



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 15 558 T2** 2008.05.08

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 376 977 B1**

(21) Deutsches Aktenzeichen: **603 15 558.8**

(96) Europäisches Aktenzeichen: **03 009 717.4**

(96) Europäischer Anmeldetag: **30.04.2003**

(97) Erstveröffentlichung durch das EPA: **02.01.2004**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **15.08.2007**

(47) Veröffentlichungstag im Patentblatt: **08.05.2008**

(51) Int Cl.⁸: **G06F 9/46** (2006.01)
H04L 29/06 (2006.01)

(30) Unionspriorität:

187221 28.06.2002 US

(73) Patentinhaber:

Microsoft Corp., Redmond, Wash., US

(74) Vertreter:

**Grünecker, Kinkeldey, Stockmair &
Schwanhäusser, 80538 München**

(84) Benannte Vertragsstaaten:

**AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB,
GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK,
TR**

(72) Erfinder:

**Alles, David, Seattle, Washington 98103, US;
Moore, George M., Issaquah, Washington 98029,
US**

(54) Bezeichnung: **Verteiltes Rechnersystem für Vorrichtungsressourcen basierend auf Identität**

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

GEBIET DER ERFINDUNG

[0001] Die Erfindung bezieht sich im allgemeinen auf Computersysteme, und im Speziellen auf Computerressourcen.

HINTERGRUND DER ERFINDUNG

[0002] In frühen Personalcomputersystemen mussten Anwendungsprogramme mit Fachwissen über verschiedene Computergeräte programmiert werden, damit die Anwendung solche Geräte verwenden kann. Zum Beispiel, um auf einem bestimmten Drucker zu drucken musste ein Textverarbeitungsprogramm wissen, wie es auf dem Drucker druckt, und lieferte im Allgemeinen einen Treiber für jeden unterstützten Drucker. Alternativ würde ein Drucker Treiber für die gängigsten Programm bereitstellen. Der Benutzer musste jedes Programm konfigurieren, um dieses Programm zu informieren, mit welchem Drucker der Personalcomputer verbunden ist.

[0003] Heutige Betriebssysteme, wie z.B. Microsoft Windows[®], stellten Entwicklern eine Möglichkeit bereit, um Anwendungen mit solchen Geräten zu koppeln, ungeachtet der spezifischen Bedürfnisse und Erfordernisse der Gerätetreiber. Zum Beispiel teilt bei dem Microsoft Windows[®]-Betriebssystem ein Benutzer oder das Gerät dem Betriebssystem mit, welcher Drucker (oder welche Drucker) verbunden ist (sind), wobei anschließend jede Anwendung in der Lage ist, mit dem Druckertreiber über das Betriebssystem zu kommunizieren. Als ein Ergebnis mussten Anwendungsprogramme sich gewöhnlich nur mit einer Reihe von standardisierten Druckfunktionen, die durch das Betriebssystem bereitgestellt wurden, befassen, um das Drucken zu bewerkstelligen. Die enormen Vorteile solch einer Abstraktion können nicht unterbewertet werden, z.B., ermöglicht diese Abstraktion irgendeinem von Tausenden von Softwareprogrammen, mit irgendeinem von Tausenden von Druckern und/oder anderen Geräten zu interagieren, ohne dass das Programm die Details über den Druck im Voraus wissen muss (oder anderweitig bereitstellen muss).

[0004] Die .NET-Technologie von Microsoft Corporation bietet einen etwas ähnlichen Abstraktionstyp mit Bezug auf die Daten eines Benutzers oder einer Firma. Im allgemeinen kann mit der .NET-Technologie irgendein .NET-geeignetes Programm auf irgendeinem Gerät mit den Daten eines Benutzers über zentralisierte, schema-basierte Dienste interagieren. Aufgrund der Schematisierung brauchen die Programme kein spezifisches Wissen über die Details dieser Daten, welche, z.B. in proprietären Dateiformaten gespeichert worden sind, sondern braucht nur in der Lage sein, mit (dem Lesen und Schreiben von) Daten, die gemäß einem oder mehreren Schemas formatiert sind, umzugehen.

[0005] Mit der .NET-Technologie steuert die Identität eines Benutzers (oder einer anderen Einheit) den Zugriff auf zentral vorgehaltene Daten, z.B. kann ein Benutzer einen .NET-Postfachdienst nutzen, der seine oder ihre elektronischen Maildaten zentral gespeichert hält (z.B. im Internet oder einem Intranet), und kann anschließend auf diese Daten über den .NET-Postfachdienst zu jeder Zeit von irgendeinem Gerät aus zugreifen, das im Stande ist, sich mit dem Internet oder einem Intranet zu verbinden, und das ein Programm aufweist, das im Stande ist, mit dem .NET-Postfachdienst zu interagieren. Die Identität des Benutzers wird verwendet, um den Benutzer mit den richtigen Daten sicher abzugleichen. Benutzer können ebenso den Zugriff auf Daten durch andere Benutzer regelbar erlauben, z.B. kann ein Benutzer wahlweise bestimmte geschäftliche Kontaktinformationen für Geschäftspartner verfügbar machen, persönliche Kontaktinformationen für Freunde verfügbar machen, und so weiter, alles basierend auf den Identitäten der anderen.

[0006] Während .NET deshalb gewaltige datenbasierte Vorteile für Benutzer, Firmen und andere Einheiten bietet, sind der zentralisierte Service und die zentralisierten Daten nicht besonders darauf bedacht, wie ein Benutzer verschiedene Geräte und die Ressourcen jener Geräte verwendet. Zum Beispiel waren Geräte wie ein Personalcomputer und Taschencomputer für einige Zeit in der Lage, sich direkt miteinander zu synchronisieren. Zum Teil ist die .NET-Technologie eine andere Möglichkeit, solche Daten synchronisiert zu halten, obwohl es indirekt geschieht, ohne dass spezialisierte und proprietäre Synchronisationsprogramme benötigt werden.

[0007] Jedoch kann ein Benutzer, selbst mit .NET, nicht einfach ein Gerät verwenden, um auf die Ressourcen eines anderen Gerätes zuzugreifen, wie z.B. die Rechenleistung, den Speicher, die Bandbreite, Eingabe- und Ausgabemechanismen, Programme und/oder andere Ressourcen eines Gerätes im Namen eines anderen Gerätes zu verwenden, z.B. um eine komplexe oder besser geeignete Aufgabe auf dem leistungsfähigeren Gerät durchzuführen. Was benötigt wird, ist eine allgemeine, konsistente Möglichkeit, Computerressourcen von verschiedenen Computergeräten gegenseitig für eine überkreuzende Ressourcenverwendung (cross-re-

source usage) dezentral (remotely) zu entdecken und darzulegen.

[0008] EP0952513 offenbart die automatische Konfiguration von einem Druckerserver, um es einem Client zu ermöglichen, den Druckertreiber, der in dem Druckerserver liegt, zu verwenden, um einen Druck auf einem entfernten Drucker durchzuführen.

[0009] US-A-5 898 870 beschreibt Lastverteilung in einem parallelen Computersystem, das eine Computergruppe, die durch eine Vielzahl von Computern gebildet wird, und einen Betriebsverwaltungsmechanismus aufweist, welcher ein Computer zum Verwalten des Betriebs der Computergruppe ist. Erstens werden Zielwerte für eine Ressourcennutzung pro Arbeitsgang in dem Betriebsverwaltungsmechanismus im Voraus gesetzt, und anschließend informieren die entsprechenden Computer der Computergruppe den Betriebsverwaltungsmechanismus über gesammelte Stati der Ressourcennutzung pro Arbeitsgang. Als Antwort auf eine Anfrage durch einen Computer, um einen Arbeitsprozess neu zu starten, wird der Zielwert für die Ressourcennutzung für den Arbeitsprozess mit dem Status der Ressourcennutzung davon auf einer Statusverwaltungstabelle für Ressourcennutzung verglichen, welche bei der Initialisierung des Systems bei dem Betriebsverwaltungsmechanismus vorbereitet wird. Desweiteren wird einer der Computer aus der Vielzahl von Computern, die die Computergruppe ausmachen, auf der Basis des Ergebnisses des Vergleichs als das optimale System ausgewählt und wird aufgefordert, den Arbeitsprozess auszuführen.

[0010] US-A-6 078 955 beschreibt ein Verfahren zur Steuerung eines Computersystems einschließlich einer Vielzahl von Computern, die durch ein Netzwerk miteinander verbunden sind und von einer Vielzahl von Nutzern verwendet wird. Das Computersystem schließt mindestens einen Server, einen Administrationsserver und eine Vielzahl von Terminals ein. Der Administrationsserver ist für die Verwaltung aller Computerressourcen und aller Arbeiten, die auf den Computern ausgeführt werden, sowie für die gesamten Benutzerinformationen in den Computersystemen verantwortlich. Der Administrationsserver sammelt und verwaltet Informationen in einer Systemressourcenverwaltungstabelle. Desweiteren, wenn sich ein Benutzer bei einem Terminal an dem System anmeldet, werden die Anmeldeinformationen und die Terminalinformationen, das für die Anmeldung verwendet wurde, an den Administrationsserver gesendet. Nachdem eine Verarbeitungsanfrage von einem Terminal zu dem Administrationsserver gesendet wurde, wird ein Server oder Terminal, welches bewertet wird, in der Lage zu sein, einen gegebenen Prozess am effizientesten handzuhaben, auf Grund der vorherrschenden Arbeitsbelastung der Server in dem Netzwerk und der Eigenschaft und dem Ort der Datei und des Programms, das für den vorgesehenen Prozess verwendet wird, ermittelt, und der Ausführungsort des Prozesses wird dem entsprechenden Terminal mitgeteilt. Das Terminal fragt anschließend einen Computer entsprechend der Mitteilung an, den Prozess auszuführen.

KURZFASSUNG DER ERFINDUNG

[0011] Es ist die Aufgabe der Erfindung, ein Verfahren und System zum Entdecken und Darlegen von Computerressourcen von verschiedenen Computergeräten einander für eine kreuzweise Ressourcenverwendung bereitzustellen.

[0012] Diese Aufgabe wird durch die Erfindung wie in den unabhängigen Ansprüchen beansprucht gelöst.

[0013] Bevorzugte Ausführungsformen sind durch die abhängigen Ansprüche definiert.

[0014] Kurz beschrieben, bietet die vorliegende Erfindung ein System und Verfahren zum Entdecken und Beschreiben von Computerressourcen, die bei verschiedenen Computergeräten, die mit einer Identität assoziiert sind, verfügbar sind, und zum Darlegen jener Ressourcen als Dienste, die durch Softwareprogramme adressierbar sind, so dass Programme auf den Geräten die Ressourcen eines anderen Gerätes wirksam einsetzen können. Die Gerätedienste sind dadurch identitätsbasiert, dass Geräte eines Gerätebesitzers sich gegenseitig und/oder andere Geräte, auf die durch einen Benutzer basierend auf der Identität des Benutzers oder eines anderen Benutzers zugegriffen werden kann, entdecken. Desweiteren kann, basierend auf der Identität, ein Gerätebesitzer (oder Administrator) steuern, welche anderen Benutzer und/oder anderen Geräte die Ressourcen von jedem, im Besitz befindlichen Gerät wirksam einsetzen können, sowie in welchem Umfang.

[0015] Computerressourcen können z.B. die Speicherkapazität der Geräte, die Bandbreite, Prozessorleistung (z.B. die CPU- und RAM-Fähigkeiten), die Eingabemechanismen und -verfahren, das Rendern und anderen Ausgabemechanismen und -verfahren (z.B. Bildschirme, Drucker und Audioausgabe), und Programme einschließen. Ein Gerät wird deshalb in seine Ressourcen abstrahiert, welche anschließend von anderen Geräten in einer zusammenstellenden Art und Weise programmatisch adressierbar sind (z.B. können die Remo-

te-Ressourcen auf solche eine Weise definiert sein, dass sie wiederum aus kleineren Ressourcen bestehen, die in der größeren Definition der Ressource enthalten sind). Mit anderen Worten, damit Geräte in einem intelligenten, verteilten Computermodell teilnehmen, sind verschiedene Aspekte von jedem teilnehmenden Gerät abstrahiert und von anderen Geräten programmatisch adressierbar gemacht worden. Ein vereinheitlichter, identitätsbasierter Mechanismus entdeckt die Liste der Remote-Ressourcen auf dem Gerät. Eine Reihe von einem oder mehreren Schemata beschreibt die Remote-Ressourcen auf dem Gerät einheitlich, und ein vereinheitlichtes Sicherheitsmodell erlaubt es dem Benutzer des Gerätes, zu steuern, wer Zugriff auf welche Ressource und in welchem Umfang hat.

[0016] Mit der vorliegenden Erfindung werden gewünschte Computerlösungen für eine große Anzahl von Beispielszenarien ermöglicht, einschließlich einer Reihe an Echtzeit-Szenarien, in welchen ein Benutzergerät mit einem oder mehreren Remote-Computergeräten in Echtzeit interagiert, und eine Reihe von geplanten oder vorhergesagten Szenarien, die unsichtbar und automatisch im Namen eines Benutzers arbeiten.

[0017] Andere Vorteile werden von der nachfolgenden detaillierten Beschreibung ersichtlich, wenn sie in Verbindung mit den Figuren genommen wird, in welchen:

KURZE BESCHREIBUNG DER FIGUREN

[0018] [Fig. 1A](#) ein Blockdiagramm ist, das allgemein ein Computersystem darstellt, in dem die vorliegende Erfindung eingebaut werden kann;

[0019] [Fig. 1B](#) ein Blockdiagramm ist, das allgemein ein anderes Computersystem darstellt, in dem die vorliegende Erfindung eingebaut werden kann;

[0020] [Fig. 2](#) ein Blockdiagramm ist, das verschiedene Komponenten in einer verteilten Computerumgebung gemäß einem Aspekt der vorliegenden Erfindung darstellt;

[0021] [Fig. 3](#) ein Blockdiagramm ist, das allgemein eine Architektur zum Entdecken und Austauschen von Ressourceinformationen gemäß einem Aspekt der vorliegenden Erfindung darstellt;

[0022] [Fig. 4](#), [Fig. 5](#) Blockdiagramme sind, die Beispiele darstellen, wie die vorliegende Erfindung die Erleben des Computer durch einen Benutzer durch das Verteilen von Computeroperationen auf Ressourcen gemäß einem Aspekt der vorliegenden Erfindung verbessern kann;

[0023] [Fig. 6](#) ein Flussdiagramm ist, das eine beispielhafte Logik und in Beziehung stehende Operationen darstellt, die durchgeführt werden, um die gemeinsame Benutzung von Ressourcen in einer verteilten Computerumgebung gemäß einem Aspekt der vorliegenden Erfindung ermöglicht; und

[0024] [Fig. 7](#) ein Blockdiagramm ist, das allgemein eine Beispielimplementation darstellt, die Ressourcen einschließt, die gemäß einem Aspekt der vorliegenden Erfindung verteilt sind.

DETAILLIERTE BESCHREIBUNG

Exemplarische Arbeitsumgebung

[0025] [Fig. 1A](#) stellt ein Beispiel einer geeigneten Computersystemumgebung **100** dar, auf der die Erfindung implementiert werden kann. Die Computersystemumgebung **100** ist nur ein Beispiel einer geeigneten Computerumgebung, und ist nicht gedacht, irgendeine Einschränkung auf den Umfang der Verwendung oder Funktionalität der Erfindung vorzuschlagen. Noch sollte die Computerumgebung **100** so interpretiert werden, als gäbe es irgendeine Abhängigkeit oder Erfordernis bezüglich irgendeiner Komponente oder einer Kombination von Komponenten, die in der exemplarischen Arbeitsumgebung **100** gezeigt sind.

[0026] Die Erfindung ist mit vielen anderen Allzweck- oder Spezialzweck-Computersystemumgebungen oder -konfigurationen betriebsbereit. Beispiele von gut bekannten Computersystemen, -umgebungen und/oder -konfigurationen, die für eine Verwendung mit der Erfindung geeignet sind, schließen ein, sind aber nicht darauf begrenzt: Personalcomputer, Servercomputer, tragbare oder Laptopgeräte, Tablet-Geräte, Multiprozessorsysteme, mikroprozessorbasierte Systeme, Set-Top-Boxen, programmierbare Unterhaltungselektronik, Netzwerk-PCs, Minicomputer, Mainframecomputer, verteilte Computerumgebungen, die irgendeins der oben genannten Systeme oder Geräte einschließen, und ähnliches.

[0027] Die Erfindung kann in dem allgemeinen Kontext von Computerausführbaren Instruktionen, wie z.B. Programmmodulen, die durch einen Computer ausgeführt werden, beschrieben werden. Im allgemeinen schließen Programmmodule Routinen, Programme, Objekte, Komponenten, Datenstrukturen und so weiter ein, die bestimmte Aufgaben durchführen oder bestimmte abstrakte Datentypen implementieren. Die Erfindung kann ebenso in verteilten Computerumgebungen praktiziert werden, wo Aufgaben durch Remote-Verarbeitungsgeräte durchgeführt werden, die durch ein Kommunikationsnetzwerk verbunden sind. In einer verteilten Computerumgebung können Programmmodule auf lokalen und/oder Remote-Computerspeicherdatenträgern liegen, einschließlich Datenspeichergeräten.

[0028] Mit Bezug auf [Fig. 1A](#) schließt ein exemplarisches System zum Implementieren der Erfindung ein Allzweckcomputergerät in der Form eines Computers **110** ein. Komponenten des Computers **110** können einschließen, sind aber nicht darauf begrenzt, eine Prozessoreinheit **120**, einen Systemspeicher **130** und einen Systembus **121**, der verschiedene Systemkomponenten einschließlich des Systemspeichers mit der Prozessoreinheit **120** koppelt. Der Systembus **121** kann irgendeiner von verschiedenen Typen von Busstrukturen, einschließlich eines Speicherbusses oder Speichercontrollers, eines Peripheriebusses und eines lokalen Busses, der irgendeine einer Vielfalt von Busarchitekturen verwendet, sein. Als Beispiel, und nicht Einschränkung, schließen solche Architekturen Industry-Standard-Architecture-Bus (ISA-Bus), Micro-Channel-Architecture-Bus (MCA-Bus), Enhanced-ISA-Bus (EISA-Bus), Video-Electronics-Standards-Association-local-Bus (VESA-local-Bus) und Peripheral-Component-Interconnect-Bus (PCI-Bus), ebenso bekannt als Mezzanine-Bus, ein. Es ist zu beachten, dass irgendeines bzw. alle der Elemente, die im Computer **110** gezeigt sind, miteinander durch standardisierte, Hochgeschwindigkeitsnetzwerke verbunden sein können, einschließlich jener, die über ein großes geografisches Gebiet verteilt sind. Zum Beispiel kann der Systemspeicher **130** an einem physikalisch unterschiedlichen Ort als die Prozessoreinheit **120** liegen, ist aber in einer zusammengesetzten Weise als logischer Computer **110** definiert.

[0029] Der Computer **110** schließt üblicherweise eine Vielfalt von Computerlesbaren Datenträgern ein. Computerlesbare Datenträger können irgendeiner von verfügbaren Datenträgern sein, auf die durch den Computer **110** zugegriffen werden kann, und schließt sowohl flüchtige als auch nicht-flüchtige Datenträger und entfernbare und nicht-entfernbar Datenträger ein. Als Beispiel, und nicht Einschränkung, können Computer-lesbare Datenträger Computerspeicherdatenträger und Datenübertragungsträger einschließen. Computerspeicherdatenträger schließen sowohl flüchtige als auch nicht-flüchtige, entfernbare als auch nicht-entfernbar Datenträger ein, die mit irgendeinem Verfahren oder Technologie zum Speichern von Informationen implementiert sind, wie z.B. Computerlesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten. Computerspeicherdatenträger schließen ein, sind aber nicht darauf begrenzt, RAM, ROM, EEPROM, Flashmemory oder andere Speichertechnologie, CD-ROM, Digital Versatile Disks (DVD) oder andere optische Diskspeicher, magnetische Kassetten, magnetische Bänder, magnetische Diskspeicher oder andere magnetische Speichergeräte oder irgendein anderer Datenträger, welcher zum Speichern der gewünschten Informationen verwendet werden kann, und auf den durch den Computer **110** zugegriffen werden kann. Datenübertragungsträger verkörpern üblicherweise Computer-lesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten in einem modulierten Datensignal, wie z.B. einer Trägerwelle oder anderen Transportmechanismen und schließen irgendetwelche Informationsliefermedien ein. Der Begriff „moduliertes Datensignal“ bedeutet ein Signal, das eine oder mehrere seiner Charakteristiken in solch einer Weise gesetzt oder verändert hat, dass Informationen in dem Signal kodiert sind. Als Beispiel und nicht Einschränkung, können Datenübertragungsträger verkabelte Datenträger, wie z.B. ein verkabeltes Netzwerk oder eine Direktverkabelte Verbindung, und kabellose Datenträger, wie z.B. Akustik-, RF-(Funk-), Infrarot- und andere kabellose Datenmedien, einschließen. Kombinationen von irgendetwelchen der oben genannten sollten in dem Umfang der Computer-lesbaren Datenträger enthalten sein.

[0030] Der Systemspeicher **130** schließt Computerspeicherdatenträger in der Form von flüchtigem und/oder nicht-flüchtigem Speicher ein, wie z.B. Festwertspeicher (read only memory – ROM) **131** und Schreib-Lese-Speicher (random access memory – RAM) **132**. Ein Basic-Input/Output-System **133** (BIOS), das die Basisroutinen enthält, die helfen, Informationen zwischen Elementen innerhalb des Computers **110** zu transferieren, wie z.B. während des Hochfahrens, ist üblicherweise in dem ROM **131** gespeichert. Der RAM **132** enthält üblicherweise Daten und/oder Programmmodule, auf die sofort durch die Prozessoreinheit **120** zugegriffen werden kann und/oder auf denen derzeit durch die Prozessoreinheit **120** gearbeitet wird. Als Beispiel, und nicht Einschränkung, stellt die [Fig. 1A](#) ein Betriebssystem **134**, Anwendungsprogramme **135**, andere Programmmodule **136** und Programmdateien **137** dar.

[0031] Der Computer **110** kann ebenso andere entfernbare/nicht entfernbare, flüchtige/nicht-flüchtige Computerspeicherdatenträger einschließen. Nur als Beispiel stellt die [Fig. 1A](#) ein Festplattenlaufwerk **141**, das von

oder zu einem nicht entfernbaren, nicht flüchtigen magnetischen Datenträger liest oder schreibt, ein magnetisches Disklaufwerk **151**, das von oder zu einer entfernbaren, nicht flüchtigen magnetischen Disk **152** liest oder schreibt, und ein optisches Disklaufwerk **155**, das von oder zu einer entfernbaren, nicht flüchtigen optischen Disk **156** liest oder schreibt, wie z.B. einer CD-ROM oder einem anderen optischen Datenträger, dar. Andere entfernbare/nicht-entfernbare, flüchtige/nicht-flüchtige Computerspeicherdatenträger, die in der exemplarischen Arbeitsumgebung verwendet werden können, schließen ein, sind aber nicht darauf begrenzt, magnetische Bandkassetten, Flash-Memorykarten, Digital Versatile Disks, digitale Videobänder, Solid-State-RAM, Solid-State-ROM und ähnliches. Das Festplattenlaufwerk **141** ist üblicherweise mit dem Systembus **121** durch eine Schnittstelle für nicht-entfernbaren Speicher, wie z.B. Schnittstelle **140** verbunden, und das magnetische Disklaufwerk **151** und optische Disklaufwerk **155** sind üblicherweise mit dem Systembus **121** durch eine Schnittstelle für entfernbaren Speicher, wie z.B. Schnittstelle **150**, verbunden.

[0032] Die Laufwerke und ihre zugehörigen Computerdatenspeicherträger, die oberhalb diskutiert und in [Fig. 1A](#) dargestellt sind, bieten Speicherplatz für Computer-lesbare Instruktionen, Datenstrukturen, Programmmodule und andere Daten für den Computer **110**. In [Fig. 1A](#) ist z.B. das Festplattenlaufwerk **141** so dargestellt, dass es das Betriebssystem **144**, Anwendungsprogramme **145**, andere Programmmodule **146** und Programmdateien **147** speichert. Es ist zu beachten, dass diese Komponenten entweder dieselben sein können oder verschieden sein können von dem Betriebssystem **134**, Anwendungsprogrammen **135**, anderen Programmmodulen **136** und Programmdateien **137**. Das Betriebssystem **144**, die Anwendungsprogramme **145**, anderen Programmmodule **146** und Programmdateien **147** sind hier verschiedene Nummern gegeben worden, um darzustellen, dass sie wenigstens unterschiedliche Kopien sind. Ein Benutzer kann Befehle und Informationen in den Computer **110** durch Eingabegeräte, wie z.B. ein Tablet, oder elektrischen Digitalisierer **164**, ein Mikrofon **163**, eine Tastatur **162** und ein Zeigergerät **161**, allgemein als Maus, Trackball oder Touchpad bezeichnet, eingeben. Andere Eingabegeräte, die in [Fig. 1A](#) nicht gezeigt sind, können einen Joystick, ein Gamepad, eine Satellitenschüssel, einen Scanner oder ähnliches einschließen. Diese und andere Eingabegeräte sind oft mit deressoreinheit **120** durch eine Benutzereingabeschnittstelle **160** verbunden, die mit dem Systembus gekoppelt ist, aber können auch durch andere Schnittstellen und Busstrukturen verbunden sein, wie z.B. einen Parallelanschluss, Gameport oder Universal Serial Bus (USB). Ein Monitor **191** oder anderer Typ an Anzeigegerät ist ebenso mit dem Systembus **121** über eine Schnittstelle verbunden, wie z.B. eine Videoschnittstelle **190**. Der Monitor **191** kann ebenso mit einem Berührungssensitiven Bildschirmpanel oder ähnlichem integriert sein. Es ist zu beachten, dass der Monitor und/oder das berührungssensitive Bildschirmpanel (touch-screen Panel) physisch an ein Gehäuse, in dem das Computergerät **110** eingebaut ist, gekoppelt sein können, wie z.B. in einem Personalcomputer des Tablet-Typs. Zusätzlich können Computer, wie z.B. das Computergerät **110**, andere periphere Ausgabegeräte einschließen, wie z.B. Lautsprecher **195** und Drucker **196**, welche durch eine Ausgabegeräteschnittstelle **194** oder ähnliches verbunden sind.

[0033] Der Computer **110** kann in einer Netzwerkumgebung unter Verwendung logischer Verbindungen zu einem oder mehreren Remote-Computern, wie z.B. einem Remote-Computer **180**, arbeiten. Der Remote-Computer **180** kann ein Personalcomputer, ein Server, ein Router, ein Netzwerk-PC, ein Peer-Gerät oder anderer bekannter Netzwerkknoten sein, und enthält üblicherweise viele oder alle der oben mit Bezug auf Computer **110** beschriebenen Elemente, obwohl nur ein Speicherdatenträgergerät **181** in [Fig. 1A](#) dargestellt worden ist. Die logischen Verbindungen, die in [Fig. 1A](#) gezeigt sind, schließen ein Local Area Network (LAN) **171** und ein Wide Area Network (WAN) **173** ein, können aber ebenso andere Netzwerke einschließen. Solche Netzwerkumgebungen sind in Büros, Unternehmensweiten Netzwerken, Intranets und dem Internet alltäglich. Zum Beispiel kann der Computer **110** in der vorliegenden Erfindung eine Quellmaschine (Ursprungsgerät – source machine) umfassen, von welchem Daten migriert werden, und der Remote-Computer **180** kann die Zielmaschine umfassen. Es ist jedoch zu beachten, dass die Quell- und Zielmaschinen nicht durch ein Netzwerk oder andere Mittel verbunden sein müssen, sondern dass stattdessen die Daten über irgendeinen Datenträger migriert werden können, der in der Lage ist, durch die Quellplattform beschrieben zu werden und durch die Zielplattform oder -plattformen gelesen zu werden.

[0034] Wenn er in einer LAN-Netzwerkumgebung verwendet wird, ist der Computer **110** mit dem LAN **171** über eine Netzwerkschnittstelle oder -adapter **170** verbunden. Wenn er in einer WAN-Netzwerkumgebung verwendet wird, enthält der Computer **110** üblicherweise ein Modem **172** oder andere Mittel zum Herstellen einer Datenübertragung über das WAN **173**, wie z.B. das Internet. Das Modem **172**, welches intern oder extern sein kann, kann mit dem Systembus **121** über die Benutzereingabeschnittstelle **160** oder anderen geeigneten Mechanismen verbunden sein. In einer Netzwerkumgebung können Programmmodule, die mit Bezug auf Computer **110** dargestellt sind, oder Teile davon in dem Remote-Datenspeichergerät gespeichert sein. Als Beispiel, und nicht Einschränkung, stellt die [Fig. 1A](#) die Remote-Anwendungsprogramme **185** so dar, dass sie auf dem Speichergerät **181** liegen. Es wird begrüßt, dass die gezeigten Netzwerkverbindungen exemplarisch sind, und

andere Mittel zum Herstellen einer Datenübertragungsverbindung zwischen den Computern verwendet werden können.

[0035] [Fig. 1B](#) stellt ein Beispiel einer anderen geeigneten Arbeitsumgebung (Computergerät **188**) dar, in dem mindestens ein Teil der vorliegenden Erfindung implementiert werden kann. Wie die [Fig. 1A](#) ist die Arbeitsumgebung aus [Fig. 1B](#) nur ein weiteres Beispiel einer geeigneten Arbeitsumgebung und ist nicht gedacht, irgendeine Einschränkung auf den Umfang der Verwendung oder Funktionalität der Erfindung vorzuschlagen. Das andere Computergerät **188** in [Fig. 1B](#) enthält üblicherweise irgendeine Form von Computerlesbaren Datenträgern. Computer-lesbare Datenträger können irgendwelche verfügbaren Datenträger sein (wie oberhalb allgemein definiert), auf die durch das Computergerät **188** zugegriffen werden kann.

[0036] [Fig. 1B](#) zeigt funktionale Komponenten des Computergerätes **188**, wie z.B. einen tragbaren (in Taschengröße oder Tablet-)Personal Digital Assistant, der einen Prozessor **189**, einen Speicher **190**, eine Anzeige **192** und eine Tastatur **193** (welche eine physische oder virtuelle Tastatur sein kann) einschließt. Der Speicher **190** enthält im Allgemeinen sowohl flüchtigen Speicher (z.B. RAM) als auch nicht-flüchtigen Speicher (z.B. ROM, PCMCIA-Karten usw.). Ein Betriebssystem **194** liegt in dem Speicher **190** und wird auf dem Prozessor **189** ausgeführt, wie z.B. das Windows[®]-CE-Betriebssystem von Microsoft[®] Corporation oder ein anderes Betriebssystem.

[0037] Ein oder mehrere Anwendungsprogramme **195** werden in den Speicher **190** geladen und laufen auf dem Betriebssystem **194**. Beispiele für Anwendungen schließen E-Mailprogramme, Terminprogramme, PIM-Programme (Personal Information Managementprogramme), Textverarbeitungsprogramme, Tabellenverarbeitungsprogramme, Internetbrowserprogramme usw. ein. Der tragbare Personalcomputer **188** kann ebenso einen Benachrichtigungsmanager **196** einschließen, der in den Speicher **190** geladen ist, und welcher auf dem Prozessor **189** ausgeführt wird. Der Benachrichtigungsmanager **196** behandelt Benachrichtigungsanfragen, z.B. von den Anwendungsprogrammen **195**.

[0038] Der tragbare Personalcomputer **188** hat eine Energieversorgung **197**, welche als eine oder mehrere Batterien implementiert ist. Die Energieversorgung **197** kann des Weiteren eine externe Energiequelle einschließen, die die eingebauten Batterien außer Kraft setzt oder wieder auflädt, wie z.B. ein Wechselstromadapter oder eine am Strom angeschlossene Basisstation.

[0039] Der exemplarische tragbare Personalcomputer **188**, der in [Fig. 1B](#) dargestellt ist, wird mit drei Typen von externen Benachrichtigungsmechanismen gezeigt: einer oder mehrerer lichtimitierenden Dioden (LEDs) **198** und einem Audio-Erzeuger **199**. Diese Geräte können direkt mit der Stromquelle **197** gekoppelt sein, so dass sie, wenn sie aktiviert werden, für eine Dauer, die durch einen Benachrichtigungsmechanismus vorgegeben wird, eingeschaltet bleiben, selbst wenn der Prozessor **189** des tragbaren Personalcomputers und andere Komponenten heruntergefahren sind, um Batteriestrom zu sparen. Die LED **198** bleibt vorzugsweise unbegrenzt angeschaltet, bis der Benutzer etwas unternimmt. Es ist zu beachten, dass heutige Versionen des Audio-Erzeugers **199** zu viel Energie für die heutigen Batterien von tragbaren Personalcomputern verwenden und deshalb ist er konfiguriert, sich auszuschalten, wenn der Rest des Systems dies tut, oder nach einer begrenzten Dauer nach der Aktivierung.

Identitätsbasiertes verteiltes Computing für Gerätere Ressourcen

[0040] Die vorliegende Erfindung ist im Allgemeinen auf ein System und Verfahren gerichtet, das die Verteilung von Ressourcen von Geräten auf andere Geräte ermöglicht. Die vorliegende Erfindung ist gewöhnlich nicht damit involviert, was die Geräte mit den Ressourcen voneinander machen, was zu dieser Zeit kommuniziert wird oder wie die Kommunikation zwischen den Geräten stattfindet. Die vorliegende Erfindung kann im Allgemeinen eher mehr als eine strukturierte Art und Weise für Geräte betrachtet werden, um etwas über ihre gegenseitigen Ressourcen, Fähigkeiten oder anderen Details herauszufinden, und um es anschließend Geräten zu erlauben, die Ressourcen ohne weitere Hilfe zu teilen. Obwohl es für die vorliegende Erfindung nicht notwendig ist, wird als Teil des Ermöglichens von Ressourcenverteilung die Sicherheit für einen praktischen Zweck bereitgestellt, und mindestens um sicherzustellen, dass gültige Berechtigungsnachweise erfordert werden, um die Ressourcen eines anderen Gerätes zu verwenden. Deshalb ist die vorliegende Erfindung vorzugsweise identitätsbasiert, so dass verteilte Ressourcen Beziehungen entsprechend den Identitätsdaten (z.B. Berechtigungsnachweise), die von den Geräten bereitgestellt werden, miteinander haben.

[0041] Im Allgemeinen ist die vorliegende Erfindung auf ein verteiltes Computing zwischen Geräten eines Benutzers gerichtet, wobei ein Benutzer (oder Besitzer/Administrator) irgendjemand ist, der autorisiert ist, die Ge-

räte zu steuern und/oder zu verbinden. Weil im Allgemeinen Geräte zunehmend disaggregiert werden, können die hier beschriebenen Verfahren und Systeme jedoch genauer als ein verteiltes Computing von Ressourcen beschrieben werden. Zum Beispiel kann ein einzelnes und verteiltes Gerät aus vielen Ressourcen bestehen, aber kann so wenig wie nur eine seiner Ressourcen teilen (z.B. nur seinen nicht-flüchtigen Speicher), um als ein Ressourcenanbieter teilzunehmen. Dennoch sind die Begriffe „Gerät“ und „Ressource“, wie sie hier verwendet werden, im Allgemeinen austauschbar, ausgenommen dort, wo es andernfalls offensichtlich ist, z.B. werden das Verwenden eines anderen Gerätes oder das Verwenden einer Ressource des anderen Gerätes, um irgendeine Computeraufgabe auszuführen, im Allgemeinen als dieselbe Sache betrachtet. Ebenso ist, wie hierin verwendet, die Bandbreite eine Geräteressource, obwohl die Bandbreite gewöhnlich nicht als einem bestimmten Gerät zugehörig betrachtet wird.

[0042] Wie es in [Fig. 2](#) allgemein dargestellt ist, schließt eine verteilte Computerumgebung **200**, die gemäß verschiedener Aspekte der vorliegenden Erfindung konstruiert wurde, ein erstes Gerät **202** (Gerät A, wie z.B. das Computersystem **110** aus [Fig. 1A](#)) und ein zweites Gerät **212** (Gerät B, wie z.B. das Computersystem **188** aus [Fig. 1B](#)) zusammen mit eventuell anderen Geräten (nicht gezeigt) ein. Jedes Gerät, das in der verteilten Computerumgebung **200** teilnimmt, enthält einen Manager, **204** und **214**, für verteiltes Computing. Im Allgemeinen, und wie unterhalb beschrieben, können die Manager **204** und **214** für verteiltes Computing einigen Code umfassen, die miteinander über irgendeinen Datenträger, z.B. kabelloses FM (Frequency Modulation), in Sichtweite wie z.B. Infrarot, Satellit, eine direkte Verbindung, das Internet oder ein Intranet kommunizieren. In der Tat kann mehr als ein solcher Datenträger verfügbar sein, und die Manager **204** und **214** für verteiltes Computing können auswählen, welcher oder welche am besten für gegebene Umstände ist/sind.

[0043] Ein anfänglicher Zweck der Manager **204** und **214** für verteiltes Computing ist, sich gegenseitig zu lokalisieren. Solch eine Lokalisierung kann automatisch sein, z.B. dadurch, dass ein Gerät gelegentlich seine Verfügbarkeit an andere sendet, und/oder durch das Abfragen (Polling) nach der Verfügbarkeit von anderen Geräten, oder kann alternativ als Antwort auf irgendeine spezifische Anfrage von einem Programm (oder Benutzer), sich mit einem anderen Gerät zu verbinden, funktionieren. Wie es in [Fig. 2](#) dargestellt ist, enthalten die Manager **204** und **214** für verteiltes Computing jeder einen Sicherheitsmechanismus **206** bzw. **216**, die Berechtigungsnachweise **207**, **217** oder ähnliches (z.B. von einem Authentifizierungsdienst **220** erlangt) austauschen, um sicherzustellen, dass der Benutzer, der derzeit das Gerät bedient, von welchem es kontaktiert wird (und/oder das Gerät selbst), in der verteilten Computerumgebung zulässig ist, und gleichermaßen, wenn notwendig, dass das Gerät zulässig ist, welches kontaktiert wird. Es ist zu beachten, dass ein Programm, das verwendet wird, ebenso mit in die Sicherheits-/Zugriffskontrolle miteinbezogen wird, z.B. kann ein erstes Programm, das auf einem Gerät läuft, die Berechtigung haben, eine Ressource oder Ressourcen eines anderen Gerätes zu verwenden, aber nicht ein anderes Programm, unabhängig von oder zusätzlich zu der Benutzeridentität und/oder der Geräteidentität. Es ist weiter zu beachten, dass der Sicherheitsmechanismus pro Identität und pro Ressource arbeiten kann, z.B. kann eine Einheit berechtigt sein, ein Display des Gerätes zu verwenden, aber nicht seinen Speicher, während eine andere Einheit imstande sein kann, irgendeine verfügbare Ressource desselben Gerätes, einschließlich seines Speichers, zu verwenden. Dies ist ähnlich zu dem Punkt .NET-Sicherheitsmodell, in welchem Dienste Verfahren bereitstellen, die den Zugriff auf Daten eines Benutzer basierend auf der Identität, der definierten Rolle und des Umfangs für diese Rolle eines anfragenden Benutzers kontrollieren. Auf diese Weise können auf Daten durch ihren Benutzer zugegriffen werden und in einem Umfang, der durch den Besitzer bestimmt wird, geteilt werden. In der vorliegenden Erfindung ist es eine Geräteressource anstatt Daten, auf die zugegriffen wird, wobei die Identifikation eines Benutzers, seine definierte Rolle und der Umfang für diese Rolle das Ausmaß, bis zu dem auf eine Ressource zugegriffen werden kann, bestimmt.

[0044] In der Tat basieren die Sicherheitsmechanismen der vorliegenden Erfindung auf jenen, im Wesentlichen auf jenen, die für die Sicherheits-/Zugriffskontrolle in .NET-Diensten eingesetzt werden, wie es allgemein in den US-Patentanmeldungen mit den Seriennummern 10/003,754 mit dem Titel „IDENTITY-CENTRIC DATA ACCESS“ und der Seriennummer 10/003,767 mit dem Titel „AUTHORIZING A REQUESTING ENTITY TO OPERATE UPON DATA STRUCTURES“, die am 22. Oktober 2001 eingereicht wurden und dem Begünstigten der vorliegenden Erfindung zugeordnet sind, beschrieben. Zu diesem Zweck läuft mindestens eine kleine Untermenge der .NET-Dienste auf jedem Gerät, die erwartet, Computerressourcen in der Umgebung **200** anzubieten und zu teilen. Zusätzlich kann ein lokal gebundenes, intern konsistentes Sicherheitsmodell auf diesen Geräten ausgeführt werden, so dass eine lokale Sicherheit selbst dann bereitgestellt wird, wenn zwei lokale Peer-to-Peer-Geräte von dem zentralisierten (z.B. Passport-)Authentifizierungsservice **220** ([Fig. 2](#)) getrennt sind.

[0045] Die vorliegende Erfindung ist nicht auf irgendein bestimmtes Übertragungsmedium beschränkt, son-

dem bietet an, dass die verteilten Ressourcedienste für ein Zielgerät direkt adressierbar sind, oder über das Internet oder ein Intranet, wie z.B. über die zentralisierte Reihe an .NET-Diensten im Web, abgefragt werden können. Wenn es mit dem Internet oder dem Intranet verbunden ist, kann ein Programm die letztbekanntesten verfügbaren Computerressourcen auf irgendeinem Gerät zugehörig zu der Identität des Benutzers abfragen, selbst wenn das Gerät offline oder anderweitig nicht erreichbar ist. Die Bandbreite ist ebenso eine Ressource und kann beeinflussen, wie die Ressourcen eines Gerätes geteilt werden, z.B. kann es effizienter sein, die Ressourcen eines Personalcomputers wirksam einzusetzen, wenn der Personalcomputer eine Hochgeschwindigkeitsverbindung mit dem Internet hat, aber nicht, wenn der Personalcomputer nur mit einer geringen Geschwindigkeit verbunden ist, wie z.B. wenn das andere Gerät eine Hochgeschwindigkeitsverbindung hat. In der Tat kann der Personalcomputer wollen, dass die Hochgeschwindigkeitsverbindung des anderen Gerätes wirksam eingesetzt wird, in welchem Falle dieses andere Gerät dann eine Ressource des Personalcomputers wirksam einsetzen kann, z.B. tauschen die Geräte Ressourcen aus, um ein optimiertes Erlebnis für den Benutzer bereitzustellen.

[0046] Gemäß einem Aspekt der vorliegenden Erfindung schließen die Manager **204** und **214** für verteiltes Computing ebenso jeder einen jeweiligen Auffindungsmechanismus **208** und **218** ein, die es den Geräten erlauben, geteilte Ressourcen/Fähigkeiten eines anderen teilnehmenden Gerätes aufzudecken, das heißt, wenn dieses andere Gerät konfiguriert ist, seine Ressourcen zu teilen. Die entsprechenden Ressourcendaten **209** und **219** sind formatiert (z.B. in XML) und werden gemäß einem definierten Schema, wie unterhalb beschrieben, ausgetauscht. Wie es in [Fig. 2](#) durch den Austausch der Schematisierten Daten **209** und **219** dargestellt ist, sind beide Geräte **202** und **204** konfiguriert, ihre Ressourcen (R_{A1} - R_{AM} und R_{B1} - R_{Bn}) mit anderen Geräten in der verteilten Computerumgebung zu teilen. Wie es jedoch begrüßt werden kann, kann in einer gegebenen Umgebung ein bestimmtes Gerät (z.B. ein Mobiltelefon) eher nur ein Konsument von Ressourcen sein als ein Anbieter von irgendeiner Ressource, oder umgekehrt, kann nur ein Anbieter und nicht ein Konsument sein.

[0047] Ein Gerät, das seine Ressourcen teilt, muss die Ressourcedaten nicht notwendigerweise direkt mit anderen Geräten kommunizieren, sondern kann stattdessen z.B. seine Ressourcen irgendwo anbieten, um von anderen Geräten entdeckt zu werden. Zum Beispiel wird ein .NET-Dienst (werden .NET-Geräte) bereitgestellt, der Geräteressourcefähigkeiten zentral speichert, so dass die Daten richtig formatiert sein können bevor zu irgendeinem Gerät gesendet werden, z.B. so dass keine hoch aufgelösten Grafiken zu einem Gerät gesendet werden, dass nicht in der Lage ist, diese anzuzeigen. Ein ähnlicher .NET-Ressourcedienst, oder ein erweiterter .NET-Gertedienst, kann bereitgestellt werden, um diese Informationen anderen Geräten anzubieten.

[0048] Die folgenden Tabellen legen manche Beispielschemata dar, die verwendet werden können (z.B. mit XML-formatierten Daten), um Ressource-(Speicher- und Prozessleistungs-)Informationen bezüglich der Geräte eines Benutzers auszutauschen, welche in diesem Beispiel einen Arbeits- und Heimpersonalcomputer ("Work PC" und "Home PC"), und ein Mobiltelefon ("Cell Phone") einschließen:

```

<myStorage>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
</myStorage>

```

```

<myProcessingPower>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
</myProcessingPower>

```

[0049] Es gilt als verstanden, dass das Schema vorzugsweise Ressourcen-basiert ist, das heißt, mehrere Geräte können für jeden Ressourcotyp aufgelistet sein, um das allgemeine Konzept einer Identität, die verteilte Ressourcen aufweist, die für eine Verwendung verfügbar sind, zu treffen, unabhängig von dem eigentlichen Gerät, auf dem die Ressource ist. Nichts desto trotz ist die vorliegende Erfindung nicht auf ein bestimmtes Schema und/oder Format beschränkt, z.B. können die Schemata gerätebasiert sein, so dass jedes Gerät stattdessen seine Liste an verfügbaren Ressourcen für andere Geräte bereitstellt, so dass dieselben Informationen bekannt sind.

[0050] Wie es allgemein in [Fig. 3](#) dargestellt ist, wird eine Architektur **300** bereitgestellt, damit Geräte in einem intelligenten, verteilten Computermodell teilnehmen können, in dem verschiedene Aspekte der teilnehmenden Geräte abstrahiert sind und von einem unterschiedlichen Gerät programmatisch adressierbar gemacht sind. Eine Universal-Plug-and-Play-(UPnP 2.0) Infrastruktur oder ähnliches kann als ein Basis-, Niedriglevelmechanismus **302** zum Entdecken dieser Ressourcen verwendet werden. Alternativ kann die Liste mit Ressourcen für verschiedene Computergeräte an einem gut bekannten Ort zwischengespeichert werden, wie z.B. einem Active-Directory-Speicher, um das Abfragen von Remote-Ressourcen selbst dann zu ermöglichen, wenn das

Gerät nur durch manche dazwischen liegende Geräte erreichbar ist, wie z.B. eine Firewall oder einen kabellosen Hub. Oberhalb des Auffindungsmechanismus liegen die Schemata **304**, z.B. ein Schema zugehörig zu jedem bestimmten Typ an Computerressource, wobei jedes dieser Schemata z.B. wie in der .NET-Architektur dargelegt (exposed) ist.

[0051] Es ist zu beachten, dass, während UPnP einen geeigneten Auffindungsmechanismus **304** bereitstellt, es nur Teil eines Prozesses des eigentlichen Verwendens der Remote-Computerressource ist. Deshalb werden die UPnP- und .NET-Technologien verwendet, um ein Teilen (gemeinsames Benutzen – sharing) der Ressource zu ermöglichen, werden aber nicht verwendet, um die Ressourcen, die auf einem Gerät verfügbar sind, tatsächlich einzusetzen. Sobald die Ressourcen bekannt sind, werden eher passende, fokussierte Protokolle für diesen bestimmten Ressourcebereich verwendet. Als Beispiel ist UPnP im Zusammenhang mit den .NET-identityHeader (Identitätskopf), endpointHeader (Endpunktkopf) und requestHeader (Anfragekopf) der SOAP-Bedeutungen (semantics) gut geeignet, um die Videobildschirmcharakteristiken eines Remote-Gerätes zu erlangen, obwohl ein Protokoll, wie z.B. RDP, verwendet wird, um Videodaten zu übertragen, nachdem das anfängliche Auffinden und Quittierung (handshaking) zwischen den Geräten stattgefunden hatte. Es ist zu beachten, dass dies ähnlich zu der Art und Weise ist, wie .NET mit Datendiensten heutzutage Kontakt aufnimmt, z.B. stellt .NET einen Mechanismus bereit, um sich mit einem elektronischen Endpunkt, der ein proprietäres Protokoll (z.B. Messenger-Protokoll) darstellt, aber sobald dieser Endpunkt verbunden ist, erlaubt es .NET dem Client, ein existierendes Echtzeitprotokoll zu verwenden. Andere Protokolle, die für andere Ressourcetypen passend sind, werden verwendet, z.B. CIFS+ zur Speicherung, welches sich sowohl für Firmen- als auch Heimspeicherszenarien gut eignet, und das Dateisystem des Benutzers befähigt, sich kontinuierlich auf andere Speichergeräte auszudehnen, ohne den Benutzer zu zwingen, mit Laufwerksbuchstaben, Namespaces (Namensräume) usw. umzugehen. Zusammengefasst stellt die vorliegende Erfindung einen identitätsbasierten Ansatz bereit, um diese zusätzlichen Ressourcen, die für eine Verwendung verfügbar sind, aufzufinden, aber hat anschließend mit deren Verwendung direkt nichts zu schaffen.

[0052] [Fig. 6](#) ist ein Beispielablaufdiagramm, das manche allgemeinen Aspekte der vorliegenden Erfindung darstellt und mit Schritt **600** beginnt, der das Empfangen einer Anfrage darstellt, eine geteilte Ressource (shared resource) zu verwenden. Die Anfrage kann allgemein sein, z.B. das Anfragen einer Liste an Ressourcen, die ein Benutzer auf Geräten verfügbar hat, die konfiguriert sind, teilzunehmen, oder kann eine Anfrage für eine bestimmte Ressource sein, z.B. als Antwort auf eine Bekanntmachung. Zum Zwecke dieses Beispiels wird die Anfrage als eine Anfrage für eine bestimmte Ressource behandelt.

[0053] Schritt **602** stellt die Evaluierung der Berechtigungsnachweise dar, wie sie z.B. durch den Microsoft-.NET-Passport-Authentifizierungsdienst bereitgestellt wird. Die Anfrage kann die Berechtigungsnachweise einschließen, oder die Berechtigungsnachweise können separat ausgetauscht werden, z.B. bevor, als Teil davon, oder nach dem Schritt **600**. Wenn es bei Schritt **602** keine Zulassung gibt geht der Schritt **602** zu Schritt **604** über, welcher das Zurückweisen der Anfrage darstellt. Es ist zu beachten, dass abhängig von einer gegebenen Umgebung die Anfrage ohne richtige Berechtigungsnachweise eher ignoriert werden kann, als speziell zurückgewiesen werden.

[0054] Wenn sie zugelassen wird, wird Schritt **606** ausgeführt, der das Ermitteln der Menge an angefragter Ressource, die verfügbar ist, darstellt. Zum Beispiel kann eine Ressource, wie z.B. Speicher begrenzt sein, weil es nur einen begrenzten freien Speicherraum gibt, und des Weiteren kann ein Gerät eingerichtet sein, nur einen Teil seines Speicherplatzes zum Teilen verfügbar zu machen, und den Rest für sich selbst zu reservieren. Eine Ressource, wie z.B. Prozessortakte können nur in dem Umfang verfügbar gemacht werden, wie der Prozessor nicht anderweitig verwendet wird und kann selbst dann noch weiter begrenzt sein, so dass andere Geräte diese nur bis zu einer maximalen Menge an nicht verwendeten Takten beanspruchen können. Irgendeine Ressource kann einen Zeitüberschreitungswert für sich assoziiert haben, welcher des Weiteren von seiner Verwendung abhängig ist, z.B. kann ein Speicher nicht für mehr als eine Stunde pro Anfrage geteilt werden, und die Anfrage wird beendet werden, wenn keine Dateisystemanfragen mindestens alle zwei Minuten erkannt werden. Auf diese Weise kann ein Gerät so eingerichtet sein, dass es seine Ressourcen in einem Maß teilt, dass durch seinen Besitzer/Administrator gesteuert wird.

[0055] Es ist zu beachten, dass der Ressourcenzugriff dynamisch sein kann, z.B. können nicht verwendeten Prozessortakte geteilt werden, jedoch ist CPU-Verwendung eine Zahl, die schwankt, und das anfragende Gastgerät kann zusammen mit anderen Geräten und/oder Prozessen auf dem Gerät, dass seine CPU teilt, priorisiert sein. Zum Beispiel kann ein Gastbenutzer nur einen kontrollierbaren Prozentsatz der Ressource konsumieren, wenn der Besitzer des Gerätes sie nicht anderweitig verwendet, wie z.B. Bandbreite oder Prozessorleistung. Sowohl CPU als auch Bandbreite sind Beispiele von Ressourcen, welche eher gemäß von Zeit-

grenzen (z.B. Takte oder Bits pro Sekunde) geplant werden, als nicht-zeitbasierte Ressourcen, wie z.B. Disk oder Speicherverwendung. Als ein Ergebnis wird der Planungsmechanismus für zeitbasierte Ressourcen im Allgemeinen grundsätzlich verschieden sein von dem für nicht-zeitbasierte Ressourcen. Es ist zu beachten, dass Kioskorte (kiosk locations) viele Szenarien ermöglichen würden, die für solche Gäste nützlich wären. Eine Ressource, wie z.B. ein Anzeigemonitor, kann entweder gänzlich verfügbar sein zum Teilen oder gar nicht, abhängig davon, ob er derzeit verwendet wird oder nicht. Obwohl es in [Fig. 6](#) nicht dargestellt wird, kann der Schritt **606** deshalb ebenso eine Anfrage zurückweisen (oder wenn passend ignorieren), wenn es keine verfügbare Menge einer angefragten Ressource zum Teilen gibt. Schritt **608** stellt das Zurückgeben einer Antwort an die anfragende Einheit dar, vorzugsweise die schemabasierte Antwort, die die Ressourcefähigkeitsdaten der anfragenden Einheit bereitstellt.

[0056] Schritt **610** stellt das Erlauben der angefragten Ressource, geteilt zu werden, dar, und schließt z.B. den Austausch von Daten zwischen dem anfragenden Gerät und dem teilenden Gerät ein, so dass das Teilende eine bestimmte, entsprechende Aufgabe für das anfragende Gerät durchführt. Schritt **612** testet, ob die Ressource verbraucht wurde, z.B. eine Zeitüberschreitung stattfand oder nicht länger verfügbar ist. Es ist zu beachten, dass dies als eine Schleife in [Fig. 6](#) für die Zwecke der Einfachheit gezeigt ist, es ist jedoch leicht verständlich, dass die Vorgänge ereignisgesteuert sein können. Des Weiteren kann ein Schritt (wie z.B. Schritt **606**) in der Schleife eingeschlossen sein/verwendet werden, um das Ereignis auszulösen, z.B. wird die Resourceverwendung dynamisch überwacht, um zu ermitteln, wann sie für die bestimmte Anfrage verbraucht ist. Es ist zu beachten, dass eine Ressource direkt erschöpft sein kann, wenn die anfragende Einheit das Gerät informiert, dass die Ressource nicht länger benötigt wird.

[0057] Schritt **614** stellt das Benachrichtigen des anfragenden Gerätes dar, dass die Ressource für seine Verwendung nicht länger verfügbar ist. Schritt **616** stellt das Deaktivieren der Resourceverwendung dar, so dass z.B. das anfragende Gerät, sobald sie erschöpft ist, eine neue Anfrage machen muss, wenn es die Ressource teilen (mitnutzen) möchte.

[0058] [Fig. 4](#) und [Fig. 5](#) stellen Beispiele für manche der vielen möglichen Verwendungen der vorliegenden Erfindung bereit. Wie es von den Beispielen der [Fig. 4](#) und [Fig. 5](#) verstanden werden wird, kann das Integrieren einer Rechen-Proxy-Unterstützung in ein Netzwerkrandgerät oder -apparat in ein verbessertes Benutzererlebnis für Benutzer eines internen Netzwerkes führen.

[0059] In einem ersten Beispiel, das in [Fig. 4](#) dargestellt ist, möchte ein kleines Mobilgerät **400** (wie z.B. ein Mobiltelefon) auf einem Drucker **402** drucken, jedoch hat das kleine Gerät **400** keinen passenden Druckertreiber. Dies kann daher kommen, dass das mobile Gerät **400** nicht ausreichend Speicher aufweist, um einen Druckertreiber, der für diesen Drucker **402** benötigt wird, zu installieren und zu führen, und/oder keinen einfachen Weg hat, den benötigten Druckertreiber zu installieren und/oder aus irgendeinem anderen Grund. Es ist zu beachten, dass dies ein echtes Szenario ist, und es eine vorgeschlagene Lösung ist, einen XHTML-PRINT als eine vereinfachte PDL (Page Description Language – Seitenbeschreibungssprache) von dem mobilen Gerät **400** an den Drucker **402** auszugeben, z.B. über irgendeine Netzwerkverbindung (durch die gestrichelte Linie 4 gezeigt). Obwohl XHTML-PRINT in einem gewissen Umfang ausreichend ist, hat diese Lösung eine Anzahl von Nachteilen, wie z.B., XHTML-PRINT ist nicht WYSIWYG (what you see is what you get), kann keine Zeilenumbrüche steuern und stellt im Allgemeinen ein schwaches Benutzererlebnis bereit.

[0060] Wenn jedoch über die vorliegende Erfindung das mobile Gerät **400** zuerst dynamisch einen Personalcomputer **404** entdeckt, der mit einem geeigneten Druckertreiber **406** ausgestattet ist, (und für welchen der Benutzer des mobilen Gerätes zur Verwendung autorisiert ist), stellt das mobile Gerät **400** stattdessen die Druckerdaten über den Personalcomputer **404** dem Druckertreiber **406** bereit, welcher wiederum eine WYSIWYG-Ausgabe unter Verwendung von PCL, Postscript oder Rasterisierung erzeugt. Wie oberhalb erwähnt stellt die vorliegende Erfindung den Auffindungsmechanismus bereit, und anschließend kann das mobile Gerät **400** auf irgendeine Art und Weise mit dem Personalcomputer **404** kommunizieren, wie z.B. durch eine leichtgewichtige PDL (light weight PDL), das eine höhere Genauigkeit (relativ zu XHTML-PRINT) hat. Das Ergebnis ist eine hoch qualitative Ausgabe, die sehr schnell durch den wirksamen Einsatz des schnelleren Prozessors und passenden Druckertreibers **406** in dem Personalcomputer **404** gemacht wird.

[0061] [Fig. 5](#) demonstriert ein zweites Beispiel, das im Allgemeinen auf das Bereitstellen einer verbesserten Benutzererfahrung gerichtet ist, wenn ein PDA (Personal Digital Assistant) **500** betrieben wird, der verwalteten Code entfernt (remote) auf einem Server **502** (z.B. ein ASP.NET-Server) ausführen möchte. Eine Möglichkeit, dies durchzuführen ist, den PDA mit dem Server zu verbinden, um den verwalteten Code auszuführen, (z.B. durch den wirksamen Einsatz von Mobile Controls (mobilen Steuerelementen), die in ASP.NET unterstützt wer-

den), wobei die Ergebnisse als HTML übermittelt werden, was durch die gestrichelte Linie in [Fig. 5](#) angezeigt wird. Der Server **502** skaliert den HTML-Inhalt auf die passende Bildschirmgröße des Gerätes **500** dynamisch, ohne dass der ISV Code mit speziellem Zweck (spezial-purpose code) für jedes Gerät **500** erzeugen muss. Dies erfordert jedoch viele Hin und Her's mit dem Server **500**, wodurch der Bildschirm für Aktualisierungen aufblitzt, und allgemein ein verringertes Benutzererlebnis bereitstellt.

[0062] Eine verbesserte Lösung, die durch die vorliegende Erfindung bereitgestellt wird, wird dadurch erzielt, dass der PDA dynamisch entdeckt, dass ein Personalcomputer **504**, der mit der Identität des Benutzer assoziiert ist, ein Proxy für eine Anzahl von Berechnungsressourcen sein kann, wie es allgemein in [Fig. 5](#) darstellt ist. In solch einem Fall gibt der PDA eine HTML-Anfrage für eine Remote-Seite über den Proxy-Personalcomputer **504** aus. Der Personalcomputer **504** fängt die Anfrage ab und startet eine Session mit dem Remote-Server **502**, um die UI zum Rendern des verwalteten Codes abzuladen (offload), z.B. unter Verwendung von WSDL als das Protokoll zu dem Server **502**. Möglicherweise wird die Bandbreite von dem Proxy-Personalcomputer **504** zu dem Server **502** weniger sein als die Bandbreite von dem PDA-Gerät **500** zu dem Proxy-Personalcomputer **504**, wodurch die geringere Datenmenge, die über die langsamere Verbindung zu und von dem Server **502** übertragen wird, zu einer antwortintensiveren Benutzerschnittstelle führt, wenn sie auf dem PDA **500** angezeigt wird, was ein verbessertes Benutzererlebnis bereitstellt. Natürlich können die zwei Bandbreiten zuerst evaluiert werden, um sicherzustellen, dass die Proxy-Lösung die optimalere ist.

[0063] Des Weiteren ist zu beachten, dass, wenn der PDA **500** mit lokal laufenden Steuerelementen ausgestattet ist, die der Server wirksam einsetzen könnte, ein noch besseres Erlebnis bereitgestellt werden kann. Zum Beispiel setzt ASP.NET automatisch irgendwelche lokalen Steuerelemente (z.B. ein Steuerelement zum Holen des Datums (date picker control)), die lokal auf einem PDA laufen, auf dem ein Windows®-basiertes Betriebssystem läuft, automatisch wirksam ein. Dies führt zu noch weniger Hin- und Her-Übertragungen zu dem Server **502** und/oder Proxy **504**, mit weniger resultierenden Bildschirmaktualisierungen/-Blitzen.

[0064] Wie es begrüßt werden kann, gibt es eine Vielzahl anderer Beispiele, die durch die Verwendung von Computerressourcen, die sich um eine Identität drehen, anstatt um die bestimmten Geräte, die jene Ressourcen mit sich bringen, ermöglicht werden. Die Resourceschemata und Bedienstrukturen sind nicht auf ein einfaches Beschreiben der Ressourcen begrenzt, sondern schließen ebenso Elemente ein, um jenen Ressourcen zu helfen, miteinander zu arbeiten. Dies kann von den folgenden zwei Typen an Szenarien verstanden werden, nämlich einer Reihe von Echtzeit-Szenarien, in denen der Benutzer mit einem Remote-Computergerät in Echtzeit interagiert, und einer Reihe von geplanten oder vorhergesagten Szenarien. Ein Beispiel eines vorhergesagten Szenarios kann einen Speicherlastausgleichsmechanismus einschließen, der gelegentlich auf einem Heimnetzwerk läuft, um sicherzustellen, dass der freie Speicherplatz eines Computergerätes nicht ausgeht, während ein anderes Laufwerk genügend Platz hat.

[0065] Deshalb kann ein ergiebiges Schema Daten bereitstellen, die über das Beschreiben der Ressourcen selbst hinausgeht, wie z.B. Verfahren zum Planen, zu einer bestimmten Zeit oder zu regelmäßigen Intervallen, einer Priorisierungsmaschinerie, und Daten, die anzeigen, wie mit möglichen Auseinandersetzungen zwischen Echtzeit- und vorhergesagten Aufgaben oder mehreren solcher Aufgaben umzugehen ist. Das Schema berücksichtigt das Konzept der Kosten, z.B. wenn zwei Verbindungen zu dem Internet verfügbar sind, ist es wertvoll, zu erkennen, dass der Benutzer einen Preis für unbegrenzten Zugriff zahlt, aber einen Preis für eine Datenmenge, die über einen unterschiedlichen Mechanismus übertragen wird, und die bessere Option ausgewählt wird.

[0066] Eine Replikation/Teilen durch ein mitschreibendes (logging) System kann verwendet werden, um zu verstehen, wie Benutzer mit ihren Geräten interagieren, und um eine konsolidierte Ansicht zu erlangen, wie ein Benutzer arbeitet, als nur getrennte Ansichten, für die Geräte getrennt, zu haben. Diese Daten sind vielen Geräten einschlägig bekannt, und sie werden eine automatische Möglichkeit bereitstellen, solche Daten zu konsolidieren. Vorhersageoptimierungen können basierend auf einer überwachten Verwendung gespeichert und ausgetauscht werden, z.B., wenn das System weiß, dass ein Benutzer bestimmte Informationen auf einen tragbaren Computer jeden Freitag manuell herunterlädt, bevor er das Büro verlässt, dann kann das System das Herunterladen während einer Geräteleerlaufzeit starten, so dass die "Synchronisation" unverzüglich stattfindet. Die Optionen/Einstellungen/Präferenzen von Endbenutzern sind ebenso wertvolle Informationen für die geteilte Umgebung, z.B., was ein wünschenswertes oder selbst notwendiges Merkmal für manche Benutzer ist, ist nicht wünschenswert oder selbst störend für andere. Durch das Führen von diesen Daten, hat der Endbenutzer die Macht, sein Erlebnis zu steuern. Ebenso gibt es in einer üblichen Umgebung mindestens einen reichhaltigen Client, der das Protokoll/Schema verwendet, um eine Systemverwaltung durchzuführen, wobei „System“ sich auf die gesamte Reihe von Ressourcen zugehörig zu einer Identität bezieht. Algorithmen und

Verfahren können das Schema zur Vorhersage solcher Operationen verwenden.

[0067] Es gilt als verstanden, dass es viele wertvolle Szenarien gibt, die durch das Abstrahieren der Geräteressourcen und das Ansammeln dieser um eine Identität herum ermöglicht werden. Eine Anzahl von Beispielen wird bereitgestellt, jedoch sollte die Liste nicht als vollständig oder in irgendeiner Weise einschränkend betrachtet werden, sondern dient eher dazu, manche der vielen Nutzen und Vorteile der vorliegenden Erfindung darzustellen. Wie die oben genannten Beispiele kann ein Gerät mit einem kleinen Computerbildschirm einen nicht verwendeten Computermonitor, der mit einem Personalcomputer verbunden ist, für eine bessere Anzeige wirksam einsetzen, Ressourcen eines Proxy verwenden, um eine bessere Anzeige auf seinem eigenen Display (**Fig. 5**) zu ermöglichen, oder auf einem Drucker zu drucken, selbst ohne einen Druckertreiber zu besitzen, und durch das Verbinden zu einem Drucker über einen Personalcomputer, der den korrekten Treiber hat (**Fig. 4**).

[0068] Andere Beispiele können darin gesehen werden, dass ein Gerät mit sehr geringer Leistung, wie z.B. ein Mobiltelefon, komplexe Berechnungen durch die Verwendung des Speichers und der CPU eines Personalcomputers durchführen kann. Gespeicherte Daten können automatisch und unsichtbar von einem ersten Festplattenlaufwerk zu einem zweiten bewegt werden, wenn das erste Festplattenlaufwerk voll wird, einschließlich, z.B. von persönlichen digitalen Videorekordern, die Videoprogramme aufnehmen, und die alte Aufzeichnungen löschen müssen, um Platz für neuere Aufzeichnungen zu machen. Die vorliegende Erfindung schiebt stattdessen jene älteren Aufzeichnungen zu einem anderen Speicher, als diese zu löschen. Ein tragbarer Audio-Abspieler muss nicht imstande sein, irgendeinen Typ an physischen Datenträgern abzuspielen, sondern kann stattdessen Audiodaten empfangen und puffern, die von einem Personalcomputer zu ihm geladen werden (streamed) und die Audiodaten von dem Speicher abspielen.

[0069] Andere Beispiele schließen ein, dass es Benutzern erlaubt wird, ihre Dateien von irgendeinem Gerät aus einfach zu erreichen, ungeachtet dessen, wo sie gespeichert sind; z.B. während sie unterwegs sind und nur einen kleinen Personalcomputer haben, sind sie in der Lage, Dateien, die auf einem Heim-Personalcomputer oder Büro-Personalcomputer gespeichert sind, einfach zu öffnen (unabhängig von einem zentralisierten .NET-Datenspeicher). Ein Benutzer kann den gesamten freien Platz auf verschiedenen Geräten sehen, einschließlich dem in verschiedenen öffentlichen Sharepoints in einem Firmennetzwerk. Ein großflächiger Fernseher, der mit einem passenden Gerät verbunden ist, kann verwendet werden, um Fotografien anzusehen, die in einem Ordner eines Personalcomputers gespeichert sind, um eine Diashow abzuspielen, um Heimfilme anzusehen oder Filmvorschauen, die auf einem Personalcomputer gespeichert sind und so weiter.

[0070] **Fig. 7** stellt eine Demonstration einer Implementierung **700** bereit, die unter Verwendung von 4 miteinander verbundenen Computersystemen **702** bis **705** läuft. Die Implementierung **700** schließt eine Befehlshülle (command shell) **710** und ein Anwendungsprogramm **712** (oder ähnlichen Funktionscode) auf dem Computergerät **704**, eine Tastatur-Ressource **714** auf dem Computersystem **702**, einen Speicher (z.B. Disk) **716** auf dem Computersystem **703** und eine Anzeige **718** auf dem Computersystem **705** ein. Die Ressourcen in diesem Beispiel sind deshalb verteilt, so dass es eine Eingaberessource auf dem Computer **702** gibt, eine Speicherressource auf dem Computer **703** und eine Ausgaberrsource auf dem Computer **704**. Der Computer **704** agiert als eine koordinierende Autorität für die Implementierung **700**. Obwohl die Ressourcen auf verschiedenen Computersystemen sind wird auf die Ressourcen gemäß der vorliegenden Erfindung zugegriffen, um im Wesentlichen ein einzelnes funktionierendes System zu bilden.

[0071] Die Shell-Ressource **710** ruft die Anwendung **712** auf, und verbindet sich ebenso zu den Ressourcen der anderen Geräte durch die Anschlüsse **724**, **726** und **728**, die konzeptionell als ein Anschluss pro Ressource in **Fig. 7** dargestellt sind, obwohl ein separater Eingangs- und Ausgangsanschluss bereitgestellt werden könnte. Anschlüsse umfassen im Wesentlichen strukturierte Warteschleifen (structured queues), auf die/zu denen über ein Handle zu dem Anschluss geschrieben (output) oder gelesen (input) werden kann. Anschlüsse sind mobil, Daten gehen nicht verloren, und Sicherheit kann mit einem Anschluss assoziiert werden, so dass niemand, außer der beabsichtigten Einheit, die Daten sehen kann, die für ihn gedacht sind.

[0072] Die Shell **710**, die in der Implementierung **700** verwendet wird, basiert auf einer Programmiersprache, die als xSpresso bezeichnet wird, wie Sie in der vorläufigen US-Patentanmeldung mit der Seriennummer 60/379,864, die am 10. Mai 2002 angemeldet wurde, beschrieben ist. Im allgemeinen ist xSpresso auf eine Programmierungssprache und runtime (Echtzeitumgebung) gerichtet, die die Steuerung und den Datenfluss zwischen asynchronen Systemkomponenten vermittelt. Neben anderem drückt xSpresso die kooperativen Dimensionen von Programmen oder Diensten aus, einschließlich des Ordnen und der Zeitsteuerung, die programmatisch überprüft werden können. Die Sprache kann programmatisch das Übergeben eines Organisati-

onsschemas ausdrücken, das aus einer benutzeranpassbaren kennzeichenbasierenden (tag-based) Sprache wie z.B. XML, gebildet wurde.

[0073] In dem Beispiel der [Fig. 7](#) stellen die Pfeile, die mit 1 bis 4 und 6 bis 10 bezeichnet sind, ein Beispiel von den verschiedenen Datenübertragungen zwischen den Anschlüssen **724**, **726** und **728** und der Shell **710** dar, um die Textdaten einer Datei in dem Speicher **716** auf dem Display **718** anzuzeigen, z.B. ähnlich zu der gut bekannten „Schreib“-Funktion (Type-Funktion). In dem Beispiel der [Fig. 7](#) initiiert ein Benutzer der Tastatur-Ressource **714** die Funktion durch das Eintippen des Wortes „Schreibe“ auf der Tastatur **714** (der Pfeil mit der Bezeichnung 1 (1)). Die Shell **710** empfängt dies über den Anschluss **724** und kommuniziert die Daten zu dem Anschluss **728** zum Anzeigen auf dem Bildschirm **718** (Pfeil mit der Bezeichnung 2 (2)), zum Beispiel um zu wiederholen (echo), was der Benutzer eingegeben hat und/oder um zur Eingabe eines Dateinamens einer Datei aufzufordern, die den Text speichert, den der Benutzer angezeigt haben möchte. In [Fig. 7](#) stellt der Pfeil, bezeichnet als 3 (3) dar, dass der Benutzer den Dateinamen eingibt, wie z.B. entsprechend einer XML-Datei, was ebenso wiederholt werden kann (nicht extra bezeichnet).

[0074] Mit dem Dateinamen schreibt die Shell **710** anschließend Daten (dargestellt durch den Pfeil bezeichnet mit 4 (4)) an den Anschluss **726**, der der Speicher- (z.B. Disk-) Ressource **716** entspricht, was zu einer Leseanfrage an das Dateisystem des Speichers führt. Parallel dazu ruft die Shell **710** (der breite Pfeil, bezeichnet mit 5 (5)) das Anwendungsprogramm **712** auf, um die gelesenen Daten zu empfangen und auszugeben. In Reaktion darauf beginnt die Speicherressource **716** mit dem Ausgeben der Daten über dem Anschluss **726** an das Anwendungsprogramm **712**, welches sie wiederum an den Anschluss **728** des Display **718** ausgibt, wie es durch die Pfeile dargestellt ist, die mit 6 (6) und 7 (7) bezeichnet sind. Die Pfeile, bezeichnet mit 8 (8) und 9 (9) stellen den Rest dar, indem die Daten in einer ähnlichen Weise übertragen werden. Der Pfeil mit der Bezeichnung 10 (10) stellt den Status-(z.B. Erfolgs-) Code dar, der an die Shell **710** zurückgegeben wird.

[0075] In Übereinstimmung mit der vorliegenden Erfindung können Speicher, Eingabe und Ausgabe auf verschiedenen Systemen platziert werden, wobei die Implementierung nicht verändert wird. Eine Interoperabilität mit Altsystemen (legacy interoperability) kann über Klassen einer allgemeinen Sprachezeit (common language runtime-CLR) bereitgestellt werden, die allgemeine Altsystemkomponenten (z.B. die Tastatur, Konsole und Speicher) repräsentieren. Die Systemverbreitung wird in einer lokalen und transparenten, verteilten Implementierung sowie das explizite Behandeln von gleichzeitigem Zugriff (concurrency) und Nachrichtübergabe bereitgestellt.

[0076] Wie es von der vorangegangenen detaillierten Beschreibung gesehen werden kann, wird ein Verfahren und System für verteiltes Computing von Gerätere Ressourcen bereitgestellt, die auf dem Konzept der Identität basieren. Das Verfahren und System abstrahiert die Ressourcen eines Computers der einzelnen Geräte in einer Art und Weise, die viele Vorteile für einen Benutzer bereitstellt und die Erfahrung mit der Verwendung des Computers für den Benutzer allgemein verbessert. Das Verfahren und System ist schematabasiert, und sie sind erweiterbar und hoch flexibel.

[0077] Während die Erfindung empfänglich für verschiedene Modifizierungen und alternative Konstruktionen ist, sind bestimmten dargestellte Ausführungsformen davon in den Zeichnungen gezeigt und sind oberhalb im Detail beschrieben worden. Es sollte verstanden sein, dass es keine Absicht zum Einschränken der Erfindung auf bestimmte offenbarte Formen gibt, sondern im Gegenteil, es die Absicht ist, alle Modifikationen, alternativen Konstruktionen und Äquivalente, die in den Umfang der Erfindung fallen, abzudecken.

Patentansprüche

1. Verfahren eines mobilen Computergeräts zum Drucken in einer Computerumgebung, wobei das Verfahren erlaubt, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das Verfahren umfasst:

Entdecken, durch das mobile Computergerät (**202**, **400**), das mit mindestens einer Identität assoziiert ist, eines anderen Computergeräts (**212**, **404**), das mit der mindestens einen Identität assoziiert ist, und das mit einem Druckertreiber (**406**) für den besagten Drucker ausgestattet ist;

Erlangen, durch das mobile Computergerät, von Daten (**209B**), die das andere Computergerät betreffen, wobei die Daten Druckertreiberinformationen des anderen Computergeräts einschließen, die gemäß einem Schema, das das mobile Computergerät versteht, formatiert sind;

Interpretieren, durch das mobile Computergerät, der Daten, um auf den Druckertreiber des anderen Computergeräts zuzugreifen; und

Ausführen, durch das mobile Computergerät, eines Druckauftrags, einschließlich des Bereitstellens von Druck-

daten zu dem Druckertreiber über das andere Computergerät.

2. Verfahren nach Anspruch 1, das des Weiteren umfasst: Übermitteln (**600**) einer Anfrage, den Druckertreiber zu verwenden, von dem mobilen Computergerät zu dem anderen Computergerät.

3. Verfahren nach Anspruch 1 oder 2, das des Weiteren umfasst: Übermitteln von Identitätsinformationen von dem mobilen Computergerät zu dem anderen Computergerät.

4. Verfahren nach Anspruch 3, wobei die Identitätsinformationen Berechtigungsnachweisen (**207A**, **207B**) entsprechen.

5. Verfahren nach Anspruch 4, wobei die Berechtigungsnachweise durch einen Authentifizierungsdienst (**220**) bereitgestellt werden.

6. Computer-lesbarer Datenträger, der Computer-ausführbare Instruktionen aufweist, um das Verfahren eines der Ansprüche 1 bis 5 auszuführen.

7. System in einer Computerumgebung zum Erlauben, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das System umfasst:

Ein erstes Computergerät (**212**, **404**), das mit mindestens einer Identität assoziiert ist, wobei das erste Gerät einen Druckertreiber aufweist und eingerichtet ist zum Bereitstellen eines Zugriffs auf den Druckertreiber durch mindestens ein zweites Gerät (**202**, **400**), das mit der mindestens einen Identität assoziiert ist, und wobei das erste Computergerät dem zweiten Gerät Informationen (**209B**) über den Druckertreiber gemäß einem Ressourcen-Schema zugehörig zu der mindestens einen Identität bereitstellt;

das zweite Gerät (**202**, **400**), wobei das zweite Gerät das mobile Computergerät ist, das eingerichtet ist, einen Zugriff auf den Druckertreiber anzufordern; und

wobei das zweite Gerät einen Druckauftrag ausführt, der das Zugreifen auf den Druckertreiber auf dem ersten Gerät einschließt, um mindestens einen Teil des Druckauftrags auszuführen.

8. System nach Anspruch 7, das des Weiteren ein drittes Computergerät (**402**), das mit dem ersten Computergerät verbunden ist, umfasst, wobei das zweite Computergerät auf den Druckertreiber zugreift, um einen Arbeitsvorgang mit dem dritten Computergerät auszuführen, und wobei das dritte Computergerät den Drucker (**402**) umfasst.

Es folgen 7 Blatt Zeichnungen

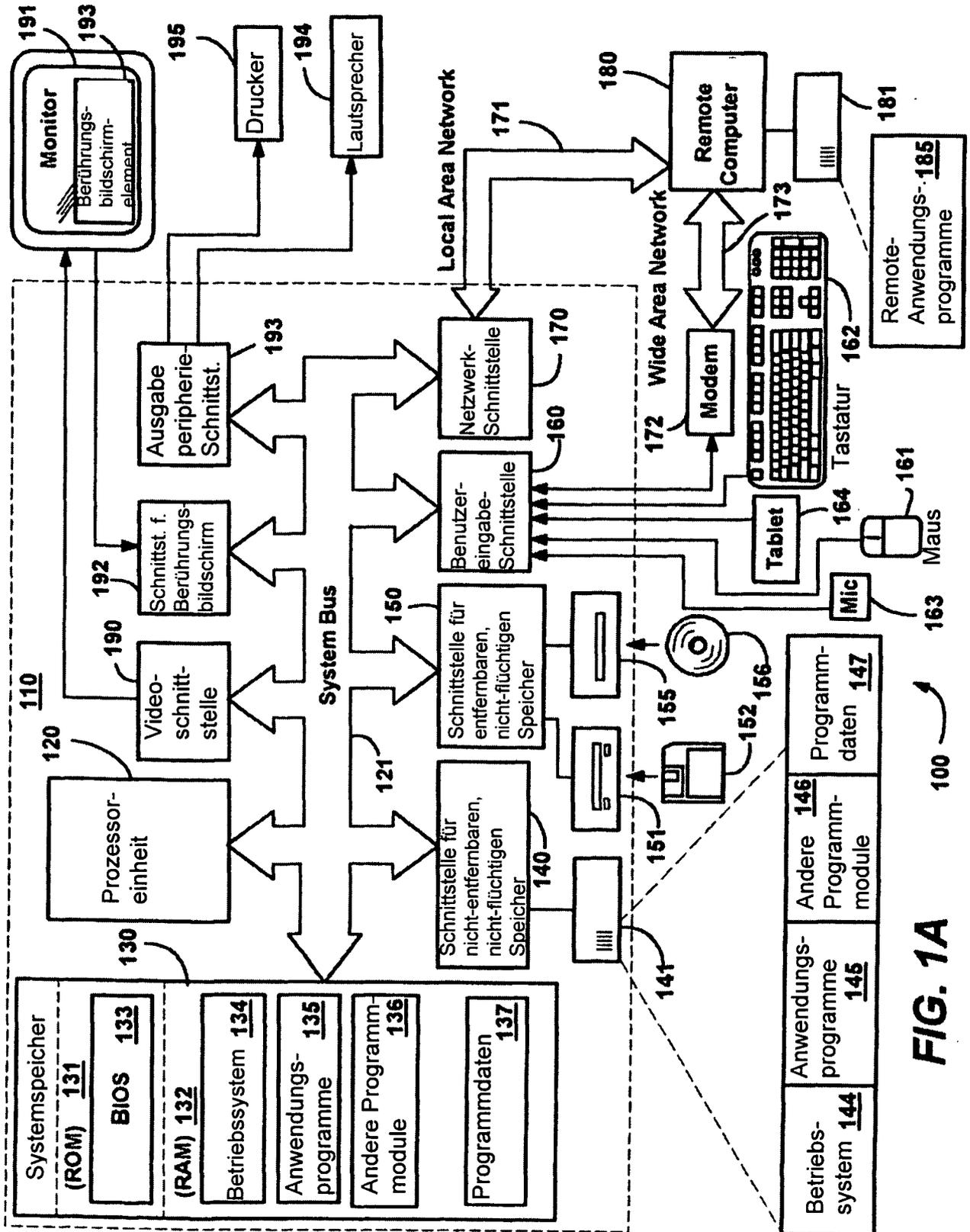


FIG. 1A

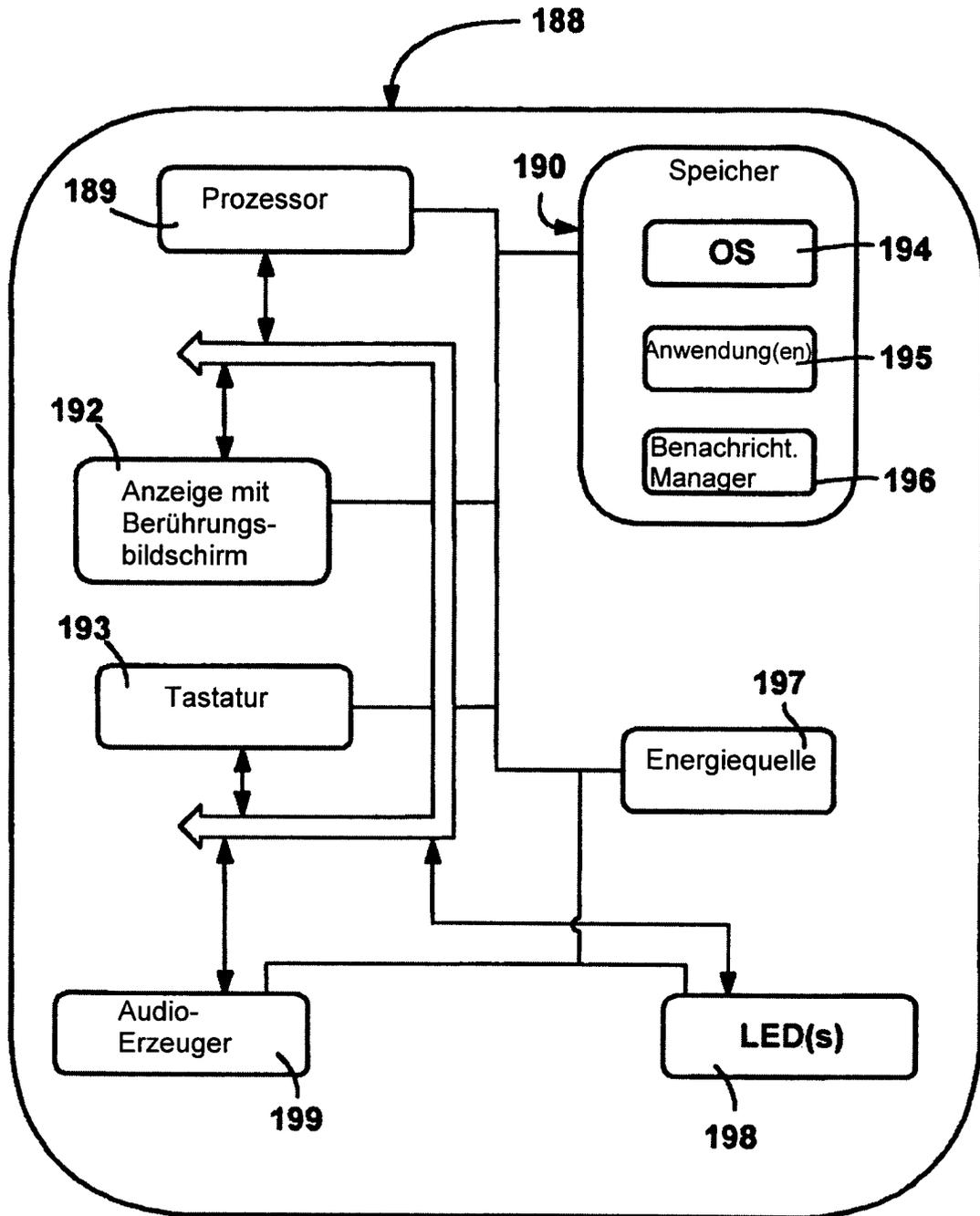


FIG. 1B

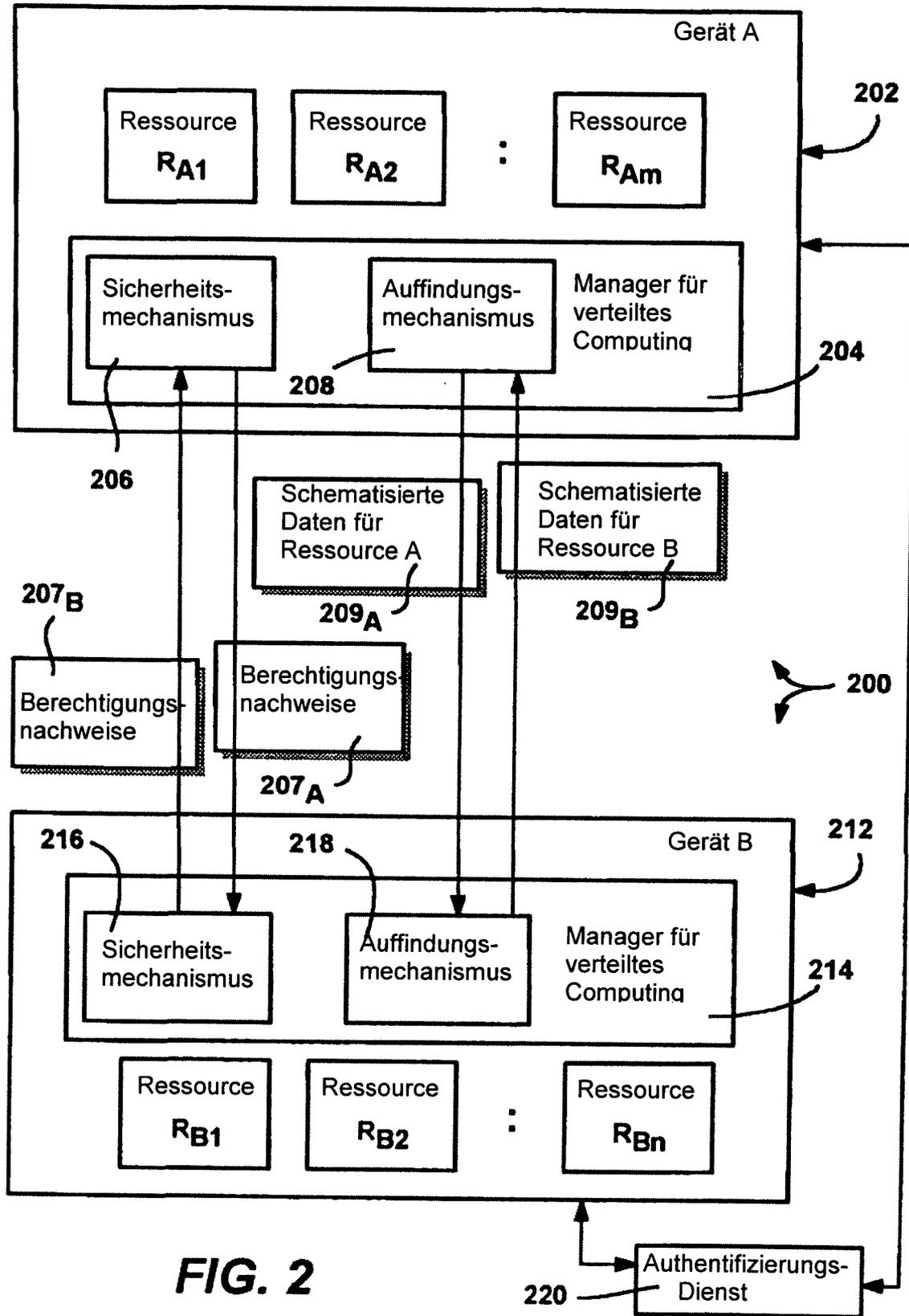


FIG. 2

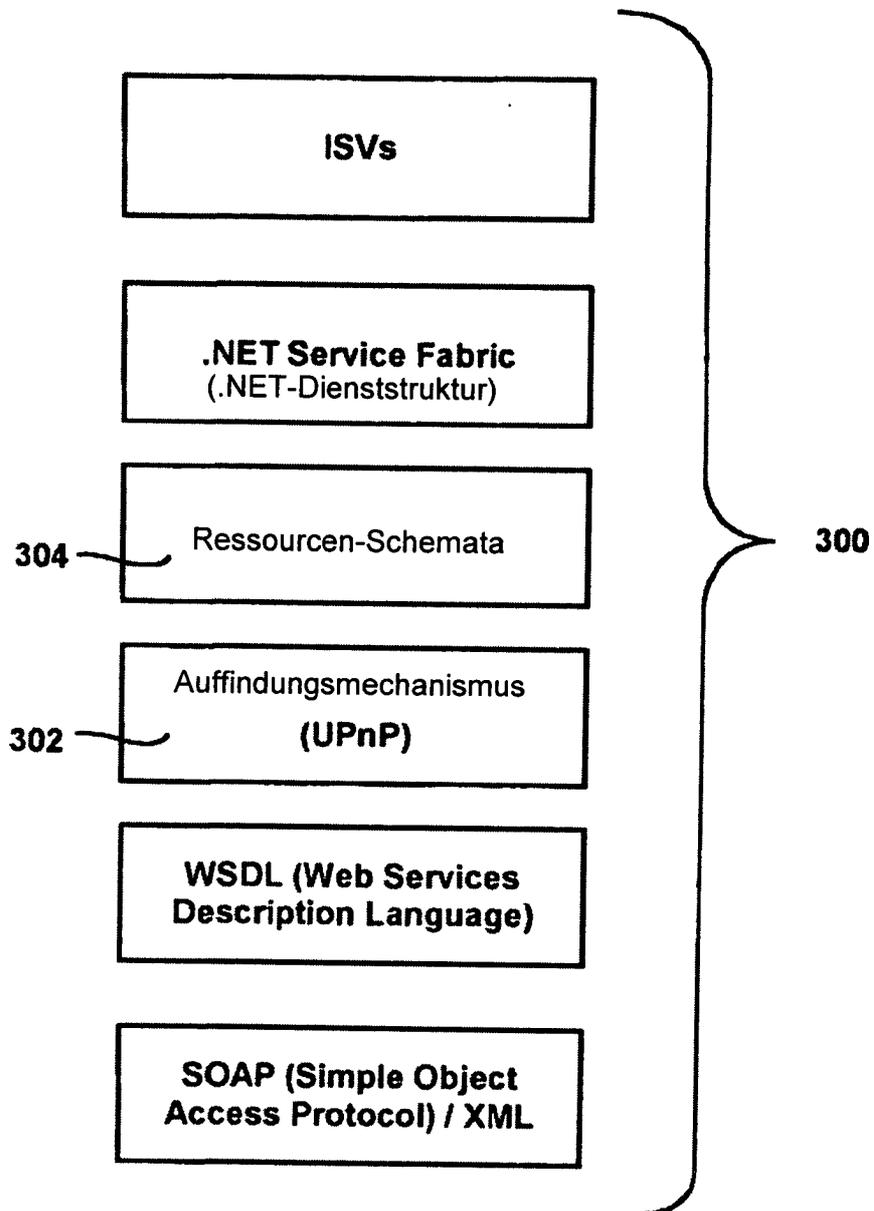


FIG. 3

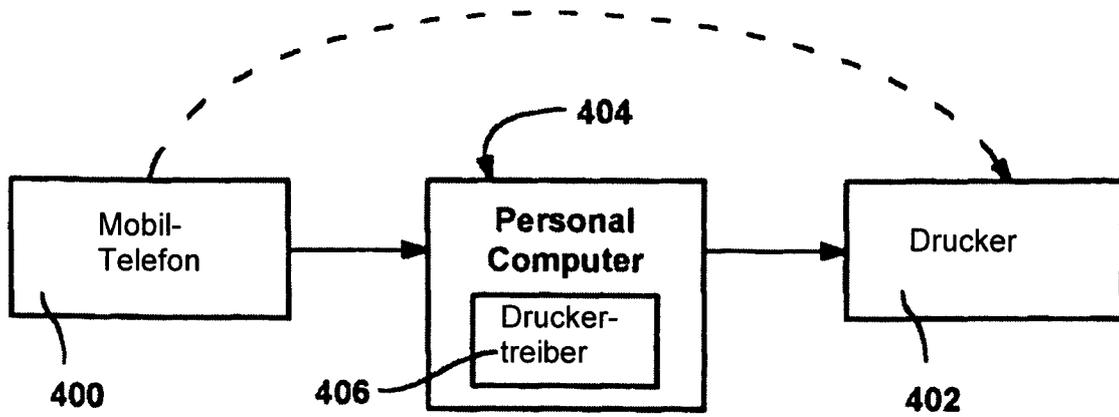


FIG. 4

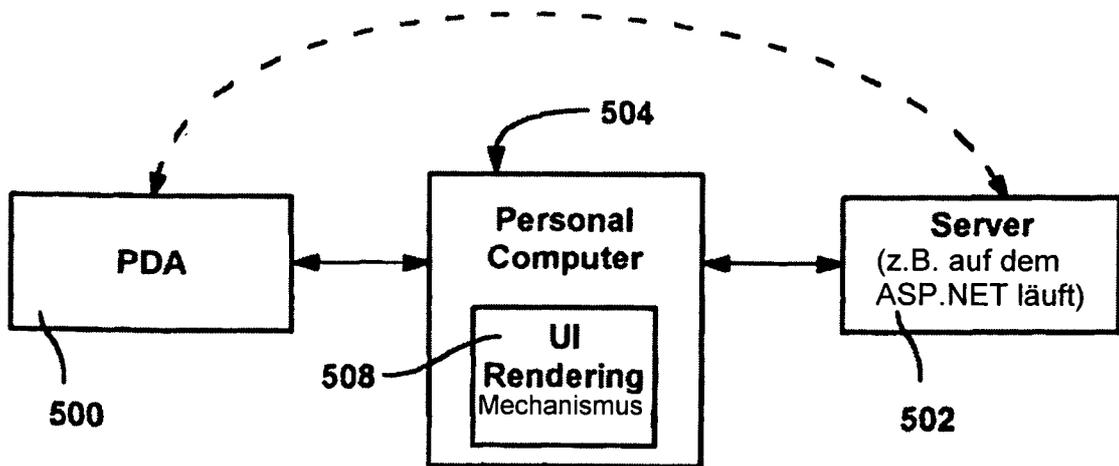
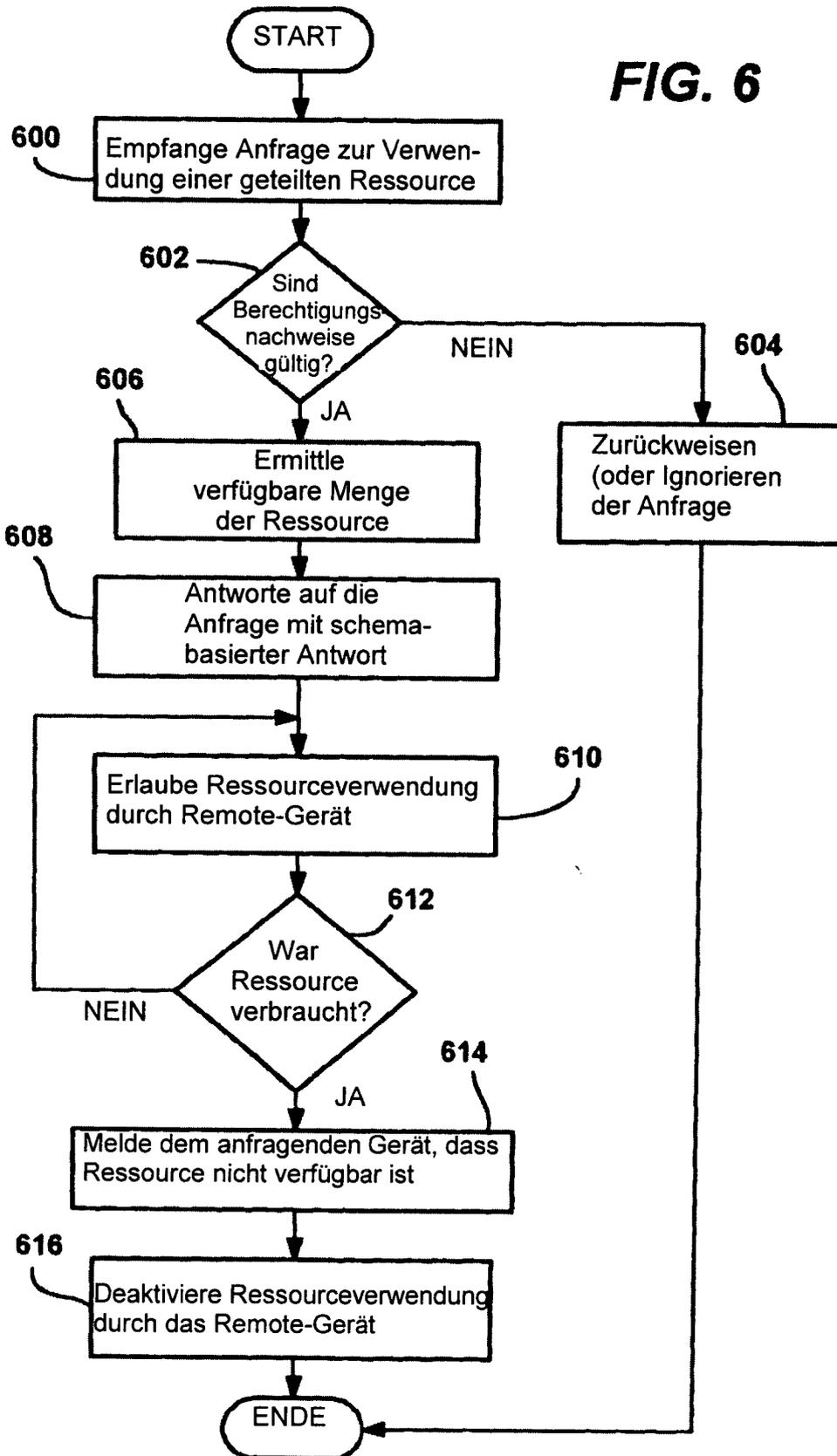


FIG. 5

FIG. 6



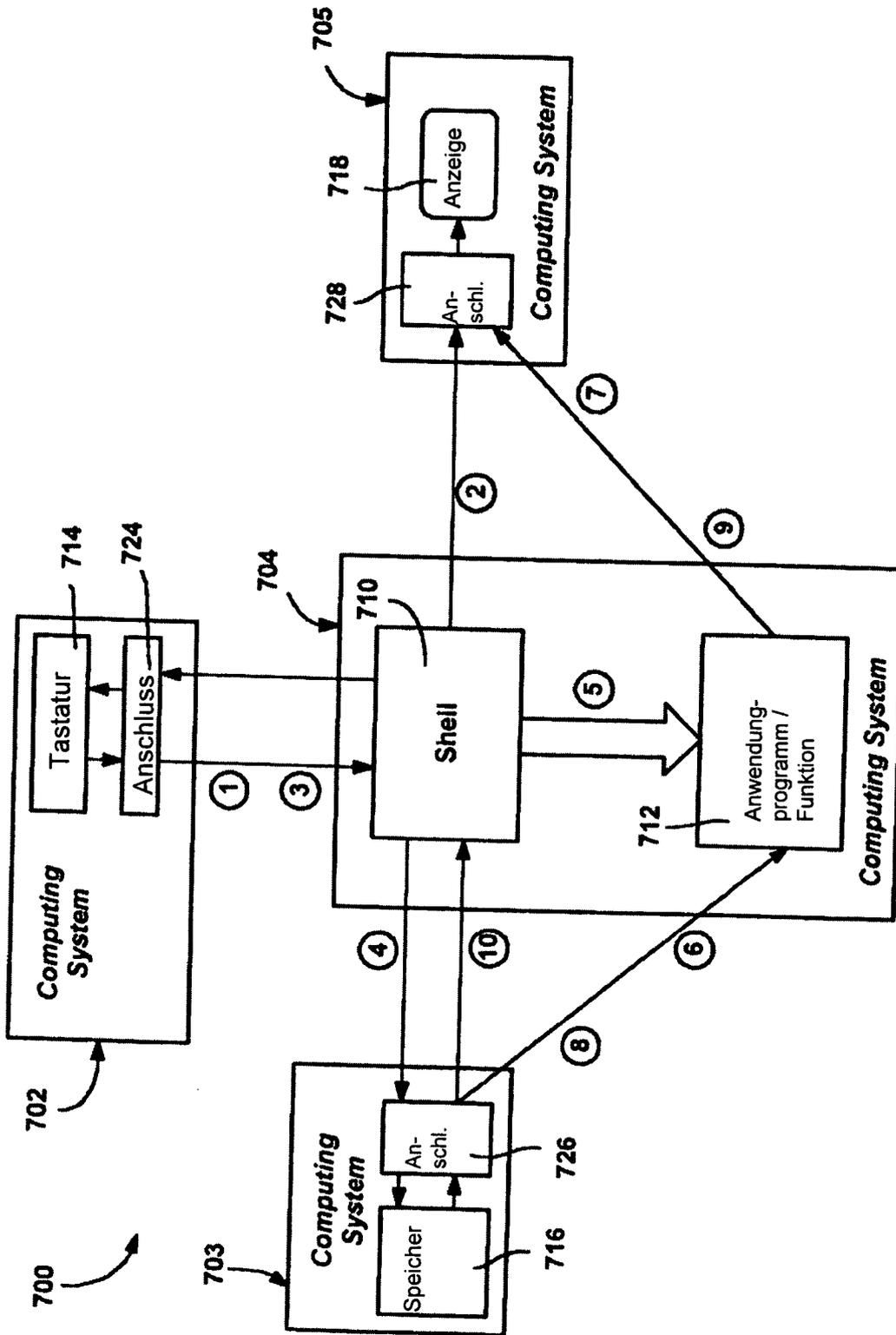


FIG. 7