(19) SCHWEIZERISCHE EIDGENOSSENSCHAFT
EIDGENÖSSISCHES INSTITUT FÜR GEISTIGES EIGENTUM

(11) CH 702 260 B1

(51) Int. Cl.: G06F 9/44 (2006.01)

Erfindungspatent für die Schweiz und Liechtenstein
Schweizerisch-liechtensteinischer Patentschutzvertrag vom 22. Dezember 1978

(12) PATENTSCHRIFT

(21) Anmeldenummer: 00730/11

(22) Anmeldedatum: 08.09.2009

(43) Anmeldung veröffentlicht: 11.03.2010

(30) Priorität: 08.09.2008 US 61/095,009

(24) Patent erteilt: 30.06.2014

(45) Patentschrift veröffentlicht: 30.06.2014

(73) Inhaber:
CREDIT SUISSE SECURITIES (USA) LLC,
11 Madison Avenue
New York, NY 10010 (US)

(72) Erfinder:
Robin Heizmann, 75116 Paris (FR)
Toni Farruggia, 8810 Horgen (CH)

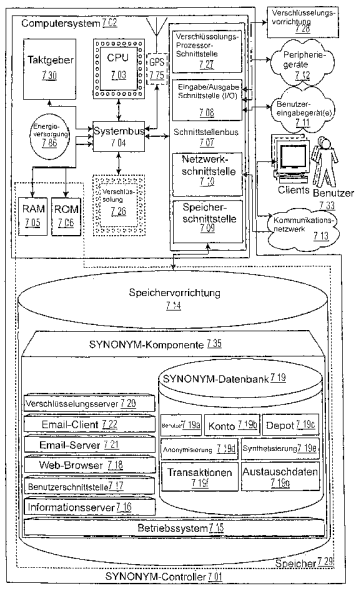
(74) Vertreter:
E. Blum & Co. AG Patent- und Markenanwälte VSP,
Vorderberg 11
8044 Zürich (CH)

(86) Internationale Anmeldung:
PCT/US 2009/056278

(87) Internationale Veröffentlichung:
WO 2010/028395

(54) Umgebungsentwicklungsvorrichtung.

(57) Die Erfindung betrifft eine Umgebungsentwicklungsvorrichtung umfassend einen Speicher, einen sich mit dem Speicher in Kommunikation befindenden Prozessor, welcher derart konfiguriert ist, dass er eine Vielzahl von im Speicher gespeicherten Verarbeitungsinstruktionen ausgibt. Der Prozessor gibt Instruktionen aus zum Empfangen einer Anfrage zur Erzeugung einer Entwicklungsumgebung, Identifizieren reeller Datensätze basierend auf die Anfrage, Erzeugen variierter Daten aus den reellen Daten und Generieren einer synthetischen Entwicklungsumgebung. Die Generierung umfasst weiter Instruktionen zum Instantiieren von Benutzerkonten, Bevölkern der synthetischen Entwicklungsumgebung mit einem Profil von Testkomponenten, Laden der variierten Daten in die synthetische Entwicklungsumgebung, und Verteilen der synthetischen Entwicklungsumgebung zu einem in der Anfrage angegebenen Standort.



Beschreibung

Zugehörige Anwendungen

[0001] Die vorliegende Anmeldung betrifft eine Umgebungsentwicklungsvorrichtung nach Anspruch 1.

[0002] Die vorliegende Anmeldung umfasst Ansprüche zur Verhinderung von nicht autorisierten Verwendungen von Daten einschliesslich der Verhinderung von Piraterie, der Verletzung der Privatsphäre, oder der nicht autorisierten Datenmodifikation (vorgeschlagene Klasse/Unterklasse: 726/26). Um aber ein Verständnis des Lesers im Hinblick auf die Beschreibungen der anderen Implementierungen zu entwickeln, wurden die Implementierungen in eine einzige Anwendung zusammengefasst, um darzustellen und klarzustellen, wie Aspekte dieser Implementierungen unabhängig voneinander arbeiten, wie zwischen einzelnen Ausführungsformen miteinander arbeiten und/oder kollektiv zusammenarbeiten. Weiter beschreibt die Anmeldung die Wechselbeziehung und die Synergien wie zwischen jeder der unterschiedlichen Implementierungen im Kontext eines allumfassenden erfinderischen Systems; das alles geschieht um weiter die Einhaltung des U.S.C. §112 zu gewährleisten.

[0003] Weiter beansprucht die Anmelderin hiermit gemäss 35USC §119 die Priorität der US-amerikanischen Provisional Patentanmeldung Nr. 61/095 009, eingereicht am 8. September 2008 mit dem Titel «SYSTEM UND VERFAHREN ZUR BEREITSTELLUNG EINER ENTWICKLUNGS- UND/ODER BANKINGUMGEBUNG EINSCHLIESSLICH SYNTHETISCHER DATEN.»

[0004] Der gesamte Inhalt der oben genannten Anmeldung wird hier ausdrücklich durch Bezug aufgenommen.

Gebiet

[0005] Die vorliegende Erfindung zielt im Allgemeinen auf Vorrichtungen des Testens von Computern und spezifischer auf VORRICHTUNGEN ZUR BEREITSTELLUNG EINER VIRTUELLEN ENTWICKLUNGS- UND IMPLEMENTIERUNGSUMGEBUNG EINSCHLIESSLICH REELLER UND SYNTHETISCHER DATEN.

Hintergrund

[0006] Das Testen von Software wird von Softwareentwicklern verwendet, um Programmierfehler in Software-Anwendungen zu reduzieren. Integrierte Entwicklungsumgebungen (IDEs), wie z.B. Microsoft Visual Studio und Eclipse, und die in solchen IDEs enthaltenen Debugger stellen ein Set von Werkzeugen dar, welche von Softwareentwicklern zum Testen von Software verwendet werden können.

Beschreibung der Erfindung

[0007] Es stellt sich die Aufgabe, eine Möglichkeit zu schaffen, Software-Anwendungen, insbesondere bestehende, komplexe Software-Anwendungen mit verteilten Modulen, zu testen, ohne dass die tatsächliche Software-Umgebung von Testabläufen beeinflusst wird, mit dem Ziel, Unterbrechungen von Arbeitsabläufen zu vermeiden bzw. zu minimieren.

[0008] Diese Aufgabe wird mit einer Umgebungsentwicklungsvorrichtung gemäss Anspruch 1 gelöst.

[0009] Die VORRICHTUNG ZUR BEREITSTELLUNG EINER VIRTUELLEN ENTWICKLUNGS- UND IMPLEMENTIERUNGSUMGEBUNG EINSCHLIESSLICH REELLER UND SYNTHETISCHER DATEN (im Folgenden «SYNONYM» genannt) stellt eine virtuelle Entwicklungs- und Implementierungsumgebung bereit. In einer Ausführungsform umfasst eine Umgebungsentwicklungsvorrichtung einen Speicher und einen sich mit dem Speicher in Kommunikation befindenden Prozessor, welcher derart konfiguriert ist, dass er eine Vielzahl von im Speicher gespeicherten Verarbeitungsinstruktionen ausgibt. Der Prozessor gibt Instruktionen aus zum Empfangen einer Anfrage zur Erzeugung einer Verteilungsumgebung, Identifizieren reeller Datensätze basierend auf die Anfrage, Erzeugen variierter Daten aus den reellen Daten und Generieren einer synthetischen Verteilungsumgebung. Die Generierung umfasst weiter Instruktionen zum Instanziieren von Benutzerkonten. Weiter umfasst die Generierung Instruktionen zum Bevölkern der synthetischen Verteilungsumgebung mit einem Profil von Testkomponenten, wobei ein solches Profil ein Profil für eine Sammlung von Software für einen Server, einen Desktop-Computer, einen Notebook-Computer, ein mobiles Gerät, usw. darstellen kann. Weiter umfasst die Generierung Instruktionen zum Laden der variierten Daten in die synthetische Verteilungsumgebung, und Verteilen der synthetischen Verteilungsumgebung zu einem in der Anfrage angegebenen Standort.

Kurze Beschreibung der Zeichnungen

[0010] Die beigefügten Anhänge und/oder Zeichnungen stellen verschiedene, nicht limitierende Beispiele, erfinderische Aspekte, in Übereinstimmung mit der vorliegenden Offenbarung dar:

[0011] Fig. 1A zeigt ein Blockdiagramm, welches die Erzeugung und die Verwendung einer virtuellen Entwicklungs- und/oder Testumgebung in einer Ausführungsform des SYNONYM darstellen;

[0012] Fig. 1B zeigt ein Flussdiagramm, welches die Transformation reeller Daten in synthetische Daten in einer Ausführungsform des SYNONYM darstellt;

[0013] Fig. 1C zeigt ein Diagramm einer Bildschirmkopie, welches eine Web-basierte Benutzerschnittstelle des SYNONYM darstellt;

[0014] Fig. 2A zeigt ein logisches Flussdiagramm, welches die Erzeugung von anonymisierten Daten in einer Ausführungsform des SYNONYM darstellt;

[0015] Fig. 2B zeigt eine Tabelle, welche Anonymisierungsregeln in einer Ausführungsform des SYNONYM darstellt;

[0016] Fig. 2C zeigt ein Diagramm einer Bildschirmkopie, welches eine Web-Formular-basierte Benutzerschnittstelle in einer Ausführungsform des SYNONYM darstellt;

[0017] Fig. 3 zeigt ein logisches Flussdiagramm, welches die Erzeugung synthetisierter Daten in einer Ausführungsform des SYNONYM darstellt;

[0018] Fig. 4 zeigt ein logisches Flussdiagramm, welches die Erzeugung einer virtuellen Entwicklungs- und/oder Testumgebung in einer Ausführungsform des SYNONYM darstellt;

[0019] Fig. 5 zeigt ein logisches Flussdiagramm, welches eine Instantiierung von Benutzerkonten in einer Ausführungsform des SYNONYM darstellt;

[0020] Fig. 6A zeigt ein Blockdiagramm, welches ein Kopieren einer synthetischen Datenbank von einer «Master» virtuellen Entwicklungs- und/oder Testumgebung in eine neue virtuelle Entwicklungs- und/oder Testumgebung in einer Ausführungsform des SYNONYM darstellt;

[0021] Fig. 6B zeigt ein Blockdiagramm, welches eine Entwickler-Server-Zone (DvSZ) und Offshore-Enwicklungszentren (ODC) in einer Ausführungsform des SYNONYM darstellt;

[0022] Fig. 6C zeigt ein Blockdiagramm, welches eine spezielle Verwendung von Hardware in einer DvSZ in einer Ausführungsform des SYNONYM darstellt;

[0023] Fig. 6D zeigt ein Blockdiagramm, welches eine Kontrolle eines physischen Zugangs beim ODC in einer Ausführungsform des SYNONYM darstellt;

[0024] Fig. 6E zeigt ein Blockdiagramm, welches eine Verwendung einer Infrastruktur für öffentliche Schlüssel (PKI) in einer Ausführungsform des SYNONYM darstellt;

[0025] Fig. 6F zeigt ein Blockdiagramm, welches einen Transfer sensibler Informationen in die DvSZ in einer Ausführungsform des SYNONYM darstellt;

[0026] Fig. 6G zeigt ein Blockdiagramm, welches eine Netzwerkstruktur einer DvSZ in einer Ausführungsform des SYNONYM darstellt;

[0027] Fig. 7 zeigt ein Blockdiagramm, welches Ausführungsformen des SYNONYM Controllers darstellt.

[0028] Die Anfangszahl jedes Bezugszeichens in den Figuren beschreibt die Figur, in welcher dieses Bezugszeichen eingeführt und/oder ausgeführt ist. Folglich würde eine detaillierte Beschreibung des Bezugszeichens 105 in Fig. 1 gefunden und/oder eingeführt werden. Das Bezugszeichen 205 wird in Fig. 2 eingeführt, usw.

Detaillierte Beschreibung

Überblick

[0029] Da Software-Anwendungen im Hinblick auf Grösse und Komplexität gewachsen sind, ist das Testen von Software-Anwendungen immer wichtiger geworden. Aufgrund einer Vielzahl von Komponenten, einschliesslich mehrerer Datenbank-Komponenten, steigt in umfangreichen Software-Anwendungen die Anzahl der Interaktionen, die zwischen Anwendungskomponenten stattfinden können, exponentiell. Daher sollten Testverfahren verwendet werden, um zu prüfen, wie Änderungen in einer Anwendungskomponente andere Anwendungskomponenten beeinflusst, mit dem Ziel, zu vermeiden, dass Programmierfehler und ineffiziente Implementierungen als Folge von Änderungen in existierenden Komponenten oder Hinzufügungen neuer Komponenten in Software-Anwendungen eingefügt werden. Die Anzahl und die Typen von Daten, welche in Software-Anwendungen verwendet werden, haben sich auch erhöht. Da die Anzahl von Benutzern und von verfügbaren Optionen für diese Benutzer zugenommen hat, steigt auch die Anzahl von einer Software-Anwendung unterstützten Sonderfällen, Ausnahmen und Sprachen. Daher sollten die Interaktionen zwischen Anwendungskomponenten, einschliesslich Datenbank-Komponenten, und Anwendungsdaten auch getestet werden, um Fehler zu vermeiden. Da sich der Umfang der durchzuführenden Tests erhöht, ist dies auch für die Kosten, die Dauer und/oder die Anzahl der zur Durchführung der Tests notwendigen Personen gültig. Manche Firmen wählen eine Auslagerung oder ein Offshoring der Tests, um die mit dem Testen verbundenen Kosten zu reduzieren. Die Auslagerung oder das Offshoring von Tests erzeugt eine weitere Menge von Herausforderungen, da Kundendaten, welche beim Testen nützlich sind, gemäss verschiedener Datenschutzgesetze und Firmenrichtlinien geschützt werden müssen. Die synthetische Implementierungsumgebung umfasst Anwendungen, welche eine Spiegelung erlauben, indem existierende Umgebungen dupliziert werden, existierende Systeme durch die Implementierung einer Anzahl von synthetischen Umgebungen, welche zusätzlich Kapazitäten schaffen, im Hinblick auf Last ausgleichen, synthetische Spieleimplementierungsumgebungen generieren, synthetische

Bildungsimplementierungsumgebungen generieren, Ausbildungsimplementierungsumgebungen generieren, synthetische medizinische Implementierungsumgebungen generieren, die Qualitätsverbesserung durch die Zulassung von Variationen in der Testumgebung, welche Produktionsumgebungen nicht beeinflusst, fördern.

SYNONYM

[0030] Fig. 1A ist ein Blockdiagramm, welches die Erzeugung und die Verwendung einer virtuellen Entwicklungs- und/oder Testumgebung in einer Ausführungsform des SYNONYM darstellt. In Fig. 1A wird eine Anfrage zur Erzeugung einer neuen virtuellen Entwicklungs- und/oder Testumgebung empfangen 105. In einer Ausführungsform kann eine solche Anfrage von einem Mitglied eines Entwicklungsteams und/oder eines Testteams eines Projekts, von einer für einen Aufbau von Entwicklungsumgebungen für Projekten verantwortlichen Person, und/oder dergleichen kommen. In einer anderen Ausführungsform kann eine solche Anfrage automatisch als Teil der Erzeugung einer neuen Software-Anwendung/Komponente, der Änderung von Merkmalen einer Anwendungskomponente, der Hinzufügung einer Unterstützung für neue Datentypen, und/oder dergleichen ausgegeben werden. Die Anfrage kann in E-Mail-Form, in der Form einer programmierten Anfrage zur Erzeugung einer neuen virtuellen Entwicklungs- und/oder Testumgebung, und/oder dergleichen vorliegen. In einer Ausführungsform kann eine Web-basierte Benutzerschnittstelle Textfelder, Kontrollkästchen, Optionsschaltflächen und/oder dergleichen bereitstellen, um Administrativinformationen und Parameter zur Angabe von Testanforderungen eines Projekts betreffende Benutzereingaben anzunehmen. Ein Diagramm einer Bildschirmkopie, welches eine Web-basierte Benutzerschnittstelle in einer Ausführungsform des SYNONYM darstellt, ist in Fig. 1C gezeigt. Solche Daten können z.B. einen Benutzernamen 181, eine Benutzerkennung 182, eine Liste erforderlicher Umgebungen 183, einen Projektnamen 184, eine Projektkennung 185, eine Liste erforderlicher Anwendungen 186, eine Liste erforderlicher Datenbanken 187 und/oder dergleichen umfassen. Die Parameter können in XML gespeichert werden und nach einer Eingabe eines Benutzers über HTTP zu einem SYNONYM-Server gesendet werden. In einer Implementierung bekommt die Anfrage die folgende Form:

[0031]

```
<XML>
    <USER>
        <USER NAME>Toni</USER NAME>
        <USER ID>U1234</USER ID>
    </USER>
    <ENVIRONMENT>MAINFRAME</ENVIRONMENT>
    <PROJECT>
        <PROJECT NAME>New_Application</PROJECT NAME>
        <PROJECT ID>P5678</PROJECT ID>
    </PROJECT>
    <REQUIRED APPLICATIONS>
        <APPLICATION>Microsoft_Visio</APPLICATION>
        <APPLICATION>Eclipse</APPLICATION>
    </REQUIRED APPLICATIONS>
    <REQUIRED DATABASES>
        <DATABASE>SYN_CUSTOMERS</DATABASE>
        <DATABASE>SYN_ACCOUNTS</DATABASE>

          <DATABASE>SYN_DEPOSITS</DATABASE>
      </REQUIRED DATABASES>
</XML>
```

[0032] In manchen Fällen kann es nützlich sein, den Zugang zu aktuellen Daten zu beschränken, dies aufgrund von Datenschutzbedenken, Richtlinienbedenken und/oder da US- und internationale Bestimmungen die Verwendung und den

Austausch persönlicher Informationen beschränkt und da verschiedene regionale, lokale, industrieweite Firmenbestimmungen und/oder Zoll die Verwendung und Offenbarung von Informationen über bestimmten geschäftlichen Beziehungen oder Transaktionen einer Person oder Instanz zu beschränken (z.B. können verschiedene Beschränkungen im Hinblick auf die Verwendung von reellen Kundendaten zum Testen existieren, insbesondere wenn ein solches Testen ausgelagert wird oder Offshore getätigt wird). Eine virtuelle Entwicklungs- und/oder Testumgebung kann daher synthetisierte Daten anstatt von reellen Daten verwenden. In einer Ausführungsform können solche synthetisierten Daten auf anonymisierten, reellen Daten basieren. In einer Implementierung können anonymisierte Daten durch den Austausch von Datenwerten in bestimmten Datenbankfeldern mit anonymisierten Versionen dieser Datenwerte erzeugt werden 107. Weitere Details betreffend die Erzeugung von anonymisierten Daten werden später in der Beschreibung offenbart. In einer Implementierung können synthetisierte Daten durch eine Synthetisierung neuer Dateneinträge, basierend auf anonymisierte reelle Daten, erzeugt werden 110. In einer Implementierung kann die Menge der in einer synthetischen Datenbank (Synbase) gespeicherten Daten verringert werden, um die Effizienz zu erhöhen. Weitere Details betreffend die Erzeugung von anonymisierten Daten werden später in der Beschreibung offenbart.

[0033] In alternativen Ausführungsformen können reelle Daten und/oder modifizierte reelle Daten wie z.B. reelle Daten mit modifizierten IP-Adressen und/oder modifizierten Domain/Computer/Server-Namen anstatt synthetisierten Daten verwendet werden. In einer Ausführungsform umfasst die synthetische Implementierungsumgebung Anwendungen, welche eine Spiegelung durch eine Duplizierung von vorliegenden Umgebungen erlauben. Reelle Daten und/oder modifizierte reelle Daten können z.B. zur Duplizierung mittels einer Synchronisationstechnologie verwendet werden, wie z.B. rsync, einen Ablauf durch einen Job-Scheduler wie cron. In einer Ausführungsform können die IP-Adressen dynamisch solchen zusätzlichen Kopien zugewiesen werden, wobei die neuen Domain/Computer/Server-Namen den vorherigen Namen verwenden und einen Schrittzähler addieren können (z.B. aus PREVIOUS_NAME würde PREVIOUS_NAME_02, PREVIOUS_NAME_03, usw. entstehen). In einer anderen Ausführungsform umfasst die synthetische Implementierungsumgebung Anwendungen, welche einen Lastenausgleich vorliegender Systeme erlauben, indem sie eine Anzahl synthetischer Umgebungen implementieren, wodurch zusätzliche Kapazitäten geschaffen werden. Zum Beispiel können reelle Daten und/oder modifizierte reelle Daten zum Lastenausgleich zwischen mehreren synthetischen Umgebungen verwendet werden, indem ein Round-Robin-Ablaufsteuerungsalgorithmus verwendet wird. In einer solchen Ausführungsform kann jede der anschliessenden duplizierten synthetischen Umgebungen mit neuen Identifikatoren geografisch ein Teil des Round-Robin-Pools werden, oder basierend auf die Round-Robin Zeitzuweisung (z.B. jede synthetische Umgebung, welche einen Teil der Arbeitslast für eine gegebene Zeitspanne bearbeitet). In noch einer weiteren Ausführungsform umfasst die synthetische Implementierungsumgebung Anwendungen, welche Grid-Computing erlauben, indem sie eine Anzahl parallel arbeitender synthetischer Umgebungen, implementieren. Zum Beispiel können reelle Daten und/oder modifizierte reelle Daten im Grid-Computing, unter Verwendung von verteilten Rechentechnologien, wie z.B. Mac OS Xgrid (z.B. im Falle, dass die duplizierten umbenannten synthetischen Umgebungen als individuelle Prozessknoten im Netz (grid) agieren), verwendet werden.

[0034] Eine virtuelle Entwicklungs- und/oder Testumgebung kann als Antwort auf die Anfrage zur Erzeugung einer neuen virtuellen Entwicklungs- und/oder Testumgebung erzeugt werden 115. In einer Ausführungsform kann die virtuelle Entwicklungs- und/oder Testumgebung beständig (ENV) sein und zwischen mehreren Projekten geteilt werden. In einer anderen Ausführungsform kann die Entwicklungs- und/oder Testumgebung auf Anfrage (TED) erzeugt und von einem einzigen Projekt verwendet werden. In noch einer weiteren Ausführungsform kann die Entwicklungs- und/oder Testumgebung eine Datenbank verwenden, welche synthetisierte Datensätze («Synbase» im Folgenden) umfasst. In einer anderen Implementierung kann die Entwicklungs- und/oder Testumgebung eine aus einer Vielzahl von Synbase-Versionen wählen, um sie zum Testen zu verwenden. In einer Ausführungsform kann die Entwicklungs- und/oder Testumgebung physikalisch und/oder logisch von einer Produktionsumgebung getrennt werden. Fig. 6C zeigt ein Blockdiagramm, welches eine dedizierte Verwendung von Hardware in einer DvSZ in einer Ausführungsform des SYNONYM darstellt. In Fig. 6C ist keine Hardware-Teilung zwischen der DvSZ 620 und dem Firmen-Intranet 621 erlaubt. Eine Ausnahme ist die Mainframe, und die Mainframe ist stattdessen logisch in RZ1622 und RZ8/9623 zwischen dem Firmen-Intranet bzw. der DvSZ geteilt.

[0035] In einer Implementierung können unterschiedliche Sicherheitszertifikate (z.B. Sicherheitszertifikate, welche auf einer Infrastruktur für öffentliche Schlüssel basieren) an Ressourcen in der virtuellen Entwicklungs- und/oder Testumgebung und in der Produktionsumgebung ausgegeben werden. Als Beispiel für eine SYNONYM-Ausführungsform zeigt Fig. 6E ein Blockdiagramm, welches die Verwendung der Infrastruktur für öffentliche Schlüssel (PKI) darstellt. In Fig. 6E gibt die PKI 640 im Firmen-Intranet 641 Sicherheitszertifikate an Computern des Firmen-Intranets 641 und der DvSZ 642 aus. Die PKI 640 gibt unterschiedliche Sicherheitszertifikate an Ressourcen des Firmen-Intranets 641 und an Ressourcen der DvSZ 642 aus. Eine Kommunikation zwischen einem Client und einem Server kann nur erfolgreich sein, wenn beide Ressourcen entweder Test-Sicherheitszertifikate oder Produktionssicherheitszertifikate haben. Solche Ressourcen können z.B. Java Anwendungsplattform-Server, COBRA-Adapter, Datenbankserver und/oder dergleichen umfassen. In einer Implementierung kann eine Entwicklungs- und Test-Server-Zone derart erzeugt werden, dass sie die Entwicklungs- und/oder Testumgebung enthält. Zum Beispiel kann ein dediziertes Netzwerk, welches vom Rest eines Intranets einer Organisation getrennt und in seiner Fähigkeit, auf das Intranet der Organisation zuzugreifen, beschränkt ist, welches Komponenten wie Firewall, VPN Gateway, Sicherheitszertifikate und/oder dergleichen verwendet, unter Verwendung von dedizierten Routern eingerichtet werden. Der Zugriff auf die DvSZ kann auf Benutzern mit hoher Authentifizierung beschränkt sein, und keine sensiblen Informationen können in der DvSZ zugänglich sein. Weiter kann die DvSZ als eine «Schwarzes

Loch»-Netzwerkzone implementiert sein. Als solche wird jeder Informationsaustausch zwischen der DvSZ und äusseren Datenquellen nur erlaubt, wenn er von ausserhalb der DvSZ initiiert wird. Fig. 6F zeigt ein Blockdiagramm, welches den Transfer sensibler Informationen in die DvSZ in einer Ausführungsform des SYNONYM darstellt. Fig. 6F zeigt, dass keine automatische Synchronisation 650 von Informationen zwischen Ablagen 651 im Firmen-Intranet 652 und Ablagen 653 in der DvSZ 654 existiert. Nur sensible Informationen, welche durch eine Qualitätsschleuse 655A-B freigegeben wurden, können in die DvSZ übertragen werden. Der Verbindungsaufbau vom Inneren der DvSZ 654 zu einer äusseren Ressource ist blockiert. Informationen, welche zur Verwendung in der DvSZ freigegeben sind, können nur für diejenigen Benutzer der DvSZ zugänglich sein, die zur Verwendung der Informationen befugt sind. Fig. 6G zeigt ein Blockdiagramm, welches eine Netzwerkstruktur der DvSZ in einer Ausführungsform des SYNONYM darstellt. Fig. 6G zeigt, wie das Netzwerk aufgebaut wird 660A–C und die für den Aufbau verwendete Hardware 661A–D. Weitere Details hinsichtlich der Erzeugung einer virtuellen Entwicklungs- und/oder Testumgebung werden später in der Beschreibung bereitgestellt. Benutzerkonten, die zur Verwendung einer virtuellen Entwicklungs- und/oder Testumgebung berechtigt sind, können instantiiert werden 120. In einer Ausführungsform können eine Vielzahl von Benutzerkontotypen, mit einer Anzahl von Rechten und Authentifizierungsanforderungen, instantiiert werden. Weitere Details hinsichtlich der Instantiierung von Benutzerkonten werden später in der Beschreibung bereitgestellt.

[0036] Autorisierte Benutzer können die notwendige Entwicklung und Tests 125 durch Verwendung einer virtuellen Entwicklungs- und/oder Testumgebung und/oder einer zugeordneten Synbase durchführen. In einer Ausführungsform kann ein solches Testen Modultests, Merkmaltests, Anwendungstests und/oder dergleichen umfassen. In einer Ausführungsform kann das Testen von dedizierten menschlichen Testern, welche Testergebnisse mit vorgegebenen Leistungsparametern vergleichen, durchgeführt werden. In einer anderen Implementierung kann das Testen mit Hilfe von automatischen Testroutinen, welche Testergebnisse mit vordefinierten Werten oder getesteten Durchschnittswerten vergleichen. Das Ergebnis einer «Überprüfe Kontostand»-Operation, welche für ein neulich eröffnetes Konto angewendet wird, kann z.B. mit einem vorgegebenen Wert von 0$ verglichen werden, um sicherzustellen, dass sich die Operation für diesen Testfall korrekt verhält. Als anderes Beispiel können die Zeit, um «Überprüfe Kontostand»-Operationen auszuführen, und der Durchschnitt der Ausführungszeit im Vergleich mit der Zeit zur Ausführung der aktuellen «Überprüfe Kontostand»-Operation, verfolgt werden, um die Ausführungsgeschwindigkeit zu testen. In einer anderen Ausführungsform können Zufallswerte in ein Eingabefeld eingefügt werden, um festzustellen, ob eine Programmkomponente Fehler richtig behandelt, indem die Programmantworten bei solchen Eingangswerten protokolliert/verfolgt werden. In einer anderen Ausführungsform kann Anwendungsquellcode im Hinblick auf Hartkodierung von sensiblen Daten überprüft werden. In einer Implementierung kann ein legaler Parser verwendet werden, um potentiell sensible Daten in einer Quellcode-Datei, geschrieben in einer Programmiersprache wie PL/I, Java und/oder dergleichen, welche eventuell aufgestellte Datenschutz- oder Geheimhaltungsvorschriften verletzen, zu identifizieren. Zum Beispiel kann eine für die Spezifikation von Datenschutzvorschriften verantwortliche Abteilung, wie eine Rechts- und Complianceabteilung, eine Liste potentiell sensibler Datentypen (z.B. Namen, CIF-Nummern, Kontonummern und/oder dergleichen) spezifizieren. Der legale Parser kann dann dazu verwendet werden, Quellcode-Dateien eines Projekts auf potentielle Übereinstimmungen zu durchsuchen und eine Meldung für jede solche Übereinstimmung bereitzustellen. Zum Beispiel können Quellcode und Kommentare unter Verwendung von Standard-Analyse-Techniken gescannt werden, wie z.B. durch Verwendung des m// Operators in Perl, z.B. m/$LegalParser/ für Einträge einer 12-stelligen Nummer, welche mit der Länge einer CIF-Nummer übereinstimmt. In einer solchen Ausführungsform kann eine Datenstruktur des legalen Parsers (z.B. eine Datei, memory struct, usw.) eine Liste von markierten Begriffen, Zahlen, usw., welche für die Überprüfung unautorisierter Verwendung dieser benutzt wird, enthalten. Eine Vielzahl von Verfahren kann zum Informationsaustausch zwischen Team-Mitgliedern während des Testens verwendet werden. In einer Implementierung können E-Mail, Sofort-Kommunikation (Instant Messaging), http Nachrichten und/oder ähnliche Protokolle zum Informationsaustausch verwendet werden. Zum Beispiel können sich Team-Mitglieder mit Hilfe von E-Mail-Software wie Microsoft Outlook, Lotus Notes und/oder dergleichen gegenseitig E-Mails schicken, um Fragen zu stellen und um Antworten zu bekommen. In einer anderen Implementierung können Technologien wie Microsoft SharePoint, welche Dokumentenverwaltung, Diskussionsforen, Umfragen und E-Mail-basierte Nachrichten bereitstellen, zum Informationsaustausch verwendet werden. In einer Ausführungsform kann die synthetische Implementierungsumgebung zur Simulation einer virtuellen Bank verwendet werden. In einer anderen Ausführungsform kann die synthetische Implementierungsumgebung dazu verwendet werden, synthetische Implementierungsumgebungen in Bereichen wie Spiele, Simulation, Bildung, Medizin, Qualitätskontrolle und/oder dergleichen zu generieren.

[0037] Eine virtuelle Entwicklungs- und/oder Testumgebung kann auf Anfrage ausser Betrieb gesetzt werden 130. Zum Beispiel kann eine virtuelle Entwicklungs- und/oder Testumgebung nach der Beendigung von Tests, welche für eine bestimmte Anwendungskomponente, Projektphase, Projekt und/oder dergleichen notwendig waren, ausser Betrieb gesetzt werden. In einer Implementierung kann eine solche Ausserbetriebnahme sicheres Löschen, einschliesslich mehrerer Datenlöschvorgänge, von mit der virtuellen Entwicklungs- und/oder Testumgebung verknüpften Benutzerkonten und Synbase umfassen. In einer anderen Ausführungsform kann die Ausserbetriebnahme eine Archivierung der Umgebung bei einer Eigentümerfirma und ein sicheres Löschen der Umgebung an dem Sitz eines Auslagerungslieferanten verwendet werden.

[0038] Fig. 1B zeigt ein Datenflussdiagramm, welches die Transformation reeller Daten in synthetischen Daten in einer Ausführungsform des SYNONYM darstellt. Um realistische Daten für die virtuelle Entwicklungs- und/oder Testumgebung bereitzustellen, können in einer Implementierung reelle Daten 150 als Basis für synthetisierte Daten verwendet werden. In einer Implementierung können die reellen Daten Felder wie Kundenname, Kundenadresse, Aktennummer der Kunde-

ninformation (CIF) und/oder dergleichen umfassen. In einer Implementierung werden reelle Daten in anonymisierte Daten 160 konvertiert. In einer Implementierung kann die Anonymisierung den Austausch des Kundennamens- und Kundenadressfelder mit anonymisierten Versionen dieser Felder, welche gemäss vordefinierten Vorschriften erzeugt wurden, umfassen. Weitere Details hinsichtlich der Erzeugung anonymisierter Daten werden später in der Beschreibung aufgeführt. In einer Implementierung werden anonymisierte Daten in synthetische Daten 170 konvertiert. In einer Implementierung kann eine Teilmenge anonymisierter Daten 165, welche repräsentativ für jeden für das Testen benötigten Kundentyp, für die Synthetisierung ausgewählt werden. In einer Ausführungsform weist die Reduzierung der Datensätze eine erhöhte Ansprechbarkeit auf, um einen bestimmten Datentyp effektiver zu testen. Es können z.B. etliche Kontotypen existieren, wobei SYNONYM nur einen oder eine Selektion einiger weniger Kontotypen zum Testen verwenden kann (z.B. können Premiumkonten eine grössere Anzahl und Vielfalt von Kontotypen enthalten und können als solche als bessere Datensätze für Stresstests eines Systems dienen). In einer anderen Ausführungsform bewirkt die Teilmenge eine Reduzierung der Menge synthetisierter Daten, um schnellere Testzyklen zu erlauben. Die selektierten Daten können in synthetisierten Daten dadurch konvertiert werden, dass eine neue CIF für jeden selektierten Kunden erzeugt wird und die CIF-Nummer des jeden Kunden derart aktualisiert wird, dass sie auf die neu erzeugte synthetische CIF zeigt. Weitere Details betreffend die Erzeugung synthetisierter Daten werden später in der Beschreibung dargelegt.

[0039] Fig. 2A zeigt ein logisches Flussdiagramm, welches die Erzeugung anonymisierter Daten in einer Ausführungsform des SYNONYM darstellt. In Fig. 2A wird eine Anfrage zur Anonymisierung von Datensätzen empfangen 205. In einer Ausführungsform kann die Anfrage eine Liste von Datenbanken 251, Datenbanktabellen 252, und Datenfelder 253, welche anonymisiert werden müssen, enthalten. Die Liste von Datenbanken, Datenbanktabellen und Datenfelder kann auf die Eingaben eines Benutzers, welcher eine neue Testumgebung angefordert hat, basieren. In einer Implementierung kann die Liste in XML-Format sein, und die Anfrage kann die folgende Form erhalten:

[0040]

```
<XML>
   <ANONYMIZATION DATABASES>
     <DATABASE>
       <DATABASE NAME>REAL_CUSTOMERS</DATABASE NAME>
       <TABLE>
          <TABLE NAME>Customer_Info</TABLE NAME>
          <TABLE FIELD>Name</TABLE FIELD>
       </TABLE>
     </DATABASE>
   </ANONYMIZATION DATABASES>
</XML>
```

[0041] In einer Ausführungsform kann das obige XML das Ergebnis von Benutzerselektionen von Einblendmenueinträgen in einer Web-basierten Benutzerschnittstelle sein. Fig. 2C zeigt ein Diagramm einer Bildschirmkopie, welches eine Web-basierte Benutzerschnittstelle in einer Ausführungsform des SYNONYM darstellt. In einer anderen Implementierung kann die Liste in einer Datenbanktabelle gespeichert werden, wobei SYNONYM über einen SQL-SELECT-Befehl darauf zugreift. In noch einer anderen Ausführungsform kann die Liste in einer Konfigurationsdatei gespeichert werden, wobei SYNONYM über I/O-Aufrufe, welche ein Lesen aus einer Datei erlauben, darauf zugreift. Basierend auf die Liste werden die zu anonymisierenden Datenbanktabellen selektiert 207, und Datensätze werden zur Verarbeitung entnommen 209. In einer Implementierung kann eine solche Selektion über einen oder mehrere SQL-SELECT-Befehle (z.B. SELECT CIF, Name FROM Customer_Info) erreicht werden, und die Entnahme kann dadurch erfolgen, dass die selektieren Datensätze in eine Datendatei abgelegt werden. Anonymisierungsvorschriften und Referenztabellen können bei 213 geladen werden. Anonymisierungsvorschriften geben an, wie anonymisierte Austauschwerte für ein bestimmtes Feld generiert werden können. Fig. 2B ist eine Tabelle, welche Anonymisierungsvorschriften in einer Ausführungsform des SYNONYM darstellt. Referenztabellen enthalten Daten, die als Austauschwerte gemäss Anonymisierungsvorschriften verwendet werden können. Zum Beispiel kann in einer Implementierung eine Referenztabelle eine Liste von 676 männlichen und weiblichen Namen enthalten, zusammen mit einer Liste von Regionen, in denen jeder Name gebräuchlich ist. In einer Ausführungsform wird eine Tabelle mit erfundenen Namen besetzt; in einer Ausführungsform kann ein Namensgenerator (z.B. www.fakenamegenerator.com) zur Generierung und/oder Besetzung einer Datenbanktabelle mit erfundenen Namen verwendet werden.

[0042] In einer Ausführungsform kann ein Datensatz, welcher reelle Daten enthält, aus einer Datendatei zur Anonymisierung selektiert werden 215. Zum Beispiel können I/O-Aufrufe, welche das Lesen aus einer Datendatei erlauben, verwendet werden. Der Datensatz kann mit Hilfe eines Analysealgorithmus analysiert werden, um das nächste vorhandene Feld

217 zu selektieren. In einer Implementierung kann eine Standard-XML-Formatierung verwendet werden, um die Analyse zu ermöglichen. Es wird ermittelt, ob das aktuell selektierte Feld einer Anonymisierung 219 basierend auf die Anonymisierungsvorschriften bedarf. Zum Beispiel kann der Name des Feldes mit der Liste von Feldnamen in den Anonymisierungsvorschriften verglichen werden. Wenn der Feldname keinem der in den Anonymisierungsvorschriften spezifizierten Feldnamen entspricht, wird der aktuelle Feldwert beibehalten 221. Anderenfalls wird eine geeignete Vorschrift für das Feld aus den Anonymisierungsvorschriften ausgesucht 223. In einer Implementierung können die Anonymisierungsvorschriften geladen werden, wobei der Vergleich mittels einer Programmiersprache wie PL/I, Perl und/oder dergleichen durchgeführt wird. In einer Implementierung kann ein Beispiel der PL/I-Vorschriften die folgende Form haben:

[0043]

```
if ($fieldType == TYPE_FIRST_NAME)
     replaceFirstName($field);
if ($fieldType == TYPE_ADDRESS)
     replaceAddress($field);
```

[0044] Basierend auf die Anonymisierungsvorschrift wird ein anonymisierter Wert für das Feld 225 generiert. In einer Ausführungsform kann der anonymisierte Wert derart generiert werden, dass ein zufälliger Wert aus einer Liste mit verwendbaren Werten aus einer in der Anonymisierungsvorschrift spezifizierten Referenztabelle, selektiert wird, z.B. aus einer erfundenen Namenstabelle. In einer anderen Ausführungsform kann der anonymisierte Wert basierend auf in den Anonymisierungsvorschriften spezifizierte Instruktionen zufällig generiert werden. Der generierte anonymisierte Wert ersetzt den originalen Feldwert 227. Wenn mehrere Felder im aktuell selektierten Datensatz vorhanden sind, kann das nächste Feld für die Anonymisierung 229 selektiert werden, anderenfalls kann der nächste vorhandene Datensatz für die Anonymisierung 231 selektiert werden. Sobald alle Datensätze anonymisiert sind, können die anonymisierten in eine oder mehreren Datendateien 233 durch Verwendung von I/O-Aufrufen, welche ein Schreiben in eine Datendatei erlauben, ausgegeben werden. In einer Implementierung werden die Daten in eine neue Datendatei geschrieben. In einer anderen Implementierung wird die existierende Datendatei mit neuen anonymisierten Daten überschrieben. Die anonymisierten Daten werden übertragen 235, um allen Teilen von SYNONYM Zugang zu den anonymisierten Daten zu erlauben. Zum Beispiel können die anonymisierten Daten in Dateien und/oder Datenbanken in Speicherabschnitten kopiert werden, welche so konzipiert sind, dass sie mit unterschiedlichen Betriebssystemen wie Microsoft Windows XP, Unix, Apple Macintosh OS und/oder dergleichen arbeiten.

[0045] Fig. 3 zeigt ein logisches Flussdiagramm, welches die Erzeugung synthetisierter Daten in einer Ausführungsform des SYNONYM darstellt. In Fig. 3 wird eine Anfrage zur Erzeugung synthetisierter Daten empfangen 310. In einer Ausführungsform kann die Anfrage eine Liste von Datenbanken, welche bei der Synthetisierung verwendet werden sollen, enthalten. Die Liste der Datenbanken kann auf die Eingabe eines Benutzers, der eine neue Testumgebung angefordert hat, basieren. In einer Ausführungsform kann die Liste im XML-Format vorliegen, und die Anfrage kann die folgende Form annehmen:

[0046]

```
<XML>
   <SYNTHESIZATION DATABASES>
      <DATABASE NAME>ANONYMIZED_CUSTOMERS</DATABASE NAME>
      <DATABASE NAME>ANONYMIZED_ADDRESSES</DATABASE NAME>
   </SYNTHESIZATION DATABASES>
</XML>
```

[0047] In einer anderen Implementierung kann die Liste in einer Datenbanktabelle gespeichert werden, wobei SYNONYM über SQL-Befehle darauf zugreifen kann. Zum Beispiel kann eine Web-Schnittstelle, welche einem Benutzer erlaubt, Datenbanken zu selektieren (z.B. über Kontrollkästchen), welche bei der Synthetisierung verwendet werden sollen, benutzt werden. Das obige XML kann zum SYNONYM geschickt werden, welches sich mit den spezifizierten Datenbanken mittels Datenbankadaptern verbindet, wie z.B. indem es durch die Open Database Connectivity (ODBC) API-Protokoll verfügbare Adapter, durch die WebObjects Enterprise Objects Framework (EOF) verfügbare Adapter und/oder dergleichen verwendet. In noch einer anderen Ausführungsform kann die Liste in einer Konfigurationsdatei gespeichert werden und vom SYNONYM unter Verwendung von I/O-Aufrufen, welche das Lesen aus einer Datei erlauben, abgegriffen werden. Basierend auf die Liste werden anonymisierte Daten zur Synthetisierung eingegeben 315. In einer Implementierung werden die anonymisierten Daten durch Bereitstellung eines Namens einer oder mehrerer Dateien, welche die anonymisierten Daten enthalten, eingegeben. In einer anderen Implementierung werden die anonymisierten Daten durch Bereitstellung des Namens einer oder mehrerer SQL-Datenbanken und/oder Tabellen, welche die anonymisierten Daten enthalten, und/oder

durch ein SQL SELECT von spezifizierten Tabellen eingegeben. In noch einer Implementierung werden die anonymisierten aus einer oder mehreren SQL-Datenbanken in eine oder mehreren Dateien exportiert, und die Namen dieser Dateien werden eingegeben. In einer Ausführungsform werden auch andere Daten, die den anonymisierten Daten zugeordnet sind, eingegeben.

[0048] Die Anfrage zur Erzeugung synthetisierter Daten kann auch spezifizieren, dass eine Teilmenge der für jeden Kundentyp repräsentativen Daten, die zum Testen angefordert sind, für die Synthetisierung ausgewählt werden sollen, um die Menge der synthetisierten Daten zu reduzieren. In einer Ausführungsform kann die Teilmenge der Daten, die selektiert werden soll, als Geschäftskonstellationen spezifiziert werden, und die Geschäftskonstellationen können als Äquivalenzklassen spezifiziert werden. In einer Implementierung kann eine Äquivalenzklasse: Titel, Geschäftsbereich, Zweigstelle, Staatsbürgerschaft, Domizil, Sprache, Kundentyp, Kundensegment, usw. umfassen, und eine Geschäftskonstellation kann eine eindeutige Kombination von Werten einer oder mehrerer Äquivalenzklassen umfassen. Zum Beispiel kann eine Geschäftskonstellation als einen männlichen Geschäftskunden aus den USA spezifiziert werden. Diese Geschäftskonstellation kann in XML-Format wie folgt als Teil der Anfrage übertragen werden:

[0049]

```
<XML>
   <BUSINESS CONSTELLATIONS>
     <BUSINESS CONSTELLATION>
       <CUSTOMER TYPE>Male</CUSTOMER TYPE>
       <CUSTOMER SEGMENT>Business</CUSTOMER SEGMENT>
       <DOMICILE>US</DOMICILE>
     </BUSINESS CONSTELLATION>
   </BUSINESS CONSTELLATIONS>
</XML>
```

[0050] Die Daten können basierend auf spezifizierte Geschäftskonstellationen 320 abgefragt werden. In einer Ausführungsform können programmierte Befehle, welche in jede Programmiersprache wie C++, Perl, SQL und/oder dergleichen geschrieben wurden, ausgegeben werden, um Datenwerte mit in einer Geschäftskonstellation spezifizierten Werten verglichen werden. In einer anderen Ausführungsform kann ein SQL SELECT für einen Kundentyp eine Teilmenge von mehreren Testergebnissen, welche verwendet werden können, bereitstellen. Eine Teilmenge von Daten, welche aussagekräftig für alle spezifizierten Geschäftskonstellationen sind, können dann basierend auf Vergleiche erlangt werden 330. In einer Implementierung wird der erste gefundene Datensatz, der allen Äquivalenzklassen einer gegebenen Geschäftskonstellation genügt, ausgewählt, um die gegebene Geschäftskonstellation zu repräsentieren. Jede Geschäftskonstellation kann auch durch mehrere Datensätze bis zu einer vorgegebenen maximalen Zahl repräsentiert werden.

[0051] Synthetisierte Daten basieren auf die Teilmenge der für alle spezifizierten Geschäftskonstellationen repräsentativen Daten. In einer Ausführungsform können neue Kundeninformationsdateien mit neuen CIF-Nummern für jeden Kunden in der Teilmenge von Daten erzeugt werden, und die CIF-Nummern in den anonymisierten Daten oder in anderen, den anonymisierten Daten zugeordneten Daten können mit neuen synthetischen CIF-Nummern 335 ausgetauscht werden. Zum Beispiel können neue CIF-Nummern zufällig und/oder seriell derart generiert werden, dass sie eindeutig sind und dasselbe Format wie echte CIF-Nummern haben. In einer Implementierung können die zum Generieren synthetisierter CIF-Nummern verwendeten Zahlenbereiche in der Produktionsumgebung verboten werden. Nachdem dies vollzogen ist, ist es nicht länger möglich, aus den neuen CIF-Nummern irgendwelche echten Kundenidentifikations-Informationen zu bestimmen oder die neuen synthetisierten CIF-Daten auf einen echten Kunden zurückzuführen. In einer Implementierung werden die resultierenden synthetisierten Daten in eine oder in mehreren Datendateien mittels I/O-Aufrufe, die ein Schreiben in eine Datendatei erlauben, ausgegeben.

[0052] Eine synthetische Datenbank kann derart erzeugt werden 345, dass sie die neuen synthetisierten Daten enthält. In einer Ausführungsform wird eine neue Synbase erzeugt, indem jede Anzahl von Datenbankplattformen wie Microsoft SQL Server, MySQL und/oder dergleichen verwendet werden. Zum Beispiel kann sie mit dem folgenden mySQL SQL-Befehl erzeugt werden:
create database Dbname;

[0053] In einer anderen Ausführungsform können Inhalte einer existierenden Datenbank gelöscht werden (z.B. durch Verwendung von SQL DELETE FROM synbase_table_name [WHERE constraint]) und/oder mit synthetisierten Daten überschrieben werden (z.B. durch Verwendung von SQL UPDATE synbase_table_name SET field_value='SomeNewValue'[WHERE constraint])

[0054] In einer alternativen Ausführungsform können andere Daten, welche den synthetisierten Daten zugeordnet sind, auf ähnliche Weise synthetisiert werden, bevor eine synthetische Datenbank in 345 erzeugt wird. In einer Implementierung können neue Konten/Depots mit neuen Konten-/Depotnummern für Konten/Depots, welche den synthetisierten CIF-Daten zugeordnet sind, erzeugt werden, und die Konto-/Depotnummern werden mit neuen Konto-/Depotnummern 340 ausgetauscht. Zum Beispiel können neue Konto-/Depotnummern nach dem Zufallsprinzip derart generiert werden, dass sie eindeutig sind und dasselbe Format wie echte Konto-/Depotnummern aufweisen. In einer Implementierung können die Zahlenbereiche, die zur Generierung von synthetisierten Konto-/Depotnummern verwendet werden, in der Produktionsumgebung verboten werden.

[0055] Die synthetisierten Daten werden in einer Synbase 350 abgelegt. In einer Implementierung können die in einer oder in mehreren Datendateien synthetisierten Daten mittels I/O-Aufrufen, welche das Lesen aus einer Datendatei erlauben, ausgelesen, analysiert, wie z.B. durch Verwendung des m//-Operators in Perl (z.B. m/| . * | / zur Extrahierung eines mit «|» eingeschlossenen Feldwertes) und in eine Synbase durch Verwendung von SQL-Befehlen abgelegt. Es wird eine Feststellung durchgeführt, ob die virtuelle Entwicklungs- und/oder Testumgebung zusätzlich zu den synthetisierten Daten 355 Transaktionsdaten aufweisen sollte. In einer Ausführungsform kann eine solche Feststellung durch eine Überprüfung der Anfrage zur Synthetisierung von Daten auf ein Vorliegen weiterer Informationen, welche Ende-des-Tages Arbeitsabläufe (EODs) spezifizieren, welche zur Erzeugung von Transaktionsdaten ausgeführt werden sollten. Zum Beispiel können in einer Implementierung diese Informationen in XML-Format als Teil der Anfrage abgegeben werden, wie folgt:

[0056]

```
<XML>
    <EODs>
        <EOD>MakePayments</EOD>
        <EOD>CancelPayments</EOD>
    </EOD>
</XML>
```

[0057] In einer Ausführungsform können die EODs die Auswahlkriterien für Transaktionen, welche für die virtuelle Entwicklungs- und/oder Testumgebung erzeugt werden sollen, angeben 360. Zum Beispiel kann in einer Implementierung ein EOD die Erzeugung eines spezifischen Typs von Transaktionsdaten anstossen (z.B. EOD MakePayments kann Transaktionsdaten, welche von Kunden getätigte Zahlungen simulieren, erzeugen, während EOD CancelPayments Transaktionsdaten erzeugen, welche von Kunden stornierte Zahlungen simulieren). Daher wird durch die Spezifizierung, dass ein gegebener EOD ausgeführt werden soll, um Transaktionsdaten zu erzeugen, auch die Art der Transaktionsdaten, die erzeugt werden sollen, festgelegt. Durch die Verwendung von EODs wird eine Synbase-Version erzeugt 365, welche Transaktionsdaten umfasst. In einer Ausführungsform können Transaktionsdaten, wie z.B. Geldüberweisungen zwischen zwei Kunden, derart mit einem EOD simuliert werden, dass eine vorgegebene Anzahl von zufälligen Geldüberweisungstransaktionen erzeugt wird (z.B. in Perl Auswahl zweier zufälligen Konten, Auswahl einer Zahl in einem vorgegebenen Bereich, welche eine zu überweisende Geldsumme darstellt, und die Überweisung dieser Geldsumme vom ersten zum zweiten Konto), und dass Datensätze, welche diese Überweisungen repräsentieren, unter Verwendung von I/O-Aufrufen ausgegeben werden. In einer Implementierung werden die Transaktionsdatensätze zusammen mit Synbase-Daten in eine neue Synbase kopiert (z.B. durch Verwendung von Adaptern, die durch den Open Database Connectivity (ODBC) API-Standard, von Adaptern, die durch die WebObjects Enterprise Objects Framework (EOF) und/oder dergleichen bereitgestellt werden), welche eine neue Synbase-Version umfasst. In einer anderen Implementierung werden nur die Transaktionsdatensätze in eine separate Datenbank oder in eine separate Datenbanktabelle kopiert, wobei, zusammen mit den synthetisierten Daten, die zwei Teile eine neue Synbase-Version umfassen.

[0058] Fig. 4 zeigt ein logisches Flussdiagramm, welches die Erzeugung einer virtuellen Entwicklungs- und/oder Testumgebung in einer Ausführungsform dies SYNONYM darstellt. In Fig. 4 wird eine Anfrage zur Erzeugung einer virtuellen Entwicklungs- und/oder Testumgebung empfangen 440. In einer Ausführungsform kann diese Anfrage, wie oben beschrieben, auf die Eingaben eines Benutzers basiert sein, der eine neue Testumgebung angefordert hat, und sie kann verschiedene vom Benutzer gewünschte Umgebungen, Anwendungen und Datenbanken angeben. In einer Implementierung hat die Anfrage die folgende Form:

[0059]

```
<XML>
    <ENVIRONMENT>MAINFRAME</ENVIRONMENT>
    <REQUIRED APPLICATIONS>
        <APPLICATION>Microsoft_Visio</APPLICATION>
        <APPLICATION>Eclipse</APPLICATION>
    </REQUIRED APPLICATIONS>
    <REQUIRED DATABASES>
        <DATABASE>SYN_CUSTOMERS</DATABASE>
        <DATABASE>SYN_ACCOUNTS</DATABASE>
        <DATABASE>SYN_DEPOSITS</DATABASE>
    </REQUIRED DATABASES>
</XML>
```

[0060] Die angeforderte Laufzeit-, Middleware-, Datenbankumgebung und/oder dergleichen werden bei 442 installiert. In einer Implementierung kann eine solche Software ein Betriebssystem umfassen, wie z.B. Microsoft Windows XP, Unix, Apple Macintosh OS und/oder dergleichen; Laufzeitumgebungen wie Java Runtime Environment, GTK+ und/oder dergleichen; Middleware-Umgebungen wie CORBA, DCOM, RPC und/oder dergleichen; und Datenbankumgebungen wie Microsoft SQL Server, MySQL und/oder dergleichen. Die angeforderten projektspezifischen Anwendungen, Infrastruktur und Testprogramme werden bei 444 installiert. In einer Implementierung kann eine solche Software Anwendungen wie z.B. einen Web-Server für web-basierte Projekte umfassen; Infrastruktur wie Concurrent Versioning System (CVS); und Testprogramme wie SilkTest. Die angeforderte Synbase-Version wird selektiert 446 und in eine oder in mehreren Datenbanken, welche von der virtuellen Entwicklungs- und/oder Testumgebung verwendet werden, kopiert 448. In einer Ausführungsform können mehrere Synbase-Versionen in einer virtuellen «Master» Entwicklungs- und/oder Testumgebung vorhanden sein. Sobald eine Anfrage zur Erzeugung einer neuen virtuellen Entwicklungs- und/oder Testumgebung empfangen wird, kann die angeforderte Synbase-Version von der virtuellen «Master» Entwicklungs- und/oder Testumgebung in die neue virtuelle Entwicklungs- und/oder Testumgebung kopiert werden. In einer Implementierung kann eine Synbase-Version ohne Transaktionsdaten von der virtuellen «Master» Entwicklungs- und/oder Testumgebung in die neue virtuelle Entwicklungs- und/oder Testumgebung als Teil der FIT-Abbildung kopiert werden. Zum Beispiel kann ein solches Kopieren mittels einer schnellen Kopieranwendung wie IBM FlashCopy durchgeführt werden. Fig. 6A zeigt ein Blockdiagramm, welches das Kopieren einer synthetischen Datenbank von einer virtuellen «Master» Entwicklungs- und/oder Testumgebung 601 in eine neue virtuelle Entwicklungs- und/oder Testumgebung in einer Ausführungsform des SYNONYM darstellt. In Fig. 6A wird Synbase-Version 1 mit Hilfe von FlashCopy von der virtuellen «Master» Entwicklungs- und/oder Testumgebung 601 in eine beständige Env1 virtuelle Entwicklungs- und/oder Testumgebung 602 kopiert, und in 450 beschriebene anwendungsspezifische Daten werden zum Datenspeicher hinzugefügt. Synbase-Version 3 wird mit Hilfe von FlashCopy von der virtuellen «Master» Entwicklungs- und/oder Testumgebung 601 in eine neue TED 1 virtuelle Entwicklungs- und/oder Testumgebung 603 kopiert und in 450 beschriebene anwendungsspezifische Daten werden zum Datenspeicher hinzugefügt.

[0061] In einer Ausführungsform können anwendungsspezifische Daten zu dem von der virtuellen Entwicklungs- und/oder Testumgebung verwendeten Datenspeicher hinzugefügt werden 450. In einer Implementierung können Wiki-Seiten, welche projektspezifische Informationen enthalten, ihre Daten in eine oder in mehreren Datenbanken speichern. In einer anderen Implementierung wird Quellcode in Programmiersprachen wie Java, PL/I und/oder dergleichen durch CVS hinzugefügt.

[0062] Fig. 5 zeigt ein logisches Flussdiagramm, welches eine Instantiierung von Benutzerkonten in einer Ausführungsform des SYNONYM darstellt. In Fig. 5 wird eine Anfrage zur Instantiierung von Benutzern empfangen 540. In einer Ausführungsform kann diese Anfrage auf die Eingaben eines Benutzers, der wie oben beschrieben eine neue Testumgebung angefordert hat, basieren, und es kann ein Set von Benutzern und jedem Benutzer zugeordnete Kontotypen spezifiziert werden. In einer Implementierung können Kontotypen normale Benutzer, Offshore-Benutzer und/oder dergleichen darstellen. Jeder Kontotyp kann verschiedene ihm zugeordnete Authentifizierungs- und Autorisierungseinstellungen aufweisen. Das spezifizierte Set von Benutzern erhält Zugangsrechte zur virtuellen Entwicklungs- und/oder Testumgebung 542, wobei Sicherheitszertifikate 544 zur Verwendung für eine Authentifizierung und/oder eine Autorisierung ausgegeben werden. In einer Implementierung können solche Sicherheitszertifikate auf Public Key Infrastructure basieren und von einer Zertifikatbehörde ausgegeben werden. In einer anderen Implementierung können solche Zertifikate auf SmartCards gespeichert werden. In einer Ausführungsform kann der Zugang zu Ressourcen auf Benutzer beschränkt sein, die diese

Ressourcen für ihre Arbeit brauchen. Folglich können Benutzer mit demselben Kontotyp Zugang zu unterschiedlichen Ressourcen haben. In einer Implementierung können Zugangsrechte zu verschiedenen, gut definierten Zeitpunkten (z.B. zum Projektabschluss) oder nach einer vorgegebenen Zeitspanne (z.B. von 6 Monaten) entzogen werden.

[0063] Eine First-Installation-Tool(FIT)-Abbildung ist eine Sammlung von Software, welche dazu verwendet werden kann, für jeden Computer in einer Organisation dasselbe Basisset von Software bereitzustellen. In einer Ausführungsform kann eine FIT-Abbildung ein Betriebssystem wie Windows XP; einen Web-Browser wie Microsoft Internet Explorer; einen Antivirus wie McAfee Antivirus; VPN Software wie Checkpoint VPN Connectivity and Integrity Client; Zugangssoftware wie IBM Personal Communications; und ein Set von Büroproduktivitätssoftware wie Microsoft Word, Microsoft Excel, Microsoft PowerPoint und Microsoft Outlook enthalten. In einer Implementierung können Entwicklungsprogramme, die für Entwicklungs- und Testprojekte benötigt werden, als vorgepackte Bündel von FIT (z.B. TIPD, SSH Clients, 3270 Terminalemulation, Rational Software, Eclipse, Clear Case, Clear Quest und/oder dergleichen) vorhanden sein. In einer Implementierung kann die in einer FIT-Abbildung enthaltene Software in Abhängigkeit von vordefinierten Profilen unterschiedlich sein. Zum Beispiel können solche Profile ein Profil für einen Server, einen Desktop-Computer, ein Profil für einen Notebook-Computer, ein Profil für ein mobiles Gerät und/oder dergleichen enthalten. In einer Implementierung können die Profile wie unten spezifiziert werden:

[0064]

```
<XML>
    <PROFILE>
        <TYPE>DESKTOP</TYPE>
        <SOFTWARE>WindowsXP</SOFTWARE>
        <SOFTWARE>Microsoft_OfficeXP</SOFTWARE>
    </PROFILE>

        <PROFILE>
            <TYPE>MOBILE</TYPE>
            <SOFTWARE>WindowsMobile6.0</SOFTWARE>
            <SOFTWARE>MicrosoftOfficeMobile</SOFTWARE>
        </PROFILE>
    </XML>
```

[0065] Zusätzliche Software kann auf Wunsch auch auf einem Computer mit einer FIT-Abbildung installiert werden. Solche zusätzliche Software kann Programme wie Microsoft Visio, Microsoft Project, Eclipse und/oder dergleichen umfassen. Es kann bei 546 festgestellt werden, welche Benutzer im spezifizierten Set von Benutzern normale Benutzer und welche Offshore-Benutzer sind. In einer Ausführungsform können lokale Benutzer eine FIT-Abbildung zusammen mit jeder zusätzlichen Software mittels eines Ferninstallationsprogramm wie Windows Deployment Services auf ihrem Computer 560 installiert haben, und sie können Zugriff auf SYNONYM haben. In einer anderen Ausführungsform kann eine VMware-Festplattenabbildung zum Packen und verteilen einer FIT-Abbildung verwendet werden. Zum Beispiel können Software Anwendungen, welche in die FIT-Abbildung einbezogen werden sollen, als VMware ThinApps gepackt werden und in die Vmware-Festplattenabbildung integriert werden. Zusätzlich benötigte Anwendungen können mit zusätzlichen ThinApps-Installationen über Basisinstallationen (z.B. Desktop, Mobil, usw.) bereitgestellt werden. Die Vmware-Festplattenabbildung kann dann an eine Workstation mittels VMware-Software verteilt werden. In einer Ausführungsform können sich von Offshore-Benutzern verwendete Computer in einem sicheren Offshore-Entwicklungszentrum (ODC), welches bei einem Drittanbieter untergebracht sein kann, befinden. In einer Implementierung sind Computer in einem ODC physisch und/oder logisch von jedem anderen Computer getrennt, und der Zugang zu diesen Computern ist auf autorisiertes Personal beschränkt. Zum Beispiel kann ein ODC in physisch getrennten Räumen mit physischer Zugangskontrolle und dedizierten Anlagen und Ressourcen installiert sein. Das Netzwerk des ODC muss nicht mit dem übrigen Netzwerk des Drittanbieters verbunden sein. Das ODC kann mit dem DvSZ mittels VPN verbunden sein und Offshore-Benutzer, die nicht im ODC arbeiten, haben unter Umständen keinen Zugang zum DvSZ. Fig. 6D zeigt ein Blockdiagramm, welches die physische Zugangskontrolle zum ODC in einer Ausführungsform des SYNONYM darstellt. In Fig. 6D ist das ODC 630 physisch vom Rest des Sitzes des Drittanbieters 631 getrennt. Nur Computern innerhalb des ODC 630 wird erlaubt, sich mit dem DvSZ 632 zu verbinden, und dies nur durch einen VPN-Tunnel 633.

[0066] Offshore-Benutzer können eine virtuelle FIT-Abbildung zusammen mit jeder zusätzlichen Software auf ihrem Computer installiert haben 548. In einer Implementierung kann Software auf den Computern von Offshore-Benutzern ein Be-

triebssystem wie Windows XP; eine Virtualisierungssoftware wie VMware Player; und eine FIT-Abbildung, welche durch die Virtualisierungssoftware (virtuelle FIT-Abbildung) verwendet wird, enthalten. In einer Implementierung kann die virtuelle FIT-Abbildung mittels eines sich im ODC befindlichen Code Servers verteilt werden. Ein Code Server kann dazu dienen, die FIT-Abbildung auf Workstations in einem ODC zu installieren, zusätzliche Software an die Workstations zu verteilen und Updates (z.B. Virusdefinitionsdatei-Updates für ein Antivirus Programm) für auf den Workstations installierten Software bereitzustellen. Zum Beispiel kann der Code Server ein Betriebssystem wie Windows XP und eine Virtualisierungssoftware wie VMware Player auf einer Workstation installieren. VMware Player kann dann dazu verwendet werden, eine virtuelle FIT-Abbildung, welche in einer VMware-Festplattenabbildung auf der Workstation gepackt ist, auszuführen. Ein VPN-Tunnel kann bei 550 vom ODC zur virtuellen Entwicklungs- und/oder Testumgebung in der DvSZ eingerichtet werden, was den Zugang von Offshore-Benutzer zu SYNONYM erlaubt. In einer Implementierung kann die Authentifizierung der VPN-Parteien auf Systemebene und nicht auf Benutzerebene geschehen. In einer Implementierung können Offshore-Benutzer nur zu Diensten innerhalb der DvSZ Zugang haben. Fig. 6B zeigt ein Blockdiagramm, welches die Entwickler-Serverzone 610 und die Offshore-Entwicklungszentren 611A–C in einer Ausführungsform des SYNONYM darstellt. In Fig. 6B ist eine DvSZ 610, welche vom Rest des Firmenintranets 612 getrennt ist, mit den ODCs 1–3 durch VPN-Tunnel 613A–C verbunden. Die sich in den ODCs 614A–F befindliche Workstations können mit Diensten innerhalb der DvSZ mittels der von den Code Servern 615A–C erhaltenen, installierten FIT-Abbildungen, kommunizieren. Die Workstations können nicht auf Dienste im Firmenintranet ausserhalb des DvSZ zugreifen.

SYNONYM-Controller

[0067] Fig. 7 zeigt erfinderische Aspekte eines SYNONYM-Controllers 701 in einem Blockdiagramm. In dieser Ausführungsform kann der SYNONYM-Controller 701 dazu dienen, Interaktionen mit einem Computer mit Hilfe einer Vielzahl von Informationstechnologien und/oder anderer verwandter Daten zu aggregieren, verarbeiten, speichern, suchen, betreuen, identifizieren, instruieren, generieren, abgleichen und/oder ermöglichen.

[0068] Typischerweise können Benutzer, die Personen und/oder andere Systeme sein können, sich Informationstechnologiesystemen (z.B. Computer) bedienen, um die Verarbeitung von Informationen zu ermöglichen. Wiederum verwenden Computer Prozessoren, um Informationen zu verarbeiten; solche Prozessoren 703 können zentrale Recheneinheiten (CPU) genannt werden. Eine Art von Prozessor wird Mikroprozessor genannt. CPUs verwenden kommunizierende Schaltungen, um binär kodierte Signale, die als Instruktionen fungieren, zu führen, um verschiedene Operationen zu ermöglichen. Diese Instruktionen können operationale und/oder Dateninstruktionen sein, die andere Instruktionen und Daten in verschiedenen Speicherbereichen 729 (z.B. Register, Cache Speicher, Arbeitsspeicher, usw.), zu denen der Prozessor Zugang hat und mit welchen er Operationen durchführen kann, enthalten und/oder referenzieren können. Solche kommunikative Instruktionen können in Bündeln (z.B. Instruktionsbündeln) als Programme und/oder Datenkomponenten gespeichert und/oder gesendet werden, um gewünschte Operationen zu ermöglichen. Diese gespeicherten Instruktionscodes, z.B. Programme, können von den Schaltungskomponenten der CPU und von anderen Motherboard- und/oder Systemkomponenten Gebrauch machen, um gewünschte Operationen durchzuführen. Ein Typ von Programm ist ein Betriebssystem, welches von der CPU auf einem Computer ausgeführt werden kann; das Betriebssystem ermöglicht und erleichtert den Zugriff auf und den Betrieb von Computerinformationstechnologien und Ressourcen durch den Benutzer. Manche Ressourcen, welche in Informationstechnologiesystemen eingesetzt werden können, sind: Ein- und Ausgabemechanismen, durch welche Daten in den und aus dem Computer wandern können; Speicher, in welchen Daten gespeichert werden können; und Prozessoren, von welchen Informationen verarbeitet werden können. Diese Informationstechnologiesysteme können zur Sammlung von Daten für spätere Abfragen, Analyse und Manipulation verwendet werden, was durch ein Datenbankprogramm ermöglicht werden kann. Diese Informationstechnologiesysteme stellen Schnittstellen bereit, die Benutzern den Zugriff und Betrieb verschiedener Systemkomponenten erlauben.

[0069] In einer Ausführungsform kann der SYNONYM-Controller 701 mit Einheiten wie: einem oder mehreren Benutzern von Benutzereingabevorrichtungen 711; Peripheriegeräten 712; einem optionalen Verschlüsselungsprozessor 728; und/oder Kommunikationsnetzwerken 713, jedoch nicht darauf beschränkt, verbunden sein und/oder damit kommunizieren.

[0070] Netzwerke sind normalerweise dazu gedacht, die Zusammenschaltung und Zusammenarbeit von Clients, Servern und Zwischenknoten in einer Graphenstruktur zu beinhalten. Es sollte angemerkt werden, dass sich der Begriff «Server», so wie er in dieser gesamten Anmeldung verwendet wird, generell auf einen Computer, auf ein anderes Gerät, Programm oder Kombination davon bezieht, welche die Anfragen von Remote-Benutzern über ein Kommunikationsnetzwerk verarbeitet und auf diese antwortet. Server stellen ihre Informationen anfragenden «Clients» zur Verfügung. Der Begriff «Client», wie hier verwendet, bezieht sich generell auf einen Computer, Programm, auf ein anderes Gerät, Benutzer und/oder eine Kombination davon, welche in der Lage ist, jegliche Antworten vom Server über ein Kommunikationsnetzwerk zu verarbeiten, Anfragen zu tätigen und die Antworten zu bekommen und zu verarbeiten. Ein Computer, ein anderes Gerät oder eine Kombination davon, welche Informationen verarbeitet, ermöglicht und anfragt und/oder sie von einem Quellenbenutzer an einen Zielbenutzer weitergibt, wird normalerweise «Knoten» genannt. Netzwerke sind normalerweise dazu gedacht, den Transfer von Informationen von Quellenpunkten zu Zielen zu ermöglichen. Ein Knoten, der insbesondere die Aufgabe hat, den Informationsfluss von einer Quelle zu einem Ziel weiterzuleiten, wird normalerweise «Router» genannt. Es gibt viele Arten von Netzwerken wie z.B. Local Area Networks (LANs), Pico-Netzwerke, Wide-Area-Netzwerke (WANs),

Drahtlose Netzwerke (WLANs), usw. Zum Beispiel wird das Internet generell als eine Zusammenschaltung einer Vielzahl von Netzwerken, wobei Remote-Clients und Server aufeinander zugreifen und miteinander arbeiten können, akzeptiert.

[0071] Der SYNONYM-Controller 701 kann auf Computersystemen basieren, welche Komponenten wie: ein Computersystem 702, welche mit dem Speicher 729 verbunden ist, umfassen kann, ist auf diese jedoch nicht beschränkt.

Computersystem

[0072] Ein Computersystem 702 kann einen Taktgeber 730, eine zentrale Recheneinheit («CPU(s)» und/oder «Prozessor(en)» (diese Begriffe sind überall in der Offenbarung austauschbar, wenn nicht anders angegeben)) 703, einen Speicher 729 (z.B. einen Festwertspeicher (ROM) 706, einen Arbeitsspeicher (RAM) 705, usw.), und/oder einen Schnittstellenbus 707 umfassen, und meistens, obwohl nicht notwendig, sind alle miteinander verbunden und/oder kommunizieren über einen Systembus 704 auf einer oder mehreren (Mother)board(s) 702, mit leitfähigen und/oder auf andere Weise transportierenden Schaltungswegen, durch welche Instruktionen (z.B. binär kodierte Signale) laufen können, um Kommunikationen, Operationen, Speichervorgänge, usw. zu bewirken. Optional kann das Computersystem an einer internen Leistungsquelle 786 angeschlossen sein. Optional kann ein Verschlüsselungsprozessor 726 am Systembus angeschlossen sein. Der Systemtaktgeber weist typischerweise einen Kristalloszillator und generiert ein Basissignal durch die Schaltungen des Computersystems. Der Systemtaktgeber ist typischerweise mit dem Systembus und verschiedenen Takt-Multiplikatoren, welche die Basisbetriebsfrequenz für andere im Computersystem zugeschaltete Komponenten erhöhen oder vermindern, gekoppelt. Der Taktgeber und verschiedene Komponenten in einem Computersystem treiben Signale, welche Informationen durch das gesamte System verkörpern. Ein solches Senden und Empfangen von Informationen verkörpernde Instruktionen durch ein gesamtes Computersystem kann normalerweise Kommunikation genannt werden. Diese kommunizierenden Instruktionen können weiter gesendet, empfangen und können ein Zurückschicken und/oder eine Antwortkommunikation jenseits des aktuellen Computersystems an: Kommunikationsnetzwerken, Eingabegeräten, anderes Computersystem, Peripheriegeräte und/oder dergleichen bewirken. Sicher können alle der obigen Komponenten direkt miteinander, mit der CPU verbunden sein und/oder in einer Vielzahl von Variationen organisiert werden, wie von einer Vielzahl von Computersystemen beispielhaft gezeigt.

[0073] Die CPU umfasst mindestens einen Hochgeschwindigkeitsdatenprozessor, welcher zur Ausführung von Programmkomponenten zur Ausführung von Benutzer- und/oder Systemgenerierten Anfragen geeignet ist. Oft umfassen die Prozessoren selbst verschiedene spezialisierte Recheneinheiten wie z.B: integrierte System(bus)-Controller, Steuereinheiten für Speicherverwaltung, Gleitkommaeinheiten, und sogar spezialisierte Rechenuntereinheiten wie Grafikrecheneinheiten, digitale Signalverarbeitungseinheiten und/oder dergleichen, wobei sie nicht darauf beschränkt sind. Zusätzlich können Prozessoren interne adressierbare Schnellzugriffsspeicher enthalten und imstande sein, ausserhalb des Prozessors selbst Speicher 529 abzubilden und zu adressieren; ein interner Speicher kann: schnelle Register, verschiedene Stufen von Cache-Speicher (z.B. Stufe 1, 2, 3, usw.), RAM, usw. umfassen, ist aber nicht darauf beschränkt. Der Prozessor kann auf diesen Speicher durch die Verwendung eines Adressraumes, welcher mittels einer Adresseninstruktion zugänglich ist, welche vom Prozessor aufgebaut und dekodiert wird, zugreifen, was ihm erlaubt, Zugriff auf einen Schaltungspfad zu einem spezifischen Speicherbereich mit einem Speicherzustand zu erlangen. Die CPU kann ein Mikroprozessor wie: Athlon, Duron und/oder Opteron von AMD; Applikationsprozessoren, eingebettete Prozessoren und Sichere Prozessoren von ARM; DragonBall und PowerPC von IBM und/oder Motorola; Zellenprozessoren von IBM und Sony; Celeron, Core (2) Duo, Itanium, Pentium, Xeon und/oder Xscale von Intel und/oder ähnliche Prozessor(en) sein. Die CPU interagiert mit dem Speicher mittels Instruktionen, welche durch leitfähige und/oder transportierende Leitungen (z.B. (gedruckte) elektronisch und/oder optische Schaltungen), um gespeicherte Instruktionen (z.B. Programmcode) gemäss üblichen Datenverarbeitungstechniken auszuführen. Ein solcher Instruktionstransport erleichtert die Kommunikation im SYNONYM-Controller und ausserhalb durch verschiedene Schnittstellen. Wenn Rechenvorschriften eine höhere Geschwindigkeit und/oder Kapazität auferlegen, können verteilte Prozessoren (z.B. verteilte SYNONYM), Mainframe, Multi-Kern, parallele und/oder Supercomputerarchitekturen auf die gleiche Weise verwendet werden. Alternativ, im Falle, dass Verteilungsanforderungen eine höhere Portabilität auferlegen, können kleinere Personal Digital Assistants (PDAs) verwendet werden.

[0074] Abhängig von verschiedenen Implementierungen können Merkmale des SYNONYM durch die Implementierung eines Mikrocontrollers wie den R8051XC2 Mikrocontroller von CAST; der MCS 51 (z.B. 8051 Mikrocontroller); und/oder dergleichen erreicht werden; und/oder dergleichen. Um auch bestimmte Merkmale des SYNONYM zu implementieren, können manche Merkmale auf eingebettete Komponenten wie; anwendungsspezifische integrierte Schaltungen (ASIC), Digital Signal Processing («DSP»), Field Programmable Gate Arrax («FPGA») und/oder auf ähnlichen eingebetteten Technologien basieren. Zum Beispiel kann jede der SYNONYM-Komponentenkollektion (verteilt oder dergleichen) und/oder können Merkmale über den Mikroprozessor und/oder über eingebettete Komponenten; implementiert sein, z.B: ASIC, Koprozessor, DSP, FPGA und/oder dergleichen. Alternativ können manche Implementierungen des SYNONYM mittels eingebetteten Komponenten, welche derart konfiguriert und verwendet werden, dass eine Vielzahl von Merkmalen oder Signalprozesse erreicht werden, implementiert werden.

[0075] Abhängig von der jeweiligen Implementierung können die eingebetteten Komponenten Softwarelösungen, Hardwarelösungen und/oder einige Kombinationen beider Software- und Hardwarelösungen umfassen. Zum Beispiel können hier beschriebene SYNONYM-Merkmale durch die Implementierung von FPGAs, welche Halbleiterbausteine sind, die programmierbare Logikkomponenten sind, «logische Blöcke» genannt, und durch programmierbare Verbindungen, wie die

Hochleistungs-FPGA-Virtex-Reihen und/oder die kostensparenden Spartan-Reihen, welche von Xilinx hergestellt werden, erreicht werden. Logikblöcke und Verbindungen können vom Kunden oder Designer programmiert werden, nachdem das FPGA hergestellt wurde, um jegliche SYNONYM-Merkmale zu implementieren. Eine Hierarchie programmierbarer Verbindungen erlauben Logikblöcken, so miteinander verbunden zu werden, wie vom SYNONYM Systemdesigner/Administrator benötigt, ähnlich wie eine Ein-Chip-programmierbare Lochrasterplatte. Ein FPGA-Logikblock kann so programmiert werden, dass er die Funktion von Basislogikblöcken wie AND und XOR oder von komplexeren kombinatorischen Funktionen wie Decoder oder einfache mathematische Funktionen ausführen kann. In den meisten FPGAs umfassen die Logikblöcke auch Speicherelemente, welche einfache Flip-Flops oder umfangreichere Speicherblöcke sein können. Unter gewissen Umständen kann SYNONYM auf regulären FPGAs entwickelt und dann in eine feste Version, welche mehr ASIC-Implementierungen ähnelt, migriert werden. Alternative oder koordinierende Implementierungen können SYNONYM-Controller Funktionen auf ein finales ASIC, anstelle der oder zusätzlich zu den FPGAs migrieren. Abhängig von der Implementierung können alle bereits genannten eingebetteten Komponenten und Mikroprozessoren als die «CPU» und/oder «Prozessor» für das SYNONYM aufgefasst werden.

Leistungsquelle

[0076] Die Leistungsquelle 786 kann jede Standardausführung zur Versorgung kleiner elektronischer Leiterplattenanordnungen, wie z.B. die folgenden Kraftzellen: alkalische, Lithium-Hybrid, Lithium-Ion, Lithium-Polymer, Nickel-Cadmium, Solarzellen und/oder dergleichen, sein. Andere Typen von AC- oder DC-Leistungsquellen können auch verwendet werden. Im Falle von Solarzellen stellt das Gehäuse in einer Implementierung eine Öffnung bereit, durch welche die Solarzelle Photonenenergie einfangen kann. Die Kraftzelle 786 ist mit mindestens mit einer der folgenden, miteinander geschalteten Komponenten des SYNONYM verbunden, wodurch sie einen elektrischen Strom aller nachfolgenden Elementen abgibt. In einem Beispiel ist die Kraftzelle 786 mit der Systembuskomponente 704 verbunden. In einer alternativen Ausführungsform wird eine äussere Leistungsquelle 786 mittels einer Verbindung über die I/O-Schnittstelle 708 bereitgestellt. Zum Beispiel trägt eine USB- und/oder IEEE-1394-Verbindung beides, Daten und Leistung, über die Verbindung und ist daher eine geeignete Leistungsquelle.

Schnittstellenadapter

[0077] Schnittstellenbus(se) 707 können eine Menge von Schnittstellenadaptern, jedoch normalerweise nicht notwendigerweise in der Form von Adapterkarten, annehmen, sich damit verbinden und/oder damit kommunizieren, wie z.B: Eingabe/Ausgabe-Schnittstellen (I/O) 708, Speicherschnittstellen 709, Netzwerkschnittstellen 710 und/oder dergleichen, jedoch nicht darauf beschränkt. Optional können Verschlüsselungsprozessorschnittstellen 727 auch mit dem Schnittstellenbus verbunden sein. Der Schnittstellenbus stellt die Kommunikation von Schnittstellenadaptern untereinander sowie mit anderen Komponenten des Computersystems sicher. Schnittstellenadapter sind für einen kompatiblen Schnittstellenbus ausgestaltet. Schnittstellenadapter verbinden sich normalerweise mit dem Schnittstellenbus über eine Slot-Architektur. Übliche Slot-Architekturen können als: Accelerated Graphics Port (AGP), Card Bus, (Extended) Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus, Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal Computer Memory Card International Association (PCMCIA) und/oder dergleichen, jedoch nicht darauf beschränkt, eingesetzt werden.

[0078] Speicherschnittstellen 709 können eine Menge von Speichergeräten annehmen, damit kommunizieren und/oder sich damit verbinden, wie z.B: Speichergeräte 714, Wechselfestplatten und/oder dergleichen, jedoch nicht darauf beschränkt. Speicherschnittstellen können Verbindungsprotokolle verwenden, wie: (Ultra) (Serial) Advanced Technology Attachment (Packet Interface) ((Ultra) (Serial(ATA(PI)), (Enhanced) Integrated Drive Electronics ((E)IDE), Institute of Electrical and Electronics Engineers (IEEE) 1394, Fiberglass, Small Computer Systems Interface (SCSI), Universal Serial Bus (USB) und/oder dergleichen, jedoch nicht darauf beschränkt.

[0079] Netzwerkschnittstellen 710 können ein Kommunikationsnetzwerk 713 annehmen, damit kommunizieren und/oder sich damit verbinden. Durch ein Kommunikationsnetzwerk 713 kann auf den SYNONYM-Controller durch Benutzer 733a mittels Remote-Clients 733b (z.B. Computer mit Web-Browsern) zugegriffen werden. Netzwerkschnittstellen können Verbindungsprotokolle wie: direkte Verbindung, Ethernet (dick, dünn, verdrillt 10/100/1000 Base T, und/oder dergleichen), TokenRing, drahtlose Verbindung wie IEEE 802.11a–x, und/oder dergleichen, jedoch nicht darauf beschränkt. Wenn Verarbeitungsanforderungen eine höhere Geschwindigkeit und/oder Kapazität erfordern, können verteilte Netzwerk-Controller (z.B. verteilte SYNONYM), Architekturen auf die selbe Weise eingesetzt werden, um die vom SYNONYM-Controller benötigte Kommunikationsbandbreite zu vereinigen, eine Lastverteilung durchzuführen und/oder auf andere Weise zu erhöhen. Ein Kommunikationsnetzwerk kann eines von und/oder eine Kombination von: einer direkten gegenseitigen Verbindung; dem Internet; einen Local Area Network (LAN); einem Metropolitan Area Network (MAN); einem Operating Missions als Knoten im Internet (OMNI); einer gesicherten proprietären Verbindung; einem Wide Area Network (WAN); einem drahtlosen Netzwerk (z.B. durch Verwendung von Protokollen wie Wireless Application Protocol (WAP), I-Mode und/oder dergleichen, aber nicht darauf beschränkt); und/oder dergleichen sein. Eine Netzwerkschnittstelle kann als eine spezialisierte Art einer Eingabe/Ausgabe-Schnittstelle angesehen werden. Weiter können mehrere Netzwerkschnittstellen 710 in Wechselwirkung mit mehreren Kommunikationsnetzwerkstypen 713 treten. Zum Beispiel können mehrere Netzwerkschnittstellen dazu eingesetzt werden, die Kommunikation über Rundfunk-, Multicast- und/oder Unicastnetzwerke zu ermöglichen.

[0080] Eingabe/Ausgabe-Schnittstellen (I/O) 708 können Benutzereingabegeräte 711, Peripheriegeräte 712, Verschlüsselungsprozessoren 728 und/oder dergleichen annehmen, damit kommunizieren und/oder sich damit verbinden. I/O kann Verbindungsprotokolle wie: Audio: Analog, Digital, Mono, RCA, Stereo und/oder dergleichen; Daten: Apple Desktop Bus (ADB), IEEE 1394a-b, seriell, Universal Serial Bus (USB); Infrarot; Joystick; Tastatur; Midi; optisch; PC AT; PS/2; parallel; Radio; Video Schnittstelle: Apple Desktop Connector (ADC), BNC, koaxial, Komponent, Komposit, Digital, Digital Visual Interface (DVI), High-definition Multimedia Schnittstelle (HDMI), RCA, HF Antenne, S-Video, VGA und/oder dergleichen; drahtlos: 802.11a/b/g/n/x, Bluetooth, Code division multiple access (CDMA), global system for mobile Communications (GSM), WiMax, usw; und/oder dergleichen annehmen, damit kommunizieren und/oder sich damit verbinden. Ein typisches Ausgabegerät kann ein Videodisplay umfassen, das typischerweise ein Cathode Ray Tube (CRT) oder einen Liquid Crystal Display (LCD)-basierten Monitor mit einer Schnittstelle (z.B. DVI Verschaltung und Kabeln), welche Signale von einer Video-Schnittstelle akzeptiert, welche verwendet werden können, umfasst. Die Video-Schnittstelle verbindet Informationen, die von einem Computersystem generiert werden und generiert auf die Verbundinformationen basierende Videosignale in einem Videospeicherframe. Ein anderes Ausgabegerät ist ein Ferseher, welcher Signale von einer Videoschnittstelle annimmt. Typischerweise stellt die Videoschnittstelle die Verbundvideoinformation durch eine Videoverbindungsschnittstelle, welche eine Videodisplayschnittstelle annimmt (z.B. einen RCA-Verbundvideokonnektor, welcher ein RCA-Verbundvideokabel annimmt; einen DVI-Konnektor, welcher ein DVI-Displaykabel annimmt, usw.).

[0081] Benutzereingabegeräte 711 können Kartenleser, Dongles, Fingerabdruckleser, Handschuhe, Grafiktabletts, Joysticks, Tastaturen, Mäuse, Fernbedienungen, Netzhautleser, Trackballs, Trackpads und/oder dergleichen sein.

[0082] Peripheriegeräte 712 können mit I/O und/oder anderen derartigen Einrichtungen wie Netzwerkschnittstellen, Speicherschnittstellen und/oder dergleichen verbunden sein und/oder kommunizieren. Peripheriegeräte können Audiogeräte, Kameras, Dongles (z.B. für Kopierschutz, zur Sicherstellung von sicheren Transaktionen mit einer digitalen Signatur und/oder dergleichen), externe Prozessoren (für zusätzliche Funktionalität), Brillen, Mikrophone, Monitore, Netzwerkschnittstellen, Drucker, Scanner, Speichergeräte, Videogeräte, Videoquellen, Visiere und/oder dergleichen sein.

[0083] Es wird angemerkt, dass obwohl Benutzereingabegeräte und Peripheriegeräte eingesetzt werden können, der SYNONYM-Controller als eingebettet, proprietär, und/oder monitorloses (z.B. Headless) Gerät ausgeführt sein kann, wobei der Zugang über eine Netzwerkschnittstellenverbindung bereitgestellt werden würde.

[0084] Verschlüsselungseinheiten wie Mikrocontroller, Prozessoren 726, Schnittstellen 727 und/oder Geräte 728, jedoch nicht darauf beschränkt, können an den SYNONYM-Controller angeschlossen werden und/oder damit kommunizieren. Ein MC68HC16-Mikrocontroller, welcher von Motorola Inc. hergestellt wird, kann für und/oder in Verschlüsselungseinheiten verwendet werden. Der MC68HC16-Mikrocontroller verwendet eine 16-Bit-Multiplikations- und Summieroperation in der 16-MHz-Konfiguration und benötigt weniger als eine Sekunde, um eine 512-Bit-RSA-Privatschlüsseloperation auszuführen. Verschlüsselungseinheiten unterstützen die Authentifizierung der Kommunikation für interagierende Parteien und sie erlauben anonyme Transaktionen. Verschlüsselungseinheiten können auch als Teil der CPU konfiguriert werden. Äquivalente Mikrocontroller und/oder Prozessoren können auch verwendet werden. Andere herkömmliche, spezialisierte Verschlüsselungsprozessoren umfassen: CryptoNetX von Broadcom und andere Sicherheitsprozessoren; nShield von nCipher, Luna PCI (z.B. 7100) Reihen von SafeNet; 40-MHz-Roadrunner 184 von Semaphore Communications; Cryptographic Accelerators (z.B. Accelerator 6000 PCIe Board, Accelerator 500 Daughtercard) von Sun; Via Nano Processor (z.B. L2100, L2200, U2400) Linie, welche imstande ist, 500+ MB/s an Verschlüsselungsinstruktionen zu verarbeiten; 33 Mhz 6868 von VLSI Technology; und/oder dergleichen.

Speicher

[0085] Im Allgemeinen wird jede Mechanisierung und/oder Ausführungsform, welche einem Prozessor erlaubt, die Speicherung und/oder das Abrufen von Informationen zu beeinflussen, als Speicher 729 angesehen. Speicher ist jedoch eine ersetzbare Technologie und Ressource, so dass jede Anzahl von Speicherausführungsformen anstatt von anderen oder zusammen eingesetzt werden kann. Es muss verstanden werden, dass der SYNONYM-Controller und/oder ein Computersystem verschiedene Formen von Speicher 729 einsetzen kann. Zum Beispiel kann ein Computersystem so konfiguriert werden, dass die Funktionalität von On-Chip CPU Speicher (z.B. Register), RAM, ROM und jedes andere Speichergerät durch ein Lochkartenband oder einem Lochkartenmechanismus bereitgestellt wird; sicher würde eine solche Ausführungsform eine extrem langsame Ausführung von Operationen bedeuten. In einer typischen Konfiguration umfasst der Speicher 729 ROM 706, RAM 705 und ein Speichergerät 714. Ein Speichergerät 714 kann jeder herkömmliche Computersystemspeicher sein. Speichergeräte können eine Trommel umfassen; ein (festes und/oder austauschbares) magnetisches Laufwerk; ein magneto-optisches Laufwerk; ein optisches Laufwerk (z.B. Blueray, CDROM/RAM/Recordable (R)/ReWritable (RW), DVD R/RW, HD DVD R/RW, usw.); ein Array von Geräten (z.B. Redundant Array of Independent Disks (RAID)); Solid-State-Speichergeräte (USB Speicher, Solid-State-Laufwerke (SSD), usw.); andere Speichermedien, die Prozessorlesbar sind; und/oder andere derartige Geräte. Also benötigt und verwendet ein Computersystem normalerweise Speicher.

Komponentensammlung

[0086] Der Speicher 729 kann eine Sammlung von Programmen und/oder Datenbankkomponenten und/oder Daten enthalten, wie z.B: Betriebssystemkomponente(n) 715 (Betriebssystem); Informationsserverkomponente(n) 716

(Informationsserver); Benutzerschnittstellenkomponente(n) 717 (Benutzerschnittstelle); Web-Browser-Komponente(n) 718 (Web-Browser); Datenbank(en) 719; Mailserverkomponente(n) 721; Mailclientkomponente(n) 722; Verschlüsselungsserverkomponente(n) 720 (Verschlüsselungsserver); die SYNONYM-Komponente(n) 735; und/oder dergleichen (z.B. insgesamt eine Komponentensammlung). Diese Komponenten können gespeichert, und es kann auf sie von den Speichergeräten und/oder von Speichergeräten, auf die mittels eines Schnittstellenbusses zugegriffen werden kann, zugegriffen werden. Obwohl nicht konventionelle Programmkomponenten, so wie diejenigen in der Komponentensammlung, typischerweise in einem lokalen Speichergerät 714 gespeichert werden, können sie auch in einem Speicher wie: Peripheriegeräten, RAM, Fernspeichereinrichtungen durch ein Kommunikationsnetzwerk, ROM, verschiedene Formen von Speicher und/oder dergleichen geladen und/oder gespeichert werden.

Betriebssystem

[0087] Die Betriebssystemkomponente 715 ist eine ausführbare Programmkomponente, welche den Betrieb des SYNONYM-Controllers ermöglicht. Typischerweise ermöglicht das Betriebssystem den Zugang des I/O, Netzwerkschnittstellen, Peripheriegeräte, Speichergeräte und/oder dergleichen. Das Betriebssystem kann ein in hohem Masse fehlertolerantes, skalierbares und sicheres System wie: Apple Macintosh OS X (Server); AT&T Plan 9; Be OS; Unix und Unix-ähnliche Systemvarianten (wie UNIX von AT&T; Berkley Software Distribution (BSD) Variationen wie FreeBSD, NetBSD, OpenBSD und/oder dergleichen; Linux Varianten wie Red Hat, Ubuntu und/oder dergleichen); und/oder ähnliche Betriebssysteme. Es können jedoch auch eingeschränktere und/oder weniger sichere Betriebssysteme wie Apple Macintosh OS, IBM OS/2, Microsoft DOS, Microsoft Windows 2000/2003/3.1/95/98/CE/Millenium/ NT/Vista/XP (Server), Palm OS und/oder dergleichen zum Einsatz kommen. Ein Betriebssystem kann mit anderen Komponenten in einer Komponentensammlung kommunizieren, einschliesslich mit sich selbst und/oder dergleichen. Meistens kommuniziert das Betriebssystem mit anderen Programmkomponenten, Benutzerschnittstellen und/oder dergleichen. Beispielsweise kann das Betriebssystem Programmkomponenten, Systeme, Benutzer und/oder Datenkommunikationen, Anfragen und/oder Antworten enthalten, damit kommunizieren, generieren, erhalten und/oder bereitstellen. Das Betriebssystem, einmal in der CPU ausgeführt, kann die Interaktion mit Kommunikationsnetzwerken, Daten, I/O, Peripheriegeräten, Programmkomponenten, Speicher, Benutzereingabegeräten und/oder dergleichen ermöglichen. Das Betriebssystem kann Kommunikationsprotokolle bereitstellen, welche dem SYNONYM-Controller erlauben, mit anderen Einheiten durch ein Kommunikationsnetzwerk 713 zu kommunizieren. Verschiedene Kommunikationsprotokolle können vom SYNONYM-Controller als ein Träger-Transportmechanismus für Interaktionen wie Multicast, TCP/IP, UDP, Unicast und/oder dergleichen, jedoch nicht darauf beschränkt, verwendet werden.

Informationsserver

[0088] Eine Informationsserverkomponente 716 ist eine gespeicherte Programmkomponente, die von der CPU ausgeführt wird. Der Informationsserver kann ein konventioneller Internet Informationsserver wie z.B. Apache, von Apache Software Foundation, Internet Information Server von Microsoft, und/oder dergleichen, jedoch nicht darauf beschränkt. Der Informationsserver kann die Ausführung von Programmkomponenten mittels Einrichtungen wie Active Server Page (ASP), ActiveX, (ANSI) (Objective-) C (++), C# und/oder .NET, Common Gateway Interface (CGI) Skripten, dynamic (D) hypertext markup languange (HTML), FLASH, Java, Javascript, Practical Extraction Report Language (PERL), Hypertext Pre-Processor (PHP), pipes, Python, wireless application protocol (WAP), WebObjects und/oder dergleichen erlauben. Der Informationsserver kann sichere Kommunikationsprotokolle wie File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure Hypertext Transfer Protocol (HTTPS), Secure Socket Layer (SSL), Nachrichtenprotokolle (z.B. America Online (AOL) Instant Messenger (AIM), Application Exchange (APEX), ICQ, Internet Relay Chat (IRC), Microsoft Network (MSN) Messenger Service, Presence and Instant Messaging Protocol (PRIM), Session Initiation Protocol (SIP) von Internet Engineering Task Force (IETF) , SIP für Instant Messaging and Presence Leveraging Extensions (SIMPLE), open XML-basierte Extensible Messaging and Presence Protocol (XMPP) (z.B. Jabber oder Instant Messaging and Presence Service (IMPS) von Open Mobile Alliance (OMA), Instant Messenger von Yahoo!), und/oder dergleichen, jedoch nicht darauf beschränkt, unterstützen. Der Informationsserver stellt Web-Browsern Ergebnisse in der Form von Web-Seiten bereit und erlaubt die manipulierte Generierung von Web-Seiten durch Interaktion mit anderen Programmkomponenten. Nach der Auflösung eines Teils einer HTTP-Anfrage, enthaltend ein Domain Name System (DNS), als einen bestimmten Informationsserver, löst der Informationsserver Anfragen nach Informationen an bestimmten Stellen auf dem SYNONYM-Controller auf, basierend auf den Rest der HTTP-Anfrage. Zum Beispiel könnte bei einer Anfrage wie http://123.124.125.126/myInformation.html der IP-Teil der Anfrage «123.124.125.126» von einem DNS Server als ein Informationsserver an dieser IP-Adresse aufgelöst werden; Dieser Informationsserver könnte hingegen weiter die http-Anfrage nach dem «/myInformation.html»-Teil der Anfrage absuchen und als eine Stelle im Speicher auflösen, welche die Information «myInformation.html» enthält. Zusätzlich können andere Informationen bereitstellende Protokolle über verschiedene Ports eingesetzt werden, wie z.B. FTP Kommunikation über Port 21 und/oder dergleichen. Ein Informationsserver kann mit anderen Komponenten in einer Komponentensammlung kommunizieren, einschliesslich mit sich selbst, und/oder mit derartigen Einrichtungen. Meistens kommunizieren der Informationsserver mit der SYNONYM-Datenbank 719, mit Betriebssystemen, mit anderen Programmkomponenten, Benutzerschnittstellen, Web-Browsern und/oder dergleichen.

[0089] Der Zugang zur SYNONYM-Datenbank kann durch eine Anzahl von Datenbank-Brückenmechanismen wie z.B. durch Skript-Sprachen, wie unten aufgezählt (z.B. CGI) und durch Kommunikationskanäle zwischen Anwendungen wie

unten aufgezählt (z.B. CORBA, WebObjects, usw.). Alle Datenanfragen durch einen Web-Browser werden mittels des Brückenmechanismus, mit Hilfe von geeigneter Grammatik, wie vom SYNONYM verlangt, abgesucht. In einer Ausführungsform würde der Informationsserver eine für einen Web-Browser zugängliche Web-Form bereitstellen. Eingaben in bereitgestellte Felder in der Web-Form werden derart gekennzeichnet, dass sie in den bestimmten Feldern eingegeben worden sind, und als solche abgesucht. Die eingegebenen Begriffe werden dann zusammen mit den Feldkennzeichnungen, welche dazu dienen, den Parser anzuweisen, die Generierung von Suchvorgängen an geeigneten Tabellen und/oder Feldern zu richten, weitergegeben. In einer Ausführungsform kann der Parser Suchvorgänge in Standard-SQL generieren, indem er ein Suchzeichen instantiiert, mittels der richtigen join/select-Befehlen, basierend auf die gekennzeichneten Texteingaben, wobei der resultierende Befehl über den Brückenmechanismus dem SYNONYM als eine Suche bereitgestellt wird. Nach der Generierung von Suchergebnissen werden die Ergebnisse über den Brückenmechanismus weitergegeben und können nach Formatierung abgesucht werden, wobei eine neue Ergebnis-Web-Seite vom Brückenmechanismus generiert wird. Eine solche neue Ergebnis-Web-Seite wird dann dem Informationsserver zur Verfügung gestellt, welcher sie dem anfragenden Web-Browser liefert.

[0090] Auch kann ein Informationsserver Programmkomponenten-, System-, Benutzer- und/oder Datenkommunikationsanfragen und/oder Antworten enthalten, damit kommunizieren, generieren, bekommen und/oder bereitstellen.

Benutzerschnittstelle

[0091] Die Funktion von Computerschnittstellen ist unter bestimmten Voraussetzungen ähnlich mit Automobilbetriebsschnittstellen. Elemente von Automobilbetriebsschnittstellen wie Lenkräder, Schaltungen und Tachos ermöglichen den Zugang, den Betrieb und die Anzeige von Automobilressourcen, Funktionalität und Status. Schnittstellenelemente zur Interaktion mit Computern, wie z.B. Kontrollkästchen, Zeiger, Menüs, Scroller und Fenster (insgesamt und normalerweise Widgets genannt) ermöglichen auf dieselbe Weise den Zugang, den Betrieb und die Anzeige von Daten und Computer-Hardware und Betriebssystemressourcen, Funktionalität und Status. Betriebsschnittstellen werden allgemein Benutzerschnittstellen genannt. Grafische Benutzerschnittstellen (GUIs) wie das Betriebssystem Aqua von Apple Macintosh, OS/2 von Microsoft, Windows 2000/2003/3.1/95/98/CE/Millenium/NT/XP/Vista/7 (z.B. Aero) von Microsoft, X-Windows von Unix (z.B. welches zusätzliche Unix Grafikschnittstellenbiblotheken und Ebenen wie K-Desktop Environment (KDE), mythTV und GNU Network Object Model Environment ((GNOME)), Web-Schnittstellenbibliotheken (z.B. Active X, AJAX, (D)HTML, FLASH, Java, Javascript, usw. Schnittstellenbibliotheken wie Dojo, jQuery(UI), MooTools; Prototype, Script.aculo.us, SW-Objects, Yahoo! Benutzerschnittstelle, jedoch nicht darauf beschränkt, alle welche verwendet werden können) stellen eine Basis und Mittel für den Zugang zu und die Anzeige von Informationen grafisch an den Benutzer.

[0092] Eine Benutzerschnittstellenkomponente 717 ist eine gespeicherte Programmkomponente, welche von der CPU ausgeführt wird. Die Benutzerschnittstelle kann eine konventionelle grafische Benutzerschnittstelle, wie von, mit und/oder auf Betriebssystemen und/oder Betriebsumgebungen, wie schon beschrieben, bereitgestellt. Die Benutzerschnittstelle kann die Anzeige, die Ausführung, die Interaktion, die Manipulation und/oder den Betrieb von Programmkomponenten und/oder Systemeinrichtungen durch textuelle und/oder grafische Einrichtungen erlauben. Die Benutzerschnittstelle stellt eine Einrichtung bereit, durch welche Benutzer ein Computersystem beeinflussen, damit interagieren und/oder es betreiben können. Eine Benutzerschnittstelle kann mit anderen Komponenten in einer Komponentensammlung kommunizieren, einschliesslich mit sich selbst und/oder anderen Einrichtungen. Meistens kommuniziert die Benutzerschnittstelle mit Betriebssystemen, anderen Programmkomponenten und/oder dergleichen. Die Benutzerschnittstelle kann Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen.

Web-Browser

[0093] Eine Web-Browser-Komponente 718 ist eine gespeicherte Programmkomponente, welche von der CPU ausgeführt wird. Der Web-Browser kann eine konventionelle Anwendung zur Darstellung von Hypertext, wie Microsoft Internet Explorer oder Netscape Navigator sein. Sicheres Internet-Surfen kann mit einer 128-Bit(oder mehr)-Verschlüsselung mittels HTTPS, SSL und/oder dergleichen ausgestattet werden. Web-Browser erlauben die Ausführung von Programmkomponenten durch Einrichtungen wie ActiveX, AJAX, (D)HTML, FLASH, Java, JavaScript Web-Browser Plug-in APIs (z.B. Firefox, Safari Plug-in und/oder dergleichen) und/oder dergleichen. Web-Browser und ähnliche Tools für das Zugreifen auf Informationen können in PDAs, Mobiltelefonen und/oder anderen mobilen Geräten integriert werden. Ein Web-Browser kann mit anderen Komponenten in einer Komponentensammlung und/oder ähnlichen Einrichtungen kommunizieren, einschliesslich mit sich selbst. Meistens kommuniziert der Web-Browser mit Informationsservern, Betriebssystemen, integrierten Programmkomponenten (z.B. Plug-ins) und/oder dergleichen; z. B. kann er Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen. Natürlich kann anstelle von Web-Browsern und Informationsservern eine kombinierte Anwendung entwickelt werden, welche dieselben Funktionen beider ausführen kann. Die kombinierte Anwendung würde auf gleiche Weise die Beschaffung und die Bereitstellung von Informationen an Benutzern, User-Agents und/oder dergleichen von den vom SYNONYM aktivierten Knoten beeinflussen. Die kombinierte Anwendung kann auf Systemen, die Standard Web-Browser einsetzen wertlos sein.

Mailserver

[0094] Eine Mailserverkomponente 721 ist eine gespeicherte Programmkomponente, welche von einer CPU 703 ausgeführt wird. Der Mailserver kann ein gewöhnlicher Internet Mailserver wie sendmail, Microsoft Exchange oder dergleichen sein, ist jedoch nicht darauf beschränkt. Der Mailserver kann die Ausführung von Programmkomponenten durch Einrichtungen wie ASP, ActiveX, (ANSI) (Objective-)C(++), C# und/oder .NET, CGI-Skripten, Java, Javascript, PERL, PHP, pipes, Python, WebObjects und/oder dergleichen erlauben. Der Mailserver kann Kommunikationsprotokolle wie: Internet message access protocol (IMAP), Messaging Application Programming Interface (MAPI)/Microsoft Exchange, post office protocol (POP3), simple mail transfer protocol (SMTP) und/oder dergleichen unterstützen, ist jedoch nicht darauf beschränkt. Der Mailserver kann ankommende und ausgehende Mail-Nachrichten, welche gesendet, weitergeleitet und/oder auf andere Weise durch und/oder zum SYNONYM gelangen, routen, weiterleiten und verarbeiten.

[0095] Den Zugang zur SYNONYM-Mail kann durch eine Anzahl von APIs, welche von der jeweiligen Web-Server-Komponente und/oder Betriebssystem bereitgestellt werden, erreicht werden.

[0096] Auch ein Mailserver kann Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen.

E-Mail-Client

[0097] Eine E-Mail-Client-Komponente 722 ist eine gespeicherte Programmkomponente, welche von einer CPU 703 ausgeführt wird. Der E-Mail-Client kann eine konventionelle E-Mail-Betrachtungsanwendung wie Apple Mail, Microsoft Entourage, Microsoft Outlook, Microsoft Outlook Express, Mozilla Thunderbird, und/oder dergleichen sein. E-Mail-Clients können eine Vielzahl von Transferprotokollen wie: IMAP, Microsoft Exchange, POP3, SMTP und/oder dergleichen unterstützen. Ein E-Mail-Client kann mit anderen Komponenten in einer Komponentensammlung und/oder ähnlichen Einrichtungen kommunizieren, einschliesslich mit sich selbst. Meistens kommuniziert der E-Mail-Client mit E-Mail-Servern, Betriebssystemen, anderen E-Mail-Clients und/oder dergleichen; z.B. kann er Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen. Normalerweise stellt der E-Mail-Client eine Einrichtung zum Verfassen und Senden von elektronischen E-Mail-Nachrichten dar.

Verschlüsselungsserver

[0098] Eine Verschlüsselungsserver-Komponente 720 ist eine gespeicherte Programmkomponente, welche von einer CPU 703, vom Verschlüsselungsprozessor 726, Verschlüsselungsprozessorschnittstelle 727, Verschlüsselungsprozessorgerät 728 und/oder dergleichen ausgeführt wird. Verschlüsselungsprozessorschnittstellen erlauben die Beschleunigung von Verschlüsselungs- und/oder Entschlüsselungsanfragen durch die Verschlüsselungskomponente; jedoch kann die Verschlüsselungskomponente alternativ auf einer normalen CPU laufen. Die Verschlüsselungskomponente erlaubt die Verschlüsselung und/oder Entschlüsselung von bereitgestellten Daten. Die Verschlüsselungskomponente erlaubt beides: symmetrische und asymmetrische (z.B. Pretty Good Protection (PGP)) Verschlüsselung und/oder Entschlüsselung. Die Verschlüsselungskomponente kann Verschlüsselungstechniken wie: digitale Zertifikate (z.B. X.509 authentication framework), Digitalsignaturen, Doppelsignaturen, Kuvertierung, Zugriffschutz mit Passwortabfrage, Management von öffentlichen Schlüsseln und/oder dergleichen, verwenden, ist jedoch nicht darauf beschränkt. Die Verschlüsselungskomponente ermöglicht eine Vielzahl (Verschlüsselung und/oder Entschlüsselung) von Sicherheitsprotokollen wie: Checksum, Data Encryption Standard (DES), Elliptical Curve Encryption (ECC), International Data Encryption Algorithm (IDEA), Message Digest 5 (MD5), welches eine Einbahn-Hashfunktion ist), Passwörter, Rivest Cipher (RC5), Rijndael, RSA (welche ein Internetverschlüsselungs- und Authentifizierungssystem ist, das ein 1977 von Ron Rivest, Adi Shamir und Leonard Adleman entwickeltes Algorithmus ist), Secure Hash Algorithm (SHA), Secure Socket Layer (SSL), Secure Hypertext Transfer Protocol (HTTPS) und/oder dergleichen, jedoch nicht darauf beschränkt. Durch die Verwendung solcher Verschlüsselungssicherheitsprotokolle kann das SYNONYM die gesamte eingehende und ausgehende Kommunikation verschlüsseln und es kann als Knoten innerhalb eines Virtual Private Network (VPN) mit einem breiteren Kommunikationsnetzwerk dienen. Die Verschlüsselungskomponente ermöglicht den Prozess der «Sicherheitsautorisierung», wobei der Zugriff auf eine Ressource von einem Sicherheitsprotokoll verhindert wird, wobei die Verschlüsselungskomponente einen autorisierten Zugriff auf die gesicherte Ressource durchführt. Zusätzlich kann die Verschlüsselungskomponente eindeutige Identifikatoren für den Inhalt bereitstellen, z.B. durch den Einsatz eines MD5-Hash, um eine eindeutige Signatur für eine digitale Audiodatei zu erhalten. Eine Verschlüsselungskomponente kann mit anderen Komponenten in einer Komponentensammlung und/oder ähnlichen Einrichtungen kommunizieren, einschliesslich mit sich selbst. Die Verschlüsselungskomponente unterstützt Verschlüsselungsschemas, welche die sichere Übertragung von Informationen über ein Kommunikationsnetzwerk erlauben, so dass es für die SYNONYM-Komponente möglich wird, in sicheren Transaktionen einzugreifen, wenn dies gewünscht ist. Die Verschlüsselungskomponente ermöglicht den sicheren Zugang zu Ressourcen auf dem SYNONYM und ermöglicht den Zugang zu sicheren Ressourcen auf fernen Systemen; z.B. kann sie als ein Client und/oder Server von sicheren Ressourcen agieren. Meistens kommuniziert die Verschlüsselungskomponente mit Informationsservern, Betriebssystemen, anderen Programmkomponenten und/oder dergleichen; z.B. kann er Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen.

Die SYNONYM-Datenbank

[0099] Die SYNONYM-Datenbankkomponente 719 kann in eine Datenbank und ihren gespeicherten Daten eingegliedert werden. Die Datenbank ist eine gespeicherte Programmkomponente, welche von der CPU ausgeführt wird; der gespeicherte Teil der Programmkomponente, welcher die CPU konfiguriert, um die gespeicherten Daten zu verarbeiten. Die Datenbank kann eine konventionelle, fehlertolerante, relationale, skalierbare, sichere Datenbank sein, wie Oracle oder Synbase. Relationale Datenbanken sind eine Erweiterung einer Einfachdatei. Relationale Datenbanken bestehen aus eine Reihe von verknüpften Tabellen. Die Tabellen sind mittels eines Schlüsselfeldes miteinander verbunden. Die Verwendung des Schlüsselfeldes erlaubt die Kombination der Tabellen durch Indexierung mit dem Schlüsselfeld; z.B. agieren die Schlüsselfelder als dimensionale Drehpunkte für die Kombination von Informationen aus verschiedenen Tabellen. Beziehungen identifizieren normalerweise zwischen den Tabellen aufrechterhaltene Links durch einen Abgleich von Primärschlüsseln. Primärschlüssel stellen Felder dar, welche eindeutig die Reihen einer Tabelle in einer relationalen Datenbank identifizieren. Um noch genauer zu sein, identifizieren sie Reihen einer Tabelle auf der «einen» Seite einer One-to-Many-Beziehung.

[0100] Alternativ kann die SYNONYM-Datenbank durch Verwendung verschiedener Standard-Datenstrukturen, wie einem Array, Hash, (verkettete) Liste, Struct, strukturierte Textdatei (z.B. XML), Tabelle und/oder dergleichen. Solche Datenstrukturen können im Speicher oder in (strukturierten) Dateien gespeichert werden. In einer anderen Alternative kann eine Objekt-orientierte Datenbank wie Frontier, ObjectStore, Poet, Zope und/oder dergleichen verwendet werden. Objektdatenbanken können eine Anzahl von Objektsammlungen umfassen, welche mittels einiger gemeinsamer Attribute zusammen gruppiert und/oder verkettet werden; sie können mit anderen Objektsammlungen mittels einiger gemeinsamer Attribute in Beziehung stehen. Objekt-orientierte Datenbanken arbeiten in ähnlicher Weise wie relationale Datenbanken, mit der Ausnahme, dass Objekte nicht nur Datenfragmente sind, sondern andere Arten von in einem gegebenen Objekt eingeschlossener Funktionalität aufweisen können. Wenn die SYNONYM-Datenbank als eine Datenstruktur implementiert ist, kann die Verwendung der SYNONYM-Datenbank 719 in eine andere Komponente, wie der SYNONYM-Komponente 735 integriert werden. Die Datenbank kann auch als eine Mischung von Datenstrukturen, Objekten und relationalen Strukturen implementiert sein. Datenbanken können durch standardisierte Datenverarbeitungstechniken in unzähligen Varianten konsolidiert und/oder verteilt werden. Teile der Datenbank, z.B. Tabellen, können exportiert und/oder importiert und daher dezentralisiert und/oder integriert werden.

[0101] In einer Ausführungsform umfasst die Datenbankkomponente 719 viele Tabellen 719a–g. Eine Benutzertabelle 719a umfasst Felder wie: Kunden-ID, CIF-Nummer, Name, eine Adresse und/oder dergleichen, ist aber nicht darauf beschränkt. Die Benutzertabelle kann mehrere Einheitskonten auf einem SYNONYM unterstützen und/oder verfolgen. Eine Kontotabelle 719b umfasst Felder wie: Kontonummer, Kontotyp, Kontoeröffnungsdatum, Vermögens-ID, Vermögenswert und/oder dergleichen, ist jedoch nicht darauf beschränkt. Eine Depottabelle 719c umfasst Felder wie: Depotnummer, Depottyp, Depotdatum und/oder dergleichen, ist jedoch nicht darauf beschränkt. Eine Anonymisierungstabelle 719d umfasst Felder wie: db-Name der Anonymisierung, db-ID der Anonymisierung, db-Anonymisierungstabelle und/oder dergleichen, ist aber nicht darauf beschränkt. Ein Synthetisierungstabelle 719e umfasst Felder wie: db-Name der Synthetisierung, db-ID der Synthetisierung, db-Synthetisierungstabelle, db-Synthetisierungsfeld und/oder dergleichen, ist jedoch nicht darauf beschränkt. Eine Transaktionstabelle 719f umfasst Felder wie: ID der Transaktion, erste Transaktionskontonummer, zweite Transaktionskontonummer, Transaktionstyp, Transaktionsbetrag und/oder dergleichen, ist jedoch nicht darauf beschränkt. Eine Tabelle für Anonymisierungsaustauschdaten 719g umfasst Felder wie: Austauschvorname, Austauschfamilienname, Austauschadresse und/oder dergleichen, ist jedoch nicht darauf beschränkt.

[0102] In einer Ausführungsform kann die SYNONYM-Datenbank mit anderen Datenbanksystemen interagieren. Zum Beispiel können beim Einsatz eines verteilten Datenbanksystems Suchvorgänge und Datenzugriff durch Suche der SYNONYM-Komponente die Kombination der SYNONYM-Datenbank, einer integrierten Datensicherheitsstufe der Datenbank als eine einzige Datenbankeinheit behandeln.

[0103] In einer Ausführungsform können Benutzerprogramme verschiedene Benutzerschnittstellenprimitive beinhalten, welche für einen Update des SYNONYM dienen können. Verschiedene Konten können auch proprietäre Datenbanktypen, abhängig von den Umgebungen und den Clienttypen, die vom SYNONYM bedient werden müssen, benötigen. Es wird angemerkt, dass jedes eindeutige Feld durchweg als Schlüsselfeld angegeben werden kann. In einer alternativen Ausführungsform wurden diese Tabellen in ihren eigenen Datenbanken und ihren eigenen jeweiligen Datenbankcontrollern (z.B. individuelle Datenbankcontroller für jede der obigen Tabellen) dezentralisiert. Durch die Verwendung von standardisierten Verarbeitungstechniken können die Datenbanken weiter an mehreren Computersystemen und/oder Speichergeräten verteilt werden. Auf die gleiche Weise können Konfigurationen der dezentralisierten Datenbankcontroller dadurch variiert werden, dass die verschiedenen Datenbankkomponenten 719a–g konsolidiert und/oder verteilt werden. Das SYNONYM kann so konfiguriert werden, dass es verschiedene Einstellungen, Eingaben und Parameter über Datenbankcontroller nachverfolgt.

[0104] Die SYNONYM-Datenbank kann mit Programmkomponenten in einer Programmkomponentensammlung kommunizieren, sie generieren, bekommen und/oder bereitstellen. Meistens kommuniziert die SYNONYM-Datenbank mit der SYNONYM-Komponente, mit anderen Programmkomponenten und/oder dergleichen. Die Datenbank kann andere Knoten und Daten betreffende Informationen enthalten, bewahren und bereitstellen.

Die SYNONYME

[0105] Die SYNONYM-Komponente 735 ist eine gespeicherte Programmkomponente, welche von der CPU ausgeführt wird. In einer Ausführungsform umschliesst die SYNONYM-Komponente eine und/oder alle Kombinationen von Aspekten des SYNONYM, welche in den vorangehenden Figuren beschrieben wurden. Als solches beeinflusst das SYNONYM das Zugreifen, das Bekommen und die Bereitstellung des Testens, der Informationen, der Dienste, der Transaktionen und/oder dergleichen über verschiedenen Kommunikationsnetzwerken.

[0106] Die SYNONYM-Komponente macht die Anonymisierung, Synthetisierung, das Testen von Produktionsumgebungen und/oder dergleichen und die Verwendung des SYNONYM möglich.

[0107] Die SYNONYM-Komponente, welche den Zugang von Informationen zwischen Knoten ermöglicht, kann durch den Einsatz von standardisierten Entwicklungstools und -sprachen entwickelt werden, wie: Apache Components, Assembler, ActiveX, ausführbare Binärdateien, (ANSI) (Objective) C (++), C# und/oder .NET, Datenbankadapter, CGI Skripten, Java, JavaScript, Mappingtools, prozedurale und Objekt-orientierte Entwicklungstools, PERL, PHP, Python, Shell-Skripten, SQL-Befehle, Web-Anwendungsservererweiterungen, Web-Entwicklungsumgebungen und -bibliotheken (z.B. ActiveX von Microsoft; Adobe AIR, FLEX & FLASH; AJAX; (D)HTML; Dojo; Java; JavaScript; jQuery(UI); MooTools; Prototype; script.sculo.us; Simple Object Access Protocol (SOAP); SWFObject; Yahoo! User Interface; und/oder dergleichen), WebObjects und/oder dergleichen, jedoch nicht darauf beschränkt. In einer Ausführungsform setzt der SYNONYM-Server einen Verschlüsselungsserver ein, um Kommunikationen zu verschlüsseln und zu entschlüsseln. Die SYNONYM-Komponente kann mit anderen Komponenten in einer Komponentensammlung und/oder ähnlichen Einrichtungen kommunizieren, einschliesslich mit sich selbst. Meistens kommuniziert die SYNONYM-Komponente mit der SYNONYM-Datenbank, mit Betriebssystemen, anderen Programmkomponenten und/oder dergleichen. Das SYNONYM kann Kommunikation, Anfragen und/oder Antworten von Programmkomponenten, System, Benutzer und/oder Daten enthalten, damit kommunizieren, sie generieren, bekommen und/oder bereitstellen.

Verteilte SYNONYME

[0108] Die Struktur und/oder der Betrieb irgendeiner der SYNONYM-Controller-Knotenkomponenten kann auf jede Weise kombiniert, konsolidiert und/oder verteilt werden, um die virtuelle Entwicklung und/oder Verteilung zu erleichtern. Auf dieselbe Weise kann die Komponentensammlung auf jede Art kombiniert werden, um die Verteilung und/oder Entwicklung zu erleichtern. Um dies zu erreichen, können die Komponenten in einer gemeinsamen Codebasis oder in einer Einrichtung, die dynamisch die Komponenten auf Anfrage laden kann, in eine integrierte Weise integrieren.

[0109] Die Komponentensammlung kann in unzähligen Varianten durch die standardisierte Datenverarbeitung und/oder Entwicklungstechniken konsolidiert und/oder verteilt werden. Mehrere Instanzen jeder beliebigen Programmkomponente in der Programmkomponentensammlung kann in einem einzigen Knoten instantiiert werden und/oder über vielen Knoten, um die Leistungsfähigkeit durch Lastverteilungs- und/oder Datenverarbeitungstechniken zu erhöhen. Weiter können einzelne Instanzen auch über mehreren Controllern und/oder Speichergeräte; z.B. Datenbanken, verteilt werden. Alle Programmkomponenteninstanzen und Controller, die zusammenarbeiten, können dies durch standardisierte Datenverarbeitungskommunikationstechniken tun.

[0110] Die Konfiguration des SYNONYM-Controllers hängt vom Kontext der Systemverteilung ab. Faktoren wie Budget, Kapazität, Standort und/oder Verwendung der zugrundeliegenden Hardwareressourcen, jedoch nicht darauf beschränkt, kann die Verteilungs- und Konfigurationsanforderungen beeinflussen. Unabhängig davon, ob sich aus der Konfiguration konsolidiertere und/oder integriertere Programmkomponenten, eine Reihe von Programmkomponenten, die noch mehr verteilt sind und/oder irgendeine Kombination zwischen einer konsolidierten und verteilten Konfiguration ergibt, können Daten kommuniziert, erhalten und/oder bereitgestellt werden. Instanzen von Komponenten, die in einer gemeinsamen Codebasis der Programmkomponentensammlung konsolidiert sind, können Daten kommunizieren, erhalten und/oder bereitstellen. Dies kann durch Datenverarbeitungskommunikationstechniken wie: Datenreferenzierung (z.B. Zeiger), internes Messaging, Kommunikation von Objektinstanzvariablen, geteilter Speicherbereich, Variablenübergabe und/oder dergleichen, jedoch nicht darauf beschränkt, zwischen Anwendungen erreicht werden.

[0111] Wenn Komponenten der Komponentensammlung diskret, geteilt und/oder bezogen aufeinander extern sind, kann die Kommunikation mit, der Erhalt und/oder die Bereitstellung von Daten für eine andere Komponente der Komponenten durch Datenverarbeitungskommunikationstechniken wie: Application Program Interfaces (API) Informationsübergabe; (verteilte) Component Object Model ((D)COM), (verteilte) Object Linking and Embedding ((D)OLE) und/oder dergleichen), Common Object Request Broker Architecture (CORBA), lokale und ferne Anwendungsprogrammschnittstellen Jini, Remote Method Invocation (RMI), SOAP, Prozesspipes, geteilte Dateien und/oder dergleichen, jedoch nicht darauf beschränkt, erfolgen. Nachrichten, die zwischen diskreten Komponenten der Komponenten für die Kommunikation zwischen Anwendungen oder innerhalb von Speicherbereichen einer einzelnen Komponente für die Kommunikation zwischen Anwendungen, können durch die Erzeugung und Analyse einer Grammatik ermöglicht werden. Eine Grammatik kann durch die Verwendung von standardisierten Entwicklungstools wie lex, yacc, XML, und/oder dergleichen entwickelt werden, welche wiederum die Basis der Kommunikationsnachrichten innerhalb und zwischen Komponenten bildet. Zum Beispiel kann eine Grammatik so ausgebildet sein, dass sie die Marken eines HTTP-Post-Befehls erkennt, z.B:
w3c -post http//...Value1

wobei Value1 als ein Parameter erkannt wird, da «http://» ein Teil der Syntax der Grammatik ist, und das, was folgt, als Teil des post-Wertes betrachtet wird. Auf dieselbe Weise kann mit einer solchen Grammatik eine Variable «Value1» in einen HTTP-Post-Befehl eingefügt und dann verschickt werden. Die Syntax der Grammatik selbst kann als strukturierte Daten dargestellt werden, welche interpretiert und/oder auf andere Weise zur Generierung des Analyse-Mechanismus (z.B. eine beschreibende Syntaxtextdatei, wie von lex, yacc, usw. verarbeitet) verwendet wird. Einmal, dass der Analysemechanismus generiert und/oder instantiiert wurde, kann es auch selbst strukturierte Daten wie : durch Zeichen (z.B. Tab) beschriebener Text, HTML, strukturierte Text-Ströme, XML, und/oder ähnliche strukturierte Daten verarbeiten. In einer anderen Ausführungsform können Datenverarbeitungsprotokolle zwischen Anwendungen selbst integrierte und/oder leicht zu bekommende Parser (z.B. den SOAP-Parser) aufweisen, die zur Analyse von Kommunikationsdaten verwendet werden können. Weiter kann die Analysegrammatik in weiterem Sinne als zur Analyse von Nachrichten verwendet werden, und zwar zur Analyse von: Datenbanken, Datensammlungen, gespeicherten Daten, strukturierten Daten und/oder dergleichen. Die gewünschte Konfiguration hängt wieder vom Kontext, der Umgebung und den Anforderungen der Systemverteilung ab.

[0112] Die Gesamtheit dieser Anmeldung (einschliesslich Deckblatt, Titel, Untertitel, Bereich, Hintergrund Zusammenfassung, kurze Beschreibung der Figuren, ausführlich Beschreibung, Ansprüche, Abstract, Figuren und andere) zeigt durch Darstellungen verschiedene Ausführungsformen, in denen die beanspruchte Erfindung ausgeführt werden kann. Die Vorteile und Merkmale der Anmeldung beziehen sich nur auf repräsentative Bespiele von Ausführungsformen und sind nicht erschöpfend und/oder abschliessend. Sie werden nur als Hilfe für das Verständnis und für die Lehre der beanspruchten Prinzipien präsentiert. Es versteht sich, dass sie nicht für alle beanspruchten Erfindungen repräsentativ sind. Folglich wurden hier bestimmte Aspekte der Offenbarung nicht beschrieben. Dass alternative Ausführungsformen für einen spezifischen Teil der Erfindung eventuell nicht beschrieben wurden oder dass weitere, nicht beschriebene alternative Ausführungsformen für einen Teil existieren können, soll nicht als Disclaimer dieser alternativen Ausführungsformen angesehen werden. Dies wird so betrachtet, dass viele jener unbeschriebenen Ausführungsformen dieselben Prinzipien der Erfindung beinhalten und dass andere äquivalent sind. Daher sollte es verstanden werden, dass andere Ausführungsformen verwendet werden können und funktionelle, logische, organisatorische, strukturelle und/oder topologische Änderungen gemacht werden können, ohne sich vom Rahmen und/oder Geist der Erfindung zu entfernen. Daher gelten in der gesamten Offenbarung alle Beispiele und/oder Ausführungsformen als nicht beschränkend. Auch sollten keine Rückschlüsse von diesen hier beschriebenen Ausführungsformen auf jene, die hier nicht beschrieben sind, gezogen werden, ausser im Hinblick auf Platzsparen und Wiederholung. Zum Beispiel soll verstanden werden, dass die logische und/oder topologische Struktur irgendeiner Kombination von irgendwelchen Programmkomponenten (eine Komponentensammlung), von anderen Komponenten und/oder von jedem vorliegenden Merkmalset wie in den Figuren beschrieben und/oder insgesamt nicht auf einen festen Betrieb und/oder Anordnung beschränkt sind, sondern dass jede offenbarte Reihenfolge eher beispielhaft ist und alle Äquivalente, unabhängig von ihrer Reihenfolge, in der Offenbarung in Betracht gezogen sind. Weiter sollte verstanden werden, dass solche Merkmale nicht auf serielle Ausführung beschränkt sind, sondern dass eher jede Anzahl von Threads, Prozessen, Diensten, Server und/oder dergleichen, welche asynchron, gleichzeitig, parallel, synchron und/oder dergleichen ausgeführt werden, von der Offenbarung in Betracht gezogen sind. Daher können einige dieser Merkmale zueinander widersprüchlich sein, insofern als dass sie nicht zur gleiche Zeit in einer einzigen Ausführungsform vorhanden sein können. Weiter sind einige Merkmale für einen Aspekt der Erfindung verwendbar und für andere Aspekte nicht. Zusätzlich umfasst die Offenbarung andere Erfindungen, welche zurzeit nicht beansprucht sind. Die Anmelderin behält sich alle Rechte für diese zurzeit nicht beanspruchten Erfindungen vor, einschliesslich des Rechts, solche Erfindungen zu beanspruchen, zusätzliche Anmeldungen, Continuations, Continuations-in-Part, Teilanmeldungen und/oder dergleichen einzureichen. Daher sollte verstanden werden, dass Vorteile, Ausführungsformen, Beispiele, funktionale Merkmale, logische, organisatorische, strukturelle, topologische und/oder andere Aspekte der Offenbarung nicht als Beschränkungen der Offenbarung, wie von den Ansprüchen definiert, oder als Beschränkung von Äquivalenten zu den Ansprüchen angesehen werden sollen.

Patentansprüche

1. Umgebungsentwicklungsvorrichtung umfassend:
   einen Speicher;
   einen sich mit dem Speicher in Kommunikation befindenden Prozessor, welcher derart konfiguriert ist, dass er eine Vielzahl von im Speicher gespeicherten Verarbeitungsinstruktionen ausgibt, wobei der Prozessor Instruktionen ausgibt zum:
   Empfangen einer Anfrage zur Erzeugung einer Verteilungsumgebung;
   Identifizieren reeller Datensätze basierend auf die Anfrage;
   Erzeugen variierter Daten aus den reellen Daten; Generieren einer synthetischen Entwicklungsumgebung, wobei die Generierung weiter Instruktionen umfasst zum:
   Instantiieren von Benutzerkonten, Bevölkern der synthetischen Entwicklungsumgebung mit einem Profil von Testkomponenten,
   Laden der variierten Daten in die synthetische Entwicklungsumgebung, und
   Verteilen der synthetischen Entwicklungsumgebung zu einem in der Anfrage angegebenen Standort.
2. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Erzeugung einer virtuellen Bank.

3. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Erzeugung von synthetischen Spielverteilungsumgebungen.
4. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Erzeugung synthetischer Simulationsverteilungsumgebungen.
5. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Erzeugung synthetischer Bildungsverteilungsumgebungen.
6. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Erzeugung synthetischer medizinischer Verteilungsumgebungen.
7. Verwendung der Umgebungsentwicklungsvorrichtung nach Anspruch 1 zur Qualitätssicherung.

Fig. 1A

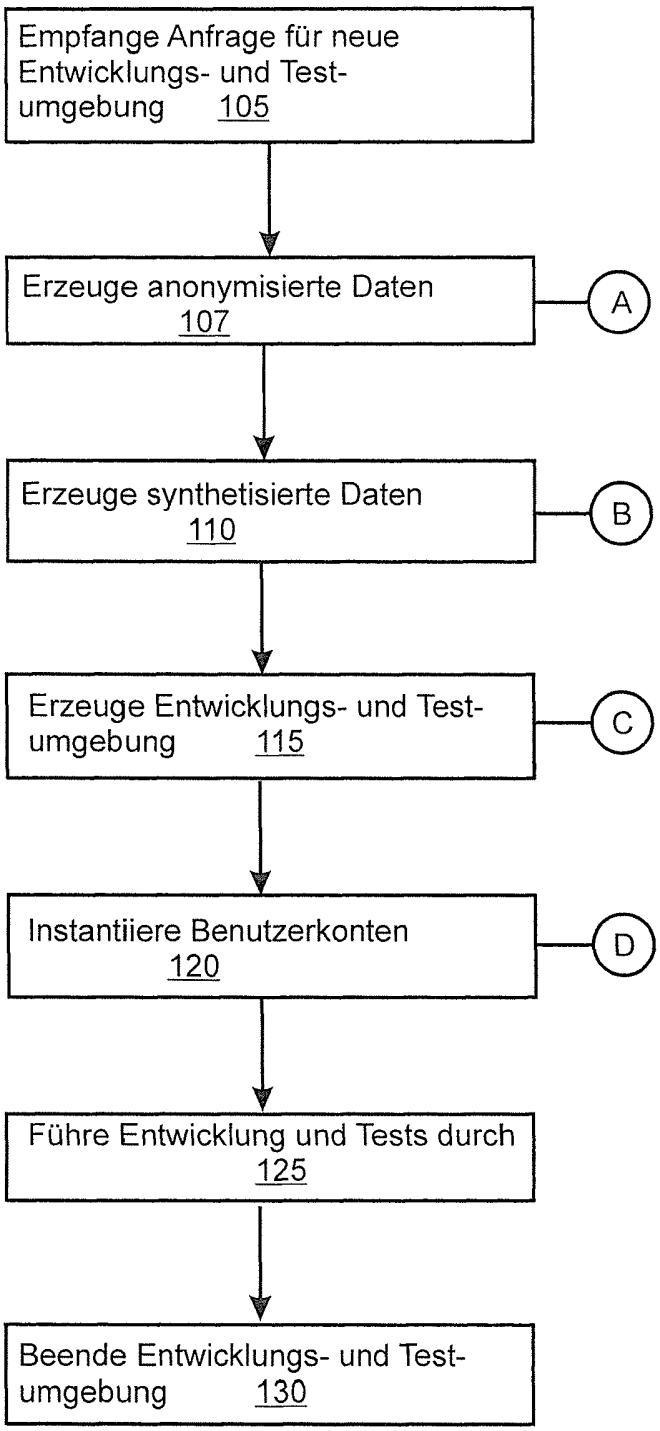

Empfange Anfrage für neue Entwicklungs- und Test-umgebung 105
Erzeuge anonymisierte Daten 107
A
Erzeuge synthetisierte Daten 110
B
Erzeuge Entwicklungs- und Test-umgebung 115
C
Instantiiere Benutzerkonten 120
D
Führe Entwicklung und Tests durch 125
Beende Entwicklungs- und Test-umgebung 130

Fig. 1B

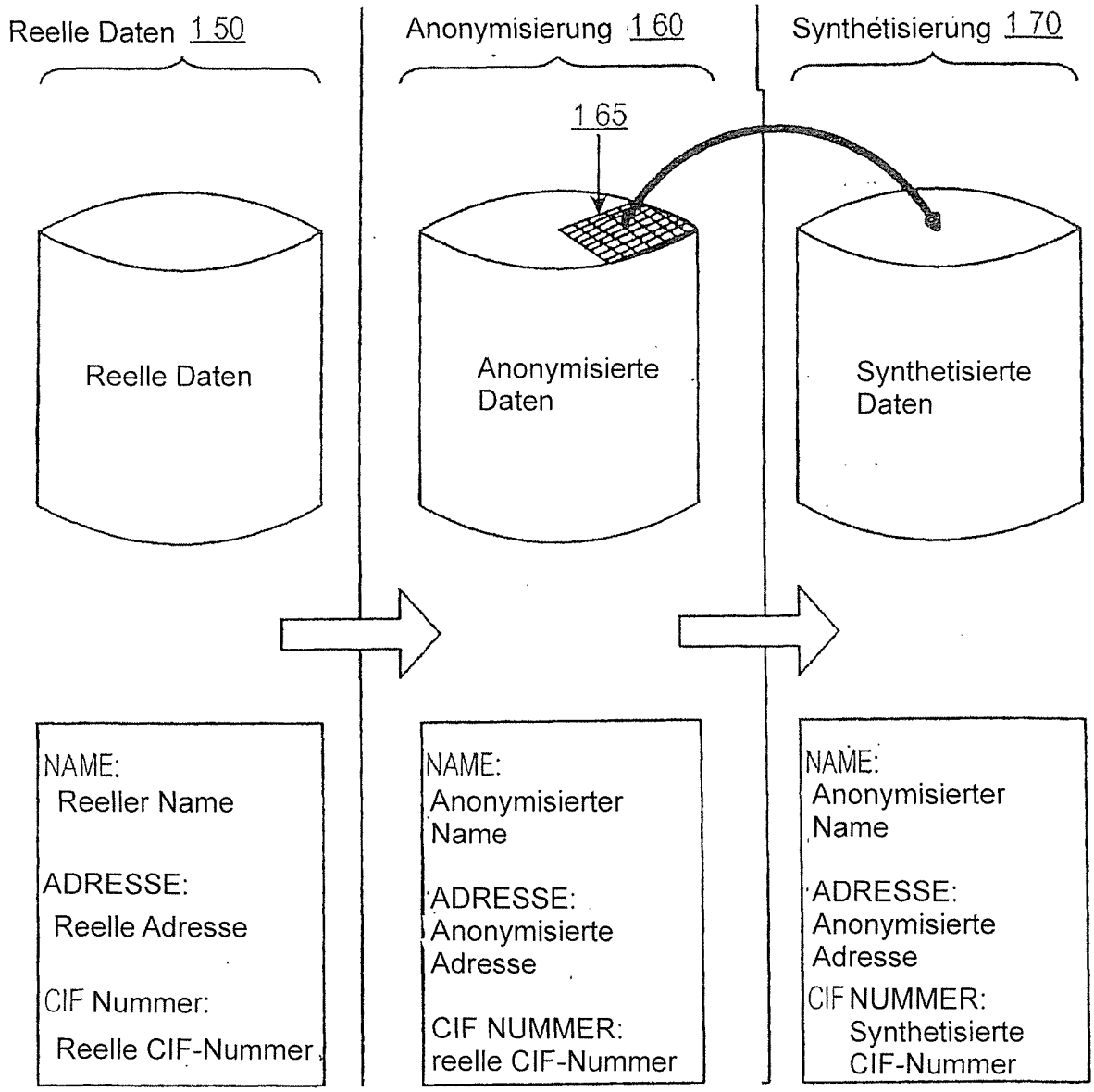

Reelle Daten 150
Anonymisierung 160
Synthetisierung 170
165
Reelle Daten
Anonymisierte Daten
Synthetisierte Daten
NAME:
Reeller Name
ADRESSE:
Reelle Adresse
CIF Nummer:
Reelle CIF-Nummer
NAME:
Anonymisierter Name
ADRESSE:
Anonymisierte Adresse
CIF NUMMER:
reelle CIF-Nummer
NAME:
Anonymisierter Name
ADRESSE:
Anonymisierte Adresse
CIF NUMMER:
Synthetisierte CIF-Nummer

Fig. 1C

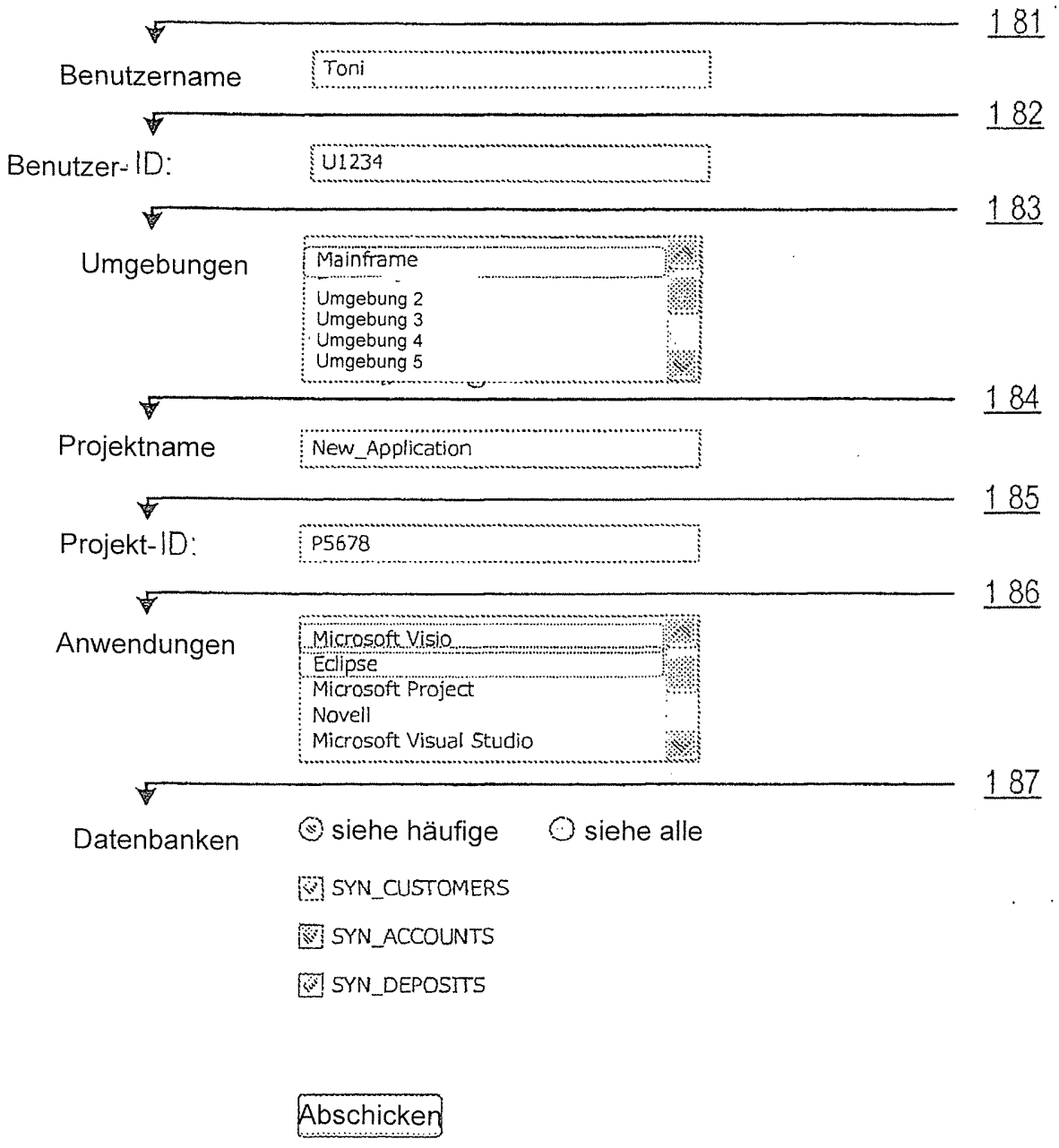

181
Benutzername
Toni
182
Benutzer-ID:
U1234
183
Umgebungen
Mainframe
Umgebung 2
Umgebung 3
Umgebung 4
Umgebung 5
184
Projektname
New_Application
185
Projekt-ID:
P5678
186
Anwendungen
Microsoft Visio
Eclipse
Microsoft Project
Novell
Microsoft Visual Studio
187
Datenbanken
siehe häufige
siehe alle
SYN_CUSTOMERS
SYN_ACCOUNTS
SYN_DEPOSITS
Abschicken

Fig. 2A

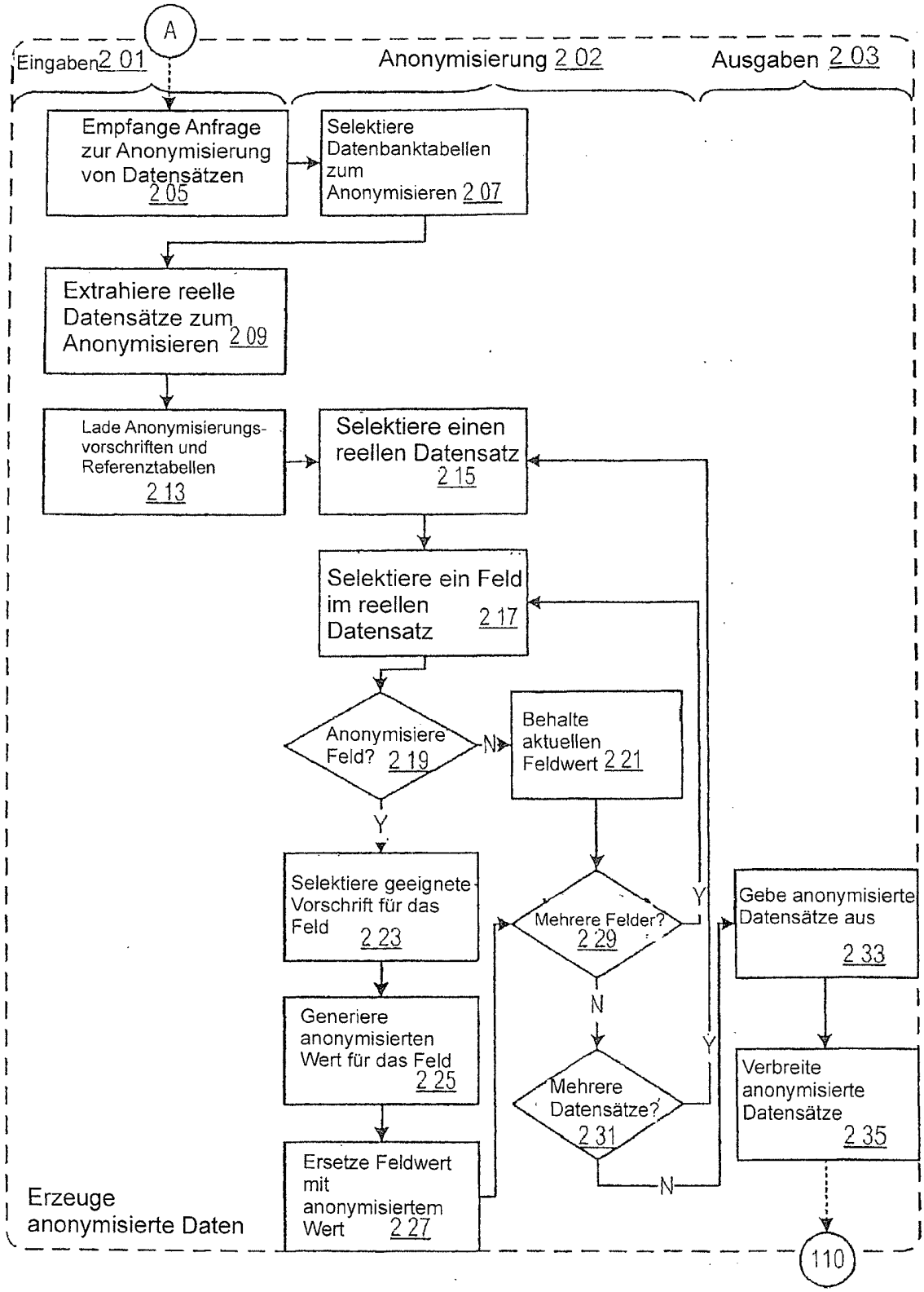

A
Eingaben 2 01
Anonymisierung 2 02
Ausgaben 2 03
Empfange Anfrage zur Anonymisierung von Datensätzen 2 05
Selektiere Datenbanktabellen zum Anonymisieren 2 07
Extrahiere reelle Datensätze zum Anonymisieren 2 09
Lade Anonymisierungs-vorschriften und Referenztabellen 2 13
Selektiere einen reellen Datensatz 2 15
Selektiere ein Feld im reellen Datensatz 2 17
Anonymisiere Feld? 2 19
N
Y
Behalte aktuellen Feldwert 2 21
Selektiere geeignete Vorschrift für das Feld 2 23
Generiere anonymisierten Wert für das Feld 2 25
Ersetze Feldwert mit anonymisiertem Wert 2 27
Mehrere Felder? 2 29
Y
N
Mehrere Datensätze? 2 31
Y
N
Gebe anonymisierte Datensätze aus 2 33
Verbreite anonymisierte Datensätze 2 35
Erzeuge anonymisierte Daten
110

Fig. 2B

Anonymsierungsvorschriften

Namen	Ersetze aus einer Referenztabelle umfassend 676 weibliche und 676 männliche Namen, unter Verwendung von in der Region gebräuchliche Namen (ökonomischer Sektor), angegeben in CIF
Familiennamen	Ersetze aus einer Referenztabelle umfassend 17576 Familiennamen (verschiedene Längen, Umlaute, Doppelnamen), unter Verwendung von in der Region gebräuchliche Namen (ökonomischer Sektor), in der CIF angegeben
Firmennamen	Ersetze mit Firmennamen, welche aus Gebiet, folgende Zahl und Überprüfungszahl aus CIF bestehen
Adressen	Ersetze aus Tabellen aus mit 900 syntaktisch korrekten Postadressen unter Verwendung von Adressen aus derselben Region (PLZ, Ort, Strasse, Hausnummer, andere). 30 bis 100 Adressen werden für ausländische Adressen verwendet.
Tel.- und FAX-Nummern	Reelle Vorwahlen werden verwendet und die verbleibenden Zahlen werden zufällig erzeugt.
Datum	Geburtsdatum, Tod, Aufnahmedatum: neues, zufällig erzeugtes Datum im selben Jahr, unter Beachtung von Altersbeschränkungen für das jeweilige Produkt.
Identifikations-nummern	Sozialversicherungs- oder persönliche Angestelltennummer (PID): neue, zufällig erzeugte Nummer der selben Länge
Freier Text	Ersetze mit einem String der selben Länge

Fig. 2C

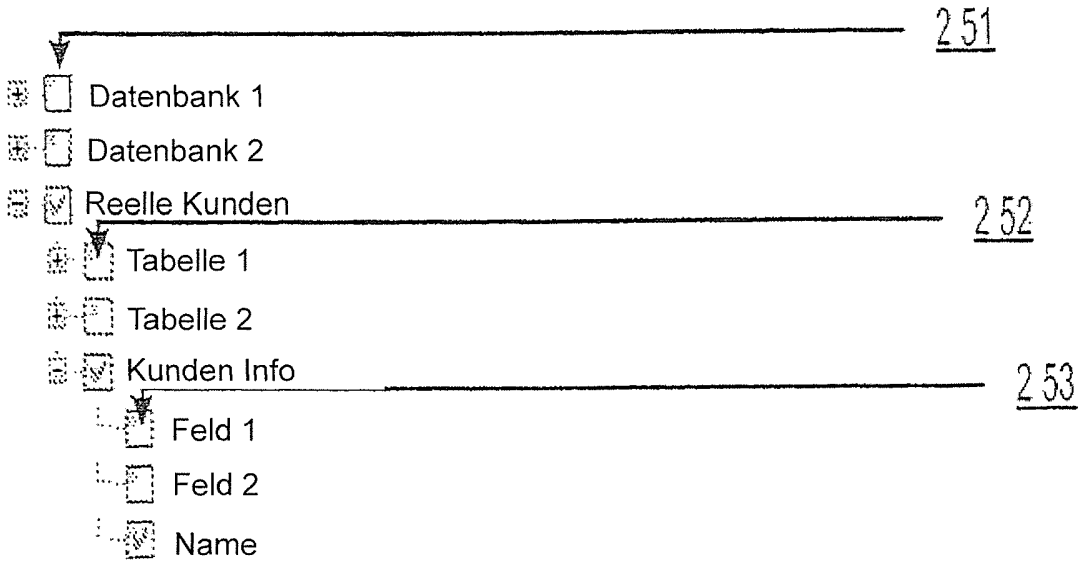

251
Datenbank 1
Datenbank 2
Reelle Kunden
252
Tabelle 1
Tabelle 2
Kunden Info
253
Feld 1
Feld 2
Name

Fig. 3

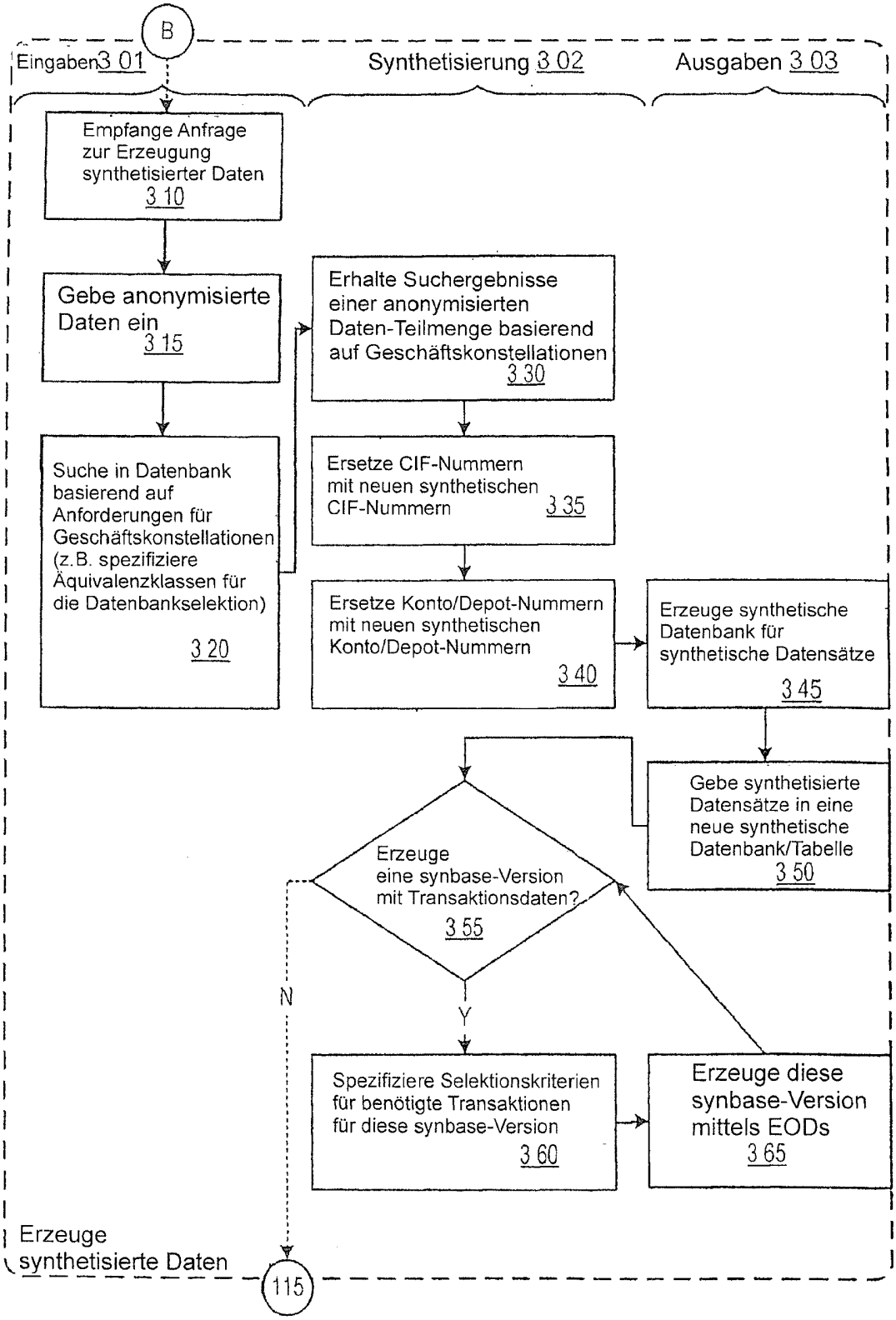

B
Eingaben 3 01
Synthetisierung 3 02
Ausgaben 3 03
Empfange Anfrage zur Erzeugung synthetisierter Daten 3 10
Gebe anonymisierte Daten ein 3 15
Suche in Datenbank basierend auf Anforderungen für Geschäftskonstellationen (z.B. spezifiziere Äquivalenzklassen für die Datenbankselektion) 3 20
Erhalte Suchergebnisse einer anonymisierten Daten-Teilmenge basierend auf Geschäftskonstellationen 3 30
Ersetze CIF-Nummern mit neuen synthetischen CIF-Nummern 3 35
Ersetze Konto/Depot-Nummern mit neuen synthetischen Konto/Depot-Nummern 3 40
Erzeuge synthetische Datenbank für synthetische Datensätze 3 45
Gebe synthetisierte Datensätze in eine neue synthetische Datenbank/Tabelle 3 50
Erzeuge eine synbase-Version mit Transaktionsdaten? 3 55
N
Y
Spezifiziere Selektionskriterien für benötigte Transaktionen für diese synbase-Version 3 60
Erzeuge diese synbase-Version mittels EODs 3 65
Erzeuge synthetisierte Daten
115

Fig. 4

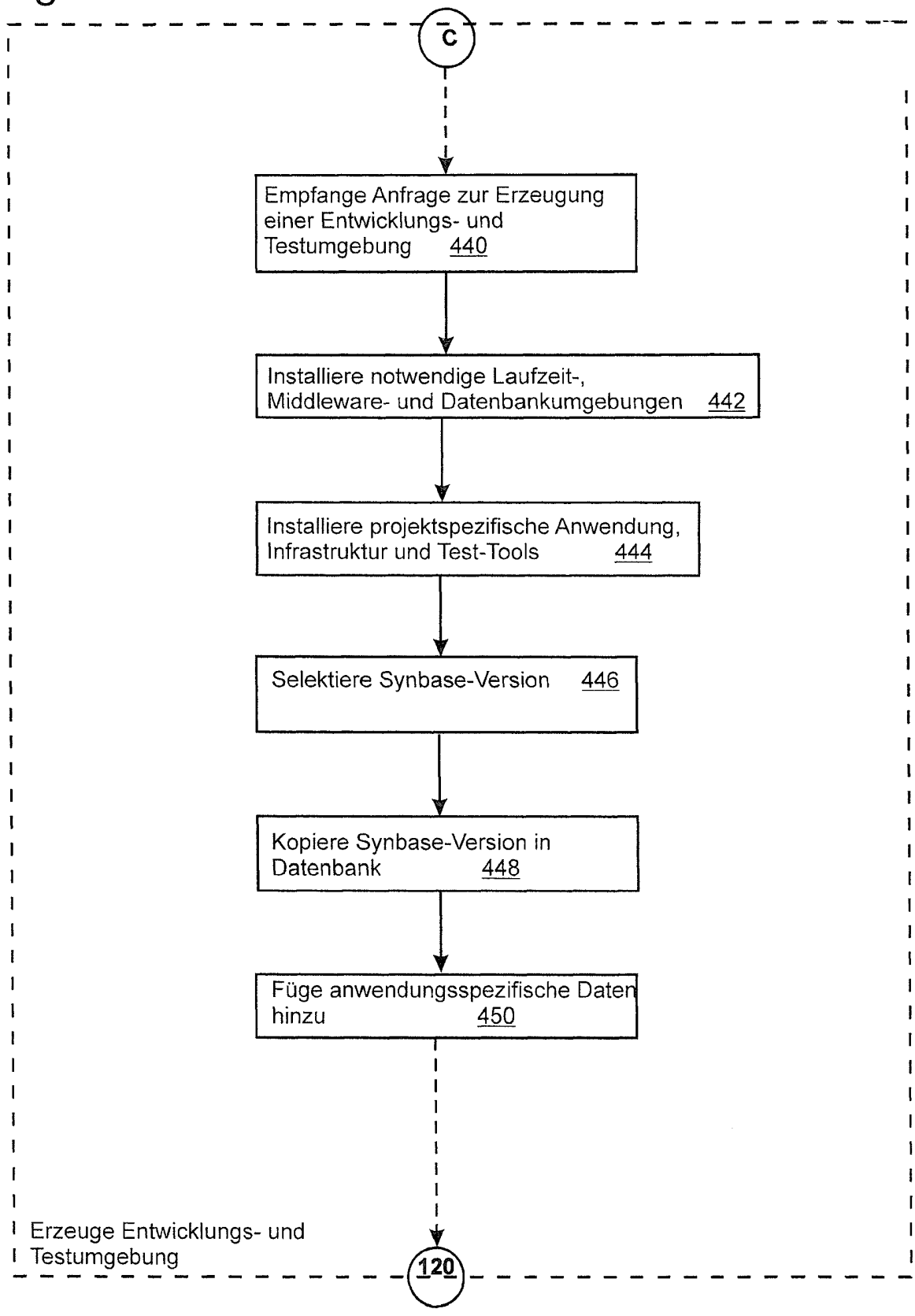

C
Empfange Anfrage zur Erzeugung einer Entwicklungs- und Testumgebung 440
Installiere notwendige Laufzeit-, Middleware- und Datenbankumgebungen 442
Installiere projektspezifische Anwendung, Infrastruktur und Test-Tools 444
Selektiere Synbase-Version 446
Kopiere Synbase-Version in Datenbank 448
Füge anwendungsspezifische Daten hinzu 450
Erzeuge Entwicklungs- und Testumgebung
120

Fig. 5

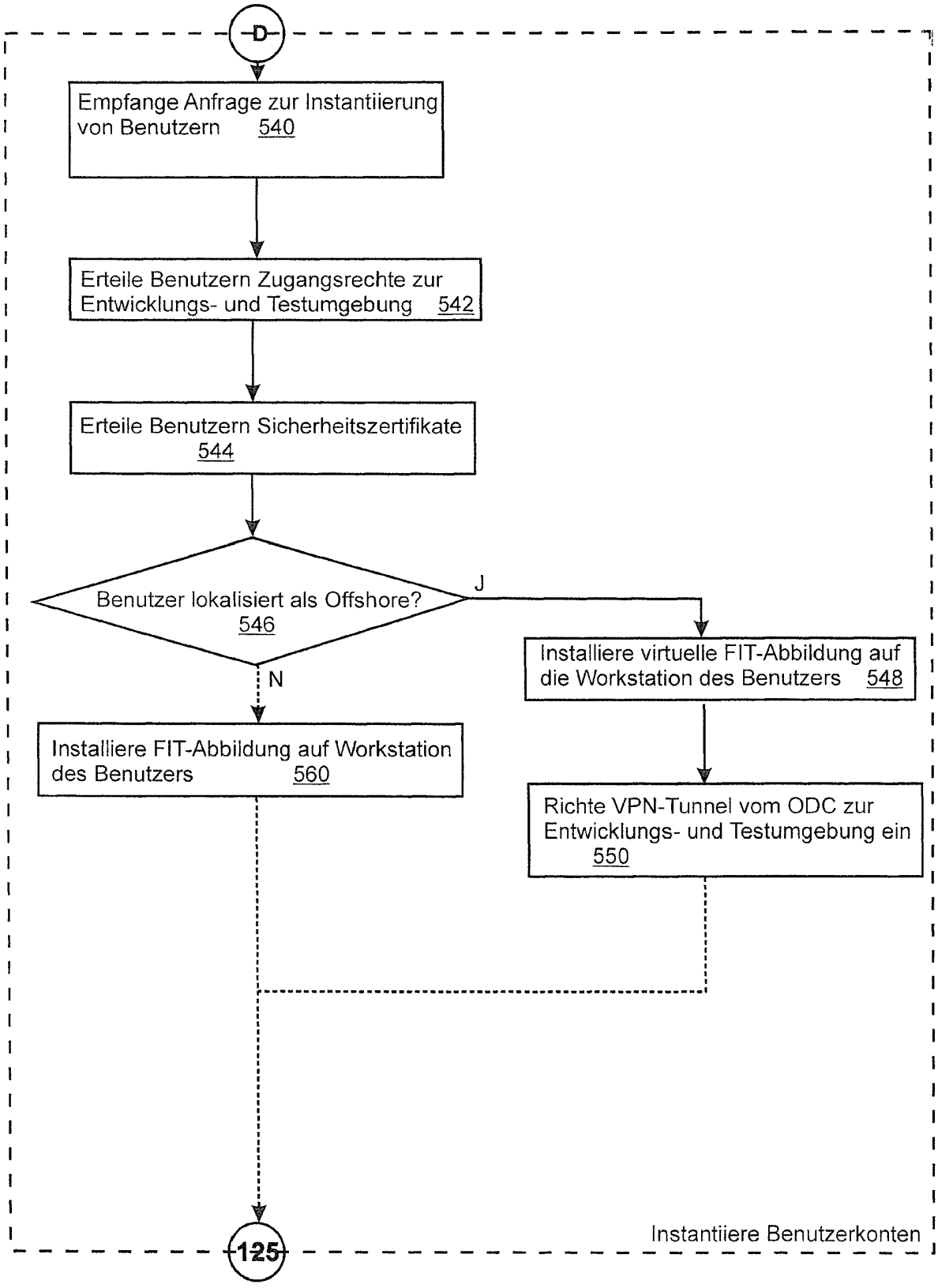

D
Empfange Anfrage zur Instantiierung von Benutzern 540
Erteile Benutzern Zugangsrechte zur Entwicklungs- und Testumgebung 542
Erteile Benutzern Sicherheitszertifikate 544
Benutzer lokalisiert als Offshore? 546
J
N
Installiere virtuelle FIT-Abbildung auf die Workstation des Benutzers 548
Installiere FIT-Abbildung auf Workstation des Benutzers 560
Richte VPN-Tunnel vom ODC zur Entwicklungs- und Testumgebung ein 550
125
Instantiiere Benutzerkonten

Fig. 6A

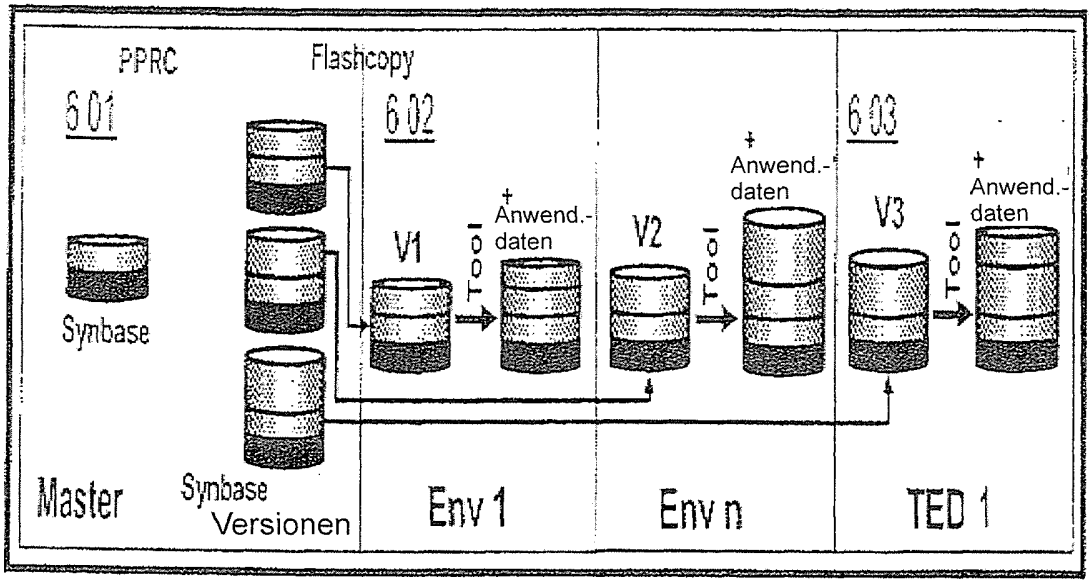

PPRC
Flashcopy
6 01
6 02
6 03
Synbase
V1
Tool
+ Anwend.-daten
V2
Tool
+ Anwend.-daten
V3
Tool
+ Anwend.-daten
Master
Synbase Versionen
Env 1
Env n
TED 1

Fig. 6B

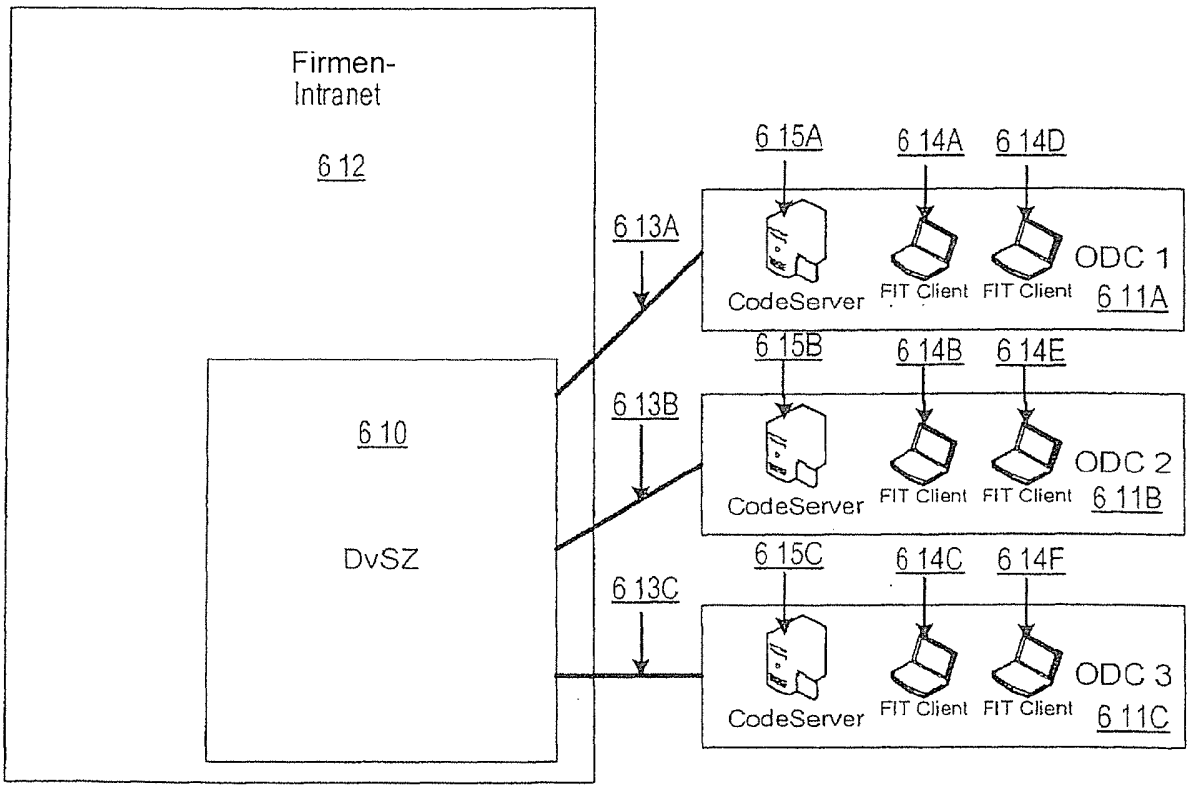

Firmen-
Intranet
6 12
6 10
DvSZ
6 13A
6 13B
6 13C
6 15A
6 14A
6 14D
CodeServer
FIT Client
FIT Client
ODC 1
6 11A
6 15B
6 14B
6 14E
CodeServer
FIT Client
FIT Client
ODC 2
6 11B
6 15C
6 14C
6 14F
CodeServer
FIT Client
FIT Client
ODC 3
6 11C

Fig. 6C

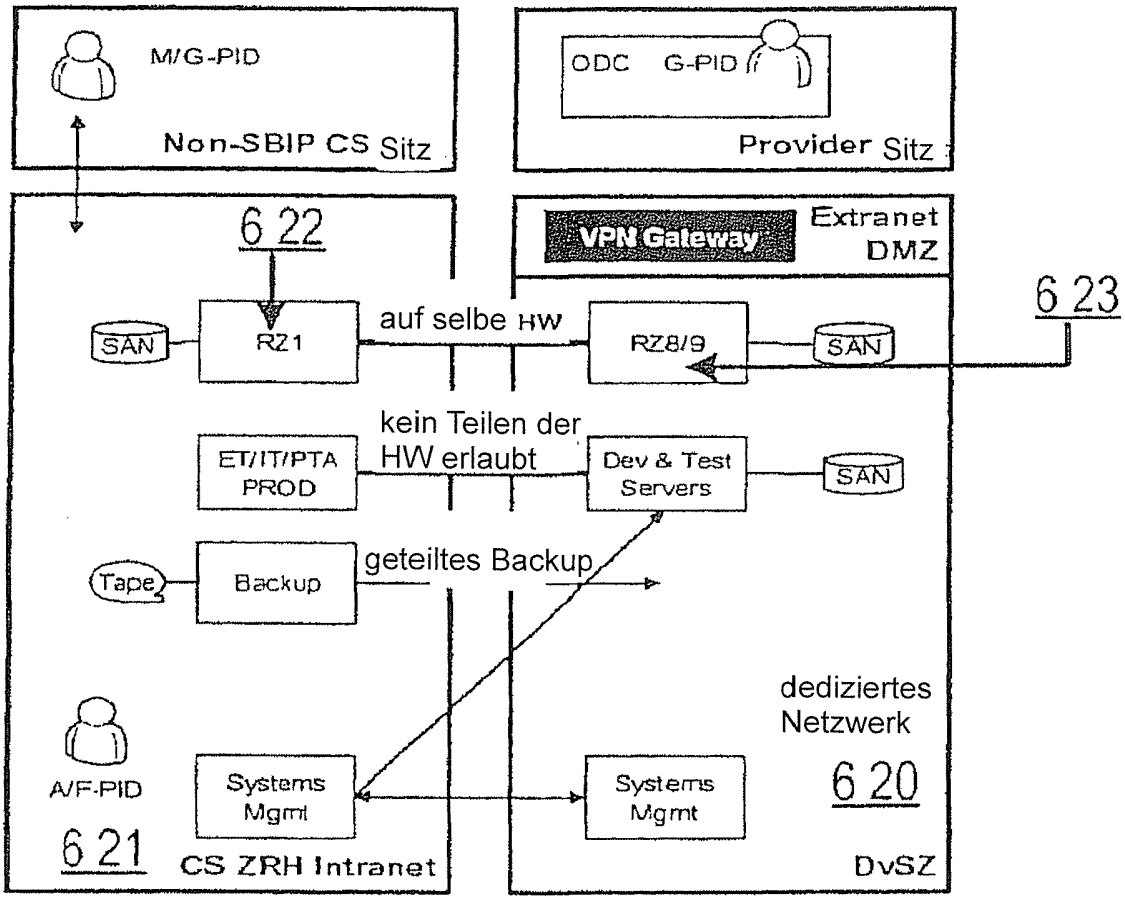

M/G-PID
Non-SBIP CS Sitz
ODC G-PID
Provider Sitz
6 22
VPN Gateway
Extranet
DMZ
6 23
SAN
RZ1
auf selbe HW
RZ8/9
SAN
kein Teilen der
HW erlaubt
ET/IT/PTA
PROD
Dev & Test
Servers
SAN
Tape
Backup
geteiltes Backup
dediziertes
Netzwerk
A/F-PID
6 21
Systems
Mgmt
Systems
Mgmt
6 20
CS ZRH Intranet
DvSZ

Fig. 6D

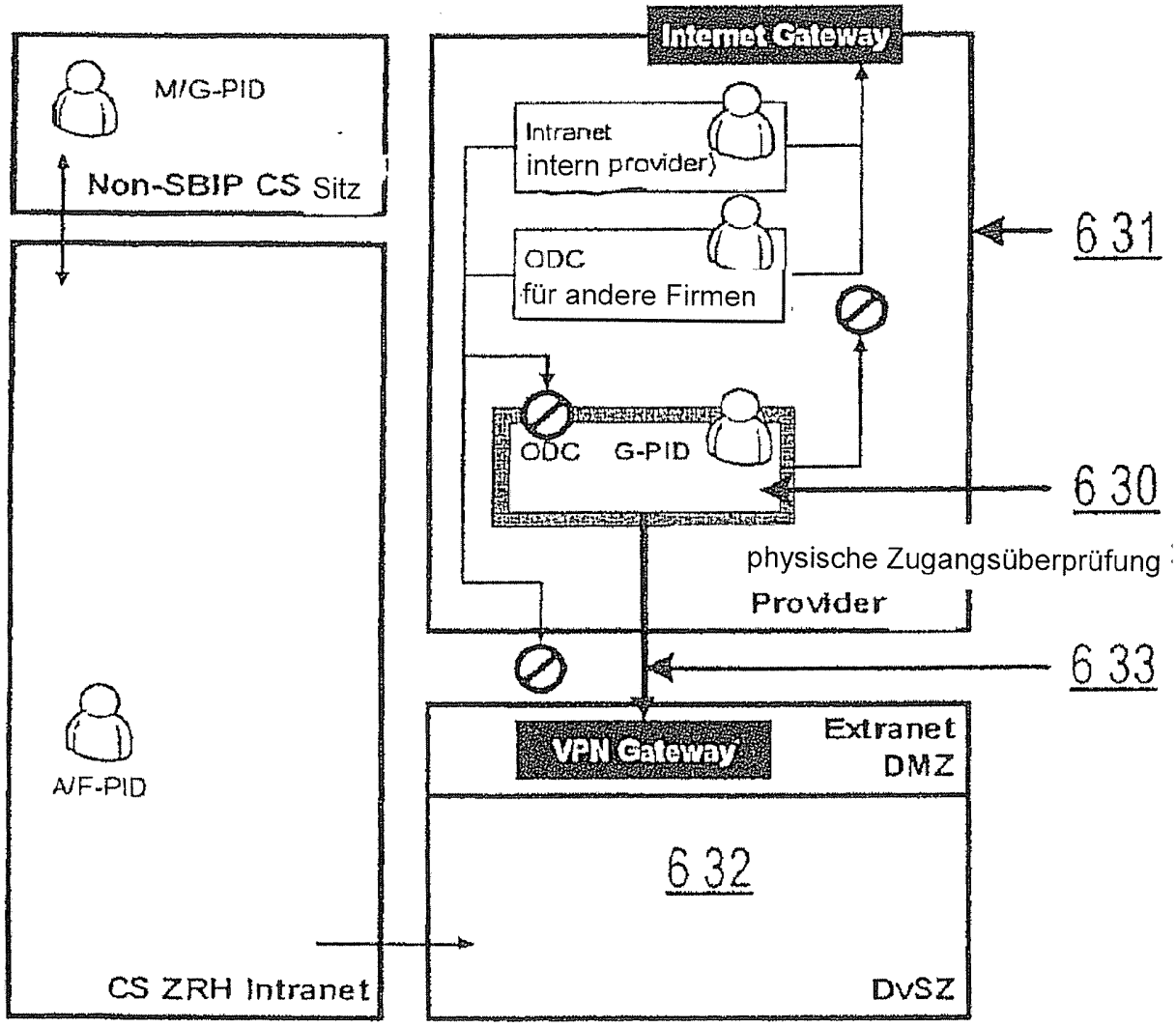

M/G-PID
Non-SBIP CS Sitz
A/F-PID
CS ZRH Intranet
Internet Gateway
Intranet
intern provider
ODC
für andere Firmen
ODC G-PID
physische Zugangsüberprüfung
Provider
VPN Gateway
Extranet
DMZ
6 32
DvSZ
6 31
6 30
6 33

Fig. 6E

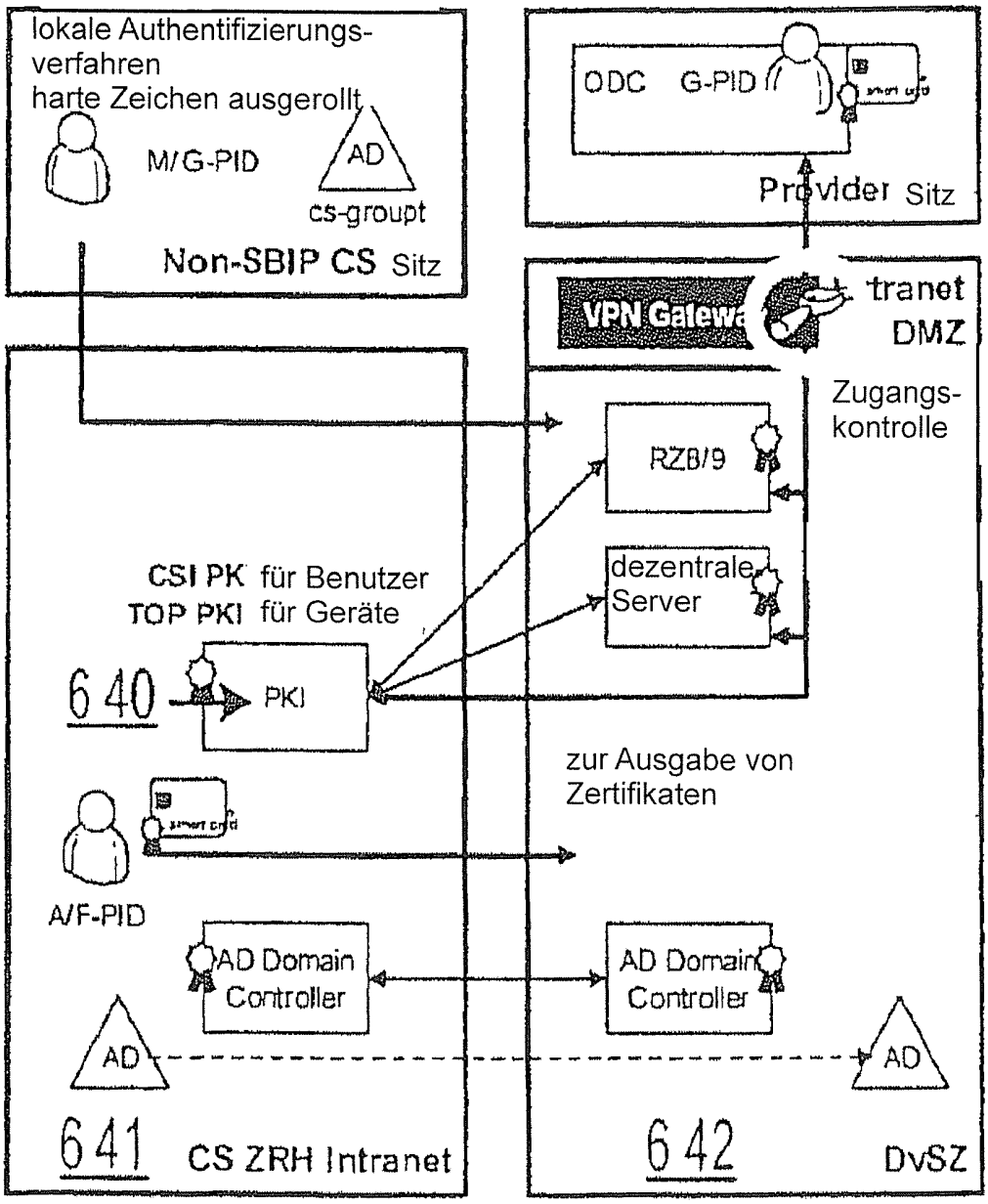

lokale Authentifizierungs-
verfahren
harte Zeichen ausgerollt
M/G-PID
AD
cs-groupt
Non-SBIP CS Sitz
ODC G-PID
Provider Sitz
VPN Gateway
tranet
DMZ
Zugangs-
kontrolle
RZB/9
dezentrale
Server
CSI PK für Benutzer
TOP PKI für Geräte
640
PKI
zur Ausgabe von
Zertifikaten
A/F-PID
AD Domain
Controller
AD Domain
Controller
AD
AD
641
CS ZRH Intranet
642
DvSZ

Fig. 6F

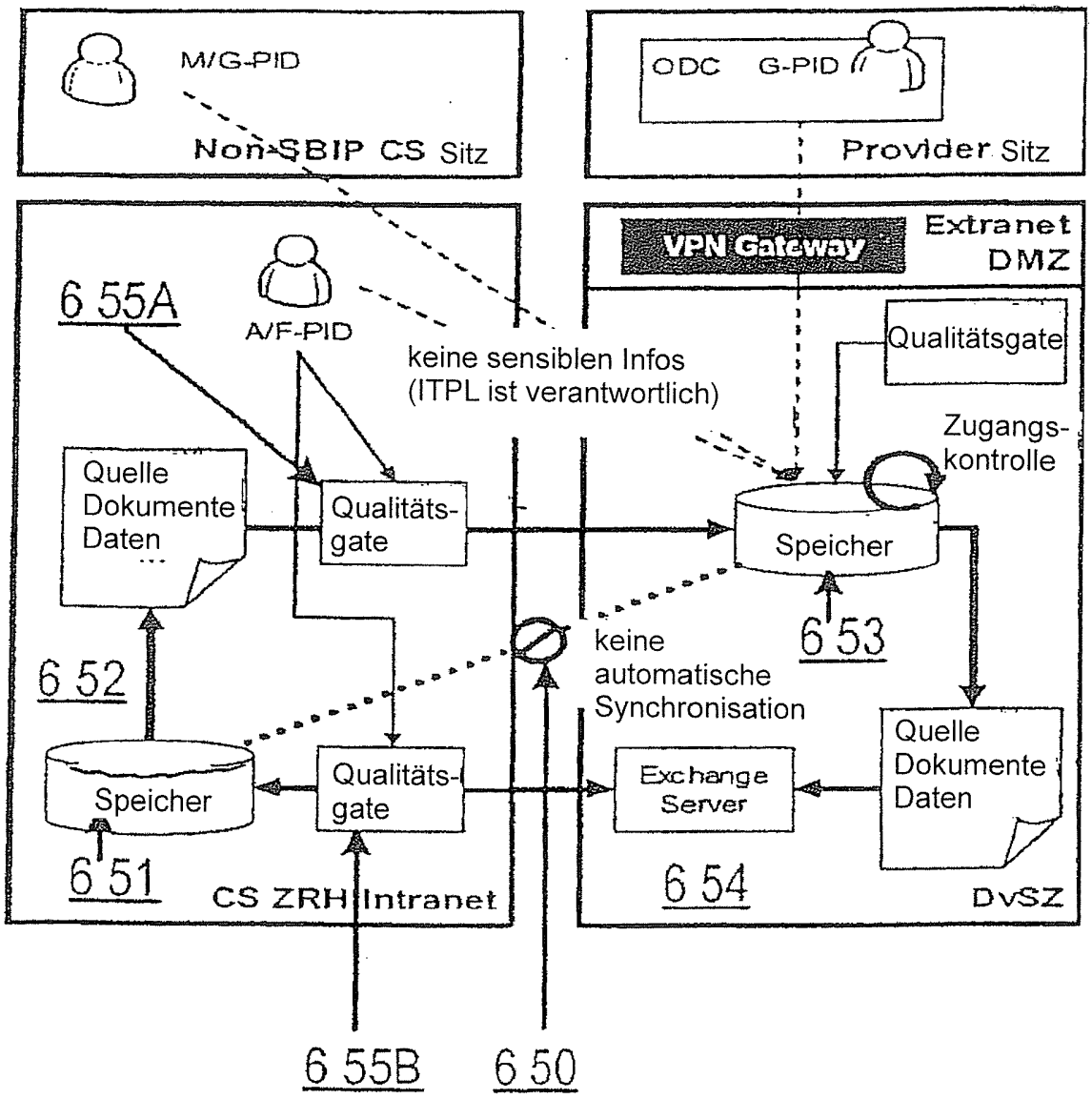

M/G-PID
Non-SBIP CS Sitz
ODC G-PID
Provider Sitz
VPN Gateway
Extranet
DMZ
6 55A
A/F-PID
keine sensiblen Infos
(ITPL ist verantwortlich)
Qualitätsgate
Zugangs-
kontrolle
Quelle
Dokumente
Daten
Qualitäts-
gate
Speicher
6 53
6 52
keine
automatische
Synchronisation
Speicher
Qualitäts-
gate
Exchange
Server
Quelle
Dokumente
Daten
6 51
CS ZRH Intranet
6 54
DvSZ
6 55B
6 50

Fig. 6G

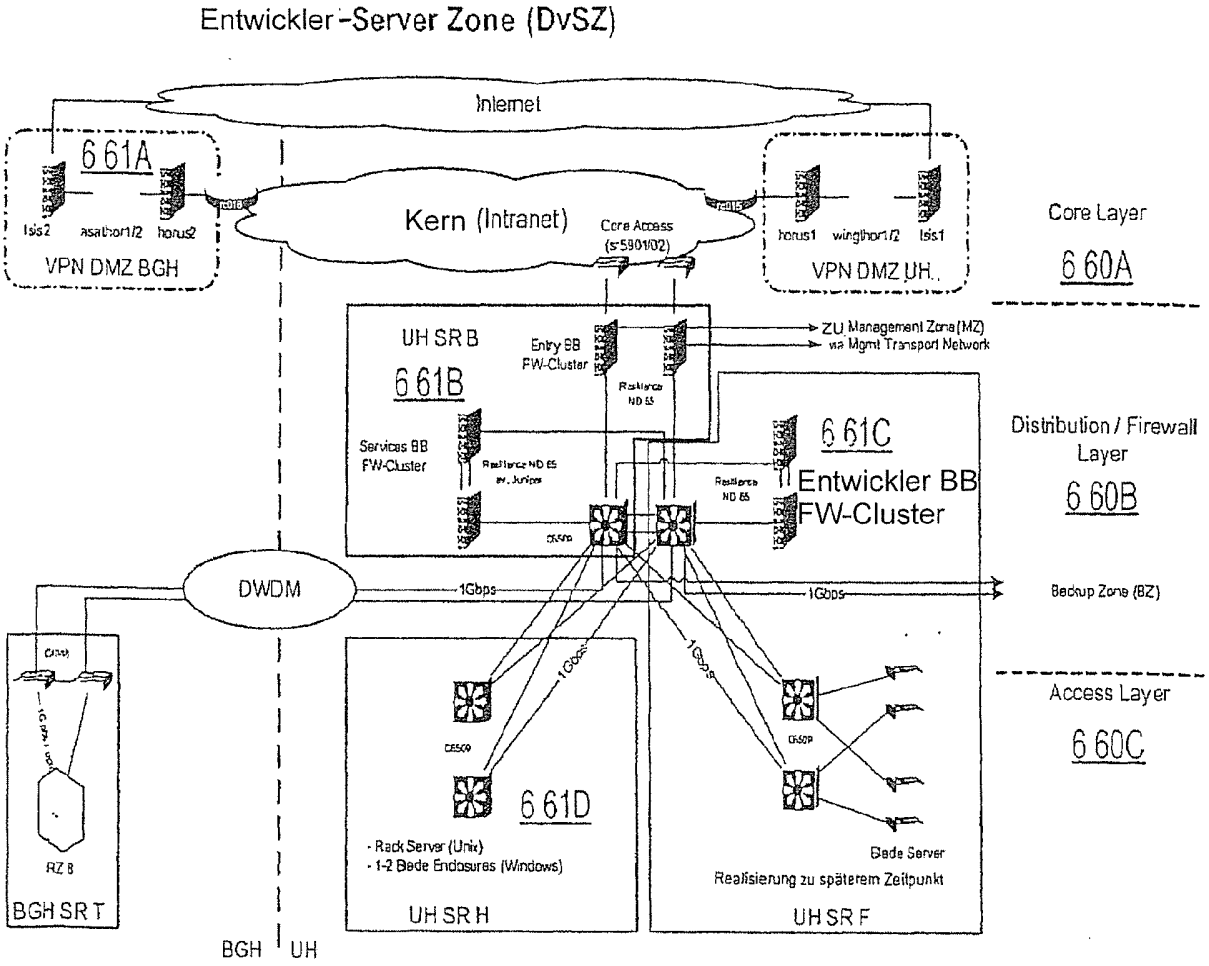
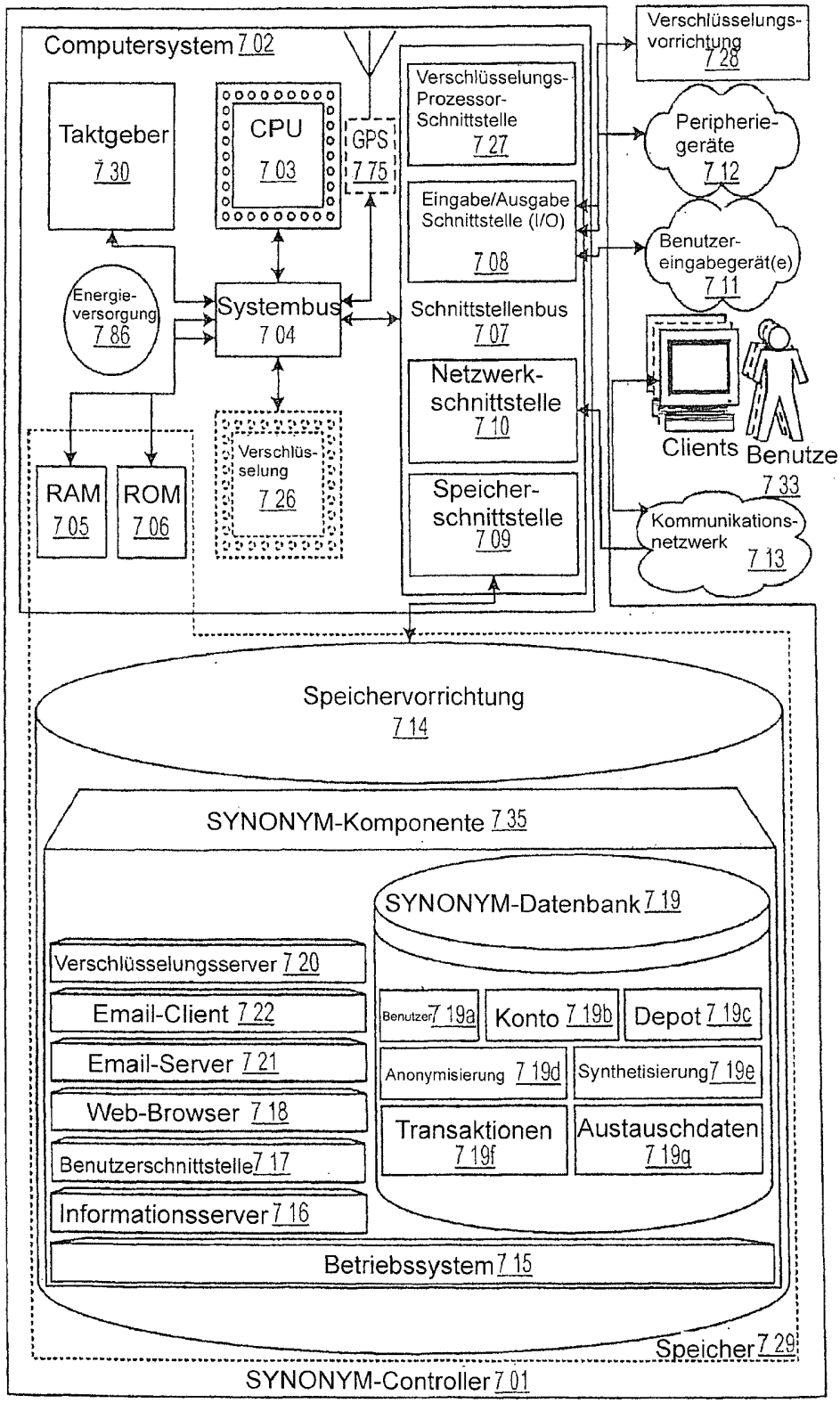

Entwickler-Server Zone (DvSZ)
Internet
661A
Isis2
asathor1/2
horus2
VPN DMZ BGH
Kern (Intranet)
Core Access
(s-5801/02)
horus1
wingthor1/2
Isis1
VPN DMZ UH
Core Layer
660A
UH SR B
661B
Entry BB
FW-Cluster
ZU Management Zone (MZ)
via Mgmt Transport Network
Services BB
FW-Cluster
661C
Entwickler BB
FW-Cluster
Distribution / Firewall
Layer
660B
DWDM
1Gbps
1Gbps
Backup Zone (BZ)
Access Layer
660C
661D
- Rack Server (Unix)
- 1-2 Bede Endosures (Windows)
Bede Server
Realisierung zu späterem Zeitpunkt
RZ B
BGH SR T
UH SR H
UH SR F
BGH
UH

Fig. 7


Computersystem 702
Taktgeber 730
CPU 703
GPS 775
Energie-versorgung 786
Systembus 704
RAM 705
ROM 706
Verschlüs-selung 726
Verschlüsselungs-Prozessor-Schnittstelle 727
Eingabe/Ausgabe Schnittstelle (I/O) 708
Schnittstellenbus 707
Netzwerk-schnittstelle 710
Speicher-schnittstelle 709
Verschlüsselungs-vorrichtung 728
Peripherie-geräte 712
Benutzer-eingabegerät(e) 711
Clients
Benutzer 733
Kommunikations-netzwerk 713
Speichervorrichtung 714
SYNONYM-Komponente 735
SYNONYM-Datenbank 719
Verschlüsselungsserver 720
Email-Client 722
Email-Server 721
Web-Browser 718
Benutzerschnittstelle 717
Informationsserver 716
Benutzer 719a
Konto 719b
Depot 719c
Anonymisierung 719d
Synthetisierung 719e
Transaktionen 719f
Austauschdaten 719g
Betriebssystem 715
Speicher 729
SYNONYM-Controller 701