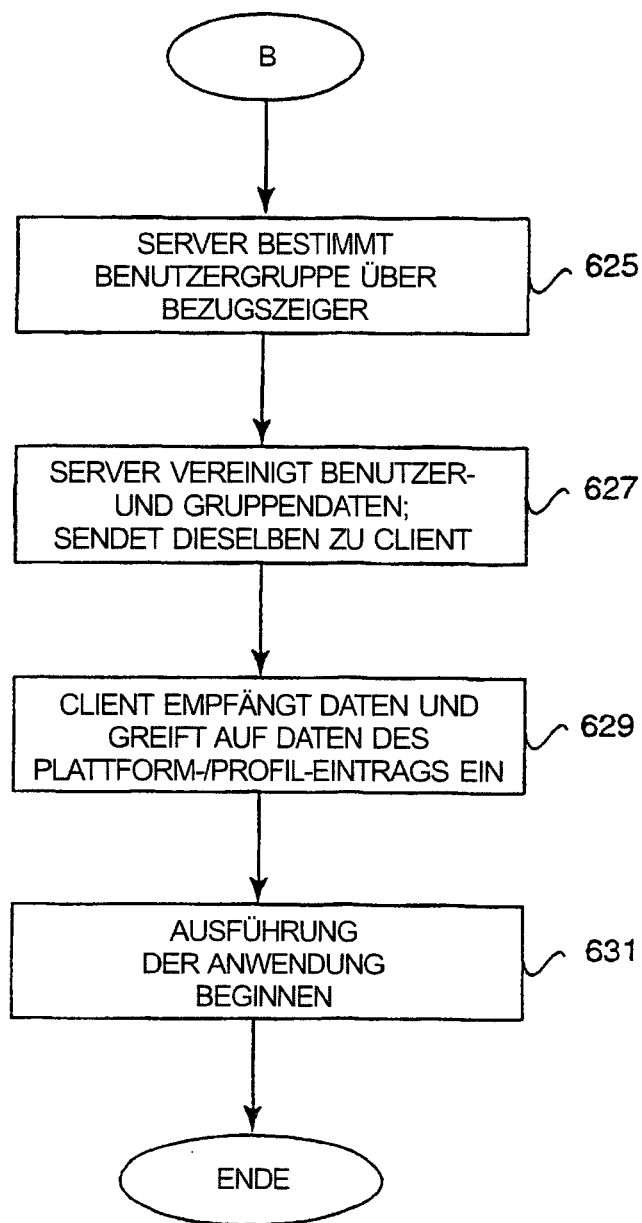


FIG. 6c

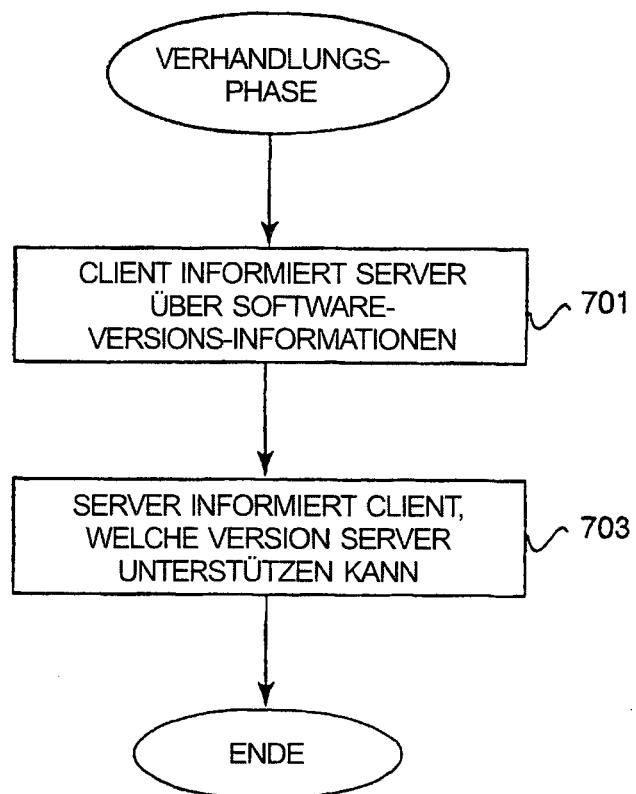


FIG. 7

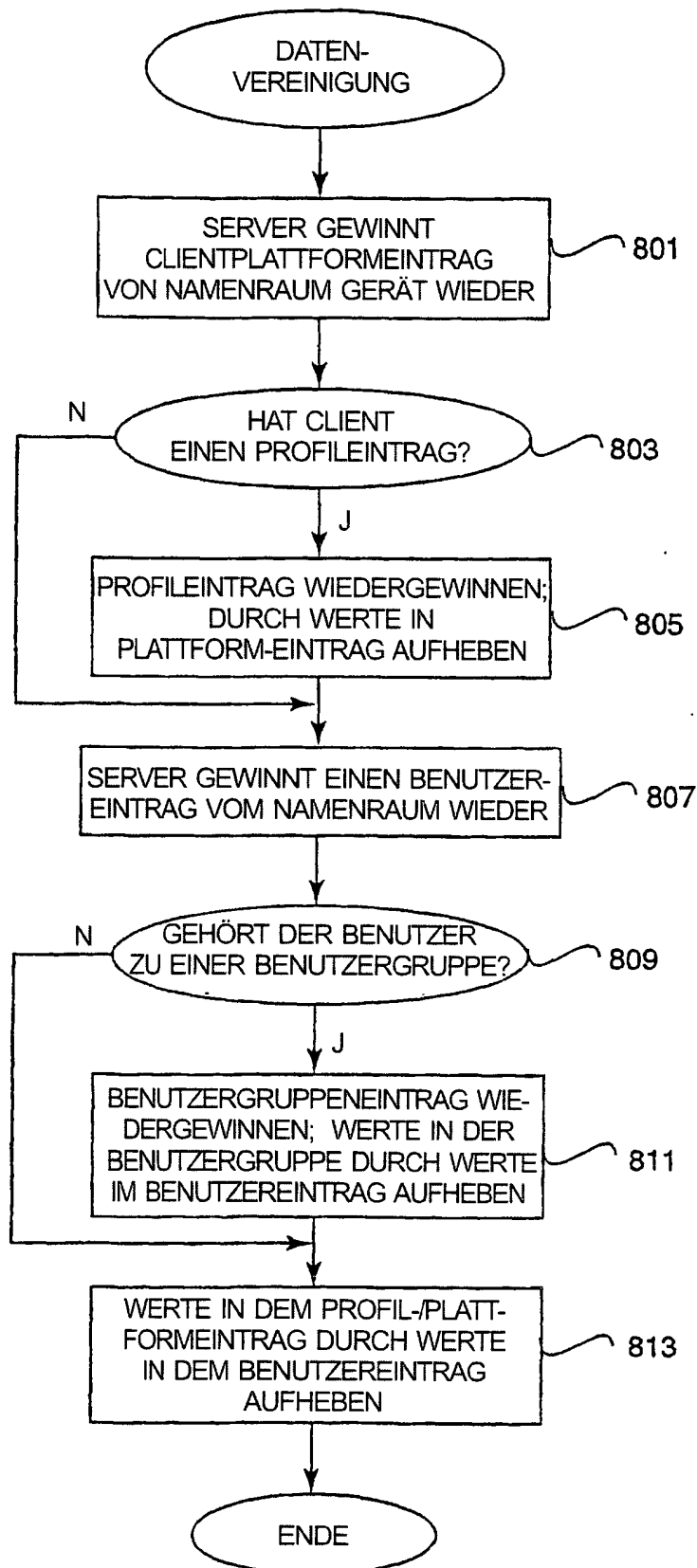


FIG. 8a

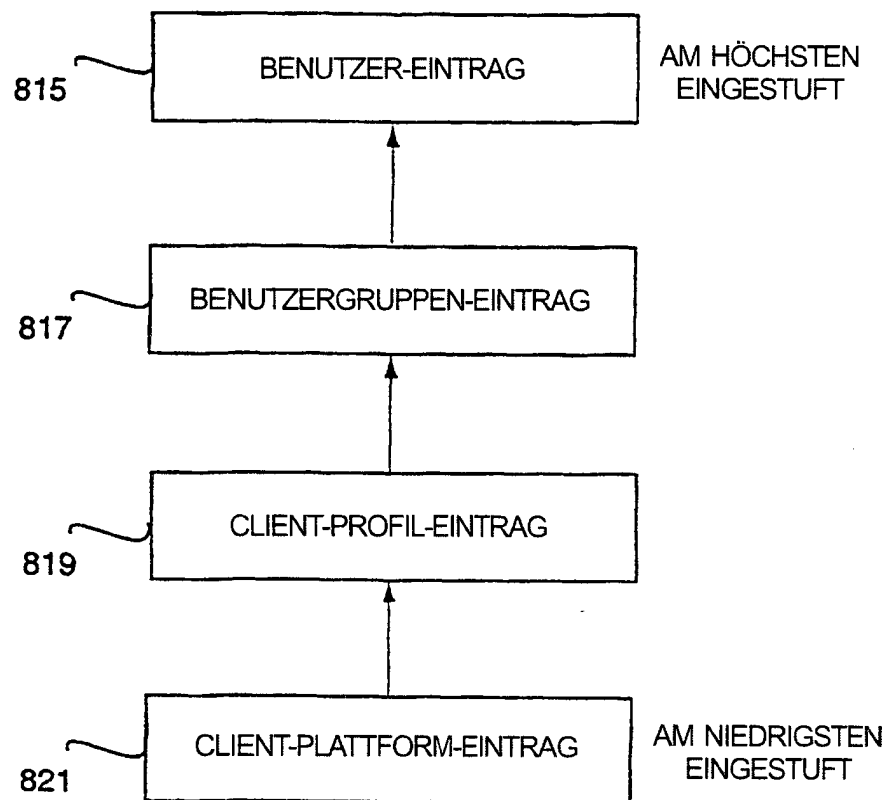


FIG. 8b

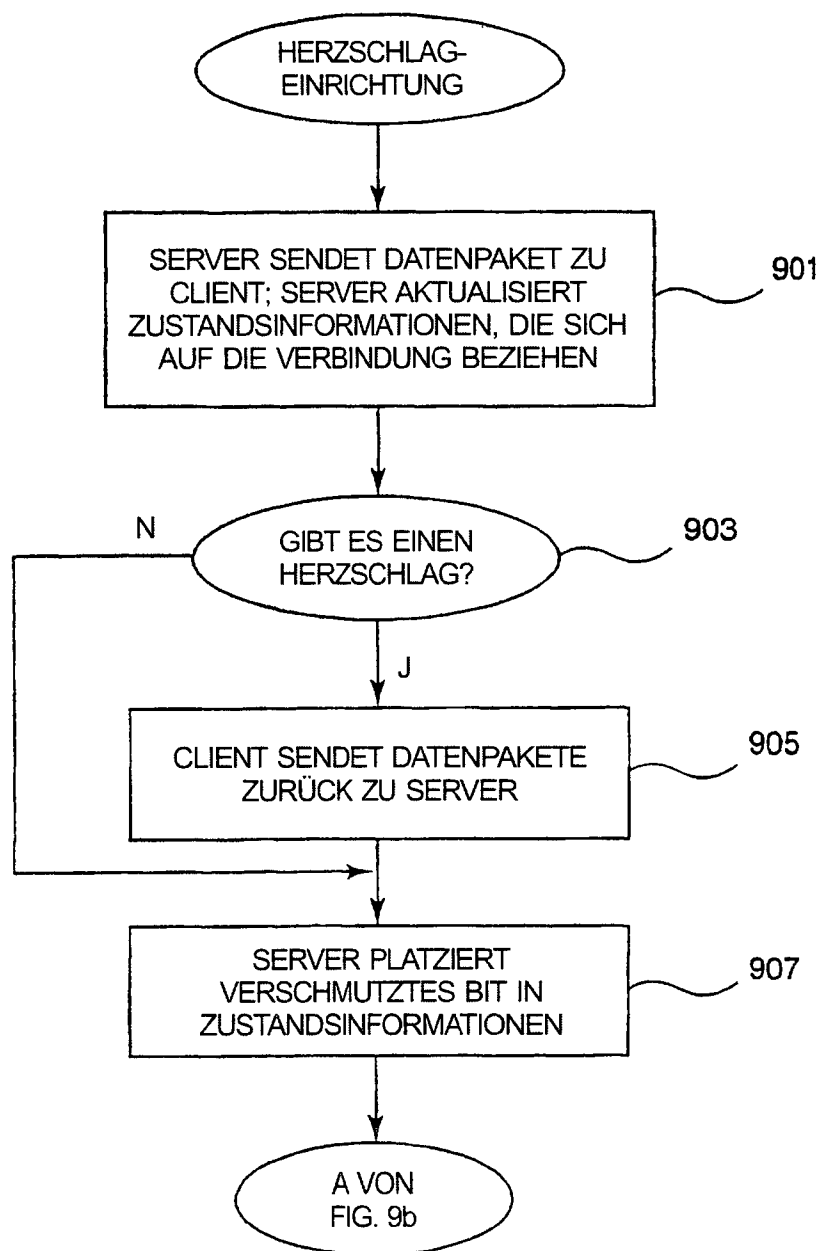


FIG. 9a

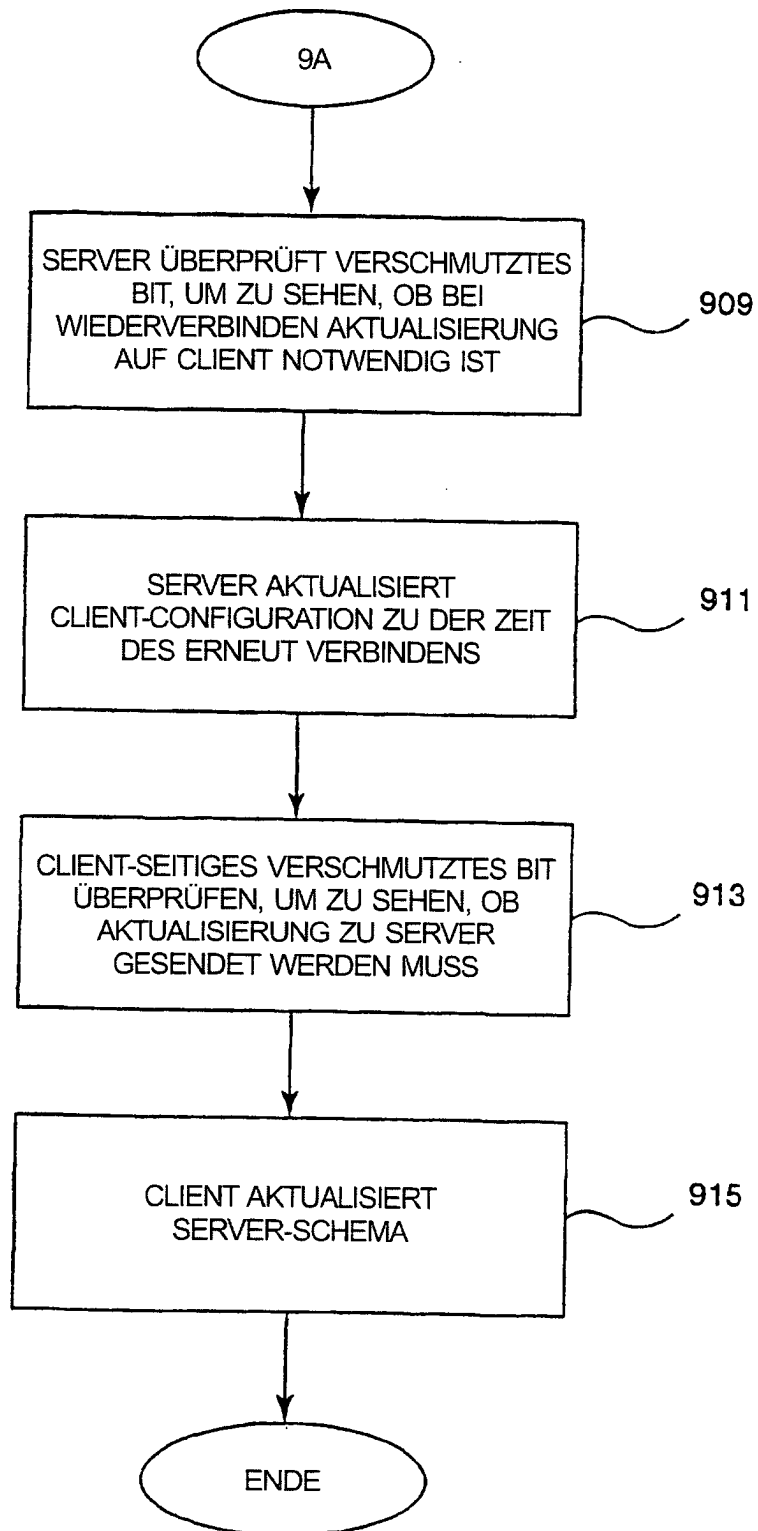


FIG. 9b

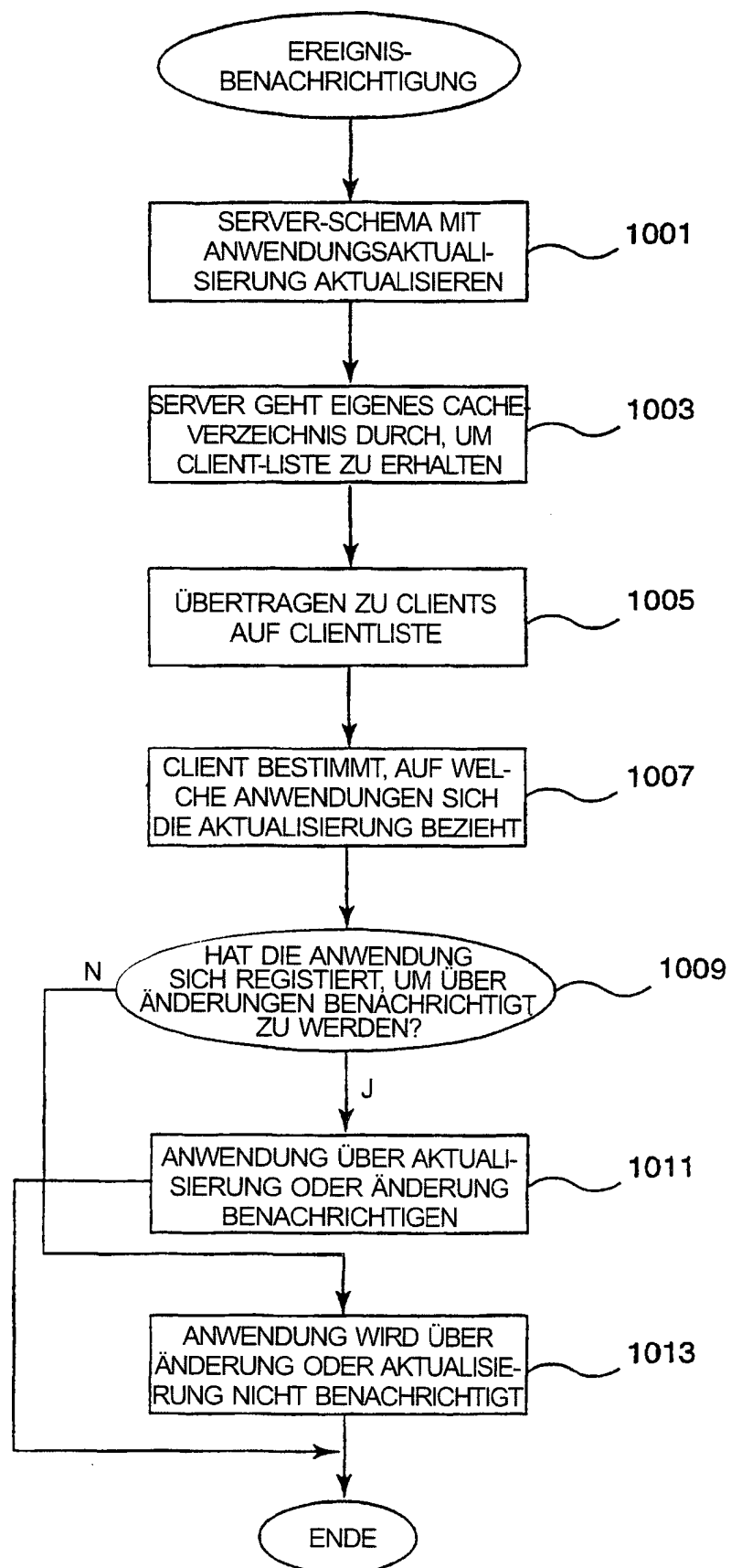


FIG. 10

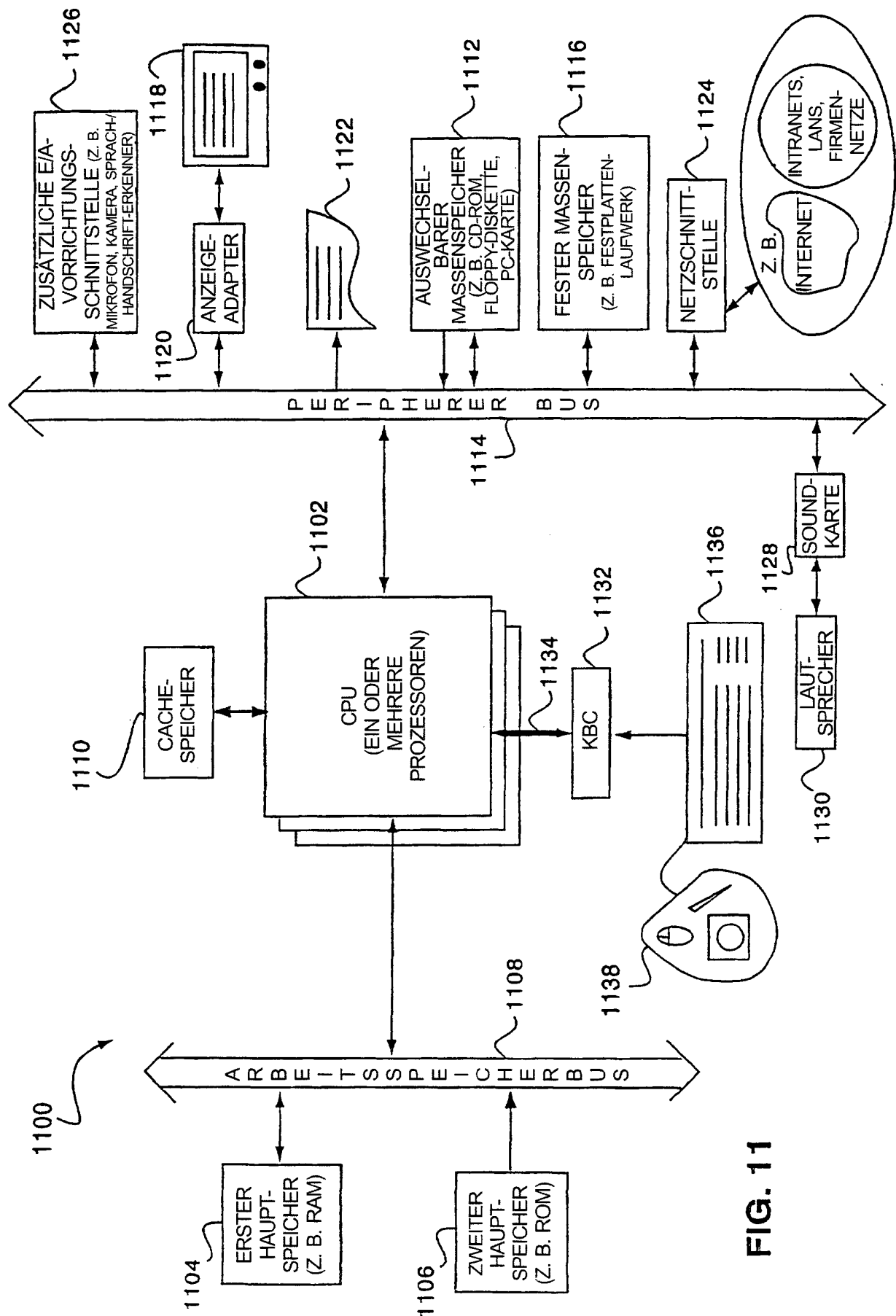


FIG. 11