



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 60 2005 000 025 T2 2006.11.30

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 558 001 B1

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

(21) Deutsches Aktenzeichen: **60 2005 000 025.9**

(96) Europäisches Aktenzeichen: **05 250 064.2**

(96) Europäischer Anmeldetag: **07.01.2005**

(97) Erstveröffentlichung durch das EPA: **27.07.2005**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **28.06.2006**

(47) Veröffentlichungstag im Patentblatt: **30.11.2006**

(30) Unionspriorität:
764754 26.01.2004 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Lucent Technologies Inc., Murray Hill, N.J., US

(72) Erfinder:
Jan W.Hellenthal, Ermelo, 3851 LA, NL;
Franciscus J.M. Panken, Bussum, 1402 PM, NL

(74) Vertreter:
derzeit kein Vertreter bestellt

(54) Bezeichnung: **Verfahren und Anordnung für den Betrieb eines offenen Netzwerks mit einem Proxy**

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 eingelebt, 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] Ausführungsformen der vorliegenden Erfindung betreffen Telekommunikationsnetzwerke. Insbesondere betreffen Ausführungsformen der vorliegenden Erfindung Telekommunikationsnetzwerke, die API-Betriebsmittel unterstützen.

BESCHREIBUNG DES STANDES DER TECHNIK

[0002] Telekommunikationsnetzwerk-Betreiber zögern oft, ihre Netzwerke für neue Gesellschaften zu öffnen, die kommunikationsbasierte Anwendungen anbieten. Ihr Zögern ist zumindest insofern etwas gerechtfertigt, als Netzwerkbetreiber riskieren, ihre eigenen Anwendungen auszuschlachten, insbesondere wenn der Preis zum Hauptunterschied zwischen ihren Anwendungen und denjenigen wird, die von neuen Gesellschaften angeboten werden. Tatsächlich laufen Telekommunikationsnetzwerk-Betreiber Gefahr, zu reinen "Bit-Transportern" zu werden, wenn sie nicht mit Anwendungen konkurrieren können, die von neuen Gesellschaften mit niedrigen Gemeinkosten angeboten werden.

[0003] Wenn Telekommunikationsnetzwerk-Betreiber ihre Netzwerke für neue Anwendungen und Gesellschaften öffnen, müssen Netzwerkbetreiber außerdem damit rechnen, die Kontrolle über ihre Netzwerke zu verlieren, indem sie Netzwerkanforderungen unterstützen müssen, die durch Anwendungen hervorgerufen werden, die von Dritten geschrieben und verwaltet werden, wie beispielsweise Diensteanbietern oder Software-Anbietern.

[0004] Obwohl ihr Zögern, die Netzwerke zu öffnen, gerechtfertigt sein mag, riskieren Telekommunikationsnetzwerk-Betreiber, wenn sie es nicht tun, lukrative Märkte zu verlieren, die neue Gesellschaften und Anwendungen schaffen können. Da die meisten Gesellschaften außerhalb der herkömmlichen Telekommunikationsnetzwerk-Grenzen operieren, können neue Anwendungen vollkommen neue Dienste schaffen, die Kunden anziehen, die das Netzwerk normalerweise nicht nutzen würden. Außerdem können staatliche Behörden, die den Wettbewerb des Telekom-Markts erweitern möchten, Netzwerkbetreiber dazu zwingen, ihre Netzwerke zu öffnen.

[0005] In einem Szenario basieren Transaktionen zwischen einem Netzwerk und Anwendungen vorteilhafterweise auf einem Set von Schnittstellen und Datentypen, die Anwendungsprogrammschnittstellen (APIs) bilden. Wenn diese APIs standardisiert und weitgehend zur Nutzung verfügbar sind, werden solche APIs als offene APIs bezeichnet. Offene APIs werden typischerweise von Software-Organisationen erstellt, die bei der Definition der APIs und der Be-

kanntmachung ihrer Verwendung sorgfältig vorgehen. Es gibt viele Sets von offenen APIs, zum Beispiel die Parlay/OSA APIs, die ursprünglich in der Parlay-Gruppe definiert wurden und im Zusammenhang mit 3GPP und ETSI standardisiert wurden. Die Parlay/OSA APIs, (wobei OSA für Open Service Architecture steht), bilden ein Set von neun orthogonalen Dienstleistungsfähigkeits-Funktionen (SCFs), von denen jede einen anderen Telekom-Bereich bearbeitet: Verbindungssteuerung, Benutzerinteraktion, Mobilität, Endgerätefähigkeiten, Datensitzungssteuerung, generische Mitteilungsübermittlung, Konnektivitäts-Management, Präsenz/Verfügbarkeit und Abrechnung.

[0006] Daher umfasst ein API-basiertes System des bisherigen Stands der Technik ein Set von offenen APIs, eine Gruppe von Benutzern, die Informationen über Server senden und empfangen, ein Telekommunikationsnetzwerk, das die Informationen und Anwendungen im Besitz von Dritten, die Informationen über die offenen APIs empfangen, transportiert, eine Anwendung basierend auf diesen APIs ausführt und antwortende Informationen über die offenen APIs an das Netzwerk zurück sendet.

[0007] Obwohl sie nützlich sind, sind API-basierte Systeme des bisherigen Stands der Technik bedeutenden Einschränkungen unterworfen. Erstens ist in API-basierten Systemen des bisherigen Stands der Technik eine Eins-zu-Eins-Beziehung zwischen offenen API-Servern und Anwendungen vorhanden. Um diese Einschränkung zu überwinden, haben API-basierte Systeme des bisherigen Stands der Technik eine Registrierungs- und Ermittlungs-Einheit aufgenommen, die von den offenen API-Servern verwendet wird, um sich selbst anzumelden, und von den Anwendungen verwendet wird, um zu ermitteln, welche APIs verfügbar sind. Wenn mehrere Registrierungs- und Ermittlungs-Einheiten vorhanden sind, besteht ein grundlegendes Problem darin, welche Registrierungs- und Ermittlungs-Einheit ein offener API-Server verwenden soll. Um dieses Problem zu vermeiden, könnte sich ein offener API-Server selbst bei allen verfügbaren Registrierungs- und Ermittlungs-Einheiten anmelden. Dies führt jedoch zu einem Problem, da zum Zuweisen eines offenen API-Servers zu mehreren Registrierungs- und Ermittlungs-Einheiten die Konfiguration des gesamten Systems bekannt sein muss, und diese ist schwierig zu bestimmen.

[0008] Ein weiteres Problem bei API-basierten Systemen des bisherigen Stands der Technik ist, nachdem eine Anwendung ermittelt hat, welchen offenen API-Server sie für ihre Dienste nutzen kann, dass die Beziehung statisch ist. Obwohl der ursprüngliche offene API-Server zu einem gewissen Zeitpunkt von Vorteil gewesen sein kann, könnte ein anschließender Vorfall, wie beispielsweise ein Ausfall eines offe-

nen API-Servers, riesige Probleme schaffen. In diesem Fall könnte die Anwendung, die einen bestimmten Dienst bereitstellt, nicht mehr ohne eine Wiederherstellungssitzung verwendet werden, die von der Anwendung initiiert wurde. Ein weiteres Problem von offenen API-Systemen des bisherigen Stands der Technik ist die Schwierigkeit, Dienstgütevereinbarungen durchzusetzen.

[0009] In Anbetracht der vorhergehenden (und weiteren) Einschränkungen wurde ein neues offenes API-basiertes System vorgeschlagen. Dieses System umfasst offene Anwendungen und offene API-Server, doch fügt das Telekommunikationsnetzwerk eine Proxy-Vorrichtung zwischen die offenen Anwendungen und die offenen API-Server ein. Dieser Proxy kann das Hochfahren zum Vermeiden von Überlappung abwickeln, bestimmen, welcher offene API-Server (bzw. welche) ein bestimmtes API-Ereignis oder einen Verfahrensauftruf abwickelt, Kommunikationslasten zwischen den verschiedenen offenen Anwendungen und offenen API-Servern im Gleichgewicht halten und Ereignisse zwischen den offenen Anwendungen und offenen API-Servern abfertigen.

[0010] Obwohl der Proxy ein vielversprechender Zusatz zu offenen API-basierten Systemen ist, scheitert der vorgeschlagene Proxy daran, andere vorhersehbare Probleme in erfolgreichen Systemen zu bearbeiten. Daher wäre ein Proxy, der andere Probleme in einem System auf offener API-Basis bearbeitet, von Vorteil. Ebenso vorteilhaft wäre ein Telekommunikationsnetzwerk mit einem Proxy, das andere Probleme eines offenen API-basierten Systems bearbeitet. Des Weiteren wäre ein Verfahren zum Betreiben eines Proxy vorteilhaft, das andere Probleme eines offenen API-basierten Systems bearbeitet. Ein computerlesbares Medium, das ein Computerprogramm speichert, welches einen Proxy betreibt, der andere Probleme eines offenen API-basierten Systems bearbeitet, wäre ebenfalls von Vorteil.

[0011] Die internationale Patentanmeldungsveröffentlichung WO 03/017619 A1 von Telefonaktiebolaget L.M. Ericsson und Ard-Jan Moerdijk offenbart einen sicheren Netzübergang mit Proxy-Dienstleistungsfähigkeits-Servern für Dienstgütevereinbarungsprüfungen. Insbesondere führt ein Framework Sicherheitsprüfungen an den Anforderungen von Anwendungen durch, auf einen oder mehrere externe Dienstleistungsfähigkeits-Server unter Verwendung von Dienstgütevereinbarungen zuzugreifen.

[0012] Der Artikel mit dem Titel "Design OSA/PAR-LAY Application Frameworks Using a Pattern of Language" von Wei Wu und anderen (International Conference on Communication Technology Proceedings (ICCT) 2003; Institute of Electrical and Electronics Engineers (IEEE), Band. 2, 9. April 2003, S. 1558–1561, XP010644263) offenbart Techniken für

eine verbesserte Anwendungsentwicklung unter Verwendung von Open Service Access (OSA)/Parlay. Insbesondere offenbart das Papier die Verwendung von Software-Muster- und -Framework-Technologien zum Aufbauen von OSA/Parlay-Anwendungs-Frameworks für den Einsatz bei einer einfacheren Anwendungsentwicklung.

[0013] Die US-amerikanische Patentanmeldungsveröffentlichung US 2002/0026473 A1 von Gourraud, die am 28. Februar 2002 veröffentlicht wurde, offenbart die Übertragung von Triggern, die auf einer Anwenderprogrammschnittstelle (API) basieren, auf einen Dienst-Manager bei Auftreten von vorbestimmten Ereignissen während Verbindungen. Insbesondere werden Informationen über Benutzer, die mit spezifischen Triggern verknüpft sind, von einer Benutzerprofil-Datenbank zur Verwendung während Verbindungen heruntergeladen. Der Dienst-Manager, der als Proxy zwischen den Anwendungen und den Netzwerkeinheiten dient, führt das Dienstinteraktions-Management in Reaktion auf die Trigger durch.

KURZDARSTELLUNG DER ERFINDUNG

[0014] Ein Verfahren und eine Vorrichtung gemäß der vorliegenden Erfindung werden in den selbstständigen Ansprüchen dargelegt, auf die der Leser im Folgenden verwiesen wird. Bevorzugte Merkmale werden in den Unteransprüchen dargelegt.

[0015] Die vorher genannten Nachteile, die mit dem bisherigen Stand der Technik verbunden sind, werden durch einen neuartigen Proxy bearbeitet und ein Telekommunikationsnetzwerk mit einem solchen Proxy, das die Einschränkungen von offenen API-basierten Systemen bearbeitet.

[0016] Die vorher genannten Nachteile, die mit dem bisherigen Stand der Technik verbunden sind, werden durch ein computerlesbares Medium bearbeitet, das ein Computerprogramm speichert, das einen Proxy betreibt und damit ein Telekommunikationsnetzwerk, so dass Einschränkungen von offenen API-basierten Systemen bearbeitet werden.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0017] Die Lehren der vorliegenden Erfindung lassen sich problemlos unter Berücksichtigung der folgenden ausführlichen Beschreibung in Verbindung mit den folgenden begleitenden Zeichnungen verstehen:

[0018] [Fig. 1](#) stellt eine Übersicht über ein API-basiertes System dar, das sich nicht in Übereinstimmung mit den Prinzipien der vorliegenden Erfindung befindet;

[0019] [Fig. 2](#) stellt eine allgemeine Ansicht eines

Proxy dar;

[0020] [Fig. 3A](#) stellt ein System zum Implementieren einer Dienstleistungsvertragssteuerung dar;

[0021] [Fig. 3B](#) stellt ein Ablaufdiagramm der Implementierung einer Dienstleistungsvertragssteuerung dar;

[0022] [Fig. 4A](#) stellt ein geografisch verteiltes offenes API-basiertes System dar;

[0023] [Fig. 4B](#) stellt die Registrierung von offenen API-Einheiten dar;

[0024] [Fig. 4C](#) stellt einen Proxy dar, der ein offenes API-System überwacht; und

[0025] [Fig. 5](#) stellt ein alternatives API-basiertes System dar, das sich in Übereinstimmung mit den Prinzipien der vorliegenden Erfindung befindet.

[0026] Zur Erleichterung des Verständnisses wurden, wo möglich, identische Bezugszeichen verwendet, um identische Elemente zu bezeichnen, die den Figuren gemeinsam sind.

AUSFÜHRLICHE BESCHREIBUNG

[0027] Die vorliegende Erfindung betrifft neuartige API-basierte Verfahren, Vorrichtungen, computerlesbare Medien und Systeme, die einen Proxy zwischen Anwendungen und offenen API-Server integrieren. Offene API-Systeme in Übereinstimmung mit der vorliegenden Erfindung können zwei Typen von Informationsströmen bearbeiten, diejenigen, die in dem Netzwerk ihren Ursprung haben und an die Anwendung weitergeleitet werden, und diejenigen, die von den Anwendungen zu dem Netzwerk fließen. Obwohl beide Ströme durch die APIs übermittelt werden, wird der erste Typ im Folgenden als ein Ereignis bezeichnet und der zweite wird als ein Verfahrensauftrag bezeichnet.

[0028] [Fig. 1](#) stellt ein offenes API-basiertes System **100** dar, das sich in Übereinstimmung mit den Prinzipien der vorliegenden Erfindung befindet. Das System **100** umfasst Anwendungen **102** und offene API-Server **104**, die Informationen über offene Anwenderprogrammschnittstellen weitergeben. Die Anwendung **102** kann ein Programm (bzw. Programme), Unternehmensanwendungssysteme oder ein anderes Betriebsmittel sein, das unter Verwendung von APIs arbeitet, die auf Netzwerkbetriebsmittel zugreifen. Die offenen API-Server **104** sind Kommunikationsknoten, die mit jeder von einer umfangreichen Bandbreite von Benutzervorrichtungen verbunden sind, wie beispielsweise Computern **106**, Handapparaten **108** oder Telefonensystemen **110**. Demzufolge sollten Benutzervorrichtungen als jede Vorrichtung

verstanden werden, die eine Anwendung **102** auszuführen sucht. Von einer Anwendung **102** angebotene Dienste werden implementiert, wenn eine der Benutzervorrichtungen einen offenen API-Server **104** kontaktiert, der dann in der im Folgenden beschriebenen Weise Informationen an die Anwendung **102** über gibt.

[0029] Wie gezeigt, befindet sich zwischen den offenen API-Servern **104** und den Anwendungen **102** ein Proxy **700**. Die offenen API-Server **104** und der Proxy **700** könnten Bestandteil eines gemeinsamen Netzwerks **103** sein. Des Weiteren, wobei die Anwendungen **102** auf einer Anwendungsebene gezeigt sind, ist es möglich, dass eine oder mehrere Anwendungen **102** sich auch im Besitz des Besitzers des Netzwerks **103** befinden oder von diesem betrieben werden. Wie im Folgenden ausführlicher beschrieben wird, bearbeitet der Proxy **700** Kommunikationen zwischen den offenen API-Servern **104** und den Anwendungen **102** transparent. Zu diesem Zweck authentifizieren die Anwendungen **102** ihre Anwesenheit zunächst an dem Proxy **700** und registrieren sie dann, was durch die Anwendungen **102** vorgenommen wird, die mit einer Registrierungs- und Ermittlungs-Vorrichtung **114** Kontakt aufnehmen, welche die Registrierungsinformationen, (wie beispielsweise IP-Adressen, Startbedingungen, geografische Standorte, zulässige APIs usw.), empfängt, – oder von einem Authentifikationsmechanismus aus bestimmen kann, – die von dem Proxy **700** angefordert werden, um die Anwendung **102** zu identifizieren, zu akzeptieren und zu verwenden.

[0030] Die Registrierung von Anwendungen **102** findet primär statt, wenn die Anwendungen **102** ihre Startbedingungen anmelden. Nachdem eine Anwendung **102** authentifiziert worden ist, kann sie die Dienste ermitteln, auswählen und zu nutzen beginnen, die von den offenen API-Servern **104** angeboten werden. Durch die Registrierung bestimmt der Proxy **700** die Fähigkeiten aller offenen API-Server **104** und kombiniert diese Fähigkeiten anschließend, um ein übergeordnetes Set von Fähigkeiten zu bilden. Der Proxy **700** registriert anschließend die Fähigkeiten des übergeordneten Sets in der Registrierungs- und Ermittlungs-Vorrichtung **114**, wodurch es der Registrierungs- und Ermittlungs-Vorrichtung **114** ermöglicht wird, ein generischeres Set von Diensten für authentifizierte Anwendungen **102** anzubieten, die Dienste ermitteln. Dies bietet auch den offenen API-Servern **104** die Möglichkeit, sich nur an einem einzigen Standort zu registrieren, und bietet auch eine zentrale Stelle zum Überwachen und Verwalten des offenen API-Netzwerks, das im Folgenden ausführlicher beschrieben wird.

[0031] [Fig. 2](#) stellt ein Blockschaltbild höchster Ebene einer Ausführungsform eines Proxy **700** dar. Der Proxy **700** umfasst einen Prozessor **710** sowie einen

Computerspeicher **720** zum Speichern von Steuerprogrammen **725** und Datenstrukturen **727**. Der Prozessor **710** wirkt mit einer herkömmlichen Unterstützungsschaltung **730**, wie beispielsweise Stromversorgungen, Taktorschaltkreisen, Cache-Speicher und dergleichen sowie Schaltkreisen zusammen, die das Ausführen der Software-Routinen unterstützen, die in dem Speicher **720** gespeichert sind. Daher wird in Erwagung gezogen, dass einige der Prozessschritte, die hierin als Software-Prozesse erläutert werden, in Hardware implementiert werden können, zum Beispiel als Schaltung, die mit dem Prozessor **710** zum Durchführen verschiedener Schritte zusammenwirkt. Der Proxy **700** enthält ebenfalls die Eingabe-Ausgabe-Schaltung **740**, die eine Schnittstelle zu dem gesamten Telekommunikationsnetzwerk, den Anwendungen **102** und den offenen API-Servern **104** bildet.

[0032] Obwohl der Proxy **700** als ein Mehrzweck-Computer dargestellt ist, der so programmiert ist, dass er verschiedene Steuerfunktionen in Übereinstimmung mit der vorliegenden Erfindung durchführt, kann die Erfindung in Hardware implementiert werden, zum Beispiel als eine anwendungsspezifische integrierte Schaltung (ASIC). Daher sollen die hierin beschriebenen Prozessschritte allgemein so interpretiert werden, dass sie äquivalent von Software, Hardware oder einer Kombination davon durchgeführt werden. Des Weiteren umfasst der Proxy **700**, wenn er in Software oder Firmware implementiert ist, ein computerlesbares Medium **750**, das Informationen speichert, die von dem Prozessor **710** ausgeführt werden können und/oder auf die dieser zugreifen kann. Gleichgültig, ob er in Hardware oder Software oder in einer Kombination aus Hardware/Software implementiert ist, arbeitet der Proxy **700**, um die im Folgenden beschriebenen Funktionen zu erfüllen.

[0033] Eine erste Funktion, die vom Proxy **700** erfüllt wird, besteht darin, Dienstverträge zwischen zwei oder mehreren der offenen API-Server **104**, den Benutzern, den Anwendungen **102** und den Telekommunikationsnetzwerk-Betreibern durchzusetzen. Beispiele für die Parameter, die Teil eines Dienstvertrags sein könnten, umfassen die Betriebsmittelnutzung, die durch einen offenen API-Server **104** gestattet wird, z.B. die maximale Anzahl von Verbindungen, die maximale Anzahl von Verbindungs-Streckenabschnitten, die pro Verbindung zulässig sind, die maximale Anzahl von offenen Briefkästen, Abrechnung, Zeitbegrenzungen und APIs, die zulässig sind. In einigen Ausführungsformen verteilt der Proxy **700** Dienstverträge dynamisch auf die offenen API-Server **104**.

[0034] [Fig. 3A](#) stellt ein System **200** zum Implementieren einer Dienstvertragssteuerung über den Proxy **700** dar, während [Fig. 3B](#) ein Ablaufdiagramm des Betriebs des Systems **200** darstellt. Im Folgenden wird auf beide dieser Figuren verwiesen. Das System

200 umfasst eine Registrierungs- und Ermittlungs-Vorrichtung **114**, die betriebsfähig mit einer Unternehmens-Benutzeroberfläche **202** und mit dem Proxy **700** über eine Datenbank **204** verbunden ist. Es sollte jedoch klar sein, dass die Datenbank **204** optional ist: in einigen Ausführungsformen ist die Datenbank **204** nicht integriert. Unter folgender Bezugnahme auf [Fig. 3B](#) beginnt ein Verfahren **245** einer Dienstvertragssteuerung in Schritt **247** und fährt mit Schritt **248** fort, indem Dienstvertragssteuerungs-Informationen erhalten und registriert werden. Dazu stellt die Unternehmens-Benutzeroberfläche **202** die Bedingungen eines Dienstvertrags als ein Set von Steuerparametern für die Registrierungs- und Ermittlungs-Vorrichtung **114** bereit, welche in Schritt **250** die Steuerparameter in der Datenbank **204** speichert. Als Nächstes ruft der Proxy **700** in Schritt **251** die Steuerparameter ab und berechnet in Schritt **252**, wie die globalen Vertragsinformationen als lokale Verträge zu den relevanten offenen API-Servern **104**, (denjenigen, die von den Bedingungen des Dienstvertrags betroffen sind), zugewiesen werden sollen. Dann werden in Schritt **253** die lokalen Verträge zu den offenen API-Servern **104** gesendet, indem der Proxy **700** in Schritt **254** einzelnen offenen API-Servern **104** ihre Implementierungsparameter sendet.

[0035] Nachdem die offenen API-Server **104** ihre Implementierungsparameter haben, kann ein Trigger in Schritt **255** veranlassen, dass ein lokaler offener API-Server **104** bestimmt, dass sein lokaler Dienstvertrag fehlerhaft ist. Wenn zum Beispiel ein lokaler offener API-Server **104** auf 10 Verbindungen beschränkt ist, kann dieser offene API-Server **104**, wenn er 8 Verbindungen bearbeitet, einen Trigger setzen, um anzugeben, dass er unter Umständen eine Berechtigung benötigt, mehr als 10 Verbindungen zu bearbeiten. In diesem Fall sendet der lokale offene API-Server **104** in Schritt **256** eine Anforderung an den Proxy **700** und fordert eine Modifizierung seines Dienstvertrags an. Indessen hat der Proxy **700** in Schritt **257** die offenen API-Server **104** überwacht, um solche Anforderungen zu identifizieren. Nachdem in Schritt **258** eine Anforderung für eine Modifizierung empfangen worden ist, fragt der Proxy **700** in Schritt **258** die relevanten offenen API-Server nach ihrer gegenwärtigen lokalen Nutzung ab. Sobald diese Nutzung bestimmt worden ist, wird zur erneuten Berechnung der lokalen Verträge der relevanten offenen API-Server eine Schleife zurück zum Schritt **252** gebildet. Auf diese Weise kann der Proxy **700** Betriebsmittel auf der Basis der gegenwärtigen Nutzung zuweisen, wobei er innerhalb des Dienstvertrags bleibt.

[0036] Es sollte klar sein, dass jede Einheit unterschiedlich auf Dienstvertragsbedingungen reagieren kann. Zum Beispiel kann der Dienstvertrag für den offenen API-Server **104c** spezifizieren, dass keine zeitkritischen Informationen zu dem oder von dem offe-

nen API-Server **104c** übergeben werden sollen. Demzufolge kann der Proxy **700** APIs für die anderen offenen API-Server **104a** und **104b** eine höhere Priorität einräumen.

[0037] Nachdem die Einheiten mit Implementierungsparametern versorgt worden sind, implementieren die verschiedenen Einheiten in Schritt **255** ihre Teile der Vertragsbedingungen. Wenn der offene API-Server **104** bestimmt, dass für seinen lokalen Dienstvertrag für ein bestimmtes Set von Anwendungen eine Aktualisierung erforderlich ist, kann dieser offene API-Server **104** den Proxy abfragen und im Verfahren **256** eine Aktualisierung des Vertrags anfordern. Wenn der Dienstvertrag für das Set von Anwendungen endet, dann endet das Verfahren in Schritt **257**. Wenn zum Beispiel ein Benutzer einen offenen API-Server **104** kontaktiert, sucht dieser Server seine Dienstvertragsparameter und bestimmt, ob er die bestimmte Verbindung oder den Dienst bearbeiten kann, der von dem Benutzer **206** angefordert worden ist. Wenn der offene API-Server **104a** zum Beispiel eine maximale Anzahl von Verbindungen hat, können alle diese Anzahl übersteigenden Verbindungen zurückgewiesen oder zu einem anderen offenen API-Server **104** umgeleitet werden.

[0038] Da das offene API-Modell immer beliebter wird, wird sich die Anzahl der offenen API-Einheiten erhöhen. Dadurch entstehen für das Modell Verwaltungs- und Integrationsprobleme. Dies stellt eine besondere Herausforderung dar, da ein offenes API-System aus verschiedenen Einheiten bestehen kann, die sich an äußerst verschiedenen geografischen Standorten befinden können. Da das Verhalten einer offenen API-Einheit überwacht werden muss, um eine systemübergreifende Übereinstimmung sicherzustellen, und im Bedarfsfall entsprechende Korrekturmaßnahmen ergriffen werden müssen, stellen verschiedene geografische Standorte ein Problem dar. Korrekturmaßnahmen sind ein Hauptproblem, da systemübergreifende Änderungen in einem großen Netzwerk äußerst schwierig, zeitaufwändig und teuer sein können. Der Proxy **700** stellt eine effektive Lösung für solche Probleme bereit. Des Weiteren kann der Proxy **700** dies auf eine transparente Weise tun.

[0039] [Fig. 4A](#) stellt eine geografische Abbildung von Europa dar, in der verschiedene offene API-Server **104** über den Kontinent verteilt sind. Alle diese offenen API-Server werden von dem gleichen Proxy **700** verwaltet. Es sollte klar sein, dass der Proxy **700** keine einzelne Einheit sein muss, sondern aus einem Netzwerk von untereinander verbundenen Vorrichtungen, Systemen und Netzwerken bestehen kann, die zusammenwirkend den Proxy **700** bilden. Die zentrale Verwaltung dieses Proxy **700** vereinfacht systemübergreifende Änderungen in hohem Maße, da diese Änderungen durch den Proxy **700** verbreitet

werden können.

[0040] [Fig. 4B](#) stellt dar, wie der Proxy **700** zur zentralen Einheit wird. In Schritt **505** akzeptiert der Proxy **700** Registrierungen und Registrierungs-Aufhebungen von den offenen API-Servern **104**. In Schritt **507** bestimmt der Proxy **700** die Dienstleistungsfähigkeiten aller offenen API-Server **104** und bestimmt dann ein übergeordnetes Set dieser Fähigkeiten. Der Proxy **700** registriert anschließend dieses übergeordnete Set bei allen bekannten Registrierungs- und Ermittlungs-Vorrichtungen **114**. Indem er als ein zentraler Registrierungspunkt arbeitet, erfährt der Proxy **700** den Status aller offenen API-Server **104**, Registrierungs- und Ermittlungs-Vorrichtungen **114** und Anwendungen in dem offenen API-Netzwerk. Dies ermöglicht es dem Proxy **700**, als Verwaltungs- und Integrations-Einrichtung des gesamten Systems zu arbeiten.

[0041] [Fig. 4C](#) stellt ein Verfahren **301** dar, wie der Proxy **700** das System verwalten und integrieren kann. Wie gezeigt, beginnt das Verfahren **301** in Schritt **303** und fährt mit Schritt **315** fort, indem der Betrieb des gesamten offenen API-Systems überwacht wird. Wenn in Schritt **317** bestimmt wird, dass das offene API-System **200** korrekt funktioniert, kehrt das Verfahren zu Schritt **315** zurück, wobei die Überwachung des gesamten Betriebs weiterhin fortgesetzt wird. Wenn in Schritt **317** jedoch eine Abweichung erfasst wird, fährt der Proxy **700** mit Schritt **319** fort, in dem eine geeignete Korrekturmaßnahme ergriffen wird. Geeignete Maßnahmen können das erneute Starten einer Registrierungs- und Ermittlungs-Vorrichtung **114**, das Informieren von Anwendungen **102** über Probleme, die im Netzwerk erfasst wurden, und/oder das erneute Zuweisen von Startbedingungen zu offenen API-Servern **104** umfassen, wenn sie wieder aktiv werden. Es ist zu beachten, dass das Informieren der Anwendungen **102** über Abweichungen verzögert oder nicht ausgeführt werden kann, wenn der Proxy alternative offene API-Server **104** findet, um die Integrität des offenen API-Systems **200** zu gewährleisten.

[0042] Eine wichtige Korrekturmaßnahme, die der Proxy ergreifen kann, besteht darin, als eine Firewall zu arbeiten, um ungeeignete oder gefährliche APIs daran zu hindern, sich über das offene API-System zu verbreiten. Dies gilt insbesondere dann, wenn eine Registrierungs- und Ermittlungs-Vorrichtung **114** innerhalb der Domäne eines fremden Netzwerkbetreibers arbeitet, der seine Netzwerk-Betriebsmittel für Dritte öffnen möchte, die APIs entwickeln könnten, die lokale oder systemweite Zusammenbrüche verursachen könnten. Den Proxy **700** als ein Filter arbeiten zu lassen, um solche APIs zu sperren, ist besonders vorteilhaft.

[0043] Wie oben erläutert, erkennt der Proxy **700**

die anderen offenen API-Systemeinheiten und kann einen speziellen offenen API-Server **104** direkt zu einer speziellen Anwendung **102** zuweisen, nicht nur anfänglich, sondern dynamisch. Zuweisungen können geändert werden, um Netzwerkbedingungen widerzuspiegeln, wie beispielsweise Hinzufügen von Anwendungen **102** und offenen API-Servern **104**, oder als ein Ergebnis von Systemausfällen. Diese dynamische Zuweisungsfähigkeit erhöht die Netzwerkverfügbarkeit beträchtlich.

[0044] Obwohl das Vorgenannte vermuten lässt, dass alle Kommunikationen zwischen den Anwendungen **102** und den offenen API-Servern **104** durch den Proxy **700** verlaufen, ist solches nicht erforderlich. [Fig. 5](#) stellt ein offenes API-System **400** dar, in welchem gewisse API-Verbindungen den Proxy **700** umgehen und direkt zwischen dem offenen API-Server **104b** und der Anwendung **102b** verlaufen. Wie gezeigt, wird eine spezielle Verbindung, die als IpCall bezeichnet wird, direkt auf Leitungen **401** übertragen. Andere API-Verbindungen werden jedoch vom Proxy **700** bearbeitet. Eine direkte Weiterleitung kann verhindern, dass der Proxy **700** zum Leistungsengpass wird.

[0045] [Fig. 5](#) zeigt des Weiteren Software- (oder Hardware-) Prozesse, die von dem Proxy **700** durchgeführt werden. Solches umfasst Dienstvertrags-Software **402**, welche die vorher beschriebenen Dienstverträge bearbeitet, Lastausgleichs-Software **404**, die ein Gleichgewicht der Kommunikationslasten zwischen den offenen API-Servern **104** und den verschiedenen Anwendungen **102** herstellt, Verwaltungs-Software **406**, welche die offenen API-Einheiten wie vorher beschrieben verwaltet, und einen Ereignis-Verteiler **408**, der die Abfertigung der verschiedenen API-Verbindungen zwischen den offenen API-Servern **104** und den Anwendungen steuert. Der Vollständigkeit halber zeigt [Fig. 5](#) auch die Registrierungs- und Ermittlungs-Vorrichtung **114** und die Datenbank **204**.

[0046] Wenn er richtig konfiguriert ist, ist der Proxy **700** transparent: ein offener API-Server **104** muss nicht wissen, dass er über den Proxy **700** kommuniziert, und eine Anwendung **102** muss nicht wissen, dass sie über den Proxy **700** kommuniziert, außer eventuell während der ersten Registrierung. Der Proxy **700** kann die richtigen Hochfahrbedingungen bereitstellen, entscheiden, welche konkurrierende Anwendung ein Ereignis empfängt, wenn mehrere Anwendungen verfügbar sind, (als Verkehrspolizist arbeiten), Verzögerungen in Weiterleitungs-Ereignisse einfügen, um potenzielle Probleme zu vermeiden, zentrale Verwaltungsfunktionen bereitstellen, Dienstverträge implementieren, das Netzwerk verbergen, es vorziehen, ausstehende oder neue Anforderungen zu anderen Betriebsmitteln weiterzuleiten, (wodurch die Netzwerkverfügbarkeit erhöht wird), als

eine Firewall arbeiten, um die Integrität des Systems aufrecht zu erhalten, Informationen von mehreren Betriebsmitteln sammeln und die Informationen nach Erfordernis und/oder zum richtigen Zeitpunkt verteilen, systemübergreifende Änderungen und ihre Integration erleichtern, und Anwendungen **102** und offene API-Server **104** zentral überwachen.

[0047] Obwohl sich das Vorgenannte auf Ausführungsformen der vorliegenden Erfindung bezieht, sind andere und weitere Ausführungsformen der Erfindung vorstellbar, ohne von ihrem Grund legenden Umfang abzuweichen, und ihr Grund legender Umfang wird durch die folgenden Ansprüche bestimmt.

Patentansprüche

1. Vorrichtung, umfassend:
einen offenen API-Server (**104**) zum Kommunizieren mit einer Benutzervorrichtung (**106, 108, 110**) und zum Senden und Empfangen von Anwendungsprogrammschnittstellenbefehlen; und
GEKENNZEICHNET DURCH:
einen Proxy (**700**) zum Empfangen von Anwendungsprogrammschnittstellenbefehlen von dem offenen API-Server (**104**), zum Senden von empfangenen Anwendungsprogrammschnittstellenbefehlen von dem offenen API-Server (**104**) zu einer Anwendung (**102**), zum Empfangen von Antworten von einer Anwendung (**102**) und zum Senden der empfangenen Antworten an den offenen API-Server (**104**);
wobei der Proxy (**700**) Dienstvertrag-Implementierungsparameter an den offenen API-Server (**104**) sendet; und
wobei der offene API-Server (**104**) das Senden von Anwendungsprogrammschnittstellenbefehlen basierend auf den Dienstvertrag-Implementierungsparametern steuert.

2. Vorrichtung nach Anspruch 1, wobei die Dienstvertrag-Implementierungsparameter sich auf einen Dienstvertrag beziehen.

3. Vorrichtung nach Anspruch 1, des Weiteren umfassend eine Registrierungs- und Ermittlungsvorrichtung (**114**), welche die Steuerparameter empfängt.

4. Vorrichtung nach Anspruch 1, des Weiteren umfassend ein computerlesbares Medium (**750**) zum Speichern von Programminformationen, die wenigstens teilweise den Proxy (**700**) steuern, um die Dienstvertrag-Implementierungsparameter zu erzeugen.

5. Vorrichtung nach Anspruch 1, wobei, basierend auf Dienstrutzung, der offene API-Server (**104**) modifizierte Dienstvertrag-Implementierungsparameter anfordert.

6. System, umfassend:

ein Telekommunikationsnetzwerk (103);
einen offenen API-Server (104) zum Senden und Empfangen von Anwendungsprogrammschnittstellenbefehlen auf dem Telekommunikationsnetzwerk (103); und

GEKENNZEICHNET DURCH:

einen Proxy (700) zum Empfangen und Senden von Anwendungsprogrammschnittstellenbefehlen von dem offenen API-Server (104) und zum selektiven Senden von Befehlen an und Empfangen von Befehlen von wenigstens einer ersten Anwendung (102) oder einer zweiten Anwendung (102);

wobei der Proxy (700) den Zustand des Systems überwacht; und

wobei der Proxy (700) die wenigstens erste Anwendung (102) oder zweite Anwendung (102) basierend auf dem Zustand des Systems dynamisch auswählt.

7. System nach Anspruch 6, wobei der Proxy (700) seine Auswahl basierend auf einer Veränderung des Systems ändert.

8. System nach Anspruch 6, wobei wenigstens ein offener API-Server (104) den Proxy (700) umgeht und einen Anwendungsprogrammschnittstellenbefehl direkt an eine Anwendung (102) sendet, um damit zu verhindern, dass der Proxy (700) ein Kommunikationsengpass ist.

9. System nach Anspruch 8, wobei der Proxy (700) vorgegebene Anwendungsprogrammschnittstellenbefehle für eine Propagierung sperrt.

10. Verfahren zum Betreiben eines Telekommunikationsnetzwerks (103), umfassend:

Erhalten (251) von Dienstvertragsbedingungen, dadurch gekennzeichnet, dass das Verfahren des Weiteren auch folgende Schritte umfasst:

Verarbeiten (252) der Dienstvertragsbedingungen zum Entwickeln von Implementierungsparametern für eine Vielzahl von offenen API-Servern (104);
Senden (253) von Implementierungsparametern an die Vielzahl von offenen API-Servern (104), wobei die zu jedem offenen API-Server (104) gesendeten Implementierungsparameter diesen offenen API-Server (104) anweisen, lokale Dienstvertragsbedingungen zu implementieren;

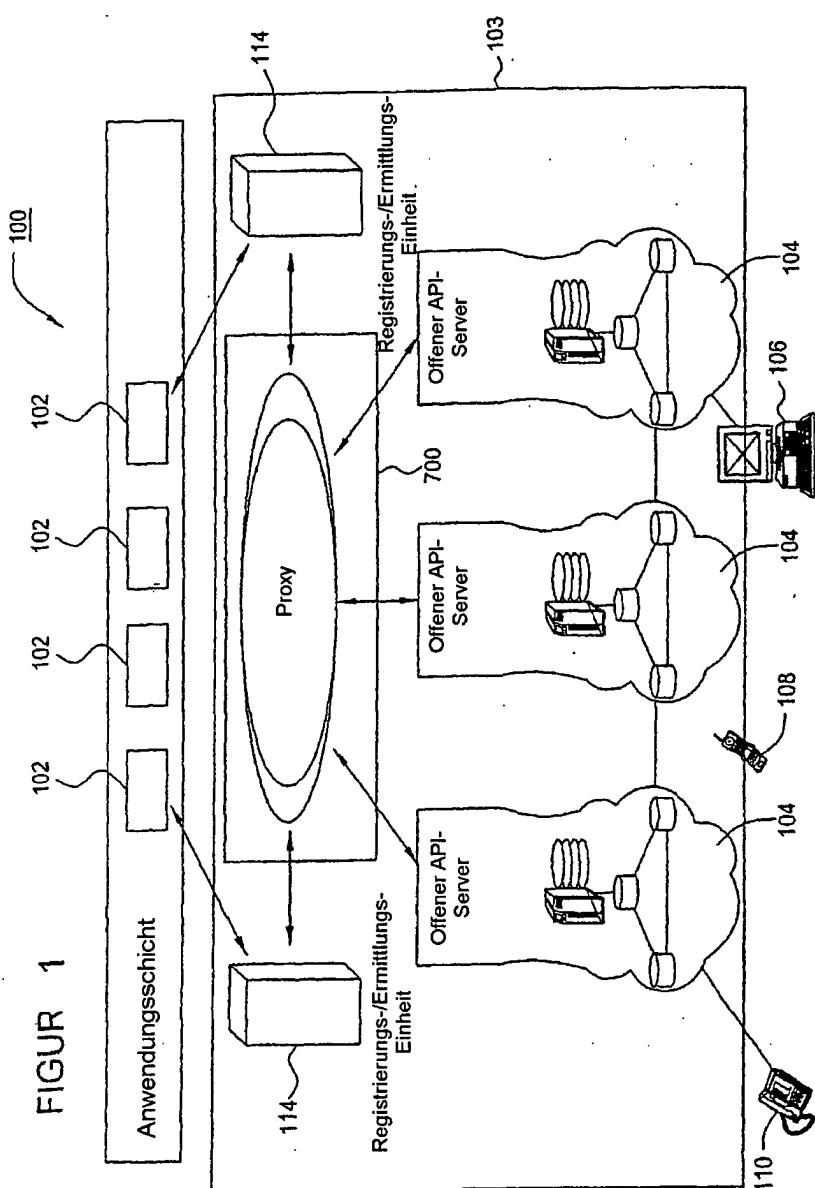
Senden und Empfangen (254, 257) von Anwendungsprogrammschnittstellenbefehlen von der Vielzahl von offenen API-Servern (104); und

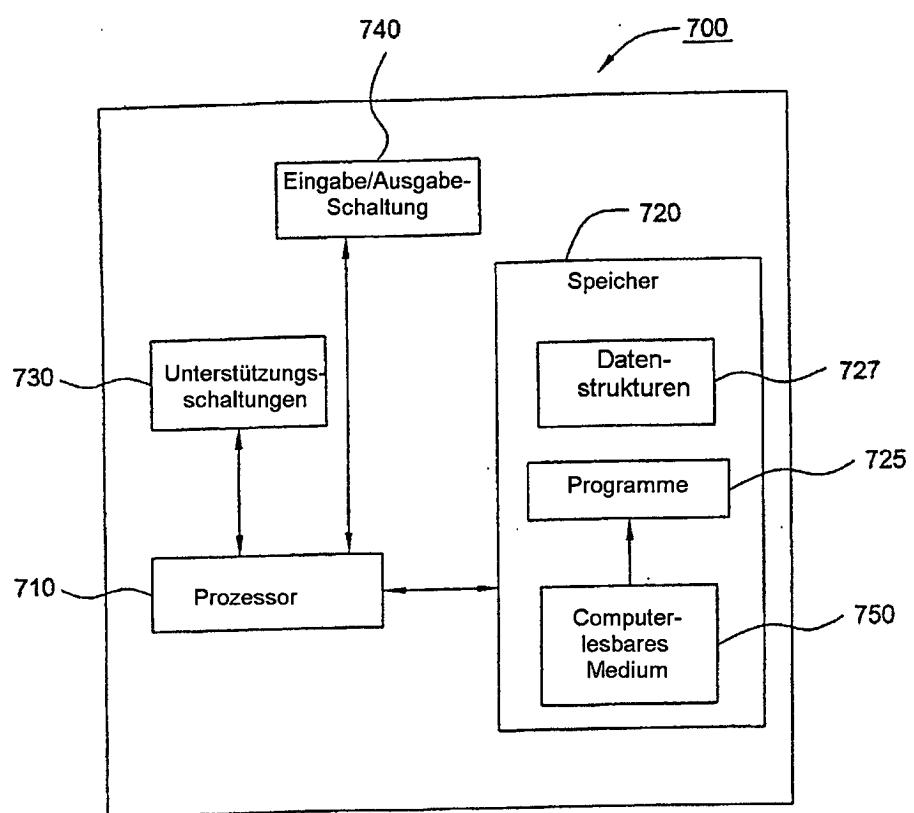
Übergeben von gesendeten und empfangenen Anwendungsprogrammschnittstellenbefehlen an wenigstens eine Anwendung (102);

wobei jeder offene API-Server (104) Anwendungsprogrammschnittstellenbefehle nur in Übereinstimmung mit seinen lokalen Dienstvertragsbedingungen sendet.

Es folgen 8 Blatt Zeichnungen

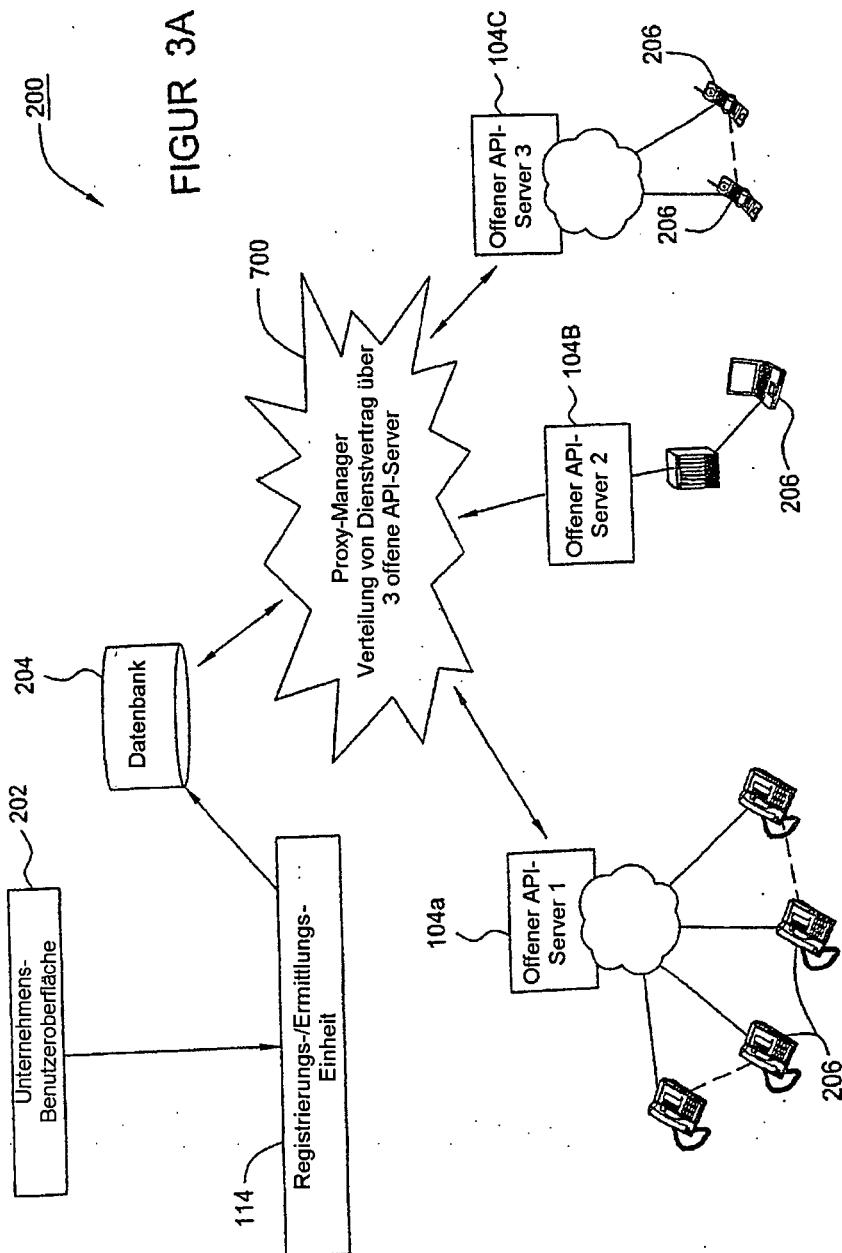
Anhängende Zeichnungen

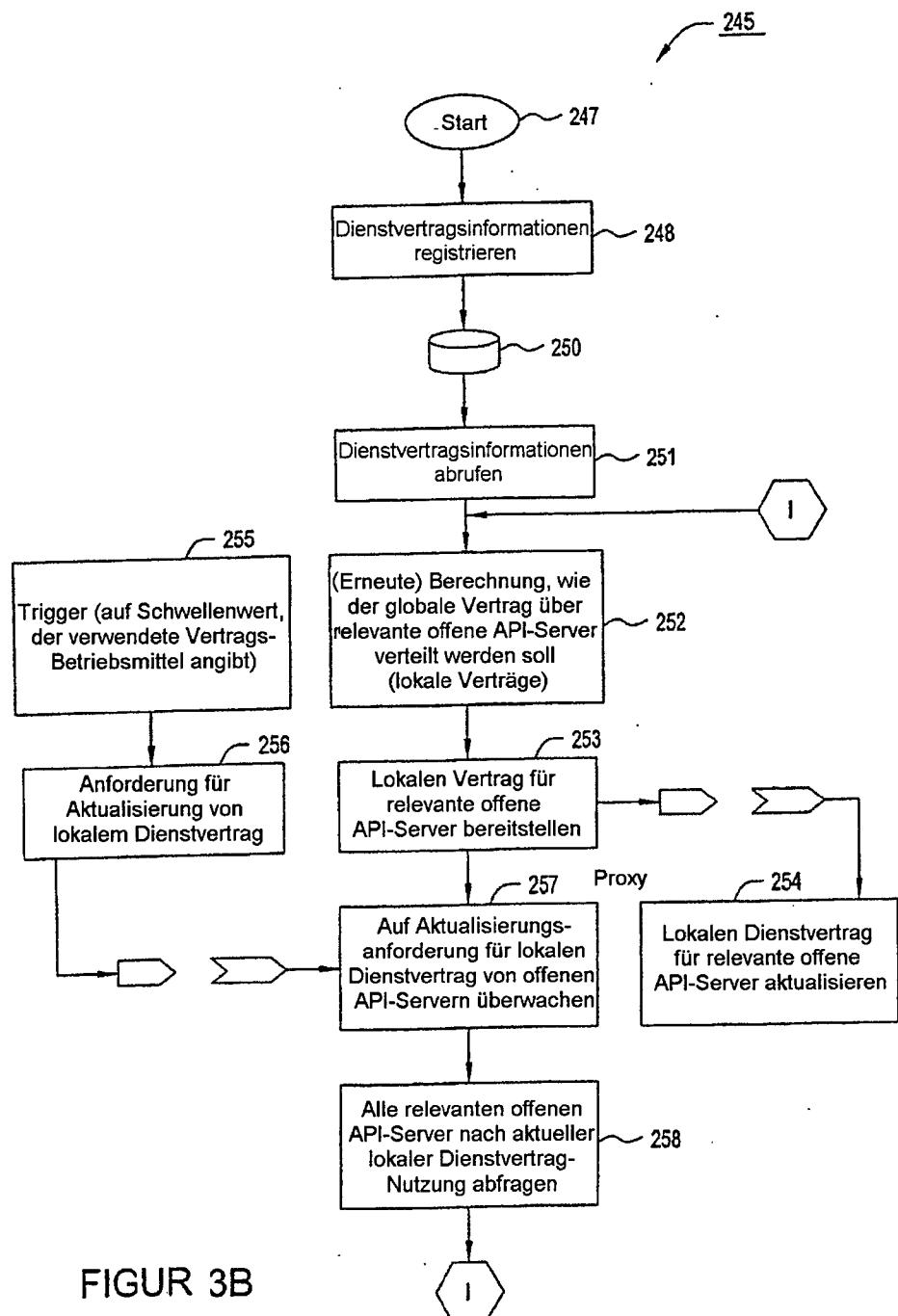




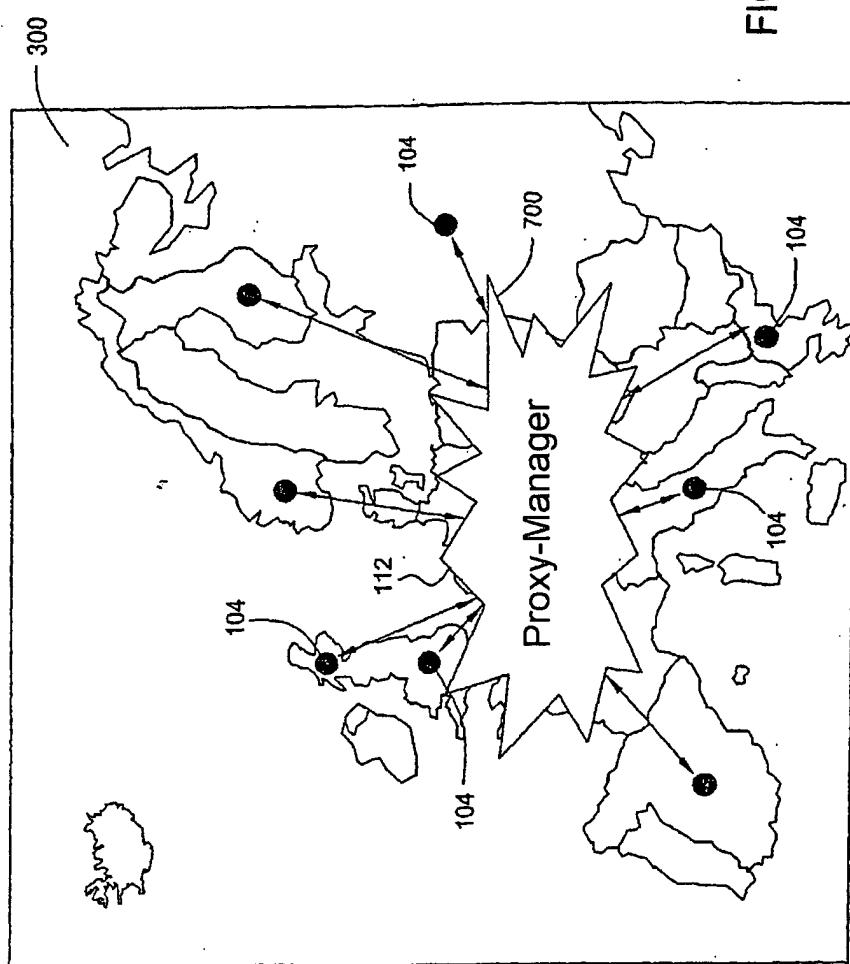
FIGUR 2

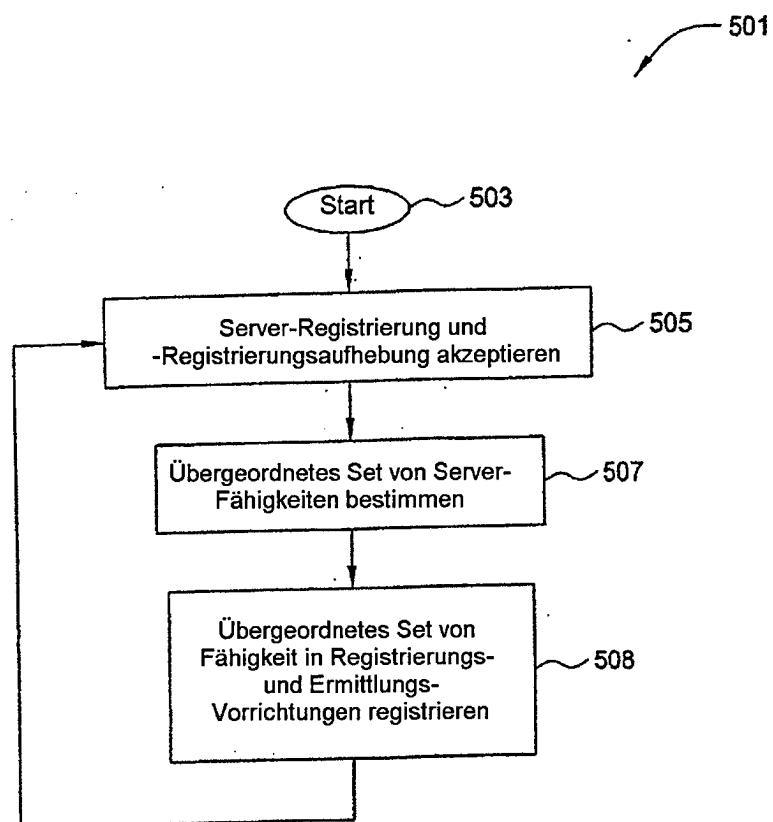
FIGUR 3A



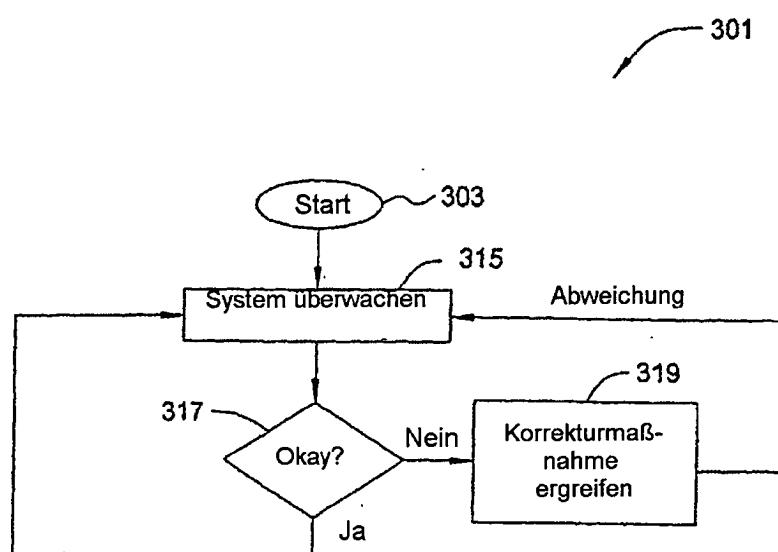


FIGUR 4A

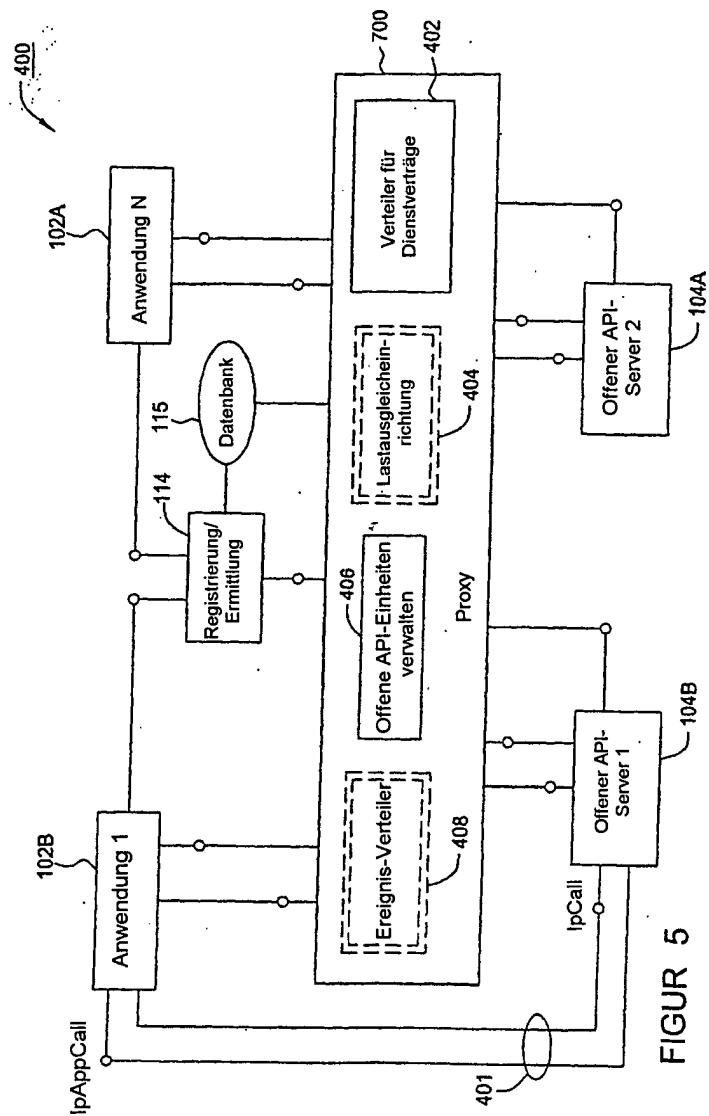




FIGUR 4B



FIGUR 4C



FIGUR 5