(19) Deutsches Patent- und Markenamt

(10) **DE 11 2014 002 799 B4** 2020.02.20

(12) **Patentschrift**

(21) Deutsches Aktenzeichen: **11 2014 002 799.5**
(86) PCT-Aktenzeichen: **PCT/JP2014/003042**
(87) PCT-Veröffentlichungs-Nr.: **WO 2014/199611**
(86) PCT-Anmeldetag: **06.06.2014**
(87) PCT-Veröffentlichungstag: **18.12.2014**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **10.03.2016**
(45) Veröffentlichungstag der Patenterteilung: **20.02.2020**

(51) Int Cl.: ***G06F 21/62*** (2013.01)
G06F 21/53 (2013.01)
G06F 9/455 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
13/916,682 **13.06.2013** **US**

(73) Patentinhaber:
INTERNATIONAL BUSINESS MACHINES CORPORATION, Armonk, N.Y., US

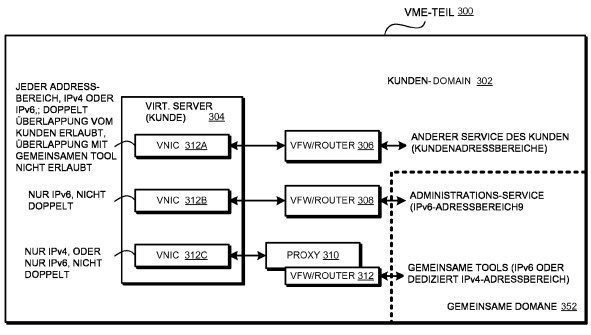
(74) Vertreter:
Spies & Behrndt Patentanwälte PartG mbB, 80687 München, DE

(72) Erfinder:
Donnellan, Sean, 71139 Ehningen, DE; Floyd III, Robart K., Essex Junction, Vt., US; Monaco, Robert P., Somers, N.Y., US; Mueller, Holger, 56412 Nomborn, DE; Robinson, Joseph D., Cary, N.C., US

(56) Ermittelter Stand der Technik:
siehe Folgeseiten

(54) Bezeichnung: **Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung**

(57) Hauptanspruch: Ein Verfahren für eine Absicherung einer Kundendomäne in einer virtuellen Mehr-Mieter-Umgebung, wobei das Verfahren aufweist:
- Konfigurieren, durch einen Prozessor, einer virtuellen Maschine (VM) für einen Kunden in der Kundendomäne;
- Konfigurieren, durch den Prozessor, eines ersten virtuellen Netzwerk-Interface (VNIC) in der VM, wobei das erste VNIC es einer ersten Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen;
- Zuweisen, durch den Prozessor, einer ersten Netzwerkadresse zu dem ersten VNIC, wobei die erste Netzwerkadresse innerhalb eines ersten Netzwerkadressbereiches liegt, der für die Kundendomäne ausgewählt ist;
- Konfigurieren, durch den Prozessor, eines zweiten VNIC in der VM, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird; und
- Konfigurieren, durch den Prozessor, eines dritten VNIC in der VM, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, ...



(56) Ermittelter Stand der Technik:

US	**8 369 333**	**B2**
US	**2009 / 0 328 061**	**A1**
US	**2011 / 0 261 828**	**A1**

Bound, J. [et al.]: Ipv6 Enterprise Network Analysis – IP Layer 3 Focus. Request for Comments: RFC 4852, April 2007. The Internet Engineering Task Force (IETF®) [online]. URL: https://tools.ietf.org/html/rfc4852 [abgerufen am 10.03.2016]

Cisco Systems, Inc.: Dual Stack Network. USA, 2010. URL: http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/IPV6at_a_glance_c45-625859.pdf [abgerufen am 10.03.2016] – Firmenschrift

OSI-Modell. In: Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 12. Juni 2013. URL: https://de.wikipedia.org/wiki/OSI-Modell?oldid=119477251 [abgerufen am 10.03.2016]

Schichtenarchitektur. In: Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 23. Mai 2013. URL: https://de.wikipedia.org/wiki/Schichtenarchitektur?oldid=118774046 [abgerufen am 10.03.2016]

Beschreibung

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich im Allgemeinen auf ein Verfahren, ein System und ein Computer-Programm-Produkt zum Verwalten einer Mehr-Mieter-Datenverarbeitungsumgebung (multi-tenant data processing environment). Insbesondere bezieht sich die vorliegende Erfindung auf ein Verfahren, ein System und ein Computer-Programm-Produkt zum Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung.

Hintergrund

[0002] Bestimmte Datenverarbeitungssysteme sind konfiguriert, um verschiedene Workloads gleichzeitig zu verarbeiten. Beispielsweise verarbeiten separate virtuelle Datenverarbeitungssysteme - wie beispielsweise separate virtuelle Maschinen (VM) - die auf einem einzelnen Host-Datenverarbeitungssystem konfiguriert sind, oft unterschiedliche Workloads für unterschiedliche Kunden oder Anwendungen.

[0003] In umfangreichen Datenverarbeitungsanlagen - wie etwa in einem Rechenzentrum - können Tausende von VMs zu einem gegebenen Zeitpunkt auf einem Host betrieben werden; dabei werden Hunderte, wenn nicht Tausende, solcher Hosts in dem Rechenzentrum gleichzeitig betrieben. Eine virtuelle Datenverarbeitungsumgebung - wie das beschriebene Rechenzentrum - wird häufig als „Cloud“ bezeichnet, welche Rechenkapazitäten und Verarbeitungs-Services mehreren Kunden auf Anforderung zur Verfügung stellt.

[0004] VMs werden auf einem Computerknoten installiert oder erzeugt, wenn es für Kunden-Workloads erforderlich ist, um Service-Level-Anforderungen zu erfüllen, bzw. aus vielen anderen Gründen. Darüber hinaus werden unterschiedliche Konfigurationen der VMs für unterschiedliche Zwecke benötigt. Wenn beispielsweise eine VM nur zur Bereitstellung einer allgemeinen Verarbeitungsplattform für einen Nutzer erzeugt wird, kann die VM nur mit dem Basisbetriebssystem und ohne Anwendungen erzeugt werden. Wenn in einem anderen Beispiel eine neue VM Anwendungs-Services liefern muss, kann die VM mit einem Betriebssystem und einem Application-Server darauf erzeugt werden.

[0005] In ähnlicher Weise können verschiedene Konfigurationen für VMs als Template-Images (Templates) vorkonfiguriert sein. Wenn eine VM, die eine spezifizierte vorbestimmte Konfiguration aufweist, auf einem Computerknoten erzeugt werden muss, wird ein geeignetes Template aus einem Template-Speicher - wie einer Datenbank oder einem File-System - ausgewählt und auf dem Computerknoten installiert, um eine VM mit der gewünschten Konfiguration zu erzeugen.

[0006] Im Kontext des hier vorgestellten Konzeptes können folgende Hintergrundinformationen genannt werden:

[0007] Das Dokument US 2011 / 0 261 828 A1 beschreibt ein Verfahren, bei dem Daten von einem virtuellen Switch, welches in einer Netzwerkvorrichtung installiert ist, in einem Cloud-Netzwerk empfangen werden. Dabei werden die Daten von einem externen Netzwerk empfangen und sind für eine oder mehrere virtuelle Maschinen in dem Cloud-Netzwerk bestimmt.

[0008] Das Dokument US 8 369 333 B2 beschreibt eine Möglichkeit zur Bereitstellung von transparentem Cloud-Computing mit einer virtualisierten Netzwerkinfrastruktur. Dabei wird es ermöglicht, eine Ressource in einem Rechenzentrum als Erweiterung eines Anwendernetzwerkes zu nutzen.

[0009] Das Dokument „Dual Stack Network“ von Cisco System (URL: http:/www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/IPV6at_a_glanc;e_c 45-625859.pdf, aufgerufen am 10.03.2016) beschreibt die Notwendigkeit einer Koexistenz zwischen einem IPv4- und einem IPv6-Netzwerk

[0010] Das Dokument von Bound, J. [et al.] „IPv6 Enterprise Network Analysis - IP Layer 3 Focus, Request for Comments“, RFC 4852, April 2007; the Internet Engineering Task Force (IETF®) [online], URL: https://tools.ietf.org/html/rfc4852 [abgerufen am 10.03.2016] beschreibt eine Analyse eines Übergangs zu IPv6 in Unternehmensnetzwerken mit einem Fokus auf den IP-Layer 3.

[0011] Das Dokument „OSI-Modell“ aus Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 12. Juni; 2013, URL: https://de.wikipedia.org/wiki/OSI-Modell?oldid=119477251 [abgerufen am 10.03.2016] beschreibt ein bekanntes Referenzmodell für Netzwerkprotokolle als Schichtenstruktur, wie es seit 1983 von der Internationalen Telekomunion genutzt wird.

[0012] Das Dokument Schichtenarchitektur. In: Wikipedia, The Free Encyclopedia. Bearbeitungsstand: 23. Mai 2013. URL: https://de.wikipedia.org/wiki/ Schichtenarchitektur?oldid=118774046 [abgerufen am 10.03.2016] beschreibt ein Schichtenarchitekturmodell, welches häufig als angewandtes Strukturierungsprinzip für die Architektur von Softwaresystemen genutzt wird. Dabei werden einzelne Aspekte des Softwaresystems konzeptionell einer Schicht zugeordnet.

[0013] Das Dokument US 2009 / 0 328 061 A1 beschreibt eine Adressierung des Problems, in dem Daten in einer anderen Domäne genutzt werden, während ein Programm in eine Skript-Sprache basierend auf Sicherheitsstandards eines HTTP-Client-Systems ausgeführt wird.

[0014] Allerdings beschreibt keines der genannten Dokumente einen Zugriff von einer VM auf eine andere bzw. eine dritte VM, sondern nur gegenseitige Datenzugriffe

[0015] Als Aufgabe des hier vorgestellten Konzeptes kann angesehen werden, dass ein verbesserter Mechanismus vorgestellt werden soll, der einen leistungsfähigen und eleganter Zugriff von einer virtuellen Maschine auf eine andere virtuelle Maschine ermöglicht.

Übersicht über die Erfindung

[0016] Das veranschaulichende Ausführungsbeispiel stellt ein Verfahren, ein System und ein Computer-Programm-Produkt für eine Bereitstellung einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung zur Verfügung. Ein Ausführungsbeispiel konfiguriert, durch einen Prozessor, eine virtuelle Maschine (VM) für einen Kunden in der Kundendomäne. Das Ausführungsbeispiel konfiguriert, durch den Prozessor, ein erstes virtuelles Netzwerk-Interface (VNIC) in der VM, wobei das erste VNIC es einer Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen. Das Ausführungsbeispiel weist, durch den Prozessor, eine erste Netzwerkadresse dem ersten VNIC zu, wobei die erste Netzwerkadresse innerhalb eines ersten Adressbereiches liegt, der für die Kundendomäne ausgewählt ist. Das Ausführungsbeispiel konfiguriert, durch den Prozessor, ein zweites VNIC in der VM, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird. Das Ausführungsbeispiel konfiguriert, durch den Prozessor, ein drittes VNIC in der VM, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abgesichert wird.

[0017] Ein anderes Ausführungsbeispiel weist einen oder mehrere computerlesbare konkrete Speichervorrichtungen auf. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, um eine virtuelle Maschine (VM) für einen Kunden in der Kundendomäne zu konfigurieren. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, um ein erstes virtuelles Netzwerk-Interface (VNIC) in der VM zu konfigurieren, wobei das erste VNIC es einer Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, um eine erste Netzwerkadresse dem ersten VNIC zuzuweisen, wobei die erste Netzwerkadresse innerhalb eines ersten Adressbereiches liegt, der für die Kundendomäne ausgewählt ist. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, um ein zweites VNIC in der VM zu konfigurieren, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, um ein drittes VNIC in der VM zu konfigurieren, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abzusichern.

[0018] Ein weiteres Ausführungsbeispiel weist einen oder mehrere Prozessoren, ein oder mehrere computerlesbare Speicher und einen oder mehrere computerlesbare konkrete Speichervorrichtungen auf. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um eine virtuelle Maschine (VM) für einen Kunden in der Kundendomäne zu konfigurieren. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausfüh-

rung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um ein erstes virtuelles Netzwerk-Interface (VNIC) in der VM zu konfigurieren, wobei das erste VNIC es einer Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um eine erste Netzwerkadresse dem ersten VNIC zuzuweisen, wobei die erste Netzwerkadresse innerhalb eines ersten Adressbereiches liegt, der für die Kundendomäne ausgewählt ist. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um ein zweites VNIC in der VM zu konfigurieren, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird. Das Ausführungsbeispiel weist weiterhin Programmanweisungen auf, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um ein drittes VNIC in der VM zu konfigurieren, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abgesichert wird.

Figurenliste

[0019] Neue Merkmale, die als charakterisierend für die Erfindung angenommen werden, sind durch die anhängenden Ansprüche beschrieben. Die Erfindung selbst dagegen sowie ein bevorzugter Ausführungsmodus, weitere zugehörige Ziele und Vorteile werden am besten durch eine Zuhilfenahme der folgenden detaillierten Beschreibung und der dargestellten Ausführungsbeispiele, wenn diese im Zusammenhang mit den begleitenden Figuren gelesen werden, verstanden.

Fig. 1 beschreibt ein Blockdiagramm für ein Datenverarbeitungssystem, in welchem die dargestellten Ausführungsbeispiele implementiert sein können.

Fig. 2 beschreibt ein Blockdiagramm eines Beispiels einer logisch partitionierten Plattform, in welcher die dargestellten Ausführungsbeispiele implementiert sein können.

Fig. 3 beschreibt ein Blockdiagramm einer Konfiguration zum Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel.

Fig. 4 beschreibt ein Flussdiagramm eines Beispielprozesses zum Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel.

Fig. 5 beschreibt ein Flussdiagramm eines anderen Beispielprozesses zum Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel.

Fig. 6 beschreibt ein Flussdiagramm eines anderen Beispielprozesses zum Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel.

Beschreibung von Ausführungsbeispielen

[0020] Eine Mehr-Mieter-Datenverarbeitungs-Umgebung (Multi-Tenant Environment) ist eine Datenverarbeitungsumgebung, bei dem ein oder mehrere Kundendatenverarbeitungssysteme „gehostet“ werden. Eine virtualisierte Mehr-Mieter-Datenverarbeitungs-Umgebung (Virtualisierungsumgebung, virtualisierte Mehr-Mieter-Umgebung, VME, Cloud) ist eine Mehr-Mieter-Umgebung, bei der virtuelle Maschinen konfiguriert sind, um Verarbeitungsaufgaben für mehrere Kunden abzuarbeiten.

[0021] Die dargestellten Ausführungsbeispiele berücksichtigen, dass ein Kunde in einer VME mit unterschiedlichen VMs für unterschiedliche Aufgaben konfiguriert ist. Beispielsweise kann ein Kunde eine VM konfiguriert haben, um einen Web-Service bereitzustellen, eine andere, um einen Backup-Anwendung-Support bereitzustellen, und eine andere als Host für eine Datenbank. Eine beliebige Anzahl von VMs kann für spezielle Aufgaben konfiguriert sein, eine gemischte Nutzung oder allgemeine Computing-Aufgaben.

[0022] Die dargestellten Ausführungsbeispiele berücksichtigen weiterhin, dass Kunden im Allgemeinen die Konfigurationen ihrer Anwendungen in einer VME steuern. Beispielsweise kann - anders als einige Ba-

sisanforderungen der VME, wo der Kunde „gehostet“ ist - ein Kunde jeden Internet-Protocol(IP)-Adressenbereich auswählen, um eine Kommunikation zwischen verschiedenen Anwendungen des Kunden, die auf einer oder mehreren VMs ausgeführt werden, zu ermöglichen. Einige Beispiele der Basisanforderungen an eine VME sind, dass die Kunden-VMs für Verwaltungsaufgaben erreichbar sein sollten, und dass die Adressierung des Kunden nicht in Konflikt mit der Adressierung stehen sollte, die für gemeinsame Services genutzt werden, die in der VME verfügbar sind.

[0023] Die dargestellten Ausführungsbeispiele berücksichtigen weiterhin, dass Kunden erwarten, dass ihr Satz von VMs und Anwendungen (Kundendomäne) sicher gegenüber anderen Kundendomänen ist. Mit anderen Worten: die Kommunikation zu und von einer Kundendomäne sollte keine Störungen mit Kommunikationen zu und von anderen Kundendomänen erzeugen, solange sie nicht beabsichtigt sind.

[0024] Die dargestellten Ausführungsbeispiele berücksichtigen, dass eine Absicherung einer Kundendomäne in einer VME ein schwieriges Problem darstellt. Ein Beispielgrund für die Schwierigkeit besteht darin, dass ein durch einen Kunden gewählter Adressbereich unbeabsichtigt mit einem Adressbereich, der durch einen anderen Kunden in der VME ausgewählt wurde, überlappt.

[0025] Die dargestellten Ausführungsbeispiele berücksichtigen, dass ein Durchsetzen einer strikten Adressbereichstrennungs-Policy praktisch nicht durchführbar, schwierig und teuer ist. Beispielsweise kann es sein, dass der Code einer Kundenanwendung geändert werden muss, um solch eine Policy durchzusetzen, was für den Kunden inakzeptabel wäre. Als anderes Beispiel sei erwähnt, dass eine Durchsetzung solch einer Policy in einer aktuell verfügbaren VME-Infrastruktur eine Konfiguration oder Rekonfiguration von einer oder mehrerer Kundendomänen erfordert, um einen neuen Kunden zu implementieren oder für eine Änderung einer Kundendomäne.

[0026] Die dargestellten Ausführungsbeispiele berücksichtigen weiterhin, dass in der derzeit verfügbaren VME-Infrastruktur Hardware-Vorrichtungen genutzt werden müssen, um die Kundendomänen von den Servern zu isolieren, welche administrativen Anwendungen und gemeinsamen Tools dienen. Folglich müssen Hardware-Elemente in einer VME-Änderung hinzugefügt, entfernt oder geändert werden, wenn sich Kundendomänen ändern. Solche Hardwareänderungen sind teuer und erfordern weiterhin strikte Policy-artige Lösungen für Adressbereichsabgrenzung, um das Problem der überlappenden Kundendomänenadressbereiche zu adressieren.

[0027] Die dargestellten Ausführungsbeispiele, die genutzt werden, um die Erfindung im Allgemeinen zu beschreiben, adressieren und lösen die oben dargestellten Probleme und andere Probleme im Zusammenhang mit Mehr-Mieter-Umgebungen. Die dargestellten Ausführungsbeispiele stellen ein Verfahren, ein System und ein Computer-Programm-Produkt für ein Bereitstellen einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung vor.

[0028] Die dargestellten Ausführungsbeispiele stellen eine Art einer Konfiguration von Komponenten in einer Kundendomäne vor, so dass die Kundendomäne sicher vor Störungen anderer Kundendomänen in der VME sind; das gilt auch dann, wenn die Kundendomäne einen Adressbereichen nutzt, der mit einer anderen Kundendomäne in der VME überlappt. Die dargestellten Ausführungsbeispiele stellen weiterhin eine Technik zur Sicherung einer Abgrenzung zwischen der Kundendomäne und den verwaltenden und gemeinsamen Tools vor, so dass Hardware-Elemente nicht erforderlich sind, um die Abgrenzung zu konfigurieren. Die hier beschriebenen Techniken, Prozeduren und die Art und Weise des Betriebes können in einem Template für eine VM implementiert sein, so dass, wenn eine VM in einer Kundendomäne konstruiert wird, die VM in einer Art und Weise konfiguriert ist, wie es durch diese Offenbarung beschrieben ist, ohne dass Hardwareänderungen in der Mehr-Mieter Umgebung erforderlich sind.

[0029] Die dargestellten Ausführungsbeispiele sind unter Berücksichtigung von bestimmten Datenverarbeitungssystemen, Umgebungen, Komponenten und Applikationen beschrieben. Jede spezifische Manifestation solcher Artefakte sind nicht dazu gedacht, die Erfindung begrenzen. Jede brauchbare Manifestation der Datenverarbeitungssysteme, Umgebungen, Komponenten und Anwendungen können aus dem Gesamtumfang der dargestellten Ausführungsbeispiele ausgewählt werden.

[0030] Darüber hinaus können dargestellte Ausführungsbeispiele unter Berücksichtigung jeder Art von Daten, Datenquellen oder Zugriff auf Datenquellen über ein Datennetzwerk implementiert werden. Jede Art von einer Datenspeichervorrichtung kann die Daten für ein Ausführungsbeispiel der Erfindung im Rahmen der Erfindung entweder lokal in einem Datenverarbeitungssystem oder über ein Datennetzwerk bereitstellen.

[0031] Die dargestellten Ausführungsbeispiele sind unter Nutzung von spezifischem Code, Design, Protokollen, Layouts, Schemata und Tools beschrieben und nicht auf die dargestellten Ausführungsbeispiele beschränkt. Darüber hinaus sind an einigen Stellen die dargestellten Ausführungsbeispiele unter Nutzung von bestimmter Software, Tools und Datenverarbeitungsumgebungen nur beispielhaft und der Klar-

heit der Beschreibung dienend beschrieben. Die dargestellten Ausführungsbeispiele können im Zusammenhang mit anderen vergleichbaren oder ähnlich gearteten Strukturen, Systemen, Anwendungen oder Architekturen genutzt werden. Ein dargestelltes Ausführungsbeispiel kann in Hardware, Software oder einer daraus bestehenden Kombination implementiert sein.

[0032] Die Beispiele in dieser Offenbarung werden nur für die Klarheit der Darstellung genutzt und begrenzen die dargestellten Ausführungsbeispiele nicht. Zusätzliche Daten, Operationen, Aktionen, Tasks, Aktivitäten und Manipulationen können aus der Beschreibung abgeleitet werden und ergeben sich aus dem Umfang der dargestellten Ausführungsbeispiele.

[0033] Jegliche hier aufgelistete Vorteile sind nur Beispiele und nicht dazu gedacht, um einschränkend auf die dargestellten Ausführungsbeispiele zu wirken. Zusätzliche oder andere Vorteile können sich durch spezifische dargestellte Ausführungsbeispiele ergeben. Darüber hinaus kann ein bestimmtes dargestelltes Ausführungsbeispiel alle oder keine der oben aufgelisteten Vorteile aufweisen.

[0034] Unter Bezugnahme auf die Figuren und insbesondere unter Bezugnahme auf die **Fig. 1** und **Fig. 2** sei erwähnt, dass diese Figuren Beispieldiagramme von Datenverarbeitungsumgebungen sind, in welchen dargestellte Ausführungsbeispiele implementiert sein können. **Fig. 1** und **Fig. 2** sind nur Beispiele, und es ist nicht beabsichtigt, irgendwelche Abgrenzungen hinsichtlich der Umgebungen, in welchen unterschiedliche Ausführungsbeispiele implementiert sein können, festzustellen oder zu implizieren. Eine bestimmte Implementierung kann viele Modifikationen hinsichtlich der beschriebenen Umgebungen und basierend auf der folgenden Beschreibung vornehmen.

[0035] **Fig. 1** stellt ein Blockdiagramm für ein Datenverarbeitungssystem dar, in denen die dargestellten Ausführungsbeispiele implementiert sein können. Das Datenverarbeitungssystem **100** kann ein symmetrisches Multiprozessorsystem (SMP) mit einer Mehrzahl von Prozessoren **101**, **102**, **103** und **104** sein, die mit dem System-Bus **106** verbunden sind. Beispielsweise kann das Datenverarbeitungssystem **101** ein IBM Power System sein, welches als Server in einem Netzwerk implementiert ist (Power Systems ist ein Produkt und ein Markenzeichen von International Business Maschinen Corporation in den Vereinigten Staaten und anderen Ländern). Alternativ kann ein einzelnes Prozessorsystem betrieben werden, und die Prozessoren **101**, **102**, **103** und **104** können Kerne des Einzelprozessor-Chips sein. Alternativ kann das Datenverarbeitungssystem **100** die Prozessoren **101**, **102**, **103**, **104** in jeder beliebigen Kombination von Prozessoren und Kernen aufweisen.

[0036] Weiterhin ist eine Speichersteuerung/Cache **108**, welcher ein Interface für eine Mehrzahl von lokalen Speichern **160**-**163** bereitstellt, mit dem System Bus **106** verbunden. Die I/O-Bus-Brücke **110** verbindet den System-Bus **106** und stellt ein Interface zum I/O-Bus **112** dar. Die Speichersteuerung/Cache **108** und die I/O-Brücke **110** können integriert sein, wie es dargestellt ist.

[0037] Das Datenverarbeitungssystem **100** ist ein logisch partitioniertes Datenverarbeitungssystem. Somit kann das Datenverarbeitungssystem **100** mehrere heterogene Betriebssysteme (oder mehrere Instanzen eines einzigen Betriebssystems) gleichzeitig ausführen. Jedes dieser mehreren Betriebssysteme kann in ihnen mehrere Softwareprogramme ausführen. Das ist logisch partitioniert, so dass verschiedene PCI-I/O-Adapter **120**-**121**, **128**-**129** und **136**, ein Grafikadapter **148** und ein Festplattenadapter **149** unterschiedlichen logischen Partitionen zugewiesen sein kann. In diesem Fall ist der Grafikadapter **148** mit einer Anzeigeeinheit (nicht dargestellt) verbunden, während der Festplattenadapter **149** mit einer Festplatte **150** verbunden ist und sie steuert.

[0038] Es sei also beispielsweise angenommen, dass das Datenverarbeitungssystem **100** in drei logische Partitionen **P1**, **P2** und **P3** aufgeteilt ist. Jeder PCI-I/O-Adapter **120**-**121**, **128**- **129**, **136**, der Grafik-Adapter **148**, der Festplattenadapter **149**, jeder der Host-Prozessoren **101**-**104** und Speicher der lokalen Speicher **160**-**163** ist einer der drei Partitionen zugewiesen. In diesen Beispielen kann der Speicher **160**-**163** die Form von Dual-In-Line-Speichermodulen (DIMMs) annehmen. DIMMS sind normalerweise auf einer pro-DIMM-Basis den Partitionen zugewiesen. Hier dagegen bekommt eine Partition einen Teil des gesamten Speichers, wie er von der Plattform aus gesehen wird. Beispielsweise können dem Prozessor **101**, einzelne Teile des Speichers von den lokalen Speichern **160**-**163** und die I/O-Adapters **120**, **128** und **129** der logischen Partition **P1** zugewiesen sein; die Prozessoren **102**-**103**, einige Teile des Speichers der lokalen Speicher **160**-**163**, und den PCI-I/O-Adaptern **121** und **136** können der Partition **P2** zugewiesen sein; und der Prozessor **104**, einzelne Teile des Speichers der lokalen Speicher **160**-**163**, des Grafik-Adapters **148** und der Festplattenadapter **149** kann der logischen Partition **P3** zugewiesen sein.

[0039] Jedes Betriebssystem innerhalb des Datenverarbeitungssystems **100** ist einer anderen logischen Partition zugewiesen. Somit kann jedes Betriebssystem, welches auf dem Datenverarbeitungssystem **100** ausgeführt wird, nur auf die I/O-Einheiten zugreifen, die sich in seiner logischen Partition befinden. Somit kann beispielsweise eine Instanz des

Advanced Interactive Executive (AIX - eingetragenes Warenzeichen) Betriebssystems innerhalb der Partition **P1** ausgeführt werden, eine zweite Instanz (Image) des AIX-Betriebssystems kann innerhalb der Partition **P2** ausgeführt werden, und eine Linux (eingetragenes Warenzeichen) oder IBM-i (eingetragenes Warenzeichen) Betriebssystem kann in der logischen Partition **P3** betrieben werden (AIX und IBM-i sind Marken der International Business Maschinen Corporation in den Vereinigten Staaten und anderen Ländern. Linux ist die Marke von Linus Torvalds in den Vereinigten Staaten und anderen Ländern).

[0040] Die Peripheral-Component-Interconnect-(PCI)-Host-Bridge **114**, die mit dem I/O-Bus **112** verbunden ist, stellt ein Interface zum lokalen PCI-Bus **115** bereit. Eine Anzahl von PCI-Input/Output-Adaptern **120**-**121** ist mit dem lokalen PCI-Bus **115** durch die PCI-zu-PCI-Brücke **116**, den PCI-Bus **116**, den PCI-Bus **119**, I/O-Slot **170** und I/O-Slot **171** verbunden. Die PCI-zu-PCI-Brücke **116** stellt ein Interface für PCI-Bus **118** und PCI-Bus **119** dar. Die PCI-I/O-Adapter **120** und **121** sind entsprechend in den I/O-Slots **170** und **171** platziert. Typische PCI-Bus-Implementationen unterstützen zwischen vier und acht I/O-Adaptern (d. h., Erweiterungs-Slots für weitere hinzufügbare Anschlüsse). Jeder PCI-I/O-Adapter **120**-**121** stellt ein Interface zwischen dem Datenverarbeitungssystem **100** und Input-/Output-Geräten - wie beispielsweise andere Netzwerk-Computer, die für das Datenverarbeitungssystem **100** Clients sind - dar.

[0041] Eine zusätzliche PCI-Host-Brücke **122** stellt ein zusätzlich es Interface für einen zusätzlichen lokalen PCI-Bus **123** bereit. Der lokale PCI-Bus **123** verbindet eine Mehrzahl von PCI-I/O-Adaptern **128**-**129**. Die PCI-I/O-Adapter **128**-**129** verbinden sich zum lokalen PCI-Bus **123** durch die PCI-zu-PCI-Brücke **124**, den PCI-Bus **116**, den PCI-Bus **127**, I/O-Slot **172** und I/O-Slot **173**. Die PCI-zu-PCI-Brücke **124** stellt ein Interface zum PCI-Bus **126** und PCI-Bus **127** dar. Die PCI-I/O-Adapter **128** und **129** sind entsprechend in den I/O-Slots **172** und **173** platziert. Auf diese Weise werden zusätzliche I/O-Geräte - wie beispielsweise Modems oder Netzwerkadapter - durch jeden der PCI-I/O-Adapter **128**-**129** unterstützt. Konsequenterweise erlaubt das Datenverarbeitungssystem **100** Verbindungen zu mehreren Netzwerk-Computern.

[0042] Der Memory-Mapped-Grafik-Adapter **148** steckt im I/O-Slot **144** und ist mit dem I/O-Bus **112** durch den PCI-Bus **144**, die PCI-zu-PCI Brücke **142**, dem lokalen PCI-Bus **141** und die PCI-Brücke **140** verbunden. Der Festplattenadapter **149** kann im I/O-Slot **175** platziert sein, der mit dem PCI-Bus **145** verbunden ist. Umgekehrt ist der PCI-Bus **145** mit der PCI-zu-PCI-Brücke **142** verbunden, welche mit der PCI-Brücke **140** durch den lokalen PCI-Bus **141** verbunden ist.

[0043] Die PCI-Brücke **130** bietet ein Interface für einen lokalen PCI-Bus **131**, um sich mit dem I/O-Bus **112** zu verbinden. Der PCI-I/O-Adapter **136** ist mit dem I/O-Slot **176** verbunden, welcher mit der PCI-zu-PCI-Brücke **132** durch den PCI-Bus **133** verbunden ist. Die PCI-zu-PCI-Brücke **132** ist mit dem lokalen PCI-Bus **131** verbunden. Der lokale PCI-Bus **131** ist auch mit der PCI-Brücke **130** verbunden, um das Prozessor-Mailbox-Interface und die ISA-Bus-Zugriff-Pass-Through-Logik **194** und die PCI-zu-PCI-Brücke **132** zu bedienen.

[0044] Das Service-Prozessor-Mailbox-Interface und die ISA-Bus-Zugriff-Pass-Through-Logik **194** leiten PCI-Zugriffe weiter, die für die PCI/ISA-Brücke **193** bestimmt sind. Der NVRAM-Speicher **192** ist mit dem ISA-Bus **196** verbunden. Der Service-Prozessor **135** ist mit dem Service-Prozessor-Mailbox-Interface und der ISA-Bus-Zugriff-Pass-Through-Logik **194** durch seinen lokalen PCI-Bus **195** verbunden. Der Service-Prozessor **135** ist auch mit den Prozessoren **101**-**104** über eine Mehrzahl von JTAG/I2C-Bussen **134** verbunden. Die JTAG/I2C-Busse **134** sind eine Kombination von JTAG/Scan-Bussen (vergleiche IEEE 1149.1) und Philips-**12C**-Bussen.

[0045] Alternativ können die JTAG/I2C-Busse **124** nur durch Philips-I2C-Busse oder nur JTAG/Scan-Busse ersetzt werden. Alle SP-ATTN-Signale der Prozessoren **101**, **102**, **103** und **104** sind gemeinsam mit einem Interrupt-Eingangssignal des Service-Prozessors **135** verbunden. Der Service-Prozessor **135** hat seinen eigenen lokalen Speicher **191** und hat Zugriff auf das Hardware-OP-Panel **190**.

[0046] Wenn das Datenverarbeitungssystem **150** zuerst mit Strom versorgt wird, nutzt der Service-Prozessor **135** die JTAG/12C-Busse **134**, um die System-(Host-)-Prozessoren **101**-**104**, die Speichersteuerung/Cache **108** und die I/O-Brücke **110** abzufragen. Am Ende dieses Schrittes hat der Service-Prozessor ein Inventar- und Topologie-Verständnis des Datenverarbeitungssystems **100**. Der Service-Prozessor **135** führt auch Build-In-Self-Tests (BISTs), Basic-Assurance-Tests (BATs) und Speichertests bezüglich aller gefundenen Elemente durch Abfrage der-Prozessoren **101**-**104**, der Speichersteuerung/Cache **108** und der I/O-Brücke **110** durch. Der Service-Prozessor **135** sammelt und berichtet jede Fehlerinformation bzgl. Fehlfunktionen, die während der BISTs, BATs und Speichertests erkannt wurden.

[0047] Wenn eine sinnvolle/gültige Konfiguration von Systemressourcen nach einer Entfernung von Elementen, die während der BISTs, BATs und Speichertests als fehlerhaft erkannt wurden, weiterhin sinnvoll sind, dann ist es dem Datenverarbeitungs-

system **100** erlaubt, damit fortzufahren, ausführbaren Code in die lokalen (Host-)-Speicher **160**-**163** zu laden. Der Service-Prozessor **135** gibt dann die Host-Prozessoren **101**-**104** für eine Ausführung von Code frei, welcher in die lokalen Speicher **160**-**163** geladen wurde. Während die Host-Prozessoren **101**-**104** Code von entsprechenden Betriebssystemen in dem Datenverarbeitungssystem **100** ausführen, geht der Service-Prozessor **135** in einen Überwachungs- und Fehlerberichtmodus über. Der Service-Prozessor **135** überwacht dabei beispielsweise Merkmale wie Geschwindigkeit und das Betriebsverhalten eines Ventilators, Temperatursensoren, Spannungsversorgungssteuerungen, behebbare und nicht behebbarer Fehler, die durch die Prozessoren **101**-**104** gemeldet werden, die lokalen Speicher **160**-**163** und die I/O-Brücke **110**.

[0048] Der Service-Prozessor **135** sichert und berichtet Fehlerinformationen, die sich auf alle überwachten Elemente des Datenverarbeitungssystems **100** beziehen. Der Service-Prozessor **135** gereift auch basierend auf der Art der Fehler und definierten Grenzwerte ein. Beispielsweise kann der Service-Prozessor **135** von überhandnehmenden behebbaren Fehlern eines Cache-Speichers eines Prozessors Notizen nehmen und entscheiden, dass dies auf einen schweren Ausfall hinweist. Der Service-Prozessor **135** kann basierend auf dieser Feststellung die entsprechende Ressource für eine die De-Konfiguration während aktuell laufender Sessions und zukünftige IPLs (Initial Program Loads) markieren. IPLs werden manchmal als „Boot“ oder „Bootstrap“ bezeichnet.

[0049] Das Datenverarbeitungssystem **100** kann unter Nutzung verschiedener kommerziell verfügbarer Computersysteme implementiert sein. Beispielsweise kann das Datenverarbeitungssystem **100** unter Nutzung von IBM-Power-Systemen von International Business Machines Corporation implementiert sein. Ein solches System kann eine logische Partitionierung unter Nutzung eines AIX-Betriebssystems unterstützen, welches auch von International Business Machines Corporation verfügbar ist.

[0050] Speicher, wie etwa der Speicher **191**, NV-RAM **192**, die lokalen Speicher **160**, **161**, **162** und **163** oder Flash-Speicher (nicht dargestellt) sind einige Beispiele von computernutzbaren Speichervorrichtungen. Die Festplatte **150**, ein CD-ROM (nicht dargestellt) und andere ähnliche nutzbare Geräte sind einige Beispiele für computernutzbare Speichervorrichtungen inklusive computernutzbare Speichermedien.

[0051] Fachleute werden erkennen, dass die Hardware, die in **Fig. 1** dargestellt ist, variieren kann. Beispielsweise können andere Peripheriegeräte - wie etwa optische Plattenlaufwerke oder Ähnliches - zusätzlich oder anstelle der dargestellten Hardware genutzt werden. In anderen Beispielen, können physische Ressourcen - wie etwa Adapter - als entsprechende virtuelle Ressourcen (nicht dargestellt) tirtualisiert werden; und die virtuellen Ressourcen können dann den verschiedenen Partitionen zugeordnet werden. In einem anderen Beispiel kann die Hardware, die in **Fig. 1** dargestellt ist, so konfiguriert sein, dass ein oder mehrere virtuelle I/O-Server (VIOS) (nicht dargestellt) genutzt werden. Die VIOS erlauben eine gemeinsame Nutzung von physischen Ressourcen, wie etwa Adaptern, Plattenlaufwerken, Steuerungen, Prozessoren, Speicher und Ähnlichem durch die unterstützten logischen Partitionen. Neben anderen Funktionen zwischen Partitionen reduziert ein gemeinsam genutztes VIOS die Notwendigkeit für große Mengen an Verkabelung, um eine Live-Migration durchzuführen. Die dargestellten Beispiele sind nicht dazu gedacht, architekturbezogene Limitation hinsichtlich der dargestellten Ausführungsbeispiele zu implizieren.

[0052] **Fig. 2** beschreibt ein Blockdiagramm eines Beispiels einer logisch partitionierten Plattform, in denen die dargestellten Ausführungsbeispiele implementiert sein können. Die Hardware der logisch partitionierte Plattform **200** kann beispielsweise mit Komponenten implementiert sein, die dem Datenverarbeitungssystem **100** in **Fig. 1** entsprechen.

[0053] Die logisch partitionierte Plattform **200** enthält partitionierte Hardware **230**, Betriebssysteme **202**, **204**, **206**, **208** und Plattform-Firmware **210**. Die Plattform-Firmware - wie etwa die Plattform-Firmware **210** - ist auch als Partitionierungs-Management-Firmware bekannt. Die Betriebssysteme **202**, **204**, **206** und **208** können mehrere Kopien eines einzigen Betriebssystems oder mehrerer heterogener Betriebssysteme sein, die gleichzeitig auf der logisch partitionierten Plattform **200** ausgeführt werden. Die Betriebssysteme können unter Nutzung von IBM-i implementiert sein, welches dazu ausgelegt ist, mit einer Partitionierungs-Management-Firmware wie beispielsweise einem Hypervisor zu interagieren. IBM-i wird nur als ein Beispiel für die dargestellten Ausführungsbeispiele genutzt. Natürlich können auch andere Arten von Betriebssystemen wie etwa AIX und Linux abhängig von der tatsächlichen Implementierung genutzt werden. Die Betriebssysteme **202**, **204**, **206** und **208** befinden sich entsprechend in den Partitionen **203**, **205**, **207** und **209**.

[0054] Ein Hypervisor ist ein Beispiel für Software, welche dazu genutzt werden kann, um die Partitionierungs-Management-Firmware **210** zu implementieren, und sie ist auch von International Business Machines Corporation erhältlich. Firmware ist „Software“, die in einem Speicher-Chip gespeichert ist, der seinen Inhalt auch ohne elektrische Stromversorgung behält, wie dies beispielsweise bei ROMs (re-

ad-only memory), programmierbaren ROMs PROM) löschbaren programmierbaren ROM (EPROM), elektrisch löschbaren programmierbaren ROMs (EEPROM) und nicht flüchtigem RAM (random access memory) (NVROM) der Fall ist.

[0055] Zusätzlich enthalten die Partitionen **203**, **205**, **207** und **209** entsprechende Partitionierungs-Firmware **210**, **213**, **215** und **217**. Die Partitionierungs-Firmware **210**, **213**, **215** und **217** kann unter Nutzung eines Inital Bootstrap Code von IEEE-1275 Standard Open Firmware und Runtime Abstraction Software (RTAS) implementiert sein, welche von International Business Machines Corporation erhältlich ist. Wenn die Partitionen **203**, **205**, **207** und **209** instanziiert sind, lädt die Plattform-Firmware **210** eine Kopie des Bootstrap-Code in die Partitionen **203**, **205**, **207** und **209**. Danach wird die Kontrolle auf den Bootstrap-Code übertragen, sodass der Bootstrap-Code dann die Open Firmware und RTAS lädt. Die Prozessoren, die mit den Partitionen assoziiert sind bzw. ihnen zugewiesen sind, werden dann auf die Partitionsspeicher verteilt, um die Partitions-Firmware auszuführen.

[0056] Partition **203** ist eine Beispiel-VM, die konfiguriert ist, um in einer Kundendomäne betrieben zu werden und um als Beispielanwendung **203A** zu dienen. Die Partition **203** ist mit mindestens drei virtuellen Netzwerk-Interface-Karten (VNICs) **212A**, **212B** und **212C** konfiguriert. Die Partition **205** ist einen weitere Beispiel-VM, die konfiguriert ist, um in einer Kundendomäne betrieben zu werden, und um als Beispiel für eine Datenbankanwendung **205A** zu dienen. Jede der Partitionen **203** und **205** ist mit mindestens drei virtuellen Netzwerk-Interface-Karten (VNICs) konfiguriert. Partition **203** umfasst VNICs **212A**, **212B** und Partition **205** enthält VNICs **214A**, **214B** und **214C**. Beispielsweise ist jede der VNICs **212A**, **212B**, **212C**, **214A**, **214B**, **214C** eine virtuelle Ressource, welche eine Kombination der physischen I/O-Adapter **248**, **250**, **252**, **254**, **256**, **258**, **260** und **262** in geeigneter Weise abbildet.

[0057] Die Partition **207** ist eine Beispiel-VM, die konfiguriert ist, um beispielhaften Administrationsservices **207A** zu dienen. Die Administrationsservices **207A** sind einsetzbar für eine Verwaltung der Partitionen **203** und **205** in der Kundendomäne. Die Administrationsservices **207A** nutzen den VNIC **216A**, um für diese Aufgabe auf die Partitionen **203** und **205** in der Kundendomäne zuzugreifen. VNIC **216A** ist eine virtuelle Ressource, die auf irgendeinen der physischen I/O-Adapter **248**, **250**, **252**, **254**, **256**, **258**, **260** und **262** in geeigneter Weise abgebildet wird. In einem Ausführungsbeispiel wird ein VNIC - wie etwa VNIC **216A** - auf eine virtuelle Ressourcen in einem VIOS-Instanz abgebildet, welche dann mit einem oder mehreren dieser physischen Adapter verbunden wird. Beispielsweise stellt die VIOS-Instanz in solch einer Implementierung Redundanz durch einen gemeinsam genutzten Ethernet-Adapter über 2 VIOS sicher. In einem Ausführungsbeispiel sind die Administrationsservices **207A** auf einem Host (nicht dargestellt) konfiguriert, der sich von der logischen Partitions-Plattform **200** unterscheidet, sind zu greifbar von der Kundendomäne über ein Datennetzwerk, und der VNIC **216A** bildet auf einen anderen physischen I/O-Adapter in dem bestimmten Host ab. In einem Ausführungsbeispiel umfasst die Kundendomäne Partitionen von VMs in anderen Host-Systemen (nicht dargestellt), die sich von der logischen Partitionierungsplattform **200** unterscheiden, und auf die von den Partitionen **203** und **205** über ein Datennetzwerk zugegriffen werden kann.

[0058] Die Partition **209** ist eine Beispiel-VM, die konfiguriert ist, um als ein Beispiel für gleichzeitig genutzte (shared) Tools **209A** zu dienen. Die shared Tools **209A** werden in der Partition **209** durch die Partitionen **203** und **205** in der Kundendomäne über ein Datennetzwerk über den VNIC **218A** erreicht. Der VNIC **218A** ist eine virtualisierte Ressource, die auf irgend einen der physischen I/O-Adapter **248**, **250**, **252**, **254**, **256**, **258**, **260** und **262** in geeigneter Weise abgebildet wird. In einem Ausführungsbeispiel sind die shared Tools **209A** auf einem Host (nicht dargestellt) konfiguriert, der sich von der logischen Partitionierungsplattform **200** unterscheidet, sind zugreifbar von der Kundendomäne über ein Datennetzwerk, und der VNIC **218A** bildet auf einen anderen physischen I/O-Adapter in dem bestimmten Host ab.

[0059] Der Domain-Name-Service (DNS) **220A** stellt eine oder mehrere kundenspezifische Ansichten eines DNS-Service dar, welcher in der VME ausgeführt wird, in der die logische Partitions-Plattform **200** betrieben wird. Kundenspezifische Ansichten von DNS **220A** exponiert nur die Routen zu denjenigen Administrationsservices **207A** und shared Tools **209A**, welche für die Kundendomäne autorisiert sind, welche die Partitionen **203** und **205** aufweisen.

[0060] Die virtuelle Firewall und der Router (VFW/Router) **220B** ist eine virtuelle Routing-Ressource in der VME. Die VFW/Router **220B** ist in der Kundendomäne instanziiert, um eine Kommunikation zwischen Servern, Anwendungen und VMs in der Kundendomäne zu ermöglichen. Mehr als eine Instanz, die der Instanz **220B** ähnlich ist, kann in der Kundendomäne erzeugt werden, ohne von dem Umfang der dargestellten Ausführungsbeispiele abzuweichen.

[0061] Der Proxy **220C** - allein oder in Verbindung mit einer Kombination der VFW/Router **220B** und DNS220A - ermöglicht eine Kommunikation zwischen den Komponenten in der Kundendomäne und den Administration-Services **207A** und den shared Tools **209A**, ohne eine Verwirrung oder Einflussnahme von anderen Kundendomänen, welche in ei-

nem Adressbereich genutzt werden, welcher mit dem Adressbereich, der von der Kundendomäne genutzt wird, überlappt. Der Proxy **220C** kann ein Reverse-Proxy, ein bidirektionaler Proxy oder eine andere geeignete Ausführung eines Proxy-Servers sein. Darüber hinaus kann der Proxy **220C** eine Transformationsvorrichtung sein - wie ein NAT - oder ein Tool, welches ein Policy-basierendes Routing zu gemeinsam genutzten Ziel-Tools auszuführen in der Lage ist. Das Policybasierende Routing erlaubt eine Überlappung von Bereichen, die basierend auf der Quell-Router/Firewall unterschiedlich geroutet und deshalb zu einem bestimmten (VM) gemeinsam genutzten Tool geroutet werden, welches aufgesetzt wurde, um entweder die überlappenden Adressierungsbereichsprobleme zu lösen oder um eine Anwendung (Tool) herzustellen, welche keine „Mehrfach-Miete“ unterstützt, wie beispielsweise durch Instanziierung.

[0062] Die partitionierte Hardware **230** umfasst eine Mehrzahl von Prozessoren **232**-**238**, eine Mehrzahl von Systemspeichereinheiten **240**-**246**, eine Mehrzahl von Eingabe/Ausgabe-(I/O)-Adaptern **248**-**262** und eine Speichereinheit **270**. Jeder der Prozessoren **232**-**238**, der Speichereinheiten **240**-**246**, des NV-RAM-Speichers **298** und der I/O-Adapter **248**-**262** kann einer der Partitionen **203**, **205**, **207**, **209** innerhalb der logisch partitionierten Plattform **200** zugewiesen sein, wobei jede der Partitionen **203**, **205**, **207** und **209** einem der Betriebssysteme **202**, **204**, **206** und **208** entspricht.

[0063] Die Partitionierungs-Management-Firmware **210** führt eine Reihe von Funktionen und Services für die Partitionen **203**, **205**, **207** und **209** aus, um die Partitionierung der logisch partitionierten Plattform **200** zu erzeugen und zu erzwingen. Die Partitionierungs-Management-Firmware **210** ist eine firmwareimplementierte virtuelle Maschine, die zu der darunterliegenden Hardware identisch ist. Somit erlaubt die Partitionierungs-Management-Firmware **210** eine gleichzeitige Ausführung von unabhängigen Betriebssystem-Images **202**, **204**, **206** und **208** durch eine Virtualisierung aller der Hardware-Ressourcen der logisch partitionierten Plattform **200**.

[0064] Der Service-Prozessor **290** kann dazu genutzt werden, um verschiedene Services bereitzustellen - wie etwa einer Verarbeitung von Plattformfehlern in den Partitionen. Diese Services können auch als ein Service-Agent agieren, um Fehler zurück zu einem Hersteller - wie International Business Machines Corporation - zu melden. Aktionen der Partitionen **203**, **205**, **207**, und **209** können durch eine Hardware-Management-Konsole gesteuert werden - wie beispielsweise der Hardware-Management-Konsole **280**. Die Hardware-Management-Konsole **280** ist ein separates Datenverarbeitungssystem von dem aus ein Systemadministrator verschiedene Funktionen inklusive einer Re-Allokation von Ressourcen zu unterschiedlichen Partitionen vornehmen kann.

[0065] Die Hardware in den **Fig. 1-2** kann abhängig von der Implementierung variieren. Andere interne Hardware oder Peripheriegeräte, wie etwa Flash-Speicher, äquivalenter nicht flüchtige Speicher oder optische Laufwerke oder Ähnliches kann zusätzlich oder anstelle von einzelnen Hardware-Elementen, die in den **Fig. 1-2** dargestellt sind, genutzt werden. Eine Implementierung der dargestellten Ausführungsbeispiele kann auch alternative Architekturen für eine Verwaltung von Partitionen nutzen, ohne von dem Umfang der Erfindung abzuweichen.

[0066] **Fig. 3** stellt ein Blockdiagramm einer Konfiguration für eine Bereitstellung einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel dar. VME-Teil **300** ist Teil einer VME, in der mehrere Kundendomänen konfiguriert sind. Die Kundendomäne **302** ist eine Beispiel-Kundendomäne in dem VME-Teil **300**. Der virtuelle Server **304** ist eine Beispielmaschine, die in der Beispiel-Kundendomäne **302** konfiguriert ist. Der virtuelle Server **304** kann unter Nutzung der Partition **302** aus **Fig. 2** implementiert sein. Die VNICs **312A**, **312B**, und **312C** sind entsprechende Ausführungsbeispiele der VNICs **212A**, **212B** und **212C**.

[0067] In einem Ausführungsbeispiel wird das VNIC **312A** betrieben, um eine Kommunikation zwischen den verschiedenen Servern zu ermöglichen, die in der Kundendomäne **302** betrieben werden. Der VNIC **312A** kommuniziert mit VFW/Router **306**, um eine oder mehrere Server, Komponenten oder Anwendungen in der Kundendomäne **302** zu erreichen.

[0068] Der VNIC **312A** ist konfiguriert, um eine Adresse aus dem Adressbereich der Kundenauswahl zu nutzen. Der ausgewählte Adressbereich und konsequenterweise die Adresse, die dem VNIC **312A** zugewiesen wird, kann eine IPv4-Adressierung oder eine IPv6-Adressierung entsprechend der Kundenwahl nutzen.

[0069] Kunden können den Adressbereich, der für Maschinen, Server, Systeme, Adapter, Komponenten oder Anwendungen, die in der gemeinsam genutzten Domäne **312** betrieben wird, nicht nutzen. Während Adressbereiche, welche sich zwischen zwei oder mehreren Kundendomänen überlappenden, unwahrscheinlich sind, wenn man IPv6 nutzt, gibt es doch eine gewisse Möglichkeit einer solchen Überlappung, wenn man sich für IPv4-Bereiche entscheidet. Überlappungen in der IPv4-Adressierung werden dadurch wahrscheinlicher, dass weniger und weniger freie eindeutige IPv4-Adressen übrig bleiben, und weil große Anteile von IPv4-Adressenräu-

men wiederverwendet werden und nicht als eindeutig verfügbar sind.

[0070] In einem Ausführungsbeispiel ist der VNIC **312A** doppelt ausgebildet, nämlich konfiguriert, um sowohl den IPv4-TCP/IP-Stack als auch den IPv6-TCP/IP-Stack zu nutzen. Vorteilhafter Weise erlaubt es die doppelt ausgeführte Konfiguration den Kunden, jede beliebige Adressierungsspezifikation zu nutzen, um von einer Adressspezifizierung zu einer anderen zu wechseln oder eine Kombination von Adressierungsspezifikationen innerhalb ihrer Domäne zu nutzen, ohne Hardware-Zusätze oder Änderungen in der unterstützten VME zu benötigen.

[0071] In einem Ausführungsbeispiel wird der VNIC **312B** betrieben, um eine Kommunikation zwischen dem virtuellen Server **304** und Servern, die Administrationsservices bereitstellen - wie etwa gemeinsam genutztenSpeicher-/Backup-/Installations-/Wiederherstellungs-Services, zu ermöglichen. Der VNIC 3N2B kommuniziert mit dem VFW/Router **308**, um einen oder mehrere Administrationsservices zu erreichen. Neben anderen Gründen nutzt ein Ausführungsbeispiel den zweiten VNIC **312B**, um umfangreichen Speicherverkehr separat von anderem Datenverkehr zu halten, um so eine Nutzung von Jumbo-Frames zu erlauben, welche extensive Routing-Pfad-Unterstützung benötigen.

[0072] Der VNIC **312B** ist konfiguriert, um eine IPv6-Adresse zu nutzen, die einem Kunden eindeutig zugewiesen ist. In einem Ausführungsbeispiel ist der VNIC **312A** konfiguriert, nur den IPv6-TCP/IP-Stack zu nutzen. Administrationsservices sind in der VME nur über IPv6-Adressen erreichbar, die eindeutig für eine Nutzung in einer gemeinsam genutzten Domäne **352** zugewiesen sind. IPv6 nutzt eine **128**-Bit-Adresse, was 2^{128} oder ungefähr 3,4 * 10^{38} Adressen ermöglicht, oder mehr als 7,9 *10^{28} mal so viele wie IPv4, welches eine 32-Adresse nutzt. Weil der IPv6-Adressraum im Vergleich zum IPv4-Adressraum extrem groß ist und weil die gleiche IPv6-Adresse nicht unterschiedlichen Entitäten zugeordnet ist, ist eine versehentliche Überlappung von Adressen höchst unwahrscheinlich, wenn nicht sogar unmöglich.

[0073] In einem Ausführungsbeispiel wird der VNIC **312C** betrieben, um eine Kommunikation zwischen dem virtuellen Server **304** und Servern, die gemeinsam genutzte Tools bereitstellen, zu ermöglichen. Der VNIC **312C** kommuniziert mit dem Proxy **310** und der VFW/Router **312**, um auf ein oder mehrere gemeinsam genutzte Tools zuzugreifen. Der Proxy **310** erlaubt eine zuverlässige Kommunikation zwischen Anwendungen in der Kundendomäne **302** und Anwendungen in der gemeinsam genutzten Domäne **352** im Falle einer Überlappung zwischen den Adressbereichen, die für die Kundendomäne **302** und andere Kundendomänen genutzt werden.

[0074] Der VNIC **312C** ist konfiguriert, um entweder eine IPv4- entsprechend einer Kundenwahl oder eine IPv6-Adresse, die eindeutig einem Kunden (oder eindeutig dem Hosting-Provider zugewiesen ist, der es dann für den Kunden allokiert) zugewiesen ist, zu nutzen, aber nicht beide. Dementsprechend ist der VNIC **312C** konfiguriert, um entsprechend entweder den IPv4-TCP/IP-Stack oder den IPv6-TCP/IP Stack zu nutzen, aber nicht beide.

[0075] Die Auswahl zwischen der IPv4-Adressierung und der IPv6-Adressierung in dem VNIC **312C** hängt wegen der Interoperabilitätsbeschränkung zwischen der IPv4-Adressierung und der IPv6-Adressierung von der Adressierung ab, die in der gemeinsam genutzten Domäne **352** für die gemeinsam genutzten Tools genutzt wird. Wenn die gemeinsam genutzten Tools über eine IPv4-Adressierung erreicht werden können, die für die gemeinsam genutzte Domäne **352** reserviert ist, dann ist der VNIC **312C** mit einer IPv4-Adresse konfiguriert. Wenn die gemeinsam genutzten Tools über eine IPv6-Adressierung erreicht werden können, dann ist der VNIC **312C** mit einer IPv6-Adresse konfiguriert.

[0076] Generell können alle oder einige Kombinationen der virtuellen Komponenten, die in diesen Figuren dargestellt sind, in einem Template für eine VM für einen Kunden spezifiziert sein. Wenn ein Kunde ein erstes Mal versorgt werden soll, wiederversorgt werden soll oder die Kundenumgebung sich geändert hat, behält die Erzeugung und Konfiguration der virtuellen Komponenten - wie beschrieben - die Sicherheit der Kundendomäne, ohne Bedenken für überlappende Kundenadressbereiche und ohne die Notwendigkeit für Hardware-Modifikationen in der VME.

[0077] **Fig. 4** beschreibt ein Flussdiagramm für einen Beispielprozess für eine Bereitstellung einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung entsprechend einem dargestellten Ausführungsbeispiel. Der Prozess **400** kann durch einen Administrationsservice - wie etwa einem der Administrationsservices **207A** von **Fig. 2** - implementiert werden.

[0078] Ein Administrationsservice beginnt den Prozess **400** durch eine Konfiguration eines Kunden-Servers als eine VM (Block **402**). Mehr als ein Kunden-Server kann einen Tier (engl. the tier) in einer Kundendomäne belegen. Beispielsweise können Web-Services einen Tier bilden, Datenbankservices können einen anderen Tier bilden und Anwendungen können einen dritten Tier bilden. Jede beliebige Anzahl von Tiers sind in ähnlicher Weise möglich, ohne vom Umfang der dargestellten Ausführungsbeispiele abzuweichen. Der Administration-Service erzeugt

drei virtuelle lokale Netzwerke (virtual local arean network - VLAN) für jeden Kundenserver-Tier. Der Prozess **400** wird unter Berücksichtigung eines Beispiel-Tier nur aus Gründen der Klarheit der Beschreibung und ohne eine Limitation der dargestellten Ausführungsbeispiele zu implizieren, beschrieben.

[0079] Dementsprechend konfiguriert der Administrationsservice einen ersten VNIC in der VM (Block **404**). Der Administrationsservice konfiguriert den ersten VNIC mit einer Adresse von irgendeinem Adressbereich nach Kundenwahl unter der Erwartung eines reservierten Adressbereiches (Block **406**). Der Administrationsservice konfiguriert den ersten VNIC mit einem IPv4-TCP/IP-Stack und einem IPv6-TCP/IP-Stack (Block **408**).

[0080] Der Administrationsservice ermöglicht es einer Anwendung auf der VM, auf eine andere Anwendung auf einem anderen virtuellen Server des Kunden zuzugreifen, und zwar unter Nutzung des ersten VNIC und einer VFW/Router, um die Kommunikation zwischen den Kundenservern auch dann abzusichern, wenn ein Kunde einen Adressbereich nutzt, der mit dem Adressbereich eines anderen Kunden überlappt (Block **410**). Der Administrationsservice beendet den Prozess **400** oder springt aus dem Prozess **400** am Exit-Punkt heraus, der mit „A“ markiert ist, um in einen anderen Prozess - wie beispielsweise Prozess **500** in **Fig. 5** - an einem korrespondierenden Eingangspunkt, der mit „A“ markiert ist, zu springen.

[0081] Fig. 5 beschreibt ein Flussdiagramm eines anderen Beispielprozesses für eine Bereitstellung einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung in Übereinstimmung mit einem dargestellten Ausführungsbeispiel. Der Prozess **500** kann in einem IPv4-Administrationsservice - wie etwa einem der Administrationsservices **207A** von **Fig. 2** - implementiert sein.

[0082] Ein Administrationsservice beginnt den Prozess **500** oder steigt in den Prozess **500** an einem Eintrittspunkt ein, der mit „A“ markiert ist, durch eine Konfiguration eines zweiten VNIC in der VM eines Kunden, wie etwa der VM, die in Block **402** von **Fig. 4** (Block **502**) konfiguriert wurde. Der Administrationsservice konfiguriert den zweiten VNIC mit einer eindeutigen IPv6-Adresse des Kunden (Block **504**). Der Administrationsservice konfiguriert den zweiten VNIC mit einem IPv6-Stack (Block **506**).

[0083] Der Administrationsservice ermöglicht einen Zugriff auf die Kunden-VM von einem Administration-Server in einer gemeinsam genutzten Domäne in der VME unter Nutzung des zweiten VNIC und einer VFW/Router (Block **508**). Der Administrationsservice beendet entweder den Prozess **500** oder springt aus dem Prozess **500** an einem Exit-Punkt heraus, der mit „B“ markiert ist, um einen anderen Prozess, wie etwa Prozess **600** aus **Fig. 6** an einem korrespondierenden Eingangspunkt, der mit „B“ markiert ist, fortzusetzen.

[0084] Fig. 6 beschreibt ein Flussdiagramm eines anderen Beispielprozesses zur Bereitstellung einer sicheren Kundendomäne in eine virtualisierte Mehr-Mieter-Umgebung in Übereinstimmung mit einem dargestellten Ausführungsbeispiel. Der Prozess **600** kann mittels eines Administrationsservice wie einem der Administrationsservices **207A** gemäß **Fig. 2** implementiert sein.

[0085] Ein Administrationsservice beginnt den Prozess **600** oder steigt in den Prozess **600** bei einem Eintrittspunkt, der mit „B“ markiert ist, ein durch eine Konfiguration eines dritten VNIC in der Kunden-VM, wie etwa der VM, die in Block **402** von **Fig. 4** (Block **602**) konfiguriert wurde. Der Administrationsservice ermittelt, ob die IPv4- oder IPv6-Adressspezifikationen bei der Konfiguration des dritten VNIC (Block **604**) genutzt werden sollen.

[0086] Wenn eine IPv6-Adressierung für den dritten VNIC genutzt werden soll („IPv6“-Weg von Block **604**) konfiguriert der Administrationsservice den dritten VNIC mit einer eindeutigen IPv6-Adresse des Kunden (Block **606**). Der Administrationsservice konfiguriert den dritten VNIC mit einem IPv6-TCP/IP-Stack (Block **608**).

[0087] Der Administrationsservice ermöglicht einen Zugriff auf ein gemeinsam genutztes Tool in der VME der Kunden-VM unter Nutzung des dritten VNIC und einer VFW/Router (Block **610**). Danach beendet der Administrationsservice den Prozess **500**.

[0088] Wenn eine IPv4-Adressierung für den dritten VNIC benutzt werden soll („IPv4“-Weg von Block **604**), konfiguriert der Administrationsservice den dritten VNIC mit einer IPv4-Adresse aus dem vom Kunden gewählten IPv4-Adressbereich, der mit einem Adressbereich eines anderen Kunden aber nicht mit einem reservierten Adressbereich überlappen kann (Block **612**). Der Administrationsservice konfiguriert den dritten VNIC mit einem IPv4-TCP/IP-Stack (Block **614**).

[0089] Der Administrationsservice ermöglicht einen Zugriff auf ein gemeinsam genutztes Tool in der VME von der Kunden-VM unter Nutzung des dritten VNIC, eines Proxy, und einer VFW/Router (Block **616**). Danach beendet der Administrationsservice den Prozess **500**.

[0090] Das Flussdiagramm und die Blockdiagramme in den Figuren beschreiben die Architektur, Funktionalität und den Betrieb von möglichen Implementierungen von Systemen, Verfahren und Compu-

ter-Programm-Produkten entsprechend unterschiedlichen Ausführungsbeispielen der vorliegenden Erfindung. In dieser Hinsicht kann jeder Block in dem Flussdiagramm oder Blockdiagrammen ein Modul, ein Segment, oder einen Code-Abschnitt repräsentieren, welcher eine oder mehrere ausführbare Instruktionen zur Implementierung der spezifizierten logischen Funktionen(en) aufweist. Es sei auch darauf hingewiesen, dass in einigen alternativen Implementierungen die Funktionen, die in dem Block dargestellt sind, in einer anderen Reihenfolge ausgeführt werden können als es in den Figuren dargestellt ist. Zum Beispiel können zwei Blöcke, die nacheinander dargestellt sind tatsächlich im Wesentlichen gleichzeitig ausgeführt werden, oder Blöcke können manchmal in umgekehrter Reihenfolge - abhängig von der betroffenen Funktionalität - ausgeführt werden. Es sei auch darauf hingewiesen, dass jeder Block der Blockdiagramme und/oder Flussdiagrammdarstellungen und Kombinationen von Blöcken in den Blockdiagrammen und/oder Flussdiagrammdarstellungen durch speziell dafür vorgesehene Hardware-basierende Systeme implementiert sein können, welche die spezifizierten Funktionen oder Aktionen oder Kombinationen von speziell dafür vorgesehene Hardware und Computer-Instruktionen ausführen.

[0091] Somit werden ein Computer-implementiertes Verfahren, ein System und ein Computer-Programm-Produkt durch die dargestellten Ausführungsbeispiele für eine Bereitstellung einer sicheren Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung vorgestellt. Ein Ausführungsbeispiel ist auf eine Bereitstellung an einen neuen Kunden, eine Wiederbereitstellung an einen Kunden oder Änderungen einer Bereitstellung für einen Kunden in einer VME bezogen, so dass die Kundendomäne mit Hinblick auf eine Kommunikation innerhalb und außerhalb der Kundendomäne abgesichert ist. Ein Ausführungsbeispiel gewährleistet die Sicherheit der Isolation von anderen Kundendomänen ohne Berücksichtigung von Adressbereichen, die durch andere Kunden in der VME genutzt werden. Ein Ausführungsbeispiel nutzt Virtualisierungskomponenten für eine Bereitstellung der sicheren Kundendomänen-Elemente. Im Ergebnis kann die Bereitstellung und Sicherheit einer Kundendomäne in einem größeren Ausmaß automatisiert werden als es derzeit möglich ist. Darüber hinaus kann jede der Änderung in der Kundendomäne, in den Administrationsservices oder in gemeinsam genutzten Tools der VME unterstützt werden, ohne dass Ergänzungen oder Änderungen an der Hardware erforderlich wären.

[0092] Ein Fachmann wird verstehen, dass Aspekte der vorliegenden Erfindung als System, Verfahren oder Computer-Programm-Produkt ausgeführt sein können. Dementsprechend können Aspekte der vorliegenden Erfindung die Form einer reinen Hardware-Implementierung, einer reinen Software-Implementierung oder eines Ausführungsbeispiels (inklusive Firmware, permanent vorhandener Software, Microcode, usw.) bestehend aus einer Kombination aus Software- und Hardware-Aspekten, auf die im allgemeinen hier als ein „Schaltkreis“, ein „Modul“ oder „System“ Bezug genommen wird, annehmen. Darüber hinaus können Aspekte der vorliegenden Erfindung die Form eines Computer-Programm Produktes annehmen, welches in einem oder mehreren computerlesbaren Speichervorrichtungen oder computerlesbaren Medien verkörpert ist, welches darin computerlesbaren Programmcode aufweist.

[0093] Jede Kombination von einem oder mehreren computerlesbaren SpeicherVorrichtungen) oder computerlesbaren Medien können verwendet werden. Das Computer-lesbare Medium kann ein computerlesbares Speichermedium sein. Ein computerlesbares Speichermedium kann beispielsweise - ohne darauf beschränkt zu sein - ein elektronisches, magnetisches, optisches, elektromagnetisches, Infrarot- oder Halbleitersystem, ein Apparat oder eine Vorrichtung oder jede andere geeignete Kombination des Aufgezählten sein. Explizitere Beispiele (eine nicht erschöpfende Liste) von computerlesbaren Speichervorrichtungen kann Folgendes aufweisen: eine elektrische Verbindung mit zwei oder mehr Drähten, eine tragbarer Computer-Diskette, ein Festplattenlaufwerk, ein RAM (random access memory), ein Nur-Lese-Speicher (ROM), ein löschbarer und programmierbarer Nur-Lese-Speicher (EPROM, oder Flash-Memory), eine optische Faser, eine tragbare Compact Disc Read-Only Memory (CD-ROM), eine optische Speichervorrichtung, eine magnetische Speichervorrichtung oder irgend eine geeignete Kombination des Aufgezählten. Im Kontext dieses Dokumentes kann eine computerlesbare Speichervorrichtung jede konkrete Vorrichtung oder Medium sein, welches ein Programm zur Nutzung durch oder in Kombination mit einem Instruktionen-ausführenden System, Gerät oder Vorrichtung enthalten oder speichern kann.

[0094] Programmcode, der in einer computerlesbare Speichervorrichtung oder in einem computerlesbaren Medium verkörpert ist, kann unter Nutzung jedes geeigneten Mediums übertragen werden - inklusive aber nicht darauf beschränkt - drahtlos, drahtgebunden, durch ein optisches Glasfaserkabel, Hochfrequenzen, usw. oder jede geeignete Kombination des Aufgezählten.

[0095] Computer-Programmcode zum Ausführen von Operationen von Aspekten der vorliegenden Erfindung können in einer Kombination von einer oder mehreren Programmiersprachen geschrieben sein, inklusive einer Objekt-orientierten Programmiersprache, wie Java, Smalltalk, C++ oder ähnlichen oder konventionellen prozeduralen Programmiersprachen wie etwa der „C“-Programmiersprache oder ähn-

lichen Programmiersprachen. Der Programmcode kann komplett auf dem Anwendercomputer, teilweise auf dem Anwendercomputer als alleinstehendes Softwarepaket, teilweise auf dem Anwendercomputer und teilweise auf einem entfernten Computer oder gänzlich auf dem entfernten Computer oder Server ausgeführt werden. In dem letzten Szenario kann der entfernte Computer mit dem Anwendercomputer durch jeder Art von Netzwerk verbunden sein, inklusive eines LAN (local area network), eines WAN (wide area network), oder eine Verbindung kann mit einem externen Computer hergestellt werden (beispielsweise durch das Internet unter Nutzung eines Internet Service Providers).

[0096] Aspekte der vorliegenden Erfindung sind hierin unter Bezugname auf Flussdiagramme und/oder Blockdiagramm oder Verfahren, Geräte (Systeme) und Computer-Programm-Produkte entsprechend den Ausführungsbeispielen der Erfindung dargestellt. Es versteht sich, dass jeder Block der Flussdiagrammdarstellung und/oder Blockdiagrammen und Kombinationen von Blöcken in den Flussdiagrammdarstellung und/oder Blockdiagrammen durch Computer-Programm-Instruktionen implementiert sein kann. Diese Computer-Programm-Instruktionen können einem oder mehreren Prozessoren von einem oder mehreren General Purpose Computern, speziell ausgelegten Computern oder anderen programmierbaren Datenverarbeitungsvorrichtungen zur Verfügung gestellt werden, um eine Maschine zu erzeugen, so dass die Instruktionen, welche durch den einen oder die mehreren Prozessoren des Computers oder anderer programmierbare Datenverarbeitungsvorrichtungen ausgeführt werden, Mittel zur Implementierung der Funktionen/Aktionen erzeugen, die in dem Flussdiagramm und/oder Blockdiagrammblock oder Blöcken spezifiziert sind.

[0097] Diese Computer-Programm-Instruktionen können außerdem auf einem oder mehreren computerlesbaren Speichervorrichtungen oder computerlesbaren Medien gespeichert sein, welche einen oder mehrere Computer, einen oder mehrere programmierbare Datenverarbeitungsvorrichtungen oder einen oder mehrere andere Vorrichtungen anweisen. in einer bestimmten Art zu arbeiten, so dass die Instruktionen, die in dem einen oder mehreren Computer lesbaren Speichervorrichtungen oder Computer lesbaren Medien gespeichert sind, einen Herstellungsartikel inklusive Instruktionen erzeugen, welcher die Funktion/Aktion, die in dem Flussdiagramm und /der dem Blockdiagrammblock oder den -blöcken spezifiziert sind, erzeugen, zu erzeugen.

[0098] Die Computer-Programm-Instruktionen können auch auf einen oder mehrere Computer, einen oder mehrere programmierbare Datenverarbeitungsvorrichtungen, oder einen oder mehrere andere Vorrichtungen geladen werden, um eine Reihe von operativen Schritten zu bewirken, die auf dem einen oder mehreren Computer, dem einen oder mehreren programmierbaren Datenverarbeitungsvorrichtungen oder der einen oder mehreren anderen Vorrichtungen ausgeführt werden, um einen Computer-implementierten Prozess zu erzeugen, so dass die Instruktionen, welche auf dem einen oder mehreren Computern, der einen oder mehreren programmierbaren Datenverarbeitungsvorrichtung(en) oder dem einen oder mehreren Vorrichtungen Prozesse für eine Implementierung der Funktion/Aktion, die in dem Flussdiagramm und/oder Blockdiagrammblock oder -blöcken spezifiziert ist.

Patentansprüche

1. Ein Verfahren für eine Absicherung einer Kundendomäne in einer virtuellen Mehr-Mieter-Umgebung, wobei das Verfahren aufweist:
- Konfigurieren, durch einen Prozessor, einer virtuellen Maschine (VM) für einen Kunden in der Kundendomäne;
- Konfigurieren, durch den Prozessor, eines ersten virtuellen Netzwerk-Interface (VNIC) in der VM, wobei das erste VNIC es einer ersten Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen;
- Zuweisen, durch den Prozessor, einer ersten Netzwerkadresse zu dem ersten VNIC, wobei die erste Netzwerkadresse innerhalb eines ersten Netzwerkadressbereiches liegt, der für die Kundendomäne ausgewählt ist;
- Konfigurieren, durch den Prozessor, eines zweiten VNIC in der VM, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird; und
- Konfigurieren, durch den Prozessor, eines dritten VNIC in der VM, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abgesichert ist, und wobei die VM, die zweite VM, die dritte VM und die vierte VM auf einer logisch partitionierten Plattform implementiert sind.

2. Das Verfahren gemäß Anspruch 1, wobei der erste Adressbereich mit einem zweiten Netzwerkadressbereich überlappt, der für die zweite Kundendomäne ausgewählt ist.

3. Das Verfahren gemäß Anspruch 1, wobei die erste Adresse eine IPv4-Adresse aufweist, und wobei der Zugriff auf die zweite Anwendung über einen ersten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

4. Das Verfahren gemäß Anspruch 1, wobei die VM einen Tier in der Kundendomäne belegt, und wobei der Tier mindestens drei virtuelle Local-Area-Networks aufweist, wobei das Verfahren weiterhin aufweist:
- Zuweisen, durch den Prozessor, einer zweiten Netzwerkadresse zu dem zweiten VNIC, wobei die zweite Netzwerkadresse eine IPv6-Adresse aufweist, und wobei der Zugriff von der dritten Anwendung über einen zweiten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

5. Das Verfahren gemäß Anspruch 4, wobei der Zugriff auf die dritte Anwendung weiterhin durch eine erste Instanz eines Domain-Name-Service (DNS) ermöglicht wird, der für die Kundendomäne erzeugt wird, wobei die erste Instanz des DNS ein Routing aufweist, welches für eine Administration der Kundendomäne autorisiert ist.

6. Das Verfahren gemäß Anspruch 1, weiterhin aufweisend:
- Zuweisen, durch den Prozessor, einer dritten Netzwerkadresse zu dem dritten VNIC, wobei die dritte Netzwerkadresse eine IPv6-Adresse aufweist, und wobei der Zugriff auf die vierte Anwendung über einen dritten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

7. Das Verfahren gemäß Anspruch 1, weiterhin aufweisend:
- Zuweisen, durch den Prozessor, einer dritten Netzwerkadresse zu dem dritten VNIC, wobei die dritte Netzwerkadresse im ersten Netzwerkadressbereich liegt, der für die Kundendomäne ausgewählt ist, und wobei der erste Netzwerkadressbereich mit einem zweiten Netzwerkadressbereich überlappt, der für eine zweite Kundendomäne ausgewählt ist, und wobei der Zugriff auf die vierte Anwendung über einen dritten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

8. Das Verfahren gemäß Anspruch 7, wobei der Zugriff auf die vierte Anwendung weiterhin durch eine zweite Instanz eines Domain-Name-Service ermöglicht wird, der für die Kundendomäne erzeugt wird, wobei die zweite Instanz des DNS ein Routing zu einem Satz von Shared-Anwendungen, die für die Kundendomäne verfügbar sind, aufweist.

9. Das Verfahren gemäß Anspruch 1, wobei jedes von dem ersten VNIC, dem zweiten VNIC, und dem dritten VNIC auf einen virtualisierten I/O-Adapter in einem virtuellen I/O-Server abgebildet wird, und wobei der virtualisierte I/O-Adapter auf einen physikalischen I/O-Adapter abgebildet wird.

10. Das Verfahren gemäß Anspruch 1, wobei einer von dem zweiten VNIC und dem dritten VNIC konfiguriert ist, um Jumbo-Frames während einer Software-Installation auf der VM zu nutzen.

11. Ein Computer-Programm-Produkt aufweisend einen oder mehrere computerlesbare konkrete Speichervorrichtungen oder computerlesbare Programmanweisungen, die in der einen oder den mehreren Speichervorrichtungen gespeichert sind, und die, wenn sie durch einen oder mehrere Prozessoren ausgeführt werden, das Verfahren gemäß einem der Ansprüche 1 bis 10 ausführen.

12. Ein Computersystem aufweisend einen oder mehrere Prozessoren, ein oder mehrere computerlesbare Speicher, ein oder mehrere computerlesbare konkrete Speichervorrichtungen und Programmanweisungen, welche auf dem einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch den einen oder mehrere Prozessoren über den einen oder die mehreren Speicher und durch den einen oder die mehreren Prozessoren das Verfahren gemäß einem der Ansprüche 1 bis 10 ausführen.

13. Ein Computer-Programm-Produkt für eine Absicherung einer Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung, wobei das Computer-Programm Produkt aufweist:
- eine oder mehrere computerlesbare konkrete Speichervorrichtungen;
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Konfigurieren, einer virtuellen Maschine (VM) für einen Kunden in der Kundendomäne;
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Konfigurieren eines ersten virtuellen Netzwerk-Interface (VNIC) in der VM, wobei das erste VNIC es einer ersten Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen;
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Zuweisen einer ersten Netzwerkadresse zu dem ersten VNIC, wobei die erste Netzwerkadresse innerhalb eines ersten Adressbereiches liegt, der für die Kundendomäne ausgewählt ist;
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Konfigurieren eines zweiten VNIC in der VM, wobei das zweite VNIC es einer dritten Anwendung außerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzu-

greifen, wobei der zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird; und
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Konfigurieren eines dritten VNIC in der VM, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei das dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abgesichert wird, und wobei die VM, die zweite VM, die dritte VM und die vierte VM auf einer logisch partitionierten Plattform implementiert sind.

14. Das Computer-Programm-Produkt gemäß Anspruch 13, wobei der erste Netzwerkadressbereich mit einem zweiten Netzwerkadressbereich überlappt, der für die zweite Kundendomäne ausgewählt ist.

15. Das Computer-Programm-Produkt gemäß Anspruch 13, wobei die erste Adresse eine IPv4-Adresse aufweist, und wobei der Zugriff auf die zweite Anwendung über einen ersten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

16. Das Computer-Programm-Produkt gemäß Anspruch 13, wobei die VM einen Tier in der Kundendomäne belegt, und wobei der Tier mindestens drei virtuelle Local-Area-Networks aufweist, weiterhin aufweist:
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Zuweisen, durch den Prozessor, einer zweiten Netzwerkadresse zu dem zweiten VNIC, wobei die zweite Netzwerkadresse eine IPv6-Adresse aufweist, und wobei der Zugriff von der dritten Anwendung über einen zweiten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

17. Das Computer-Programm-Produkt gemäß Anspruch 16, wobei der Zugriff auf die dritter Anwendung weiterhin durch eine erste Instanz eines Domain-Name-Service ermöglicht wird, der für die Kundendomäne erzeugt wird, wobei die erste Instanz des DNS ein Routing aufweist, welches für eine Administration der Kundendomäne autorisiert ist.

18. Das Computer-Programm-Produkt gemäß Anspruch 13, weiterhin aufweisend:
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Zuweisen einer dritten Netzwerkadresse zu dem dritten VNIC, wobei die dritte Netzwerkadresse eine IPv6-Adresse aufweist, und wobei der Zugriff auf die vierte Anwendung über einen dritten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

19. Das Verfahren gemäß Anspruch 1, weiterhin aufweisend:
- Programmanweisungen, die auf mindestens einem der einen oder der mehreren Speichervorrichtungen gespeichert sind, zum Zuweisen einer dritten Netzwerkadresse zu dem dritten VNIC, wobei die dritte Netzwerkadresse im ersten Netzwerkadressbereich liegt, der für die Kundendomäne ausgewählt ist, und wobei der erste Netzwerkadressbereich mit einem zweiten Netzwerkadressbereich überlappt, der für eine zweite Kundendomäne ausgewählt ist, und wobei der Zugriff auf die vierte Anwendung über einen Dritten virtuellen Router ermöglicht wird, der in der Kundendomäne instanziiert ist.

20. Ein Computersystem, zur Absicherung einer Kundendomäne in einer virtualisierten Mehr-Mieter-Umgebung, wobei das Computersystem aufweist:
- einen oder mehrere Prozessoren, einen oder mehrere computerlesbare Speicher und einen oder mehrere computerlesbare konkrete Speichervorrichtungen;
- Programmanweisungen, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um eine virtuelle Maschine (VM) für einen Kunden in der Kundendomäne zu konfigurieren;
- Programmanweisungen, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um ein erstes virtuelles Netzwerk-Interface (VNIC) in der VM zu konfigurieren, wobei das erste VNIC es einer ersten Anwendung auf der VM ermöglicht, auf eine zweite Anwendung in einer zweiten VM in der Kundendomäne zuzugreifen;
- Programmanweisungen, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um eine erste Netzwerkadresse zu dem ersten VNIC zuzuweisen, wobei die erste Netzwerkadresse innerhalb eines ersten Netzwerkadressbereiches liegt, der für die Kundendomäne ausgewählt ist;
- Programmanweisungen, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um ein zweites VNIC in der VM zu konfigurieren, wobei das zweite VNIC es einer dritten Anwendung au-

ßerhalb der Kundendomäne ermöglicht, auf die VM in der Kundendomäne zuzugreifen, wobei das zweite VNIC konfiguriert ist, eine Adressierungsspezifikation zu nutzen, die von einem Server der dritten Anwendung verwendet wird; und

- Programmanweisungen, die in mindestens einer der einen oder mehreren Speichervorrichtungen gespeichert sind, zur Ausführung durch mindestens einen der einen oder mehreren Prozessoren über mindestens einen des einen oder der mehreren Speicher, um eine drittes VNIC in der VM zu konfigurieren, wobei das dritte VNIC es der ersten Anwendung ermöglicht, auf eine vierte Anwendung zuzugreifen, die außerhalb der Kundendomäne ausgeführt wird, und wobei der dritte VNIC konfiguriert ist, um eine Adressierungsspezifikation zu nutzen, die von einem Server für die vierte Anwendung verwendet wird, wodurch eine Datenkommunikation, die der Kundendomäne zugehörig ist, gegenüber Einflüssen von einer Datenkommunikation, die einer zweiten Kundendomäne zugehörig ist, abgesichert ist, und wobei die VM, die zweite VM, die dritte VM und die vierte VM auf einer logisch partitionierten Plattform implementiert sind.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1

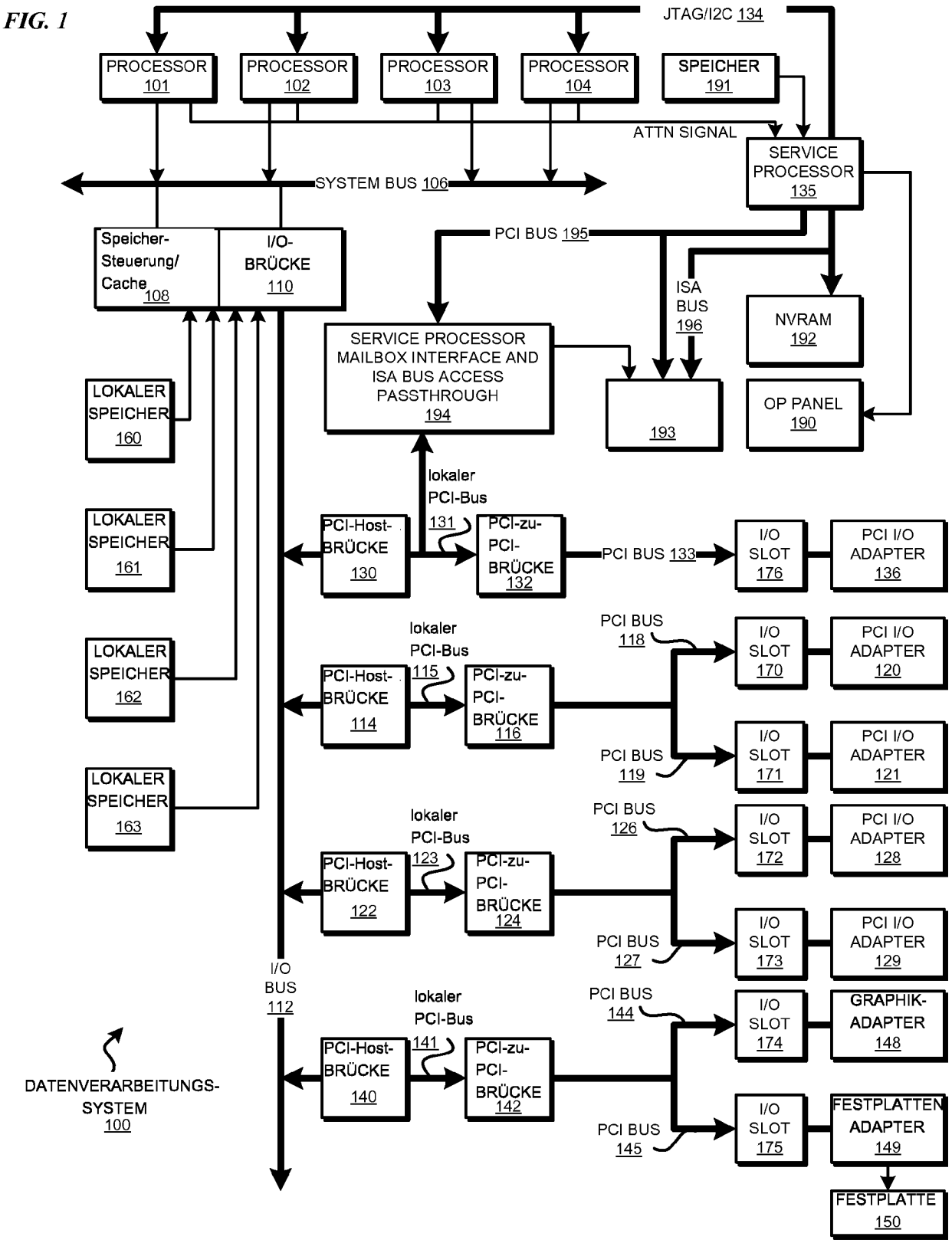


FIG. 2

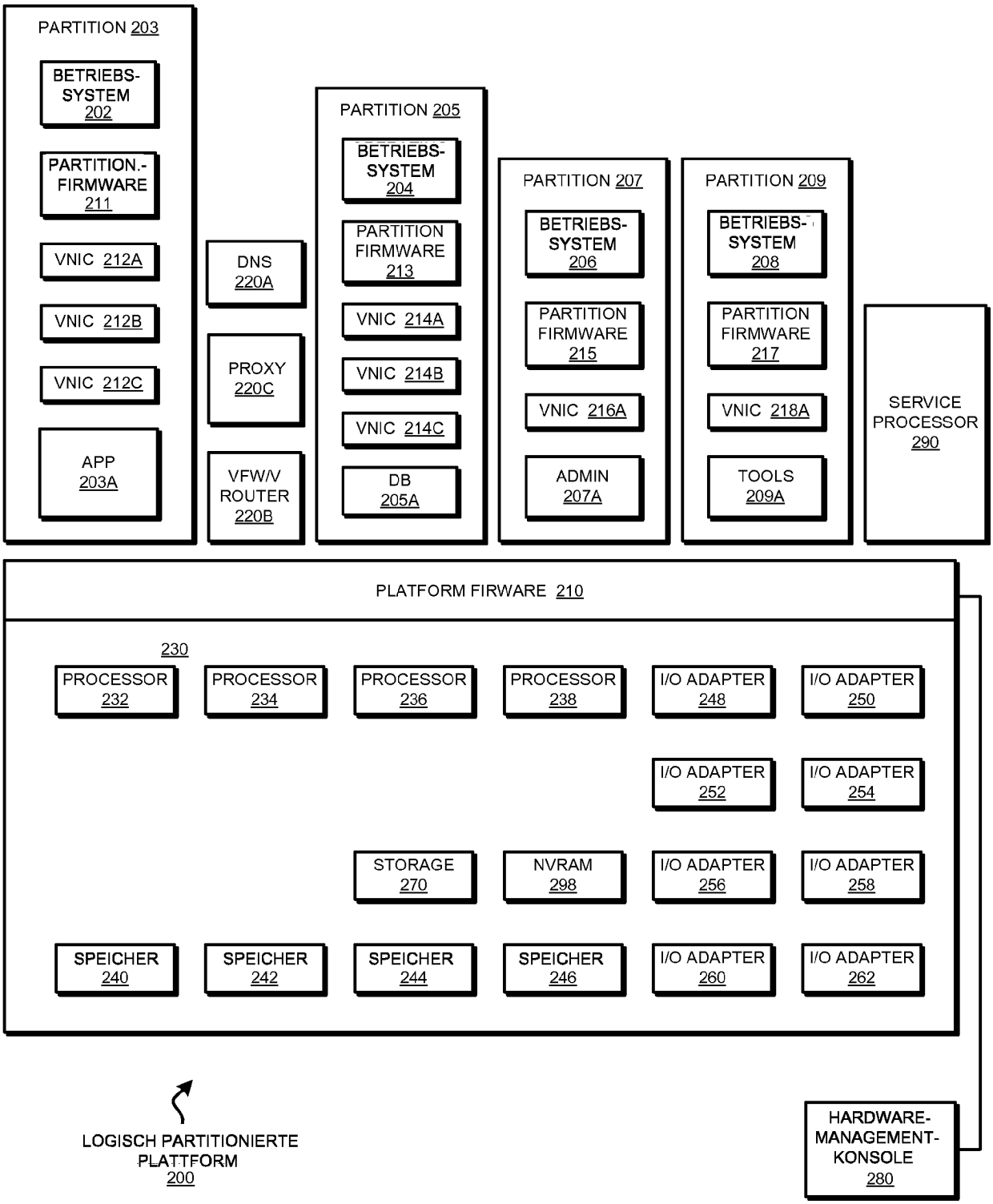

PARTITION 203
BETRIEBS-SYSTEM 202
PARTITION.-FIRMWARE 211
VNIC 212A
VNIC 212B
VNIC 212C
APP 203A
DNS 220A
PROXY 220C
VFW/V ROUTER 220B
PARTITION 205
BETRIEBS-SYSTEM 204
PARTITION FIRMWARE 213
VNIC 214A
VNIC 214B
VNIC 214C
DB 205A
PARTITION 207
BETRIEBS-SYSTEM 206
PARTITION FIRMWARE 215
VNIC 216A
ADMIN 207A
PARTITION 209
BETRIEBS-SYSTEM 208
PARTITION FIRMWARE 217
VNIC 218A
TOOLS 209A
SERVICE PROCESSOR 290
PLATFORM FIRWARE 210
230
PROCESSOR 232
PROCESSOR 234
PROCESSOR 236
PROCESSOR 238
I/O ADAPTER 248
I/O ADAPTER 250
I/O ADAPTER 252
I/O ADAPTER 254
STORAGE 270
NVRAM 298
I/O ADAPTER 256
I/O ADAPTER 258
SPEICHER 240
SPEICHER 242
SPEICHER 244
SPEICHER 246
I/O ADAPTER 260
I/O ADAPTER 262
LOGISCH PARTITIONIERTE PLATTFORM 200
HARDWARE-MANAGEMENT-KONSOLE 280

FIG. 3

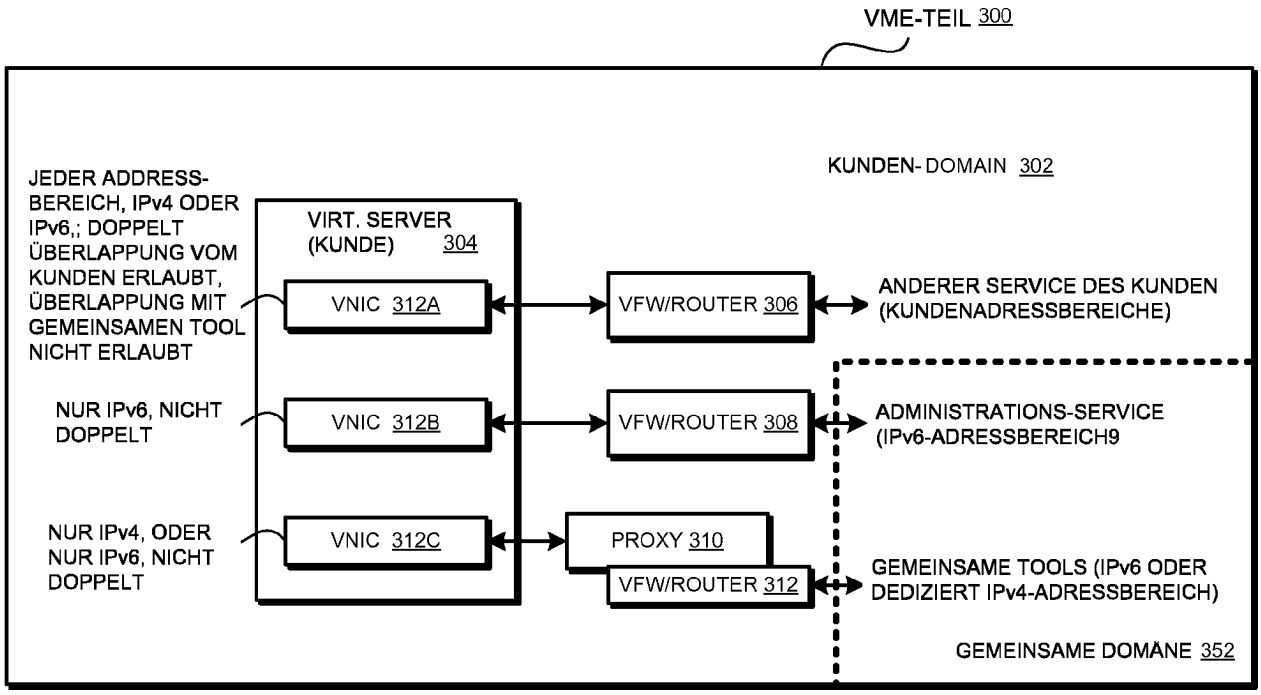

VME-TEIL 300
KUNDEN- DOMAIN 302
JEDER ADDRESS-BEREICH, IPv4 ODER IPv6,; DOPPELT ÜBERLAPPUNG VOM KUNDEN ERLAUBT, ÜBERLAPPUNG MIT GEMEINSAMEN TOOL NICHT ERLAUBT
VIRT. SERVER (KUNDE) 304
VNIC 312A
VFW/ROUTER 306
ANDERER SERVICE DES KUNDEN (KUNDENADRESSBEREICHE)
NUR IPv6, NICHT DOPPELT
VNIC 312B
VFW/ROUTER 308
ADMINISTRATIONS-SERVICE (IPv6-ADRESSBEREICH9
NUR IPv4, ODER NUR IPv6, NICHT DOPPELT
VNIC 312C
PROXY 310
VFW/ROUTER 312
GEMEINSAME TOOLS (IPv6 ODER DEDIZIERT IPv4-ADRESSBEREICH)
GEMEINSAME DOMÄNE 352

FIG. 4

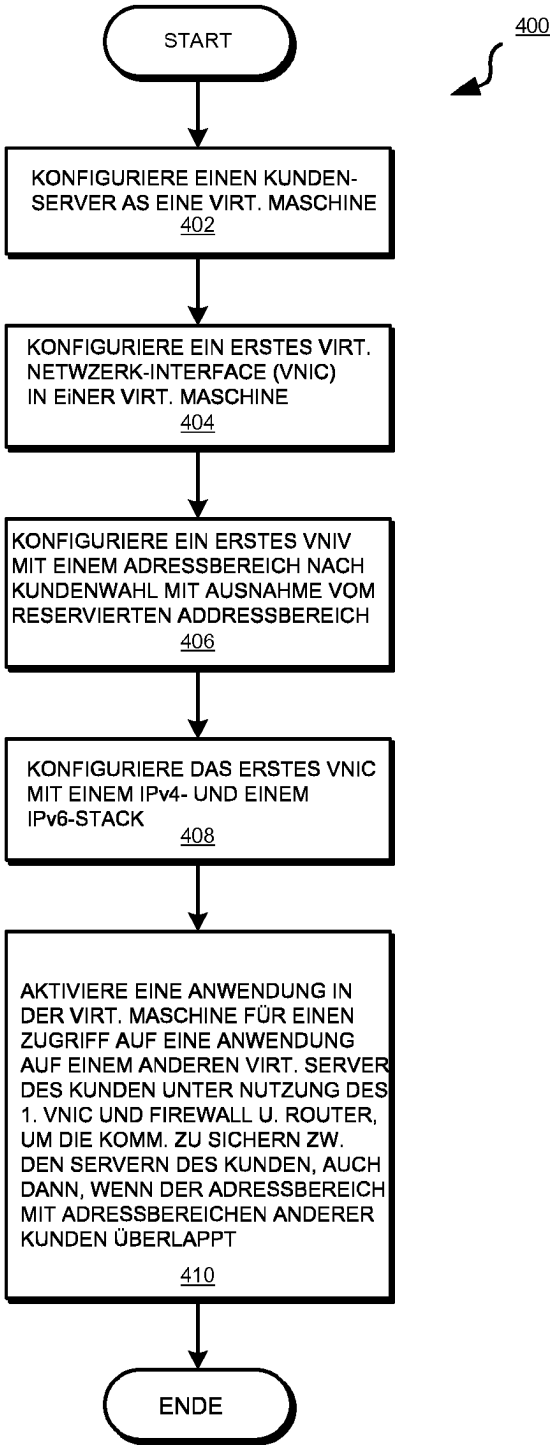

400
START
KONFIGURIERE EINEN KUNDEN-SERVER AS EINE VIRT. MASCHINE
402
KONFIGURIERE EIN ERSTES VIRT. NETWZERK-INTERFACE (VNIC) IN EINER VIRT. MASCHINE
404
KONFIGURIERE EIN ERSTES VNIV MIT EINEM ADRESSBEREICH NACH KUNDENWAHL MIT AUSNAHME VOM RESERVIERTEN ADDRESSBEREICH
406
KONFIGURIERE DAS ERSTES VNIC MIT EINEM IPv4- UND EINEM IPv6-STACK
408
AKTIVIERE EINE ANWENDUNG IN DER VIRT. MASCHINE FÜR EINEN ZUGRIFF AUF EINE ANWENDUNG AUF EINEM ANDEREN VIRT. SERVER DES KUNDEN UNTER NUTZUNG DES 1. VNIC UND FIREWALL U. ROUTER, UM DIE KOMM. ZU SICHERN ZW. DEN SERVERN DES KUNDEN, AUCH DANN, WENN DER ADRESSBEREICH MIT ADRESSBEREICHEN ANDERER KUNDEN ÜBERLAPPT
410
ENDE

FIG. 5

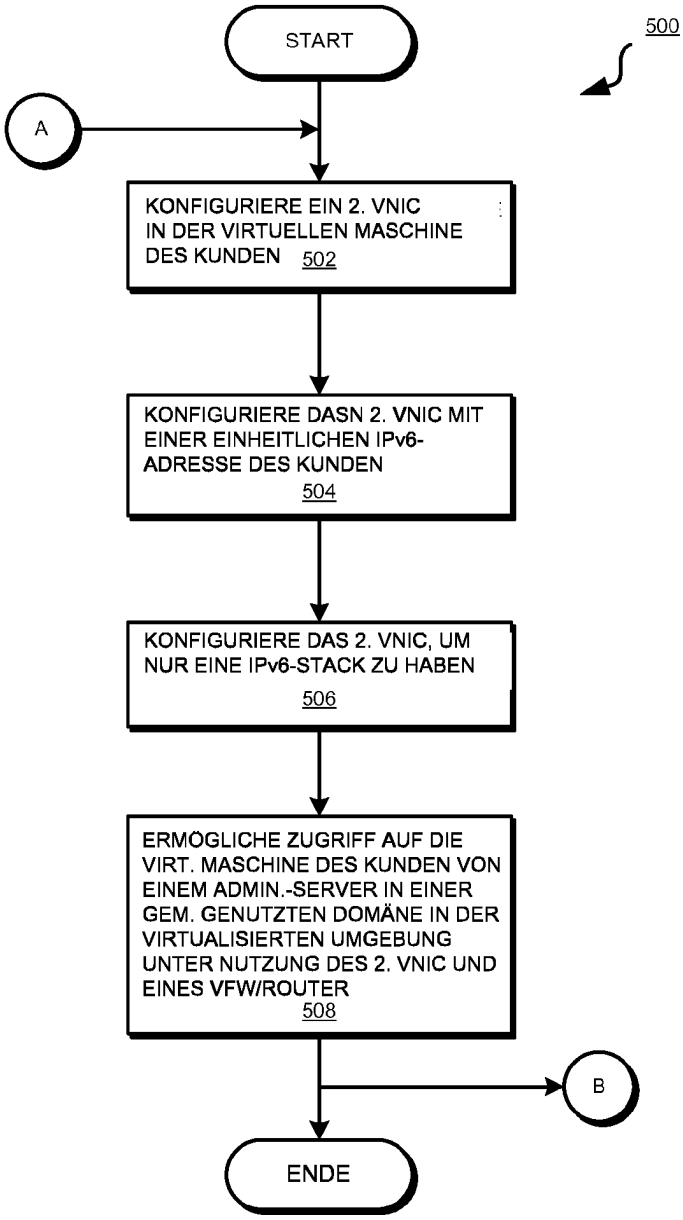
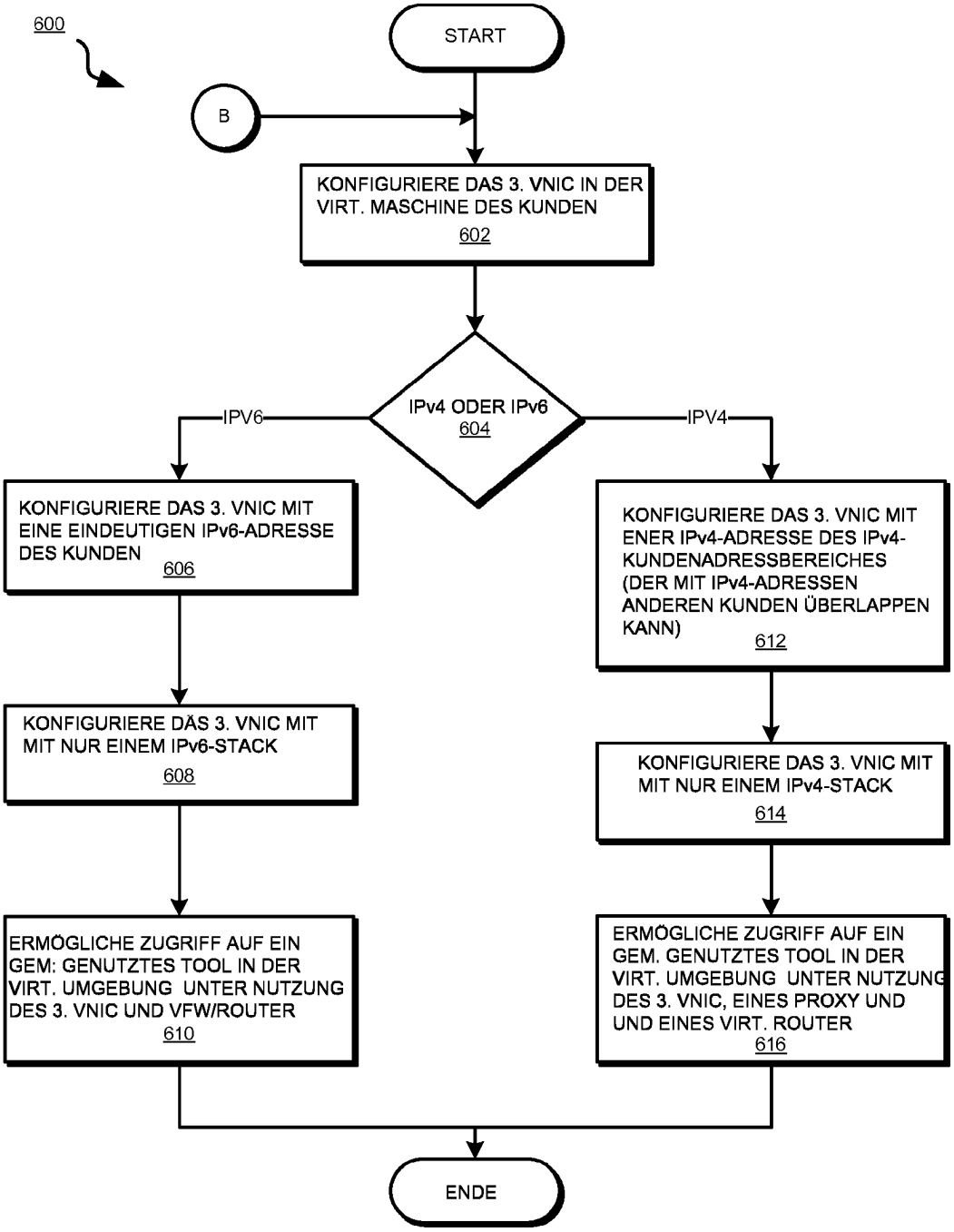

START
500
A
KONFIGURIERE EIN 2. VNIC IN DER VIRTUELLEN MASCHINE DES KUNDEN 502
KONFIGURIERE DASN 2. VNIC MIT EINER EINHEITLICHEN IPv6-ADRESSE DES KUNDEN 504
KONFIGURIERE DAS 2. VNIC, UM NUR EINE IPv6-STACK ZU HABEN 506
ERMÖGLICHE ZUGRIFF AUF DIE VIRT. MASCHINE DES KUNDEN VON EINEM ADMIN.-SERVER IN EINER GEM. GENUTZTEN DOMÄNE IN DER VIRTUALISIERTEN UMGEBUNG UNTER NUTZUNG DES 2. VNIC UND EINES VFW/ROUTER 508
B
ENDE

FIG. 6


600
START
B
KONFIGURIERE DAS 3. VNIC IN DER VIRT. MASCHINE DES KUNDEN
602
IPv4 ODER IPv6
604
IPV6
IPV4
KONFIGURIERE DAS 3. VNIC MIT EINE EINDEUTIGEN IPv6-ADRESSE DES KUNDEN
606
KONFIGURIERE DAS 3. VNIC MIT ENER IPv4-ADRESSE DES IPv4-KUNDENADRESSBEREICHES (DER MIT IPv4-ADRESSEN ANDEREN KUNDEN ÜBERLAPPEN KANN)
612
KONFIGURIERE DÄS 3. VNIC MIT MIT NUR EINEM IPv6-STACK
608
KONFIGURIERE DAS 3. VNIC MIT MIT NUR EINEM IPv4-STACK
614
ERMÖGLICHE ZUGRIFF AUF EIN GEM: GENUTZTES TOOL IN DER VIRT. UMGEBUNG UNTER NUTZUNG DES 3. VNIC UND VFW/ROUTER
610
ERMÖGLICHE ZUGRIFF AUF EIN GEM. GENUTZTES TOOL IN DER VIRT. UMGEBUNG UNTER NUTZUNG DES 3. VNIC, EINES PROXY UND UND EINES VIRT. ROUTER
616
ENDE