



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 601 25 599 T2 2007.11.29**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 399 848 B1**

(51) Int Cl.⁸: **G06F 17/30 (2006.01)**

(21) Deutsches Aktenzeichen: **601 25 599.2**

(86) PCT-Aktenzeichen: **PCT/US01/25640**

(96) Europäisches Aktenzeichen: **01 964 079.6**

(87) PCT-Veröffentlichungs-Nr.: **WO 2002/017140**

(86) PCT-Anmeldetag: **15.08.2001**

(87) Veröffentlichungstag
der PCT-Anmeldung: **28.02.2002**

(97) Erstveröffentlichung durch das EPA: **24.03.2004**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **27.12.2006**

(47) Veröffentlichungstag im Patentblatt: **29.11.2007**

(30) Unionspriorität:
642632 18.08.2000 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Apple Inc., Cupertino, Calif., US

(72) Erfinder:
**SERLET, Bertrand, Palo Alto, CA 94301, US;
TEVANI, Jr., Avadis, Los Altos Hills, CA 94022,
US; WARNER, H., Clark, Mountain View, CA 94043,
US**

(74) Vertreter:
**Zenz, Helber, Hosbach & Partner GbR, 45128
Essen**

(54) Bezeichnung: **VERFAHREN UND SYSTEM ZUM TRANSPARENTEN ZUGREIFEN AUF FERNGESPEICHERTE DA-
TEIEN**

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

Gebiet der Erfindung

[0001] Diese Erfindung betrifft das Gebiet von vernetzten Dateisystemen und Personal-Computern. Insbesondere betrifft die Erfindung ein System und ein Verfahren, die es Benutzern von Personal-Computern mit einem beliebigen Programm auf dem lokalen Personal-Computer ermöglichen, über ein Netzwerk auf Dateien zuzugreifen, die auf einem entfernten Computersystem angeordnet sind.

Hintergrund

[0002] Üblicherweise wurden Computer bei der Arbeit verwendet und konnten in einem Unternehmen über ein lokales Netzwerk (LAN) verbunden sein. Das LAN des Unternehmens konnte mit anderen Büros oder Partnern des Unternehmens über ein Fernnetzwerk (WAN) verbunden sein, oder Computer des Unternehmens konnten direkt mit einem WAN verbunden sein. Solche Verbindungen ermöglichten es Unternehmen, Daten einfach auszutauschen, indem über private Netzwerke Daten auf Remote-Computern gespeichert und von diesen abgerufen wurden. Ferner wird ein über ein Netzwerk verfügbarer entfernter Plattenspeicher verwendet, um Daten von lokalen Computern zu sichern, wodurch lokaler Plattenspeicher freigegeben wird.

[0003] Das Internet ist ein öffentlich zugängliches globales Fernnetzwerk. Das Internet und Personal-Computer sind in der modernen Gesellschaft omnipräsent geworden. Menschen greifen über ihre Personal-Computer aus beliebigen Gründen regelmäßig auf Web-Seiten zu. Obgleich das Internet in verschiedenen Formen für viele Jahre existiert hat, wurde das Internet erst durch die Einführung des World Wide Web als ein Massenkommunikationsmittel populär. Das World Wide Web ist aus der Sicht des Benutzers, ein einfacher Weg, einen Remote-Computer zu identifizieren, mit dem Remote-Computer eine Verbindung herzustellen und auf dem Remote-Computer gespeicherte Informationen zu betrachten. Jedoch haben bis vor kurzem Benutzer von Personal-Computern das Internet und Web-Seiten nicht für eine persönliche Datenspeicherung und -abrufung verwendet.

[0004] Bei der Benutzung des Internets sind es die verschiedenen Kommunikationsprotokolle, die das Internet, versteckt vor dem Benutzer, funktionieren lassen. Verschiedene Komitees und Adhoc-Gruppen, bekannt als Workgroups, koordinieren und steuern das Internet. Die Internet Engineering Task Force (IETF) ist der Protokollkonstruier- und -entwicklungsarm des Internets. Workgroups unter dem IETF be-

stimmen die Regeln und Protokolle für die grundlegende Funktionsweise des Internets und veröffentlichen diese als üblicherweise als RFCs bezeichnete Diskussionsschriften. Jede Workgroup macht ihre RFCs über das Internet auf verschiedenen Web-Seiten verfügbar. Eine zentrale Stelle für den Erhalt der unten genannten RFCs ist die IETF-Web-Seite, www.ietf.org (von der IETF wird keine Mail-Adresse oder geographischer Ort angegeben). Darüber hinaus wurde eine das World Wide Web Konsortium (W3C) genannte Organisation gegründet, um die Arbeit der IETF fortzusetzen, obgleich die IETF und das W3C nebeneinander existieren (siehe www.w3c.org; mit dem W3C kann Kontakt aufgenommen werden über Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, Massachusetts 02139).

[0005] Web-Seiten werden durch eine Textbeschreibung oder einen Namen, der als Uniform Resource Locator (URL) bezeichnet wird, spezifiziert, die bzw. der nun von dem Begriff Uniform Resource Identifier (URI) umfaßt ist (siehe Uniform Resource Identifieres (URI): Generic Syntax, RFC 2396, August 1998, Draft Standard). Informationen werden über das World Wide Web mit dem Übertragungssteuerungsprotokoll/Internetprotokoll, welches üblicherweise als TCP/IP bezeichnet wird, kommuniziert (für weitere Informationen siehe A TCP/IP Tutorial, RFC 1180, Januar 1991). Ein auf der Grundlage von TCP/IP laufendes Anwendungs-Level-Protokoll, welches einen Zugriff auf Remote-Computer über das Internet durch Spezifizieren einer URI ermöglicht, ist das Hypertext Transfer Protocol (HTTP, siehe HTTP/1.1, RFC 2616, Juni 1999, Draft Standard). Es ist die weit verbreitete Akzeptanz und Verwendung von HTTP, die das World Wide Web, wie wir es kennen, möglich gemacht haben. Erweiterungen zu HTTP für ein Distributed Authoring and Versioning über das World Wide Web, bezeichnet als WebDAV, beginnen verbreitet verwendet zu werden (siehe WebDAV, RFC 2518, Februar 1999, Proposed Standard). Die WebDAV-Erweiterungen zu HTTP erfordern, dass eine Kommunikation gemäß dem WebDAV-Protokoll entsprechend der Extensible Markup Language (XML) formatiert ist (siehe XML 1.0, verfügbar über www.w3.org/TR/REC-xml). WebDAV ermöglicht es Menschen, die WebDAV unterstützende Programme verwenden, auf Dateien zuzugreifen, die auf einem WebDAV-fähigen HTTP-Server gespeichert sind und üblicherweise als Web-Seiten bezeichnet werden. WebDAV stellt ein Lesen, Schreiben (d.h. Erzeugen), teilweises Lesen, teilweises Schreiben, Sperren, Property-Änderungen und andere Zugriffe auf entfernt gespeicherte Dateien bereit.

[0006] Verschiedene Unternehmen haben begonnen, den Internet-Nutzern freien Speicherplatz auf Remote-Servern anzubieten. Diese Remote-Server sind Web-Seiten, die mit einem WebDAV-fähigen

HTTP laufen, und welche zusätzliche Software zum Bereitstellen einer Web-basierten Schnittstelle für den Plattenspeicher, der auf dem Remote-Server verfügbar gemacht wird, aufweisen. Unternehmen, wie beispielsweise Xythos Software, Inc. aus Redwood City, Kalifornien, die eine Web-Seite namens Sharemation (www.sharemation.com) bereitstellt, My Docs Online! Inc. in Naples, Florida (www.MyDocsOnline.com) und Driveway Corporation aus San Francisco, Kalifornien (www.driveway.com), ermöglichen es Benutzern von Personal-Computern, ein Verzeichnis auf der Web-Seite des Unternehmens zu erzeugen und Dateien für eine sichere persönliche Verwendung zu speichern. Diese Unternehmen stellen jedem Benutzer eines Personal-Computers einen Zugriff auf eine Remote-Speichereinrichtung bereit, und stellen eine Einrichtung bereit, die es Benutzern von Personal-Computern ermöglicht, über den Remote-Computer Dateien zu schreiben und abzurufen, wodurch die gleichen Vorzüge bereitgestellt werden, die historisch gesehen lediglich für Unternehmen oder Betriebe über private Netzwerke zugänglich waren.

[0007] Jedoch stellen diese Unternehmen für öffentlich zugängliche Speicherung keinen nahtlosen Weg für einen Benutzer eines Personal-Computers bereit, mit sämtlichen Anwendungsprogrammen auf dessen Personal-Computern auf entfernt gespeicherte Daten zuzugreifen. Die Unternehmen ermöglichen es einem Benutzer lediglich, Dateien mittels "drag and drop" oder anderweitig zu speichern, oder von der Web-Seite abzurufen, wenn der Benutzer außerhalb des Anwendungsprogrammes ist. Einige dieser Web-Seiten für öffentlich zugängliche entfernte Speicherung ermöglichen über Erweiterungen eines Anwendungsprogrammes einen Zugriff mit einem speziellen Anwendungsprogramm, oder erfordern, dass die Anwendung zusammen mit einem Internet-Web-Browser (wie beispielsweise Internet Explorer 5.0 von Microsoft Corporation aus Redmond, Washington) betrieben werden. Obwohl diese Unternehmen für Internet-Nutzer eine Fernspeicherung bereitstellen, wird ein einfacher Zugriff für sämtliche Anwendungsprogramme auf einem Personal-Computer nicht bereitgestellt.

[0008] Weitere Beispiele bekannter Techniken sind in WO 97/46956 und EP 1 003 110 beschrieben.

KURZE ZUSAMMENFASSUNG DER ERFINDUNG

[0009] Gemäß der vorliegenden Erfindung wird bereitgestellt ein System mit einer Mehrzahl von mit dem Internet gekoppelten Web-Servern, auf denen das Distributed Authoring and Versioning-fähige Hypertext Transfer Protocol (HTTP), WebDAV, läuft, und mit einer Mehrzahl von mit dem Internet gekoppelten Personal-Computern, die ein Dateisystem-Plugin aufweisen, das mit einem Dateisystem ei-

nes Betriebssystems kommunikativ gekoppelt ist, das von einer Anwendung empfangene Dateisystemanfragen, die auf einem der Mehrzahl von Web-Servern gespeicherte Dateisysteme betreffen, an ein Netzwerkzugriff-Anwendungsprogramm auf dem Personal-Computer weiterleitet, das die Dateisystemanfragen jeweils als eine WebDAV-Anfrage an einen geeigneten der Mehrzahl von Web-Servern sendet, wobei die Dateisystemanfragen einen Dateinamen einer Datei umfassen, auf die zugegriffen wird, und wobei, wenn der Dateiname ein oder mehrere Zeichen enthält, die von dem besagten einen der Mehrzahl von Web-Servern nicht erkannt werden, das Netzwerkzugriff-Anwendungsprogramm die nicht erkannten Zeichen des Dateinamens in eine oder mehrere, von dem besagten einen der Mehrzahl von Web-Servern erkennbare Escape-Sequenzen umwandelt, bevor die umgewandelten Dateisystemanfragen an den besagten einen der Mehrzahl von Web-Servern übermittelt werden, wobei die eine oder die mehreren Escape-Sequenzen das eine oder die mehreren nicht erkannten Zeichen darstellen, wobei die Dateisystemanfragen von dem Dateisystem-Plugin ohne Kenntnis von der Anwendung weitergeleitet werden, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind.

[0010] Ferner wird gemäß der vorliegenden Erfindung bereitgestellt ein Verfahren, aufweisend: Empfangen einer ein entferntes Dateisystem betreffenden Dateisystemanfrage von einem Dateisystem-Plugin, das kommunikativ mit einem Dateisystem eines Betriebssystems gekoppelt ist, wobei das Dateisystem-Plugin die Dateisystemanfrage von einer Anwendung als eine reguläre Dateisystemanfrage in einem anderen Format als dem Distributed Authoring And Versioning-Protokoll, WebDAV, empfängt, Erzeugen einer Anfrage im WebDAV-Format in Reaktion auf die Dateisystemanfrage, Umwandeln eines oder mehrerer Zeichen der Dateisystemanfrage, die von einem WebDAV-fähigen HTTP-Server nicht erkannt werden, in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind, Weiterleiten der Anfrage im WebDAV-Format an den entsprechenden WebDAV-fähigen HTTP-Server ohne Kenntnis der Anwendung, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind, Empfangen einer Antwort von dem WebDAV-fähigen HTTP-Server, und Übermitteln von in der Antwort enthaltenen Informationen an das Dateisystem-Plugin, wobei die Informationen als eine reguläre Systemantwort präsentiert werden.

[0011] Ferner wird gemäß der vorliegenden Erfindung ein Maschinen-lesbares Medium bereitgestellt, mit darauf gespeicherten Befehlen, welche bei Ausführung durch einen Prozessor die Maschine veranlassen, Operationen auszuführen:

Empfangen einer ein entferntes Dateisystem betreffenden Dateisystemanfrage von einem Dateisystem-Plugin, das kommunikativ mit einem Dateisystem eines Betriebssystems gekoppelt ist, wobei das Dateisystem-Plugin die Dateisystemanfrage von einer Anwendung als eine reguläre Dateisystemanfrage in einem anderen Format als dem Distributed Authoring And Versioning-Protokoll, WebDAV, empfängt,

Erzeugen einer Anfrage im WebDAV-Format in Reaktion auf die Dateisystemanfrage,

Umwandeln eines oder mehrerer Zeichen der Dateisystemanfrage, die von einem WebDAV-fähigen HTTP-Server nicht erkannt werden, in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind,

Weiterleiten der Anfrage im WebDAV-Format an den entsprechenden WebDAV-fähigen HTTP-Server ohne Kenntnis der Anwendung, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind,

Empfangen einer Antwort von dem WebDAV-fähigen HTTP-Server, und

Übermitteln von in der Antwort enthaltenen Informationen an das Dateisystem-Plugin, wobei die Informationen als eine reguläre Systemantwort präsentiert werden.

[0012] Ebenso wird gemäß der vorliegenden Erfindung ein Computersystem bereitgestellt, aufweisend: zumindest ein Anwendungsprogramm, ein eine Dateisystemschnittstelle bereitstellendes Betriebssystem,

ein mit der Dateisystemschnittstelle gekoppeltes Dateisystem-Plugin zum Empfangen einer Anfrage für eine entfernt gespeicherte Datei von der Dateisystemschnittstelle des Betriebssystems, die von dem Anwendungsprogramm initiiert wird, und zum Weiterleiten der Anfrage für die entfernt gespeicherte Datei, ein Netzwerkzugriff-Anwendungsprogramm zum Empfangen der Anfrage für die entfernt gespeicherte Datei von dem Dateisystem-Plugin, zum Übersetzen von in der Anfrage spezifizierten Dateinameinformationen von einer lokalen Dateisystemsyntax in eine Syntax eines entfernten Servers, und zum Verpacken der Anfrage gemäß eines Distributed Authoring and Versioning-Protokolls, WebDAV, zum Zugreifen auf eine entfernte Datei eines benutzerspezifischen entfernten WebDAV-fähigen HTTP-Servers über das Internet,

wobei die Anfrage einen Dateinamen einer Datei umfasst, auf die zugegriffen wird, und wobei, wenn der Dateiname ein oder mehrere Zeichen umfasst, die

von dem WebDAV-fähigen HTTP-Server nicht erkannt werden, das Netzwerkzugriff-Anwendungsprogramm die nicht erkannten Zeichen des Dateinamens in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind, umwandelt vor einer Übermittlung der konvertierten Dateisystemanfragen an den WebDAV-fähigen HTTP-Server, wobei die eine oder die mehreren Escape-Sequenzen das eine oder die mehreren nicht erkannten Zeichen darstellen,

wobei die Dateisystemanfragen von dem Dateisystem-Plugin ohne Kenntnis von der Anwendung weitergeleitet werden, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0013] [Fig. 1](#) stellt ein Computersystem und eine Netzwerkumgebung dar, bei welchem bzw. welcher ein Ausführungsbeispiel des Verfahrens und des Systems des Seamless-Dateisystems ausgeführt ist.

[0014] [Fig. 2](#) zeigt die Software-Architektur eines Ausführungsbeispiels des Seamless-Dateisystems.

[0015] [Fig. 3](#) zeigt den Ablauf von von einem Personal-Computer-Benutzer vorgenommenen Handlungen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems.

[0016] [Fig. 4A](#) zeigt den Ablauf von von einem Seamless-Dateisystem durchgeführten Vorgängen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems.

[0017] [Fig. 4B](#) zeigt den Ablauf von von einem Seamless-Dateisystem durchgeführten Vorgängen gemäß einem Ausführungsbeispiel des ein Caching umfassenden Seamless-Dateisystemverfahrens und -systems.

[0018] [Fig. 5](#) zeigt den Ablauf von von einer Plugin-Erweiterung eines Seamless-Dateisystems durchgeführten Vorgängen bei einem Betriebssystem gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems.

[0019] [Fig. 6](#) zeigt den Ablauf von von einem Seamless-Dateisystem-Netzwerkzugriff-Anwendungsprogramm durchgeführten Vorgängen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems.

DETAILLIERTE BESCHREIBUNG

[0020] Diese Erfindung stellt ein System und ein Verfahren bereit, mit welchem Benutzer mit Programmen auf einem Computer nahtlos auf entfernt gespei-

cherte Dateien auf anderen Computern zugreifen können, auf denen ein bekanntes Dateizugriffprotokoll läuft. Als solches wird das Verfahren und das System als das Seamless-Dateisystem oder SFS bezeichnet. Das Ziel des SFS ist es, von sämtlichen Personal-Computer-Dateisystem-Clients einen nahtlosen Zugriff auf entfernt gespeicherte Dokumente zu ermöglichen. Es ermöglicht es, dass sämtliche auf einem Personal-Computer laufende Programme auf entfernte Dateien zugreifen können, und zwar so einfach und in der gleichen Weise, wie sie auch auf Dateien in dem Personal-Computer-Dateisystem zugreifen können, ohne dass irgendwelche Änderungen an dem Verfahren des Programms zum Kommunizieren mit dem bestehenden Dateisystem des Computers erforderlich sind. Bei einem Ausführungsbeispiel ermöglicht es SFS Benutzern von Personal-Computern nahtlos unter Verwendung der Eigenschaften von WebDAV-fähigen HTTP über das Internet auf Dateien zuzugreifen.

A. Die Umgebung, in welcher ein Ausführungsbeispiel des SFS läuft

[0021] **Fig. 1** zeigt ein Computersystem und eine Netzwerkumgebung, bei welchem bzw. welcher ein Ausführungsbeispiel des Verfahrens und des Systems des Seamless-Dateisystems ausgeführt ist. SFS läuft auf einem Personal-Computer **10**. Der Personal-Computer **10** kann eine beliebige Recheneinrichtung sein, die Software-Programme ausführen kann und auf das Internet zugreifen kann, einschließlich, aber nicht begrenzt auf Mobiltelefone, persönliche digitale Assistenten, Desktop-Personal-Computer, tragbare Computer, Computer-Workstations, etc. Der Personal-Computer **10** weist einen Prozessor **12** zum Ausführen von Software-Programmen auf. Der Prozessor **12** kann ein beliebiger Computer-Prozessor sein. Beim Ausführen von Programmen verwendet der Prozessor einen Speicher **14**, und der Speicher **14** kann eine beliebige Form von flüchtigem Speicher mit wahlfreiem Zugriff (RAM) sein. Informationen werden von einer lokalen Speichereinrichtung **16**, die über eine Steuereinrichtung **18** mit dem Personal-Computer gekoppelt ist, gelesen und auf diese geschrieben. Die Speichereinrichtung **16** kann eine Einrichtung sein, die ein maschinen-lesbares Medium, wie beispielsweise ein beschreibbares Plattenlaufwerk einschließlich beispielsweise einer Festplatte oder einer lesbaren und schreibbaren Kompakt-Disk (CDRW), und andere Einrichtungen, wie beispielsweise Bandler/-schreiber und Speicherkartenleser/-schreiber umfaßt. Der Prozessor kann Befehle an die Anzeigesteuereinrichtung **20** zum Anzeigen von Bildern auf einer Anzeigeeinrichtung **22** kommunizieren. Die Anzeigesteuereinrichtung **20** kann eine beliebige, dem Fachmann bekannte Anzeigesteuereinrichtung sein, und die Anzeigeeinrichtung **22** kann ein beliebiger, dem Fachmann bekannter Anzeigemonitor sein, einschließlich aber nicht be-

schränkt auf einen Kathodenstrahlröhre(CRT)-Anzeigemonitor oder Dünnschichttransistor(TFT)-Anzeigemonitor. Ein Benutzer greift auf den Personal-Computer **10** über eine beliebige, dem Fachmann bekannte Computer-Eingabeeinrichtung, wie beispielsweise eine Tastatur **24** und eine Maus **26**, zu, welche über eine Eingabe/Ausgabe(I/O)-Steuereinrichtung **28** mit dem Computer gekoppelt ist.

[0022] Um auf nicht auf einer lokalen Speichereinrichtung **16** gespeicherte Informationen zuzugreifen, weist der Computer **10** eine Netzwerkzugriffseinheit **30** auf, welche es dem Personal-Computer ermöglicht, über ein Netzwerk, wie beispielsweise das Internet **32**, mit einem Remote-Computer **34** zu kommunizieren und auf auf einer Remote-Speichereinrichtung **36** gespeicherte Informationen zuzugreifen. Die Netzwerkzugriffseinheit **30** kann ein Modem oder eine beliebige andere Einrichtung zum Verbinden mit einem Netzwerk, und zwar entweder direkt oder über eine Telefonwählverbindung, sein. Der Remote-Computer **34** kann eine beliebige Art von dem Fachmann bekanntem Computer sein, einschließlich aber nicht beschränkt auf Personal-Computer und Server. Die Remote-Speichereinrichtung **36** kann ein beliebiges, dem Fachmann bekanntes lesbares Speichermedium sein, wie beispielsweise Festplattenspeicher oder eine Matrix von Festplattenspeichern. Obwohl lediglich ein Personal-Computer und ein Remote-Computer gezeigt sind, können mehrere Personal-Computer und mehrere Remote-Computer mit dem Internet **32** verbunden sein. Der Prozessor **12**, der Speicher **14**, die lokale Speichereinrichtung **16**, die Anzeigesteuereinrichtung **20**, die I/O-Steuereinrichtung **28** und die Netzwerkzugriffseinheit **30** sind über einen Bus **11** miteinander gekoppelt und kommunizieren über diesen miteinander. Der Bus **11** kann ein beliebiger, dem Fachmann bekannter Bus sein. Obwohl lediglich ein Bus gezeigt ist, können in dem Personal-Computer **10** mehrere Busse verwendet sein. Ferner können andere, dem Fachmann bekannte (nicht gezeigte) Komponenten und Steuereinrichtungen oder mehrere der gezeigten Komponenten und Steuereinrichtungen in dem Personal-Computer **10** enthalten sein.

B. Die Software-Architektur eines Ausführungsbeispiels eines Seamless-Dateisystems

[0023] **Fig. 2** zeigt die Software-Architektur eines Ausführungsbeispiels des Seamless-Dateisystems. Bei einem Ausführungsbeispiel umfasst der Personal-Computer **10**, auf dem das SFS läuft, eine als SFS-Plugin **50** bezeichnete Erweiterung des Betriebssystems und ein als das SFS-Netzwerkzugriff-Anwendungsprogramm (oder das SFS-Netzwerkzugriffprogramm) **52** bezeichnetes Anwendungs-Level-Programm. Über eine Dateisystemschnittstelle **56** empfängt das SFS-Plugin **50** Anfragen von einem Anwendungsprogramm **54**, welche das Anwendungsprogramm **54** an die Dateisystem-

schnittstelle **56** des Personal-Computers richtet, welche üblicherweise als eine Anwendungsprogramm-schnittstelle (API) bezeichnet wird. Das SFS-Plugin **50** stellt ein Dateisystemaufruf-Parsing bereit und kommuniziert, zusätzlich zur Dateisystemschnittstelle **56** des Betriebssystems **48**, mit dem SFS-Netzwerkzugriffprogramm **52**. Bei einem Ausführungsbeispiel findet die Kommunikation zwischen dem SFS-Plugin **50** und dem SFS-Netzwerkzugriffprogramm **52** über lokale Sockets statt.

[0024] Das SFS-Netzwerkzugriff-Anwendungsprogramm **52** ist verantwortlich für ein Erstellen von Netzwerkverbindungen und eine Protokollkommunikation mit Remote-Computern, wie beispielsweise dem Remote-Computer **34**, auf dem ein bekanntes Anwendungs-Level-Internetprotokoll läuft. Bei einem Ausführungsbeispiel ist das bekannte Protokoll WebDAV-fähiges HTTP. Bei einem solchen Ausführungsbeispiel ist das SFS-Netzwerkzugriffprogramm **52** ein WebDAV-Client, der mit der TCP/IP-Transportschicht **58** des Betriebssystems **48** über ein Netzwerk, wie beispielsweise das Internet **32**, mit WebDAV-fähigen HTTP-Servern, wie beispielsweise dem Remote-Computer **34**, kommuniziert. Bei solch einem Ausführungsbeispiel umfasst der Remote-Computer **34** eine WebDAV-erweiterte **60** HTTP-Anwendungs-Level-Serversoftware **62**, welche über das Internet **32** eine World Wide Web-Kommunikation und einen Dateizugriff bereitstellt. Bei solch einem Ausführungsbeispiel kommuniziert der Remote-Computer **34** über eine TCP/IP-Transportschicht **64**, welche Teil eines Betriebssystems **66** ist.

C. Verwendung eines Seamless-Dateisystems

[0025] **Fig. 3** zeigt den Ablauf von von einem Personal-Computer-Benutzer vorgenommenen Handlungen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems. Nachdem ein Benutzer einen Personal-Computer startet, wie es in Block **300** gezeigt ist, stellt der Benutzer eine Verbindung mit dem Internet her, und zwar gemäß einem dem Fachmann bekannten Verfahren, wie es in Block **302** gezeigt ist. Wenn der Benutzer eine "always on"-Internetverbindung hat, kann dieser Schritt übersprungen werden. Der Benutzer kann dann ein SFS-Startprogramm ausführen, wie es in Block **304** gezeigt ist, um den Remote-Server zu mounten, so dass er erscheint, als wäre er ein lokales Laufwerk. Der Benutzer kann das SFS-Startprogramm gemäß einem beliebigen, dem Fachmann bekannten Verfahren ausführen, einschließlich, aber nicht begrenzt auf Pull-Down-Menüs oder Desktop-Icons. Das SFS-Startprogramm verlangt, dass der Benutzer ein zu einer Liste von verfügbaren Laufwerken hinzuzufügendes Remote-Dateisystem spezifiziert, wie es in Block **306** gezeigt ist. Das SFS-Startprogramm kann bei einem Ausführungsbeispiel von dem Benutzer eine Texteingabe des Namens der Web-Seite, der

URI, auf welcher die Remote-Dateien des Benutzers liegen, erhalten. Bei einem anderen Ausführungsbeispiel verlangt das SFS-Startprogramm von dem Benutzer, einen vollständigen Pfad oder ein Verzeichnis auf einer Remote-Web-Seite, die einen URI umfasst, zu spezifizieren. In noch einem anderen Ausführungsbeispiel kann das SFS-Startprogramm ferner eine Eingabe aufnehmen, die private Netzwerkdatei-Server spezifiziert. Wenn der private Netzwerk-Server kein WebDAV-fähiger HTTP-Server ist, werden bei diesem Ausführungsbeispiel die Informationen an das Betriebssystem weitergegeben, um gemäß bekannten Verfahren bearbeitet zu werden. Bei einem Ausführungsbeispiel kann das SFS-Startprogramm nach Empfang eines Remote-Web-Seite spezifizierenden URI den Benutzer auffordern, den Namen des Verzeichnisses der Web-Seite einzugeben, in welchem sich die entfernt gespeicherten Dateien befinden. Bei einem anderem Ausführungsbeispiel kann das SFS-Startprogramm den Benutzer graphisch auffordern, ein Remote-Dateisystem und/oder -Verzeichnis auszuwählen, indem auf ein graphisches Bild gezeigt und geklickt wird, und kann dann nach einem Passwort fragen.

[0026] Bei einem anderen Ausführungsbeispiel kann der Benutzer von dem SFS-Startprogramm ferner aufgefordert werden, das Remote-Dateisystem zu benennen. Dies wird durch dem Fachmann bekannte Verfahren erreicht, wie beispielsweise durch Aufforderung an den Benutzer, zur gleichen Zeit, zu der das SFS-Startprogramm eine URI anfragt, einen Textnamen zur Darstellung des Remote-Dateisystems einzugeben, oder als einen Schritt nachdem die URI bereitgestellt ist.

[0027] Bei einem Ausführungsbeispiel werden diese Informationen, nachdem der Benutzer den Namen des Remote-Dateisystem-Servers und des Remote-Dateisystem-Verzeichnisses angegeben hat, derart gespeichert, dass das Remote-Dateisystem automatisch gemountet wird, wann immer der Benutzer eine Verbindung mit dem Internet herstellt. Bei einem anderen Ausführungsbeispiel, bei welchem der Benutzer eine "always on"-Verbindung mit dem Internet hat, werden diese Informationen, nachdem der Benutzer den Namen des Remote-Dateisystem-Servers und, bei einigen Ausführungsbeispielen, des Remote-Dateisystem-Verzeichnisses angegeben hat, derart gespeichert, dass wann immer der Benutzer den Computer neu startet, das Remote-Dateisystem automatisch gemountet wird. Bei diesen Ausführungsbeispielen kann der Benutzer von dem SFS-Startprogramm gefragt werden, ob ein einen autorisierten Zugriff auf das Remote-Dateisystem bereitstellendes Passwort gespeichert werden soll, und bei zukünftigen Verbindungen mit dem Remote-Server nicht angefragt werden soll, ob immer nach dem Fassung gefragt werden soll, wenn das Remote-Dateisystem gemountet und auf dieses zugegriffen werden soll.

Bei einem anderen Ausführungsbeispiel kann ein Key Chain, wie beispielsweise der mit dem Mac OS® 9-Betriebssystem, verfügbar von Apple Computer, Inc. aus Cupertino, Kalifornien, bereitgestellte, zum Speichern von Passwörtern für mehrere Remote-Dateisysteme verwendet werden, so dass eine Bereitstellung eines einzigen Passwortes für den Key Chain Zugriff auf sämtliche der Passwörter auf dem Key Chain bereitstellt.

[0028] Nach dem Mounten des Remote-Dateisystems betreibt der Benutzer beliebige Programme, wie es in Block **308** gezeigt ist, und kann dann auf dem Computer beliebige Programme zum Zugreifen auf entfernt gespeicherte Dateien auf die gleiche Weise benutzen, als würde der Benutzer auf Dateien zugreifen, die lokal gespeichert sind, wie es in Block **310** gezeigt ist. Auf das Remote-Dateisystem wird zugegriffen, als wäre es ein weiterer Festplattenspeicher in dem Computer des Benutzers. Bei einem Ausführungsbeispiel kann das SFS den Benutzer dann auffordern, ein Passwort zum Authentifizieren bereitzustellen, wenn ein Benutzer zum ersten Mal auf eine Remote-Datei zugreift.

D. Aktionen eines Seamless-Dateisystems

[0029] **Fig. 4A** zeigt den Ablauf von von einem Seamless-Dateisystem durchgeführten Aktionen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems. Ein Benutzer macht über ein Anwendungsprogramm, wie beispielsweise einen Wortprozessor, eine Dateisystemanfrage mit einem Pfadnamen und einer Datei, die sich auf einen WebDAV-fähigen HTTP-Server bezieht, wie es in Block **400** gezeigt ist. Da das SFS-Startprogramm bereits ausgeführt wurde, oder eine andere SFS-Initialisierung auftritt, leitet die Dateisystemschnittstelle der Anwendungsprogrammchnittstelle des Betriebssystems die Anfrage an das SFS-Plugin, wie es in Block **402** gezeigt ist. Anschließend bearbeitet das SFS-Plugin die Anfrage und sendet sie an das SFS-Netzwerkzugriffprogramm, wie es in Block **404** gezeigt ist. Bei einem Ausführungsbeispiel stellt das SFS-Plugin eine Socket-Verbindung mit dem SFS-Netzwerkzugriffprogramm her und sendet den Pfadnamen oder einen anderen Identifier mit den entsprechenden Parametern, sowie einen Anfragetyp über den Socket an das SFS-Netzwerkzugriffprogramm. Das SFS-Netzwerkzugriffprogramm bearbeitet die Anfrage und sendet sie an den entsprechenden WebDAV-fähigen HTTP-Server, um die angefragte Aktion zu erzielen, wie es in Block **406** gezeigt ist. Bei einem Ausführungsbeispiel bearbeitet das SFS-Netzwerkzugriffprogramm die Anfrage, indem der Anfragetyp untersucht wird, die Anfrage in das entsprechende WebDAV- oder HTTP-Format umformatiert wird und über das Internet mit dem entsprechenden WebDAV-fähigen HTTP-Server kommuniziert wird. Anfragte Aktio-

nen umfassen beispielsweise Lösche eine Datei, Lese eine Datei, Bewege eine Datei zwischen Verzeichnissen auf dem Server, etc. Das SFS-Netzwerkzugriffprogramm empfängt dann eine Antwort von dem WebDAV/HTTP-Server, welche der Anfrage entsprechende Informationen enthält, wie es in Block **408** gezeigt ist. Die Antwort kann einen WebDAV/HTTP-Statuscode umfassen. Das SFS-Netzwerkzugriffprogramm bearbeitet dann die Antwort und gibt Statusinformationen, und in gewissen Fällen andere Informationen, an das SFS-Plugin zurück, und zwar bei einem Ausführungsbeispiel über einen Socket, wie es in Block **410** gezeigt ist. Das SFS-Plugin formatiert, wenn notwendig, die zurückgegebenen Informationen und sendet Status und zurückgegebene Informationen, wenn vorhanden, an das anfragende Benutzeranwendungsprogramm, und zwar über die Dateisystemschnittstelle des Betriebssystems, wie es in Block **412** gezeigt ist. Bei einem Ausführungsbeispiel kann das SFS-Plugin einen WebDAV/HTTP-Statuscode als einen lokalen Systemfehlercode umformatieren.

[0030] **Fig. 4B** zeigt den Ablauf von von einem Seamless-Dateisystem vorgenommenen Aktionen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems, das ein Caching umfasst. Ein Benutzer macht über ein Anwendungsprogramm eine Dateisystemanfrage nach einer entfernt gespeicherten Datei auf einem WebDAV-fähigen HTTP-Server, wie es in Block **420** gezeigt ist. Bei einem Ausführungsbeispiel richtet die Dateisystemschnittstelle der Anwendungsprogrammchnittstelle des Betriebssystems die Anfrage an das SFS-Plugin, wie es in Block **422** gezeigt ist, da das SFS-Startprogramm gestartet wurde oder eine andere SFS-Initialisierung auftritt. Das SFS-Plugin bestimmt dann, ob die Datei in dem lokalen Cache ist, wie es in Block **424** gezeigt ist. Wenn die entfernt gespeicherte Datei nicht lokal gecacht ist, sendet das SFS-Plugin die Anfrage an das SFS-Netzwerkzugriffprogramm, wie es in Block **426** gezeigt ist. Das SFS-Netzwerkzugriffprogramm bearbeitet die Anfrage und sendet sie an den entsprechenden WebDAV-fähigen HTTP-Server, um die angefragte Aktion zu erzielen, wie es in Block **428** gezeigt ist. wenn die Anfrage eine Schreibanfrage, wie beispielsweise Schreibe Datei, Benenne Datei um, Benenne Verzeichnis um, etc. ist, kann die Anfrage Dateidaten und Dateiinformationen umfassen, die zum Ausführen der Schreibanfrage benötigt werden. Wenn die Anfrage eine Leseanfrage ist, kann die Anfrage einen Dateinamen, einen Dateideskriptor oder -identifier etc. enthalten. Das SFS-Netzwerkzugriffprogramm empfängt dann eine Antwort von dem WebDAV/HTTP-Server, welche der Anfrage entsprechende Informationen enthält, wie es in Block **432** gezeigt ist. Das SFS-Netzwerkzugriffprogramm bearbeitet dann die Antwort und sendet gleichzeitig die Anfrage beantwortende Informationen an das SFS-Plugin,

wie es in Block **436** gezeigt ist, und cached die Dateiinformationen und Dateidaten in einem lokalen Cache. Die Dateiinformationen und -daten werden in Abhängigkeit von den von dem HTTP/WebDAV-Server erhaltenen Informationen gespeichert und können etwas der Datei oder die gesamte Datei selbst oder einen Teil davon, die Dateigröße, den Dateinamen, den Pfadnamen, die Dateierstellungszeit und das Dateierstellungsdatum, Autoreninformationen, Zugriffsinformationen, etc. enthalten. Das SFS-Plugin formatiert die Informationen, wenn notwendig, und sendet sie über die Dateisystemschnittstelle des Betriebssystems an das anfragende Benutzeranwendungsprogramm, wie es in Block **438** gezeigt ist.

[0031] Wenn die angefragte Datei oder Dateiinformationen bereits in dem lokalen Cache vorhanden waren, wie es in Block **424** gezeigt ist, wird ein Test durchgeführt, um zu bestimmen, welche Art von Anfrage gemacht wurde, wie es in Block **440** gezeigt ist. Bei einem Ausführungsbeispiel kann definiert werden, dass die Dateianfragen zu Gruppe 1 oder Gruppe 2 gehören. Gruppe 1-Dateianfragen sind solche Anfragen, die um Informationen über die internen Inhalte von Dateien nachsuchen, oder Informationen einschließlich den internen Inhalten der Dateien modifizieren. Beispiele von Gruppe 1-Anfragen umfassen ein Lesen einer Datei, Erhalten einer Verzeichnisaufstellung, Schreiben einer Datei, etc. Gruppe 2-Anfragen sind solche Anfrage, die Informationen über die Datei beeinflussen oder die Datei als ein Objekt behandeln. Gruppe 2-Anfragen umfassen Anfragen zum Erzeugen oder Löschen einer Datei oder eines Verzeichnisses, Umbenennen einer Datei oder eines Verzeichnisses, Öffnen einer Datei zum Vorbereiten dieser für einen I/O, etc. Wenn die Anfrage eine Gruppe 1-Anfrage ist, geht die Ausführung weiter, wie es oberhalb zu Block **426** beschrieben wurde. Wenn die Anfrage eine Gruppe 2-Anfrage ist, erlangt das SFS-Plugin die angefragte Datei oder die Dateiinformationen von dem lokalen Cache, wie es in Block **442** gezeigt ist, wobei jegliche Kommunikation mit dem Remote-WebDAV/HTTP-Server umgangen wird. Das SFS-Plugin formatiert die aus dem Cache enthaltenden Informationen dann, wenn notwendig, und sendet sie zu dem Benutzeranwendungsprogramm. In Abhängigkeit von der Anfrage können die dem Benutzeranwendungsprogramm gesendeten Informationen eine Verzeichnisaufstellung, eine Datei, der Inhalt einer Inhalt, etc. sein.

E. Eine Seamless-Dateisystem-Plugin-Erweiterung für ein Betriebssystem

[0032] **Fig. 5** zeigt den Ablauf von von einer Seamless-Dateisystem-Plugin-Erweiterung für ein Betriebssystem vorgenommenen Aktionen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems. Wenn das SFS installiert wurde, umfasst das SFS die SFS-Plugin-Erwei-

terung für das Betriebssystem des Computersystems. Insbesondere ist das SFS-Plugin eine Erweiterung für das Dateisystem des Betriebssystems. Auf eine Benutzeranfrage über ein Anwendungsprogramm nach einer Datei auf einem Remote-Dateisystem gibt das Anwendungsprogramm über eine bekannte Anwendungsprogrammchnittstelle eine Anfrage an das Dateisystems des Computers aus. Das Dateisystem empfängt diese Anfrage, wie es in Block **500** gezeigt ist, und wenn sich die Dateianfrage auf eine lokale Datei bezieht, wie es in Block **510** gezeigt ist, wird die Anfrage an das Betriebssystem weitergegeben, welches dann auf das lokale Dateisystem zugreift, wie es in Block **512** gezeigt ist. Wenn die Dateianfrage nicht auf eine lokale Datei gerichtet ist, wie es in Block **510** gezeigt ist, richtet die lokale Dateisystemschnittstelle die Anfrage an das SFS-Plugin. Um die Diskussion hier zu vereinfachen, werden der Stand der Technik und bekannte Verfahren zum Antworten auf Anfragen nach entfernt gespeicherten Dateien, die über andere Protokolle, Systeme und Verfahren zugänglich sind, nicht beschrieben, können jedoch konkurrierend mit SFS bestehen. Das SFS-Plugin empfängt die Dateisystemanfrage, die von einer Benutzeranfrage eines Anwendungsprogramms von der Dateisystemschnittstelle an das Betriebssystem stammt, wie es in Block **514** gezeigt ist. Das SFS-Plugin gibt die Anfrage dann an das SFS-Netzwerkzugriffprogramm weiter, wie es in Block **516** gezeigt ist. Bei einem Ausführungsbeispiel bearbeitet das SFS-Plugin die Anfrage auch und formatiert sie, bevor die Anfrage an das SFS-Netzwerkzugriffprogramm weitergegeben wird. Das SFS-Plugin empfängt dann eine Antwort von dem SFS-Netzwerkzugriffprogramm, wie es in Block **518** gezeigt ist. Das SFS-Plugin kann die Antwort bearbeiten und Informationen aus der Antwort über die lokale Dateisystemschnittstelle an das anfragende Benutzeranwendungsprogramm weitergeben, wie es in Block **520** gezeigt ist.

F. Ein Seamless-Dateisystem-Netzwerkzugriffanwendungsprogramm

[0033] **Fig. 6** zeigt den Ablauf von von einem Seamless-Dateisystem-Netzwerkzugriff-Anwendungsprogramm vorgenommenen Aktionen gemäß einem Ausführungsbeispiel des Seamless-Dateisystemverfahrens und -systems. Wie oben beschrieben, empfängt das SFS-Netzwerkzugriffprogramm eine Anfrage bezüglich einer entfernt gespeicherten Datei von dem SFS-Plugin, nachdem ein Benutzer über ein Anwendungsprogramm eine Anfrage nach einer entfernt gespeicherten Datei gemacht hat, wie es in Block **600** gezeigt ist. Das SFS-Netzwerkzugriffprogramm formatiert die Anfrage in ein bekanntes Format, um die angefragte Aktion zu erzielen, wie es in Block **610** gezeigt ist. Wie oberhalb beschrieben, wird bei einem Ausführungsbeispiel die Anfrage als eine HTTP-Anfrage formatiert, welche bei einigen Bei-

spielen ein WebDAV-Verfahren umfassen kann, welches in einigen Beispielen weitere Informationen im XML-Format enthalten kann. Das SFS-Netzwerkzugriffprogramm sendet die Anfrage dann über ein Netzwerk an ein Remote-Computersystem, wie es in Block **612** gezeigt ist. Bei einem Ausführungsbeispiel leitet das SFS-Netzwerkzugriffprogramm die Anfrage über das Internet an einen Benutzer-spezifisierten WebDAV/HTTP-Server. Das SFS-Netzwerkzugriffprogramm empfängt eine Antwort auf die Anfrage von dem Remote-Computersystem, wie es in Block **614** gezeigt ist. Bei einem Ausführungsbeispiel bestimmt das SFS-Netzwerkzugriffprogramm, ob die Antwort eine substantielle Antwort anstatt eine Fehlermeldung, ein Fehlercode oder ein Timeout ist, wie es in Block **616** gezeigt ist. Wenn die Antwort eine substantielle Antwort ist, extrahiert das SFS-Netzwerkzugriffprogramm Informationen aus der Antwort, wie es in Block **618** gezeigt ist, und sendet diese Informationen an das SFS-Plugin, wie es in Block **620** gezeigt ist.

[0034] Wenn die Antwort, die das SFS-Netzwerkzugriffprogramm von dem Remote-Computersystem empfangen hat, keine substantielle Antwort ist, wie es in Block **616** gezeigt ist, prüft das SFS-Netzwerkzugriffprogramm, um zu ermitteln, ob eine Fehlermeldung, ein Fehlercode oder ein Timeout zurückgegeben wurde, wie es in Block **622** gezeigt ist. Wenn die Antwort eine Fehlermeldung oder ein Fehlercode war, geht die Ausführung weiter bei den Blöcken **618** und **620**, wie es bereits beschrieben wurde, so dass die extrahierte Fehlermeldung oder der extrahierte Fehlercode übersetzt werden kann und an das SFS-Plugin weitergeleitet wird. Wenn die nicht-substantielle Antwort ein Timeout ist, wird die Anfrage erneut gesendet, wie es im Block **624** gezeigt ist, und die Ausführung geht bei Block **614** weiter, wie es oberhalb beschrieben ist.

[0035] Bei einem Ausführungsbeispiel hält das SFS-Netzwerkzugriffprogramm interne Datenstrukturen, um zu ermöglichen, dass es effizient auf Anfragen von Benutzeranwendungsprogrammen antworten kann und Antworten von Remote-Computersystemen bearbeiten kann. Bei einem Ausführungsbeispiel cached das SFS lokal Remote-Dateien und Remote-Dateisysteminformationen, um die Begrenzungen der WebDAV-Erweiterungen von HTTP zu überwinden. Bei diesem Ausführungsbeispiel erzeugt das SFS-Netzwerkzugriffprogramm und hält Cache-Dateien, die als der SFS-Cache bezeichnet werden, so dass Anfragen bezüglich entfernt gespeicherter Dateien an lokal gespeicherten Kopien der Dateien von dem SFS-Plugin ausgeführt werden können. Nachdem Arbeitsvorgänge an einer bestimmten Datei beendet sind, oder bei anderen entsprechenden Umständen, wird die modifizierte Cache-Datei dann an den Remote-Server kommuniziert, wodurch die auf dem Remote-Computersystem

gespeicherte Datei geupdated wird und die lokal gespeicherte Cache-Kopie mit der entfernt gespeicherten Datei synchronisiert wird. Ein solches Cachen ist gegenüber dem SFS-Benutzer verborgen. Bei einem Ausführungsbeispiel beispielsweise führt eine Anfrage zum Öffnen einer Datei dazu, dass das SFS-Netzwerkzugriffprogramm eine Cache-Datei erzeugt, die Inhalte der Remote-Datei enthält. Bei solch einem Ausführungsbeispiel gibt das SFS-Netzwerkzugriffprogramm einen Datei-Deskriptor an die neu erzeugte Cache-Datei zurück, welcher für nachfolgende Lese- und Schreibvorgänge von dem anfragenden Benutzeranwendungsprogramm und dem SFS-Plugin verwendet wird.

[0036] Der SFS-Cache kann gemäß dem Fachmann bekannten Verfahren implementiert werden. Bei einem Ausführungsbeispiel wird "unsichtbares Cachen" verwendet. Bei solch einem Ausführungsbeispiel wird unmittelbar nachdem das SFS-Netzwerkzugriffprogramm eine lokale Cache-Datei erzeugt und öffnet, die Cache-Datei gelöscht, so dass die Daten für den lokalen Cache gültig bleiben, jedoch der Name der lokal gecachten Datei aus dem lokalen Dateisystemverzeichnis entfernt wird. Ein Vorteil des "unsichtbaren Cachens" ist es, dass Benutzer davor bewahrt werden, auf Cache-Dateien zu stoßen, da der gesamte SFS-Cache für den Benutzer unsichtbar ist. Ein weiterer Vorteil der Verwendung des "unsichtbaren Cachens" ist, dass sämtliche lokalen Cache-Dateien in einem einzigen Verzeichnis erzeugt werden, wodurch das Management des SFS-Caches vereinfacht wird. Bei solch einem Ausführungsbeispiel können die die Verzeichnisstruktur bei dem Remote-Server wiedergebenden Informationen als eine unsichtbare Datei gespeichert werden.

[0037] Bei einem anderen Ausführungsbeispiel wird ein "parallel hierarchisches Caching" verwendet. Da die Kombination von Namensraumdefinitionen von WebDAV und der Domain-Name-Eintragung tatsächlich garantieren, dass sämtliche URIs einzigartig sind, kann ein hierarchischer lokaler Cache auf der Grundlage der von den Benutzern gelieferten URIs erzeugt werden. Bei diesem Ausführungsbeispiel sind die Verzeichnisse in dem Cache dauerhaft und verbleiben in dem Namensraum. Dieses Ausführungsbeispiel erzeugt eine parallele lokale Dateihierarchie, auf die jeder WebDAV-Server zugreift. Bei diesem Ausführungsbeispiel müssen lediglich die tatsächlich von Benutzern referenzierten Dateien in dem lokalen SFS-Cache erscheinen. Bei diesem Ausführungsbeispiel können lokale Cache-Dateien wiederkehrend verwendet werden, was die Anzahl von Dateianfragen an und Datei-Downloads von dem Netzwerk reduziert. Ein Vorteil der Verwendung des "parallel hierarchischen Cachings" ist, dass das SFS-Netzwerkzugriffprogramm keine Abbildung des lokal replizierten Remote-Dateisystems halten muss.

[0038] Bei noch einem weiteren Ausführungsbeispiel wird ein "Map-Caching" verwendet. Bei diesem Ausführungsbeispiel werden Cache-Dateien mit Standardnamen, wie beispielsweise SFS-000001, erzeugt, und eine Abbildung von URIs auf Cache-Dateien wird von dem SFS-Netzwerkzugriffprogramm und/oder dem SFS-Plugin gehalten. Genau wie beim "parallel hierarchischen Caching" werden bei diesem Ausführungsbeispiel lokale Cache-Dateien wiederkehrend gehalten, wodurch die Anzahl von Aufrufen an und Downloads von dem Netzwerk vermindert wird. Ein solches Ausführungsbeispiel erfordert weniger Aufrufe an das lokale Dateisystem als ein "parallel hierarchisches Caching" zum Aufrechterhalten des SFS-Caches.

[0039] Darüber hinaus hält bei einem Ausführungsbeispiel das SFS-Netzwerkzugriffprogramm ein Array von mit URIs gepaarten Datei-Deskriptoren. Wenn eine Datei geöffnet wird und das SFS-Netzwerkzugriffprogramm eine lokale Cache-Datei öffnet, ordnet bei einem Ausführungsbeispiel das SFS-Netzwerkzugriffprogramm den Datei-Deskriptor zusammen mit der speziellen URI in dem nächsten verfügbaren Array-Element an. Bei solch einem Ausführungsbeispiel gibt das SFS-Netzwerkzugriffprogramm den Index als einen Datei-Handle an das SFS-Plugin zusammen mit einem Datei-Deskriptor an das Array zurück, wenn eine Öffne-Datei-Anfrage beantwortet wird. Das SFS-Plugin verwendet den Datei-Deskriptor zum Zugreifen auf die gecachte Datei. Bei diesem Ausführungsbeispiel führen nachfolgende Arbeitsvorgänge, welche eine Aktivität des SFS-Netzwerkzugriffprogramms erfordern, dazu, dass das Datei-Handle von dem SFS-Plugin an das SFS-Netzwerkzugriffprogramm weitergegeben wird, so dass das SFS-Netzwerkzugriffprogramm die entsprechende URI identifiziert, indem auf das Array zugegriffen wird, und anschließend mit dem entsprechenden Remote-Server kommuniziert.

[0040] Wie oberhalb beschrieben werden Remote-Datei-Anfragen von dem SFS-Plugin abgefangen und an das SFS-Netzwerkzugriffprogramm weitergegeben. Die zwischen dem SFS-Netzwerkzugriffprogramm und dem SFS-Plugin weitergegebenen Informationen sind für jeden unterstützten Arbeitsvorgang spezifisch, umfassen jedoch allgemein, bei einem Ausführungsbeispiel, die URI der Remote-Datei oder ausreichende Informationen für das SFS-Netzwerkzugriffprogramm, um die URI zu rekonstruieren, wie es in dem vorhergehenden Paragraphen dargelegt ist. In den meisten Fällen geben die Remote-Datei-Anfragen entweder einen Erfolg oder Standardfehler zurück, die von dem SFS-Netzwerkzugriffprogramm aus den von dem WebDAV/HTTP-Server zurückgegebenen Werten übersetzt werden.

[0041] Aufgrund der transparenten Natur der Benutzerschnittstelle kann der Benutzer zum Bereitstellen

einer einheitlichen Schnittstelle für sämtliche Dateianfragen durch einen Benutzer eines Computersystems, gleichgültig ob entfernte oder lokale Dateien, sämtliche Dateien, und zwar sowohl entfernte als auch lokale, auf die gleiche Weise unter Verwendung der für das lokale Dateisystem erforderlichen Syntax spezifizieren. Das heißt Benutzer von Systemen mit SFS und lokalen Dateisystem-Clients können entfernt gespeicherte Dateien Anfragen, als wären die Dateien gemäß der von dem lokalen Dateisystem und dem lokalen Betriebssystem definierten Syntax lokal gespeichert. Insbesondere definiert ein Betriebssystem zusammen mit einem lokalen Dateisystem, welche Zeichen in Dateinamen verwendet werden können und welche nicht. Jedoch kann der Remote-Server ggf. lediglich die Verwendung eines Untersatzes der bei dem lokalen Dateisystem erlaubten Zeichen unterstützen. Bei einem Ausführungsbeispiel ermöglicht das lokale Dateisystem die Verwendung eines Freizeichens, einer linken eckigen Klammer "[", einer rechten eckigen Klammer "]", eines Rautezeichens "#", eines Fragezeichens "?" und anderer Spezial- oder nicht-alphanumerischen Zeichen in Dateinamen. Bei verschiedenen Ausführungsbeispielen können die Dateinamen von dem Betriebssystem gemäß dem American Standard Code for Information Exchange(ASCII)-Spezifikation, dem Unicode Transformation Format 8 Bit (UTF-8-Codierungsstandard) etc. dargestellt werden. Jedoch akzeptieren WebDAV-fähige HTTP-Server lediglich Dateinamen gemäß der in der URI-Spezifikation definierten Form (siehe oberhalb). Die URI-Spezifikation begrenzt die Verwendung von bei Dateinamen zu verwendenden Zeichen auf ASCII-alphanumerische Zeichen. Die URI-Spezifikation definiert bestimmte reservierte oder spezielle Verwendungen für eine Reihe spezieller Zeichen. Um die Begrenzungen des zum Darstellen von Dateinamen bei WebDAV/HTTP-Servern erlaubten Zeichensatzes zu überwinden, übersetzt das SFS-Netzwerkzugriffprogramm ausgehende Anfragen von dem lokalen Betriebssystem/Dateisystemformat aus erlaubten Zeichen in eine Sequenz von URI-erlaubten Zeichen, bei welchen die speziellen Zeichen als Escape-Sequenzen dargestellt sind. Ähnlich rückübersetzt das SFS-Netzwerkzugriffprogramm eingehende Informationen von dem entfernten WebDAV/HTTP-Server im URI-Format, in das Format des lokalen Betriebssystems/Dateisystems, wobei Escape-Sequenzzeichen in ASCII, UTF-8 etc. übersetzt werden.

[0042] Die URI-Spezifikation ermöglicht es, dass die reservierten Spezialzeichen durch Escape-Sequenzen dargestellt werden können. Dies ermöglicht die Verwendung der Spezialzeichen auf eine "versteckte" Weise und vermeidet jegliche unerwünschte Interpretation der Spezialzeichen. Die URI-Spezifikation definiert eine Escape-Sequenz als ein Prozentzeichen "%" gefolgt von einer zweistelligen hexadezimalen Darstellung des ASCII-Codes des Spezialzei-

chens. Wenn ein Remote-Dateiname beispielsweise von einem Benutzer eines lokalen Computersystems als ein Freizeichen enthalten definiert wird, wie beispielsweise der Name "meine Datei", würde der Dateiname übersetzt und dargestellt als "meine%20Datei", da 32 (dezimal) der ASCII-Code für ein Freizeichen ist. Ein weiteres Beispiel ist "Brief #1 (an Bob)", was übersetzt wird in "Brief%20%231%20%28an%20Bob%29". Bei Empfang der Dateinamen, Verzeichnisnamen, Pfade etc. von einem WebDAV/HTTP-Server in URI-Syntax werden die Dateinamen, Verzeichnisnamen, Pfade etc. aus der URI-Syntax in die Syntax des lokalen Betriebssystems, wie beispielsweise ASCII oder UTF-8, zurückübersetzt. Bei einem Ausführungsbeispiel führt das SFS-Netzwerkzugriffprogramm die Übersetzung durch, wenn das SFS-Netzwerkzugriffprogramm die Anfrage in ein bekanntes Format formatiert, wie es in Block 610 gezeigt ist, und kann die Rückübersetzung gemäß einem beliebigen der Blöcke 614, 618 oder 620 ausführen.

G. Einige SFS-unterstützte Operationen

[0043] Verschiedene Dateisystemoperationen können von SFS überstützt werden. Solche Dateisystemoperationen sind aus dem Stand der Technik bekannt. Beispiele einiger Dateisystemoperationen, und wie sie bei einem Ausführungsbeispiel von SFS implementiert werden können, folgen.

1. Öffnen und Schließen von Verzeichnissen und Dateien

[0044] Bei einem Ausführungsbeispiel dient das SFS-Plugin als ein Pass-Through-Dateisystem. Bei diesem Ausführungsbeispiel fängt das SFS-Plugin an ein Remote-Dateisystem gerichtete Dateisystemanfragen ab, manipuliert die Argumente und instruiert das lokale Dateisystem, eine Sequenz von Operationen an der, einer angefragten Remote-Datei oder einem angefragten Remote-Verzeichnis entsprechenden, lokal gespeicherten Cache-Datei auszuführen. Wenn eine Remote-Datei geöffnet wird, bewirkt der Öffnen-Befehl, dass das SFS-Netzwerkzugriffprogramm eine Cache-Datei erzeugt, ferner die entsprechende WebDAV-Methode zum Verschließen der Datei exklusiv auf dem Remote-Server ausgibt, die entsprechende WebDAV/HTTP-Methode zum Erlangen der Inhalte der Zielfile von dem Remote-WebDAV/HTTP-Server ausgibt und die erhaltenen Daten in die neu erzeugte Cache-Datei schreibt. Das SFS-Netzwerkzugriffprogramm sendet ein Datei-Handle an das SFS-Plugin, wenn die angefragte Operation beendet ist. Das SFS-Plugin kann das Datei-Handle der Cache-Datei für einen zukünftigen lokalen Zugriff auf eine Kopie der Remote-Datei aufbewahren. Bei diesem Ausführungsbeispiel können nachfolgende Datei-Deskriptor-Operationen von dem SFS-Plugin an die Cache-Datei umgeleitet werden.

Die Dateisystemschnittstelle des lokalen Dateisystems, auf welchem die lokale Cache-Datei liegt, wird zum Ausführen der Datei-Deskriptor-Operationen aufgerufen. Bei FSYNC oder CLOSE gibt das SFS-Netzwerkzugriffprogramm die entsprechende HTTP-Methode aus, um die modifizierten Inhalte der lokal gecachten Datei auf den Remote-Server zu schreiben. Das SFS-Netzwerkzugriffprogramm verwendet den Datei-Handle um zu identifizieren, welche Remote-Datei zu schließen oder zu entsperren ist.

[0045] Die von dem SFS zum Öffnen eines Verzeichnisses verwendete Sequenz von Aktionen ist ähnlich. Bei einem Ausführungsbeispiel gibt das SFS-Netzwerkzugriffprogramm, da das WebDAV-Protokoll eine Verzeichnis-Aufzählung nicht unterstützt, eine WebDAV-PROPFIND-Anfrage nach sämtlichen Eigenschaften unter Verwendung der Verzeichnis-URI und Spezifizierung einer Header-Tiefe von eins aus. In Beantwortung gibt der WebDAV/HTTP-Server die spezifizierten Eigenschaften sämtlicher der Items in der Sammlung zurück, das heißt sämtlicher Dateien in dem Verzeichnis auf dem Server. Bei einem Ausführungsbeispiel paart das SFS-Netzwerkzugriffprogramm die zurückgegebene XML-Property-Liste, um die Namen der Einträge in dem Verzeichnis zu bestimmen, und erzeugt dann eine lokale Cache-Datei, welche es ermöglicht, dass das Remote-Dateisystem vollständig mit Verzeichniseinträgen gemäß der Verzeichnisaufstellungsart des lokalen Computersystems repliziert wird. Der Datei-Deskriptor für die die Remote-Verzeichnisaufstellung darstellende gecachte Datei wird zurückgegeben.

[0046] Bei einem Ausführungsbeispiel bewirkt eine Anfrage zum Schließen eines Verzeichnisses, das das SFS-Netzwerkzugriffprogramm die das Verzeichnis darstellende lokal gecachte Datei schließt. Bei solch einem Ausführungsbeispiel ist keine Kommunikation mit dem WebDAV/HTTP-Server notwendig.

2. Erzeugen von Dateien

[0047] Wenn dem SFS-Netzwerkzugriffprogramm von dem SFS-Plugin eine Erzeuge-Anfrage weitergeleitet wird, sendet das SFS-Netzwerkzugriffprogramm die URI der Zielfile mit einer WebDAV-PUT-Methode an den Remote-WebDAV/HTTP-Server. Wenn bereits eine Cache-Datei für das Verzeichnis, in dem der neue Dateieintrag zu erzeugen ist, existiert, wird bei einem Ausführungsbeispiel die Cache-Datei geupdatet, um den neu erzeugten Eintrag zu umfassen. Bei einem anderen Ausführungsbeispiel kann die Cache-Datei geleert und neu erzeugt werden, um die neuen Dateinformationen zu erhalten.

3. FSYNC

[0048] Bei einem Ausführungsbeispiel ist FSYNC die primäre Synchronisationsroutine, die zum Bewegen von Daten von dem lokalen Cache zu dem Server verwendet wird. FSYNC-Aufrufe werden von dem SFS-Plugin an das SFS-Netzwerkzugriffprogramm unter Verwendung des von dem SFS-Netzwerkprogramm in Antwort auf eine Öffnen-Anfrage gelieferten Datei-Handle kommuniziert. Das SFS-Netzwerkzugriffprogramm gibt die entsprechende web-DAV-Methode aus, um die Daten zurück auf den Remote-Server zu schieben. Das SFS-Netzwerkzugriffprogramm gibt ferner sämtliche von dem Web-DAV/HTTP-Server empfangene Fehler an das SFS-Plugin zurück.

4. Erlangen und Setzen von Attributen

[0049] Bei einem Ausführungsbeispiel werden Attribute von dem SFS-Netzwerkzugriffprogramm über PROFIND- bzw. PROPATCH-WebDAV-Protokollmethoden erhalten und gesetzt.

5. Erzeugen von Verzeichnissen

[0050] Bei einem Ausführungsbeispiel bewirken Verzeichnis-Erzeugen-Aufrufe, welche als MKDIR bekannt sein können, dass das SFS-Netzwerkzugriffprogramm eine WebDAV-MKCOL-Anfrage an den WebDAV-Server zum Durchführen einer Sammlung ausgibt.

6. Lesen einer Datei

[0051] Bei einem Ausführungsbeispiel bewirken Lese-Anfragen, dass Daten aus der lokalen, der entfernt gespeicherten mit der Lese-Anfrage spezifizierten korrespondierenden Datei Cache-Datei gelesen werden. Wenn es keine lokal gecachte Kopie der angefragten Remote-Datei gibt, unterstützen die Web-DAV-Erweiterungen des HTTP-Protokolls die Fähigkeit, Teile einer Datei ohne Erlangen einer gesamten Datei zu erlangen. Wenn somit Dateien für nur einen Lesezugriff zu öffnen sind, kann das SFS-Netzwerkzugriffprogramm lediglich den angefragten Abschnitt einer Datei von dem spezifizierten Web-DAV-HTTP-Server für jede Leseanfrage erlangen.

7. Schreiben einer Datei

[0052] Bei einem Ausführungsbeispiel werden sämtliche Schreib-Anfragen an eine lokal gecachte Kopie der spezifizierten entfernt gespeicherten Datei durchgeführt. Ein Schreiben in Dateien wird bei FSYNC oder CLOSE synchronisiert. Bei anderen Ausführungsbeispielen kann eine häufigere Synchronisation zwischen der lokal gecachten Version der Datei und der entfernt gespeicherten Datei erreicht werden, wenn bestimmte Bedingungen erfüllt sind,

wie beispielsweise zum Ende einer vorgegebenen Zeitperiode oder wenn ein vordefinierter Prozentsatz der lokal gecachten Datei geändert wurde.

8. Lesen eines Verzeichnisses

[0053] Das Lesen eines Verzeichnisses kann über den allgemein bekannten READDIR-Befehl angefragt werden. Bei einem Ausführungsbeispiel kann eine Anfrage zum Lesen eines Verzeichnisses nur gemacht werden, nachdem das Verzeichnis geöffnet wurde. Bei diesem Ausführungsbeispiel werden sämtliche Lese-Verzeichnis-Anfragen von dem SFS-Plugin zum Zugreifen auf eine lokal gecachte Darstellung des bereits geöffneten angeforderten Remote-Verzeichnisses umgeleitet.

9. Entfernen eines Verzeichnisses und Löschen einer Datei

[0054] Nach Empfang einer Anfrage zum Entfernen eines Verzeichnisses oder Löschen einer Datei auf einem Remote-Server gibt das SFS-Netzwerkzugriffprogramm bei einem Ausführungsbeispiel unter Spezifizierung der angefragten URI und, wenn angebracht, der zu löschende Datei eine WebDAV-DELETE-Anfrage an den WebDAV/HTTP-Server. Wenn eine lokal gecachte Darstellung des Remote-Verzeichnisses für das Verzeichnis mit der zu löschenden Datei existiert, wird bei einem Ausführungsbeispiel die lokal gecachte Kopie der Remote-Verzeichnisaufstellung zum Entfernen des gelöschten Eintrags geupdated. Bei einem anderen Ausführungsbeispiel leert das SFS-Netzwerkzugriffprogramm die lokal gecachte Kopie der Remote-Verzeichnisaufstellung und erzeugt eine lokal gecachte Kopie der Verzeichnisaufstellung, indem für die entsprechende URI und beliebige Unterverzeichnisse eine WebDAV/PROPFIND-Anfrage an den WebDAV/HTTP-Server ausgegeben wird. Wenn eine lokal gecachte Kopie der Remote-Datei existiert, leert das SFS-Netzwerkzugriffprogramm die lokal gecachte Kopie der entfernt gespeicherten Datei.

10. Trunkieren

[0055] Bei einem Ausführungsbeispiel werden sämtliche Trunkier-Anfragen an einer lokal gespeicherten Kopie der spezifizierten entfernt gespeicherten Datei durchgeführt. Trunkierte Dateien werden bei FSYNC oder CLOSE synchronisiert.

Patentansprüche

1. Ein System mit einer Mehrzahl von mit dem Internet (32) gekoppelten Web-Servern (34), auf denen das Distributed Authoring and Versioning-fähige Hypertext Transfer Protocol (HTTP), WebDAV, läuft, und einer Mehrzahl von mit dem Internet gekoppelten Personal-Computern (10), wobei jeder Perso-

nal-Computer (10) ein Dateisystem-Plugin aufweist (50), das mit einem Dateisystem (56) eines Betriebssystems (48) kommunikativ gekoppelt ist, das von einer Anwendung (54) empfangene Dateisystemanfragen, die auf einem der Mehrzahl von Web-Servern gespeicherte Dateisysteme betreffen, an ein Netzwerkzugriff-Anwendungsprogramm (52) auf dem Personal-Computer weiterleitet, das die Dateisystemanfragen jeweils als eine WebDAV-Anfrage an einen geeigneten der Mehrzahl von Web-Servern (34) sendet, wobei die Dateisystemanfragen einen Dateinamen einer Datei umfassen, auf die zuzugreifen ist, und wobei, wenn der Dateiname ein oder mehrere Zeichen enthält, die von dem besagten einen der Mehrzahl von Web-Servern (34) nicht erkannt werden, das Netzwerkzugriff-Anwendungsprogramm (52) die nicht erkannten Zeichen des Dateinamens in eine oder mehrere, von dem besagten einen der Mehrzahl von Web-Servern erkennbare Escape-Sequenzen umwandelt, bevor die umgewandelten Dateisystemanfragen an den besagten einen der Mehrzahl von Web-Servern (34) übermittelt werden, wobei die eine oder die mehreren Escape-Sequenzen das eine oder die mehreren nicht erkannten Zeichen darstellen, wobei die Dateisystemanfragen von dem Dateisystem-Plugin (50) ohne Kenntnis von der Anwendung (54) weitergeleitet werden, und wobei die Anwendung (54) unabhängig in Bezug auf das Dateisystem-Plugin (50) läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung (54) notwendig sind.

2. Das System nach Anspruch 1, wobei in Reaktion auf die Dateisystemanfragen das Netzwerkzugriff-Anwendungsprogramm (52) eine lokale Cache-Datei erzeugt, die entfernte Datei auf dem Web-Server (34) für einen exklusiven Zugriff sperrt, um Andere an dem Zugriff auf die entfernte Datei bei dem Web-Server (34) zu hindern, den im wesentlichen gesamten Inhalt der entfernten Datei unter Verwendung eines oder mehrerer WebDAV-Befehle in die lokale Cache-Datei zwischenspeichert, und ein Datei-Handle von der lokalen Cache-Datei an das Dateisystem-Plugin (50) zurückgibt, um dem Dateisystem-Plugin (50) zu ermöglichen, die Dateisystemanfragen unter Verwendung des Inhalts der lokalen Cache-Datei zu verarbeiten.

3. Das System nach Anspruch 2, wobei die lokale Cache-Datei mit der entfernten, auf dem Web-Server (34) gespeicherten Datei synchronisiert wird, wenn eine vorgegebene Zeitspanne verstrichen ist oder ein vorgegebener Anteil des Inhaltes der lokalen Cache-Datei verändert worden ist.

4. Das System nach Anspruch 2, wobei nach der Verarbeitung der Dateisystemanfrage das Netzwerkzugriff-Anwendungsprogramm (52) modifizierte Abschnitte der lokalen Cache-Datei in die entfernte Datei zurückschreibt, die entfernte Datei freigibt und die

entfernte Datei unter Verwendung eines oder mehrerer WebDAV-kompatibler Befehle schließt.

5. Das System nach Anspruch 2, wobei das Netzwerkzugriff-Anwendungsprogramm (52) ein Array mit einer Mehrzahl von Einträgen unterhält, um Datei-Deskriptoren von lokalen Cache-Dateien zusammen mit einer URI (Uniform Resource Indicator) der auf dem Web-Server (34) gespeicherten entfernten Datei, die mit der entsprechenden lokalen Cache-Datei verknüpft ist, zu speichern, wobei das Netzwerkzugriff-Anwendungsprogramm einen Index des korrespondierenden Eintrages als einen Datei-Handler und den Datei-Deskriptor an das Dateisystem-Plugin zurückgibt, und wobei der Datei-Deskriptor von dem Dateisystem-Plugin verwendet wird, um auf die entsprechende lokale Cache-Datei zuzugreifen, und der Index von dem Dateisystem-Plugin als eine Referenz verwendet wird, um das Netzwerkzugriff-Anwendungsprogramm zu instruieren, auf die entfernte Datei unter Verwendung der zugehörigen URI zuzugreifen.

6. Ein Verfahren, aufweisend:
Empfangen einer ein entferntes Dateisystem betreffenden Dateisystemanfrage von einem Dateisystem-Plugin, das kommunikativ mit einem Dateisystem eines Betriebssystems gekoppelt ist, wobei das Dateisystem-Plugin die Dateisystemanfrage von einer Anwendung als eine reguläre Dateisystemanfrage in einem anderen Format als dem Distributed Authoring And Versioning-Protokoll empfängt,
Erzeugen einer Anfrage im WebDAV-Format in Reaktion auf die Dateisystemanfrage,
Umwandeln eines oder mehrerer Zeichen der Dateisystemanfrage, die von einem WebDAV-fähigen HTTP-Server nicht erkannt werden, in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind,
Weiterleiten der Anfrage im WebDAV-Format an den entsprechenden WebDAV-fähigen HTTP-Server ohne Kenntnis der Anwendung, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind,
Empfangen einer Antwort von dem WebDAV-fähigen HTTP-Server, und
Übermitteln von in der Antwort enthaltenen Informationen an das Dateisystem-Plugin, wobei die Informationen als eine reguläre Systemantwort präsentiert werden.

7. Das Verfahren nach Anspruch 6, ferner aufweisend:
Erzeugen einer lokalen Cache-Datei in Reaktion auf die Dateisystemanfrage,
Sperren einer entfernten Datei, auf die von der Dateisystemanfrage zugegriffen wird, für einen exklusiven Zugriff, um Andere an dem Zugriff auf die entfernte

Datei zu hindern,
Zwischenspeichern des im wesentlich gesamten Inhaltes der entfernten Datei in der lokalen Cache-Datei unter Verwendung von einem oder mehreren WebDAV-kompatiblen Befehlen, und
Zurückgeben eines Datei-Handlers der lokalen Cache-Datei, wobei der Datei-Handler zum Verarbeiten der Dateisystemanfrage unter Verwendung des Inhaltes der lokalen Cache-Datei verwendet wird.

8. Das Verfahren nach Anspruch 7, ferner aufweisend ein Synchronisieren des Inhaltes der lokalen Cache-Datei und der entfernten Datei, wenn eine vorgegebene Zeitspanne verstrichen ist oder ein vorgegebener Anteil des Inhaltes der lokalen Cache-Datei verändert worden ist.

9. Das Verfahren nach Anspruch 7, ferner aufweisend:
Zurückschreiben von zumindest modifizierten Abschnitten der lokalen Cache-Datei in die entfernte Datei nachdem die Dateisystemanfrage verarbeitet worden ist,
Freigeben der entfernten Datei, und
Schließen der entfernten Datei unter Verwendung von einem oder mehreren WebDAV-kompatiblen Befehlen.

10. Das Verfahren nach Anspruch 7, ferner aufweisend ein Konvertieren eines WebDAV-Status-Codes in einen entsprechenden lokalen Fehlercode des Betriebssystems.

11. Das Verfahren nach Anspruch 7, ferner aufweisend:
Halten eines Arrays mit einer Mehrzahl von Einträgen, wobei jeder Eintrag einen der entsprechenden lokalen Cache-Datei zugeordneten Datei-Deskriptor umfasst, wobei der Datei-Deskriptor gepaart ist mit einer URI, die einem Ort der auf einem Web-Server gespeicherten entfernten Datei zugeordnet ist.

12. Das Verfahren nach Anspruch 11, wobei der Datei-Deskriptor von dem Dateisystem-Plugin verwendet wird, um auf die entsprechende lokale Cache-Datei zuzugreifen, und die URI wird von dem Netzwerkzugriff-Anwendungsprogramm verwendet, um auf die entfernte Datei zuzugreifen.

13. Das Verfahren nach Anspruch 12, wobei ein Index eines Eintrages von dem Dateisystem-Plugin verwendet wird, um das Netzwerkzugriff-Anwendungsprogramm zu instruieren, auf die entfernte Datei unter Verwendung der in dem durch den Index referenzierten Eintrag gespeicherten URI zuzugreifen.

14. Ein maschinen-lesbares Medium mit darauf gespeicherten Befehlen, welche bei Ausführung durch einen Prozessor die Maschine veranlassen, Operationen auszuführen:

Empfangen einer ein entferntes Dateisystem betreffenden Dateisystemanfrage von einem Dateisystem-Plugin, das kommunikativ mit einem Dateisystem eines Betriebssystems gekoppelt ist, wobei das Dateisystem-Plugin die Dateisystemanfrage von einer Anwendung als eine reguläre Dateisystemanfrage in einem anderen Format als dem Distributed Authoring And Versioning-Protokoll empfängt,
Erzeugen einer Anfrage im WebDAV-Format in Reaktion auf die Dateisystemanfrage,
Umwandeln eines oder mehrerer Zeichen der Dateisystemanfrage, die von einem WebDAV-fähigen HTTP-Server nicht erkannt werden, in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind,
Weiterleiten der Anfrage im WebDAV-Format an den entsprechenden WebDAV-fähigen HTTP-Server ohne Kenntnis der Anwendung, und wobei die Anwendung unabhängig in Bezug auf das Dateisystem-Plugin läuft, ohne dass Änderungen der mit dem Dateisystem kommunizierenden Anwendung notwendig sind,
Empfangen einer Antwort von dem WebDAV-fähigen HTTP-Server, und
Übermitteln von in der Antwort enthaltenen Informationen an das Dateisystem-Plugin, wobei die Informationen als eine reguläre Systemantwort präsentiert werden.

15. Das maschinen-lesbare Medium nach Anspruch 14, wobei das Empfangen einer Dateisystemanfrage umfasst:
Erhalten von zumindest einem Uniform Resource Identifier (URI) und eines Anfragetyps.

16. Das maschinen-lesbare Verfahren nach Anspruch 15, wobei das Erzeugen umfasst:
Wählen eines geeigneten WebDAV-Verfahrens in Reaktion auf den Anfragetyp.

17. Das maschinen-lesbare Verfahren nach Anspruch 14, wobei die von dem Prozessor ausgeführten Befehle das System veranlassen, Operationen auszuführen, ferner aufweisend:
Extrahieren von Informationen aus der Antwort.

18. Das maschinen-lesbare Verfahren nach Anspruch 14, wobei das Extrahieren umfasst:
Konvertieren eines WebDAV-Status-Codes in einen entsprechenden Fehlercode des lokalen Betriebssystems.

19. Das maschinen-lesbare Verfahren nach Anspruch 17, wobei die von dem Prozessor ausgeführten Befehle das System veranlassen, Operationen auszuführen, ferner aufweisend:
Erzeugen einer lokalen Cache-Datei zum Speichern von zumindest den Informationen.

20. Das maschinen-lesbare Medium nach An-

spruch 19, wobei das Übertragen umfasst:
Übergeben eines Datei-Handle an die lokale Cache-Datei, oder zumindest einen Abschnitt der Informationen.

21. Das maschinen-lesbare Medium nach Anspruch 17, wobei die von dem Prozessor ausgeführten Befehle das System veranlassen, Operationen auszuführen, ferner aufweisend:
Updaten einer lokalen Cache-Datei in Reaktion auf die Informationen.

22. Ein Computersystem, aufweisend:
zumindest ein Anwendungsprogramm (54),
ein eine Dateisystemschnittstelle bereitstellendes Betriebssystem (48),
ein mit der Dateisystemschnittstelle (56) gekoppeltes Dateisystem-Plugin (50) zum Empfangen einer Anfrage für eine entfernt gespeicherte Datei von der Dateisystemschnittstelle des Betriebssystems, die von dem Anwendungsprogramm (54) initiiert wird, und zum Weiterleiten der Anfrage für die entfernt gespeicherte Datei,
ein Netzwerkzugriff-Anwendungsprogramm (52) zum Empfangen der Anfrage für die entfernt gespeicherte Datei von dem Dateisystem-Plugin, zum Übersetzen von in der Anfrage spezifizierten Dateinameinformationen von einer lokalen Dateisystemsyntax in eine Syntax eines entfernten Servers, und zum Verpacken der Anfrage gemäß eines Distributed Authoring and Versioning-Protokolls zum Zugreifen auf eine entfernte Datei eines benutzer-spezifisierten entfernten WebDAV-fähigen HTTP-Servers über das Internet (32),
wobei die Anfrage einen Dateinamen einer Datei umfasst, auf die zugegriffen wird, und wobei, wenn der Dateiname ein oder mehrere Zeichen umfasst, die von dem WebDAV-fähigen HTTP-Server (34) nicht erkannt werden, das Netzwerkzugriff-Anwendungsprogramm (52) die nicht erkannten Zeichen des Dateinamens in eine oder mehrere Escape-Sequenzen, die von dem WebDAV-fähigen HTTP-Server erkennbar sind, umwandelt vor einer Übermittlung der konvertierten Dateisystemanfragen an den WebDAV-fähigen HTTP-Server, wobei die eine oder die mehreren Escape-Sequenzen das eine oder die mehreren nicht erkannten Zeichen darstellen,
wobei die Dateisystemanfragen von dem Dateisystem-Plugin (50) ohne Kenntnis von der Anwendung (54) weitergeleitet werden, und wobei die Anwendung (54) unabhängig in Bezug auf das Dateisystem-Plugin (50) läuft, ohne das Änderungen der mit dem Dateisystem kommunizierenden Anwendung (54) notwendig sind.

23. Das Computersystem nach Anspruch 22, wobei das Netzwerkzugriff-Anwendungsprogramm (52) eine von dem benutzerspezifisierten entfernten Computersystem empfangene Antwort neu formatiert, was eine Rückübersetzung sämtlicher Dateinamen-

informationen aus einer Syntax eines entfernten Servers in eine lokale Dateisystemsyntax umfasst, und eine neu formatierte Antwort an das Extension-Programm des Betriebssystems weiterleitet.

24. Das Computersystem nach Anspruch 23, wobei die Syntax des entfernten Servers die Syntax eines Uniform Resource Identifiers (URI) ist.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

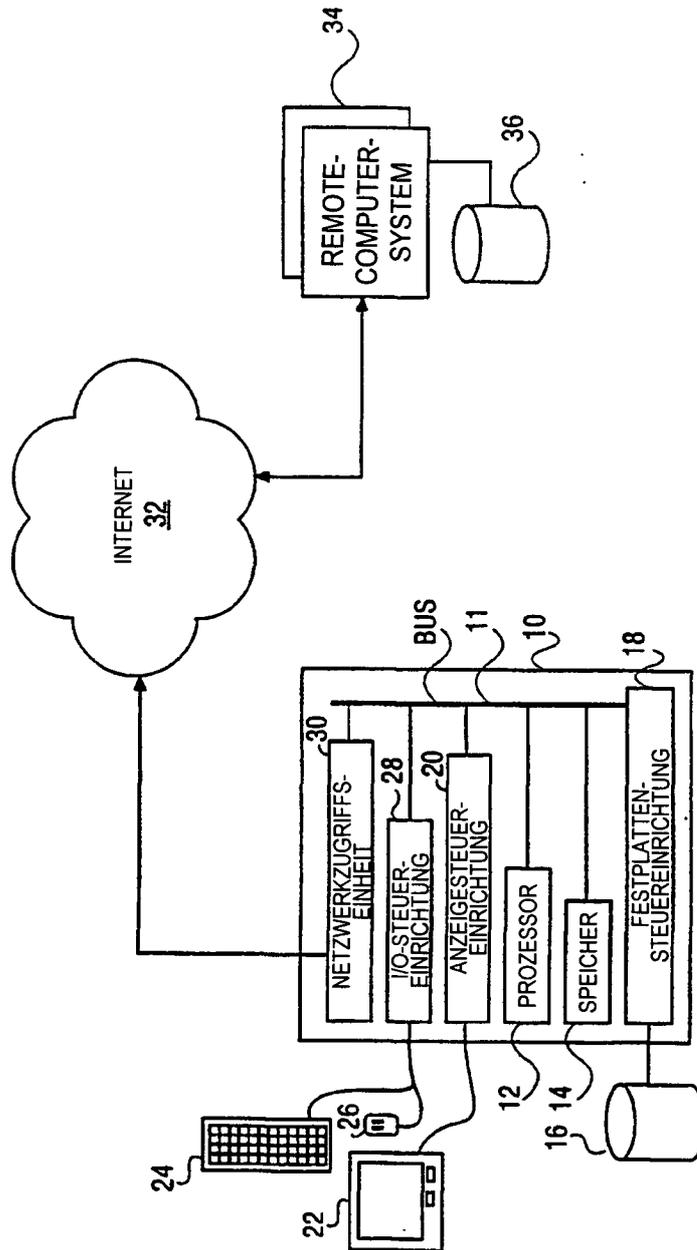


FIG. 1

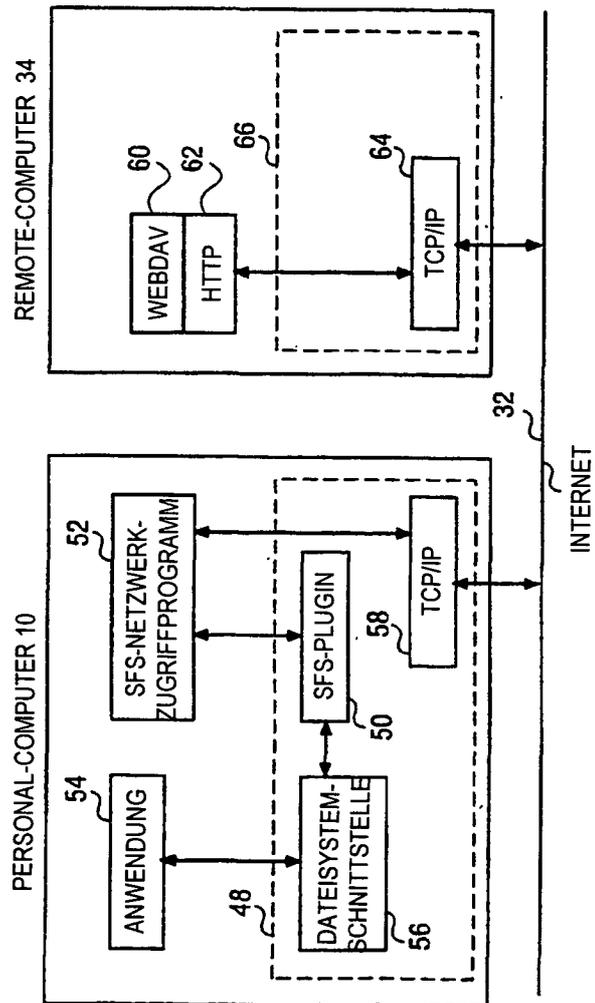


FIG. 2

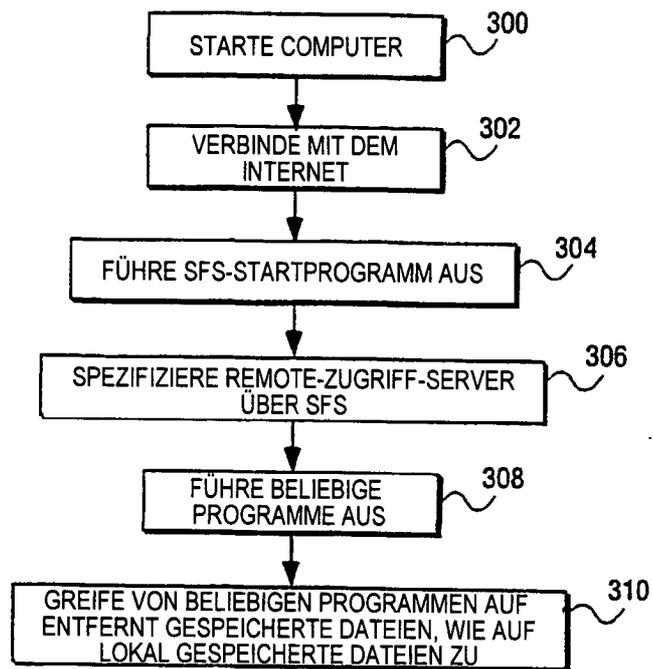


FIG. 3

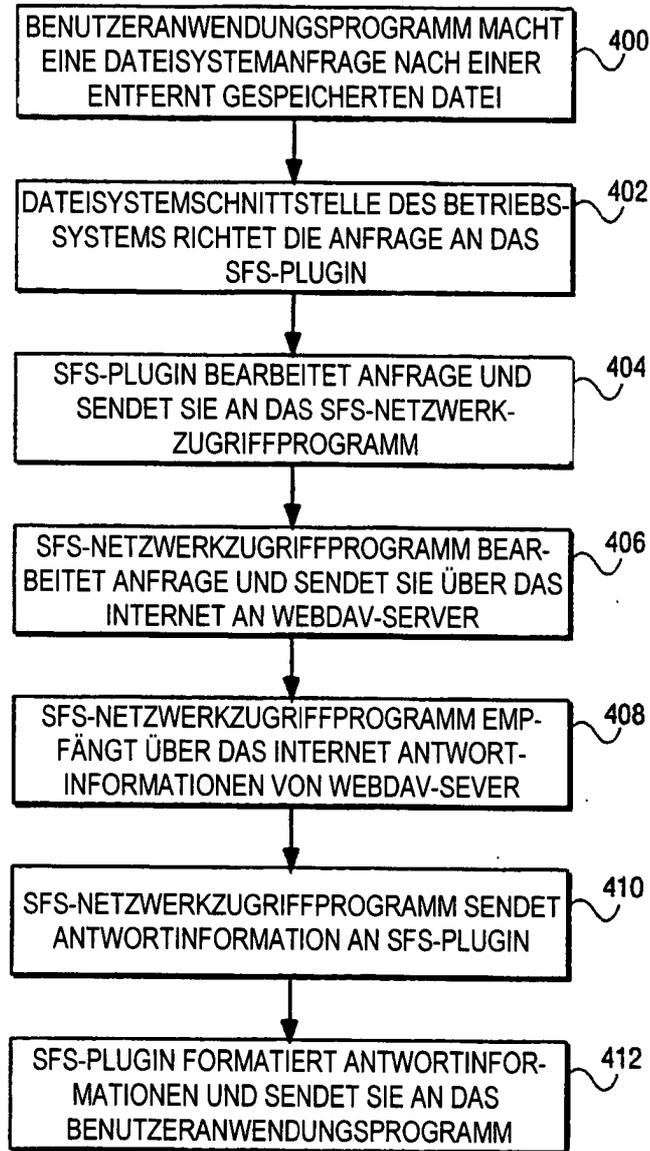


FIG. 4A

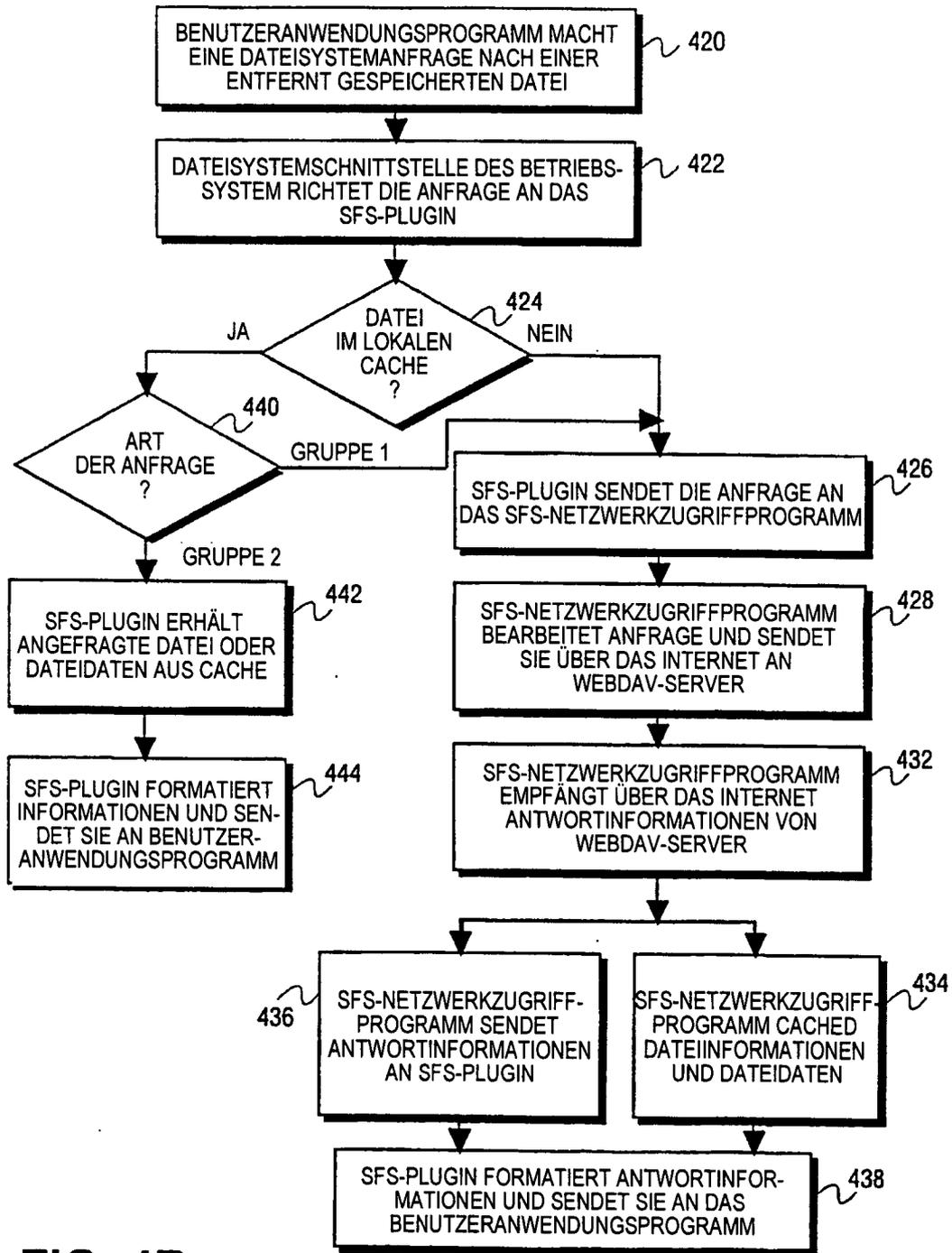


FIG. 4B

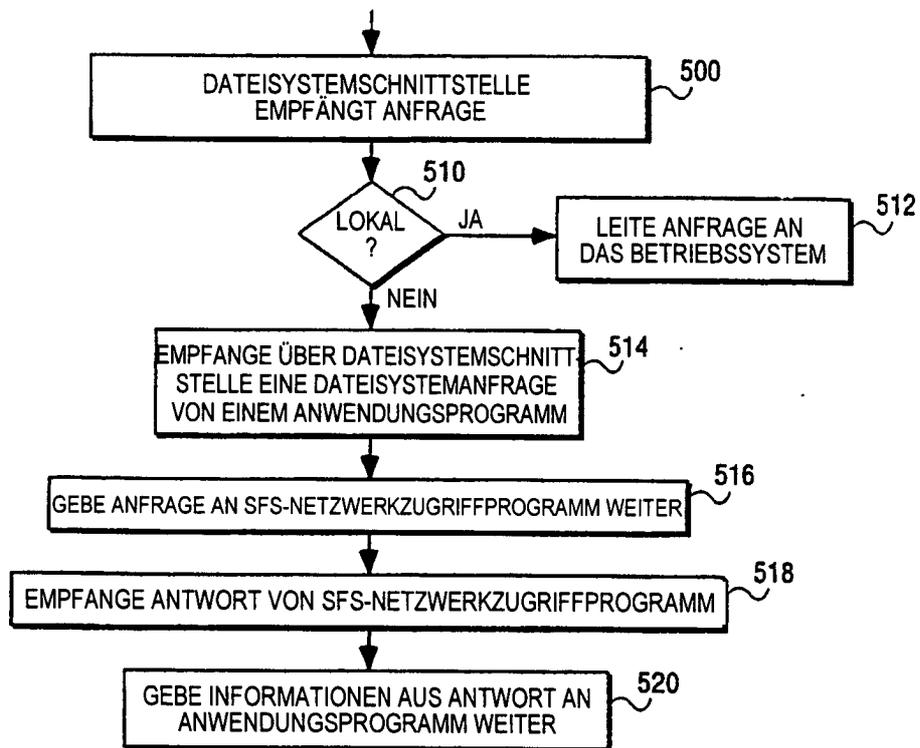


FIG. 5

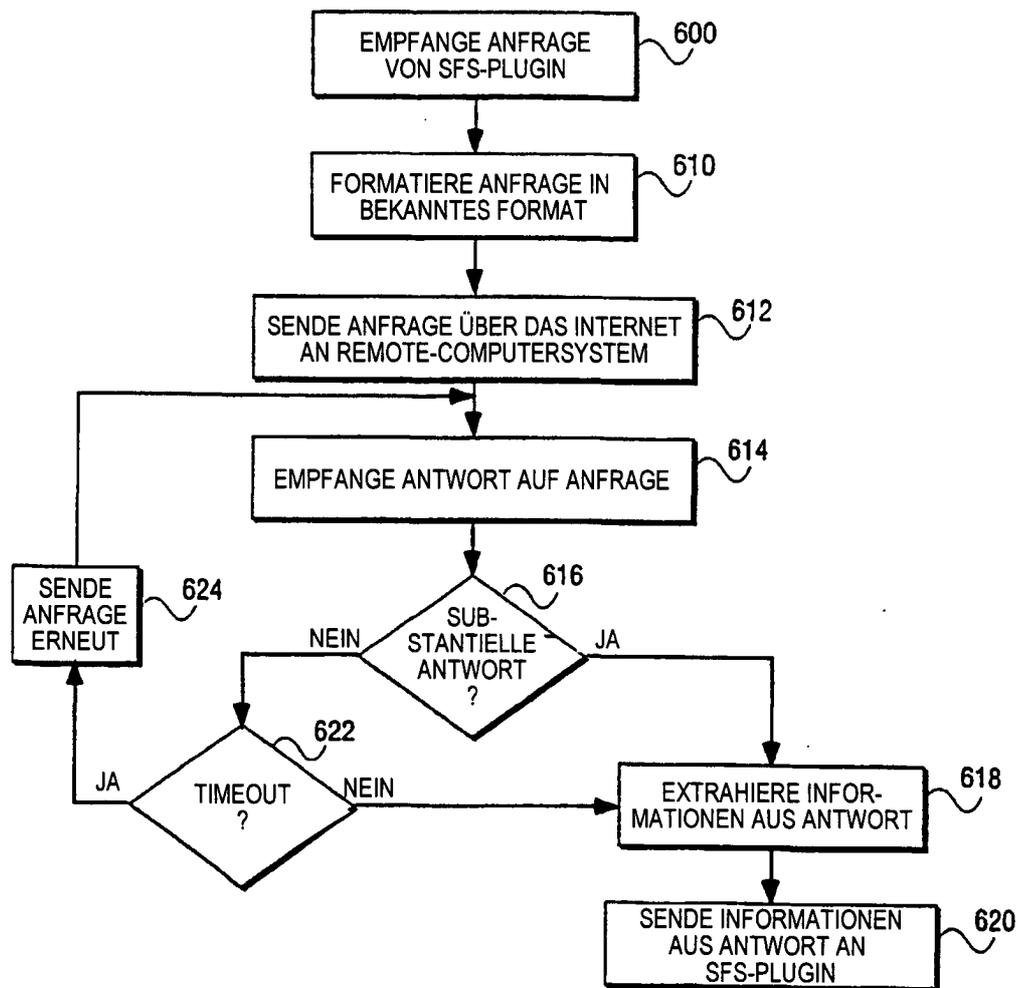


FIG. 6