

Betriebssysteme

Kapitel 7 Virtualisierung und Cloud

- Einführung
 - Historischer Überblick
 - Betriebssystemkonzepte
- Prozesse und Threads
 - Einführung in das Konzept der Prozesse
 - Prozesskommunikation
 - Scheduling von Prozessen
 - Threads
- Speicherverwaltung
 - Einfache Speicherverwaltung
 - Virtueller Speicher
 - Segmentierter Speicher
- Dateien und Dateisysteme
 - Dateien
 - Verzeichnisse
 - Implementierung von Dateisystemen
- Grundlegende Eigenschaften der I/O-Hardware
 - Festplatten
 - Terminals
 - Die I/O-Software
- Deadlocks/Verklemmungen
- Virtualisierung und die Cloud
- IT-Sicherheit
- Multiprozessor-Systeme

- Überlegen Sie welche Gründe es gibt, daß Virtualisierung sich in der Informatik durchgesetzt hat?

Bedingung:

→ 2 min



2 min

Antwort:

Vorteile Virtualisierung

- Energie einsparen
- Kleinere Rechenzentren
- Nutzung der ,übrigen‘ Server für andere Zwecke (Test,...)
- Schnellere Server Provisionierung
- Reduzierung der Abhängigkeit zu Herstellern
- Erhöhte Verfügbarkeit
- Virtuelles Disaster/Recovery
- Lebenszeit der Alt-Anwendungen lässt sich verlängern
- Ausrichtung in Richtung Cloud

Virtualisierung

Nachteile Virtualisierung

- Viel HW-Leistung
- Lizenzierungsproblematik
- Schnelles Netzwerk notwendig
- Programme, die nicht virtualisiert laufen
- Erhöhtes Ausfallrisiko (es fallen gleich viele Server aus)
- Anpassung des Toolings

- ‚Geschichte‘ der Virtualisierung
 - 1969: CP-67 ("control program") für IBM System/360-67 (1969!) CP/CMS wurde mit Einplatzsystem CMS für Teilnehmerbetrieb eingesetzt und später VM/370 genannt (Mainframe S/390, zSeries)
 - 1972: z/VM wird Nachfolger von VM/370 für die aktuelle Großrechner-Familie IBM zSeries
 - 1999: VMware WSX für Intel x86
 - 2003: Xen realisiert Paravirtualisierung für x86
 - ...

Virtualisierung

- Definition Virtualisierung (Wikipedia)

Virtualisierung bezeichnet in der [Informatik](#) die Nachbildung eines Hard- oder Software-„Objekts“ durch ein ähnliches Objekt vom selben Typ mit Hilfe einer Software-Schicht. Dadurch lassen sich virtuelle (d. h. nicht-physische) Dinge wie [emulierte Hardware](#), [Betriebssysteme](#), [Datenspeicher](#) oder [Netzwerkressourcen](#) erzeugen. Dies erlaubt es etwa, Computer-Ressourcen (insbesondere im [Server](#)-Bereich) transparent zusammenzufassen oder aufzuteilen, oder ein [Betriebssystem](#) innerhalb eines anderen auszuführen.

- Definition Virtualisierung

(<https://www.ionos.de/digitalguide/server/konfiguration/virtualisierung/>)

Bei Virtualisierung handelt es sich um die Abstraktion physischer IT-Ressourcen wie Hardware, Software, Speicher und Netzwerkkomponenten. Ziel ist es, diese Ressourcen auf virtueller Ebene bereitzustellen und ebenso flexibel wie bedarfsgerecht an verschiedene Abnehmer zu verteilen. Dies soll für eine verbesserte Auslastung der IT-Ressourcen sorgen.

- **Definition Virtualisierung**

(<https://www.computerweekly.com/de/definition/Virtualisierung>)

Virtualisierung erstellt eine virtuelle anstelle einer tatsächlichen Version von etwas, zum Beispiel ein Betriebssystem, einen Server, ein Speichergerät oder eine Netzwerkressource. Sie verwendet dafür Software, die eine Hardwarefunktionalität simuliert, um ein virtuelles System zu erstellen.

- So funktioniert Virtualisierung

- Technologie, die eine Anwendung, ein Gastbetriebssystem (Guest) oder einen Datenspeicher von der tatsächlich zugrunde liegenden Hardware oder Software abstrahiert.
- Bei der Servervirtualisierung emuliert eine Softwareschicht – Hypervisor genannt – die zugrunde liegende Hardware emuliert (Arbeitsspeicher, die Ein- und Ausgabe und den Netzwerkverkehr der CPU.
- Hypervisoren nehmen die physischen Ressourcen und virtualisieren sie.
- Sie können auf einem Betriebssystem sitzen oder direkt auf der Hardware installiert werden.
- Mit Hilfe des Hypervisors führt das Gastbetriebssystem, das normalerweise direkt mit der Hardware interagiert, seine Operationen jetzt mit einer Softwareemulation dieser Hardware durch. Auf der Ebene des Gastbetriebssystems ist nicht sichtbar, ob es sich um physische oder virtuelle Hardware handelt.
- Die Leistung dieses virtuellen Systems entspricht nicht der Leistung des Betriebssystems, das auf echter Hardware ausgeführt wird.
- Das Konzept der Virtualisierung funktioniert gut, da die meisten Gastbetriebssysteme und Anwendungen die zugrundeliegende Hardware nicht vollständig benötigen.

Virtualisierung

- Hardwarevirtualisierung
 - Der Begriff Hardware-Virtualisierung verweist auf Technologien, die es ermöglichen, Hardware-Komponenten mittels Software unabhängig von ihrer physischen Grundlage bereitzustellen. Das Paradebeispiel für eine Virtualisierung von Hardware ist die virtuelle Maschine (kurz VM)
 - Systemvirtualisierung (Computersystemvirtualisierung oder Betriebssystemvirtualisierung)
 - Computervirtualisierung: mithilfe von Software wird ein virtuelles Computersystem erstellt
 - Betriebssystemvirtualisierung erstellt virtuelle Instanzen einer Betriebsumgebung
 - Prozessorvirtualisierung
 - Speichervirtualisierung

Quelle: [http://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](http://de.wikipedia.org/wiki/Virtualisierung_(Informatik))

- Systemvirtualisierung (Computersystemvirtualisierung oder Betriebssystemvirtualisierung)
 - Dieses Verfahren bedient sich der Partitionierung des vorhandenen Betriebssystems und nicht der vorhandenen Hardware.
 - Bei der Erstellung der Gäste werden nur die Individualdaten für die virtuelle Maschine (VM) angelegt.
 - Alle **"gleichen" Betriebssystemdaten** (etwa Betriebssystem-Bibliotheken) des Host-Systems werden demnach von den Gästen mitgenutzt. Das gilt auch für die laufenden Prozesse. Abweichende Daten werden im Heimatverzeichnis der VM abgelegt.
 - Eine VM in der Grundausstattung besteht nur aus sehr wenigen Dateien und verbraucht deutlich weniger Festplatten- und Hauptspeicher als ein normal installierter Server.
 - Betriebssystem-Virtualisierung steht für sehr hohe Ausnutzung der Systemressourcen beziehungsweise **geringen Virtualisierungsschwund** (ein bis drei Prozent).
 - Aufgrund der geringen Leistungsansprüche der Gäste ist eine sehr hohe Anzahl VMs (größer Faktor zehn im Vergleich zur Komplettvirtualisierung) bei verhältnismäßig bescheidender Hardwareausstattung möglich.
 - Größter Nachteil ist die Abhängigkeit des Gastsystems vom Host, was eine homogene Basis an Betriebssystemen voraussetzt.
 - Wichtig: Alle Gäste müssen getrennt und unabhängig voneinander agieren können.

Quelle: <https://www.tecchannel.de/a/virtualisierungs-grundlagen-varianten-und-unterschiede,2029842,7>

- Speichervirtualisierung (<https://www.datacore.com/de/storage-virtualization/>)
 - Speichervirtualisierung (auch als Software-definierter Speicher oder virtuelles SAN bezeichnet) ist die Konsolidierung mehrerer physischer Speicher-Arrays aus SANs, sodass sie als einheitliches virtuelles Speichersystem erscheinen.
 - Es kann ganz **unterschiedliche Hardware** unterschiedlicher Netzwerke, Anbieter oder Rechenzentren umfassen und diese als logische Einheit abbilden, die über eine einheitliche Bedienfläche verwaltet wird.
 - Bei virtualisierten Speicherlösungen wird die **Speichermanagementsoftware** von der grundlegenden Hardware-Infrastruktur getrennt, um die Flexibilität zu vergrößern und skalierbare Pools von Speichersystemen zu schaffen.
 - Die Technologie kann die Speicherhardware (Arrays und Festplatten) in virtuellen Pools abstrahieren, so wie Computer-Virtualisierungssysteme (VMWare ESX oder Hyper-V) die Computer-Hardware (Server) als virtuelle Maschinen (VMs) abstrahieren.

Virtualisierung

- Prozessorvirtualisierung (<https://powercampus.de/dokumentation/virtualisierung-mit-powervm/praktische-einfuehrung/prozessor-virtualisierung/>)
 - Im einfachsten Fall werden einer LPAR dedizierte Prozessoren zugeordnet.
 - Die LPAR kann die zugeordneten dedizierten Prozessoren dann zu 100% nutzen und muß **nicht mit anderen LPARs um Prozessor-Ressourcen konkurrieren.**
 - Meist werden Prozessoren (bzw. Prozessor-Cores) zwischen LPARs geteilt, um eine bessere Auslastung der Prozessoren zu erzielen. Man spricht dann von sogenannten Shared Prozessoren, im Unterschied zu den dedizierten Prozessoren. Die Aufteilung eines Shared Prozessors in Anteile, welche dann den LPARs zugeordnet werden, wird als Micro-Partitioning bezeichnet.
 - Beispiel: Der POWER Hypervisor teilt die Shared Prozessoren über ein Zeitscheiben-Verfahren auf die zugehörigen LPARs auf.

- Unterstützende Hardwaretechnologien
 - Blades
 - Blades nutzen die gleichen Ressourcen, werden zentral verwaltet und haben eine gemeinsame Stromversorgung und Lüftung. So kann beispielsweise ein Standard-42U-19"-Rack bis zu 84 Blades aufnehmen und bis zu 1344 Prozessorkerne enthalten.
 - Server muss keine Lüfter in den Blades haben, sondern die Lüfter sind redundant an der Rückseite des Gehäuses (Blade-Chassis) eingebaut
 - Beispiel: Bladesystem (HP), BladeCenter (Lenovo), Cisco UCS

Bladecenter Rückansicht

- Unterstützende Hardwaretechnologien

BladeCenter (Beispiel)

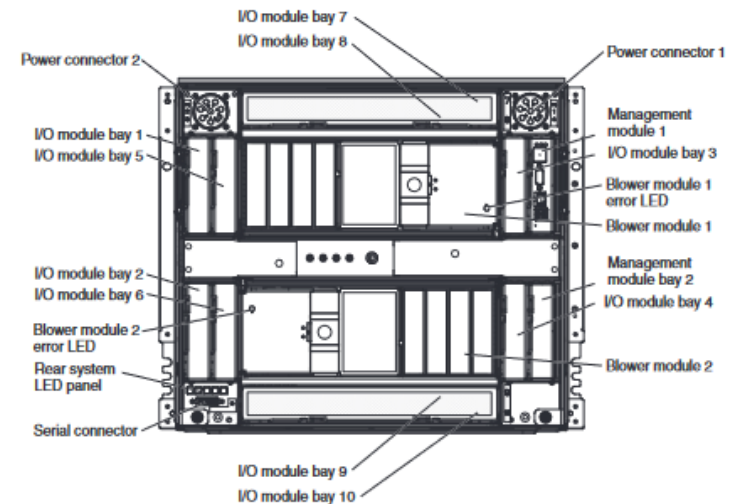
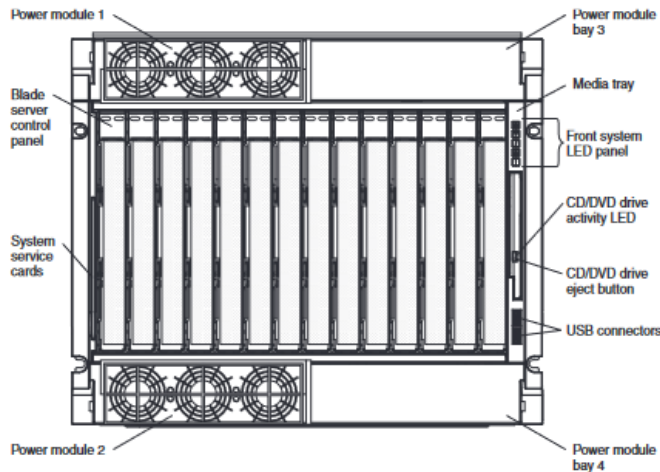
Bladecenter Vorderansicht



Bladecenter Rückansicht



Schlüsselkomponenten
Blade-Chassis



- Unterstützende Hardwaretechnologien

BladeCenter und Blade-Einschübe (Beispiel)

Bladecenter Vorderansicht



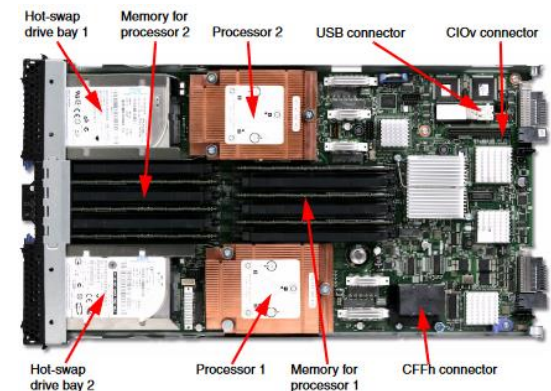
Bladecenter Rückansicht



Blades

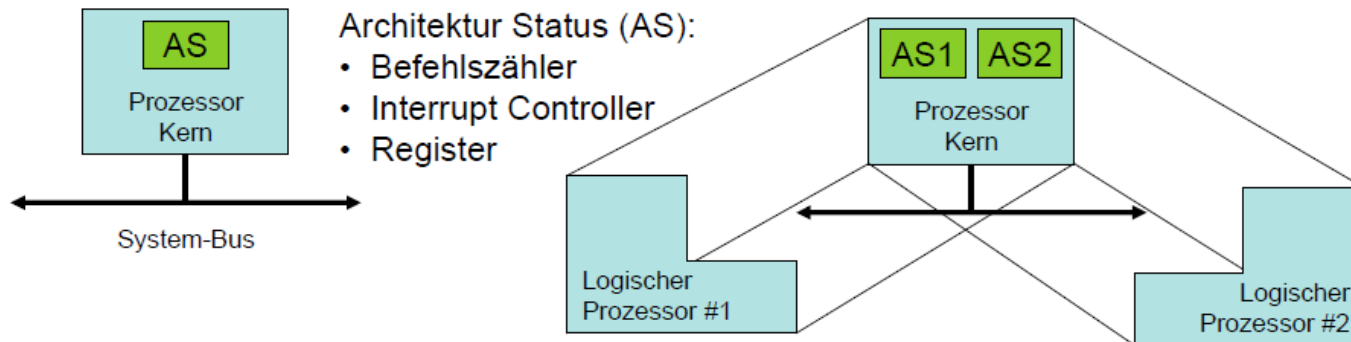


- ▶ Supports up to two Intel Xeon processor E5-2600 product family CPUs
- ▶ Sixteen DIMM slots for up to 512 GB of memory (using 32 GB DIMMs)
- ▶ Up to 1866 MHz memory speeds
- ▶ Supports up to two hot-swap HDDs
- ▶ Internal USB port for embedded hypervisor support
- ▶ Integrated dual-port Gigabit Ethernet and dual-port 10 Gb Ethernet Virtual Fabric
- ▶ Up to 18 I/O ports per blade with virtual NICs
- ▶ PCI Express connectors for attaching various I/O expansion cards
- ▶ Interposer Card connector



- Unterstützende Hardwaretechnologien: Hyperthreading (1/2)
 - Hyperthreading
 - Eine von Intel erfundene Technologie, mit der sich ein physischer Mikroprozessor wie **zwei logische, virtuelle Kerne** verhält.
 - Hyperthreading wurde von AMD übernommen und heisst bei AMD **Simultaneous Multi-Threading (SMT)**.
 - Einem Prozessor ist es somit möglich, in einem Zustand zwei verschiedene Threads zu verwalten und innerhalb von Nanosekunden zwischen diesen Threads hin- und herzuschalten.
 - Das Hyperthread-Verfahren **erhöht die CPU-Leistung** und sorgt für eine bessere Auslastung des Rechners.
 - Beispiel: Einer der Prozesse muss ein Datenwort aus dem Speicher lesen; während der Wartezeit kann eine Multithread-CPU zu einem anderen Prozess umschalten.
 - Softwareseitig verhält sich eine **CPU mit Hyper-Threading wie ein Symmetrisches Multiprozessorsystem**
 - Hyper-Threading bringt jedoch nur für **Anwendungen** einen Geschwindigkeitsvorteil, deren Berechnungen **parallelisierbar** sind, das heißt die Berechnung eines Threads ist nicht abhängig vom Ergebnis eines anderen.
 - Multithreading ist **keine echte Parallelität**; es läuft jeweils nur ein Prozess; nur das Umschalten der Threads reduziert sich auf Nanosekunden.

- Unterstützende Hardwaretechnologien: Hyperthreading (2/2)
 - Durchschnittliche Auslastung einer Execution-Unit: ca. 35 %*
 - Hyper-Threading Technologie



- Leistungssteigerung durch Hyper-Threading ca. 30 %*
- Grenzen bestehen durch sharing von Prozessor-Ressourcen; z.B. L1-, L2-Cache, Execution Pipeline

*Quelle: Intel

Virtualisierung

- Softwarevirtualisierung
 - Hardware-Emulation
 - Betriebssystemvirtualisierung mittels Betriebssystem-Container
 - Systemvirtualisierung mittels Virtual Machine Monitor (VMM)
 - Hardware-Virtualisierung (native Virtualisierung, full Virtualisierung)
 - Paravirtualisierung

- Anwendungsvirtualisierung

Quelle: [http://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](http://de.wikipedia.org/wiki/Virtualisierung_(Informatik))

Wir sprechen in der IT immer wieder von Simulation, Emulation und Virtualisierung.

Beschreiben Sie die Begriffe in Ihren eigenen Worten.

- Was ist eine Simulation?
- Was ist eine Emulation?
- Was ist eine Virtualisierung?
- Geben Sie Beispiele!

Bedingung:

→ 2 min



2min

Antwort:

Definition Simulation (Wikipedia)

Die **Simulation** oder **Simulierung** ist eine Vorgehensweise zur Analyse von Systemen, die für die theoretische oder formelmäßige Behandlung zu komplex sind. Dies ist überwiegend bei *dynamischem* Systemverhalten gegeben.

Bei der Simulation werden Experimente an einem Modell durchgeführt, um Erkenntnisse über das reale System zu gewinnen. Im Zusammenhang mit Simulation spricht man von dem zu simulierenden System und von einem **Simulator** als Implementierung oder *Realisierung* eines Simulationsmodells. Letzteres stellt eine Abstraktion des zu simulierenden Systems dar (Struktur, Funktion, Verhalten). Der Ablauf des Simulators mit konkreten Werten (Parametrierung) wird als Simulationsexperiment bezeichnet. Dessen Ergebnisse können dann interpretiert und auf das zu simulierende System übertragen werden.

Definition Simulation

(<https://www.twi-global.com/locations/deutschland/was-wir-tun/haeufig-gestellte-fragen/was-ist-simulation>)

Eine Simulation imitiert die Funktionsweise realer Prozesse oder Systeme mit Hilfe von Modellen. Das Modell stellt die wichtigsten Verhaltensweisen und Merkmale des ausgewählten Prozesses oder Systems dar, während die Simulation zeigt, wie sich das Modell unter verschiedenen Bedingungen im Laufe der Zeit entwickelt.

Simulationen sind in der Regel computergestützt und verwenden ein softwaregeneriertes Modell, um die Entscheidungen von Managern und Ingenieuren zu unterstützen

Simulationstechniken erleichtern das Verständnis und das Experimentieren, da die Modelle sowohl visuell als auch interaktiv sind.

Zu den Simulationssystemen gehören die diskrete Ereignissimulation, die Prozesssimulation und die dynamische Simulation. Unternehmen können alle diese Systeme auf verschiedenen Ebenen der Organisation einsetzen.

Antwort:

Definition Emulation (Wikipedia)

Als **Emulator** (von lat. *aemulari*, „nachahmen“) wird in der Computertechnik ein System bezeichnet, das ein anderes in bestimmten Teilaspekten nachbildet.

Das nachgebildete System erhält die gleichen Daten, führt vergleichbare Programme aus und erzielt die möglichst gleichen Ergebnisse in Bezug auf bestimmte Fragestellungen wie das zu emulierende System.

Definition Emulation

(<https://it-talents.de/it-wissen/virtualisierungstechnologien-hardware-emulation/>)

Emulation bedeutet das Übersetzen von Befehlen einer Gastmaschine in die Befehle einer Hostmaschine, die auf dieser dann ausgeführt werden.

- Virtualisierungstechnik: Emulation

- **Hardware-Emulator:** Alle Operationen, auch diejenigen der CPU, werden emuliert. Der Emulator nimmt Instruktionen in einem bestimmten Befehlssatz und stellt diese Instruktionen auf einer anderen Maschine dar. Das ist mit Rechenaufwand verbunden und mindert somit die verfügbare Rechenleistung für die eigentliche Anwendung. Häufig benutzt, wenn ein Programm für eine Maschine ausgeführt werden soll, die aber nicht zur Verfügung steht. Das kann auch ein altes Gerät sein, das gar nicht mehr erhältlich ist.

Ein Hardware-Emulator ist ein elektronisches Gerät, das ein System wie einen Drucker oder einen Prozessor (CPU) funktionell, elektrisch oder mechanisch (Gehäuse und Pins) nachbilden kann. Die Verbindung zur Prozessorbaugruppe wird mittels Sockel und passendem Stecker erstellt. Er wird auch als In-Circuit-Emulator (ICE) bezeichnet.

- Virtualisierungstechnik: Emulation

- **Terminalemulator***: Systemprogramm, das einem „fremden“ Rechner oder Terminal erlaubt, die Ressourcen eines bestimmten Rechners zu nutzen, indem das Kommunikationsverhalten eines anschließbaren Terminals nachgebildet wird.
- **Software-Emulator**: Programme, die einen Computer oder ein Betriebssystem nachbilden und es so ermöglichen, Software für diesen Computer auf einem anderen Computer mit einer anderen Architektur zu verwenden oder zu erstellen.

Besipiele:

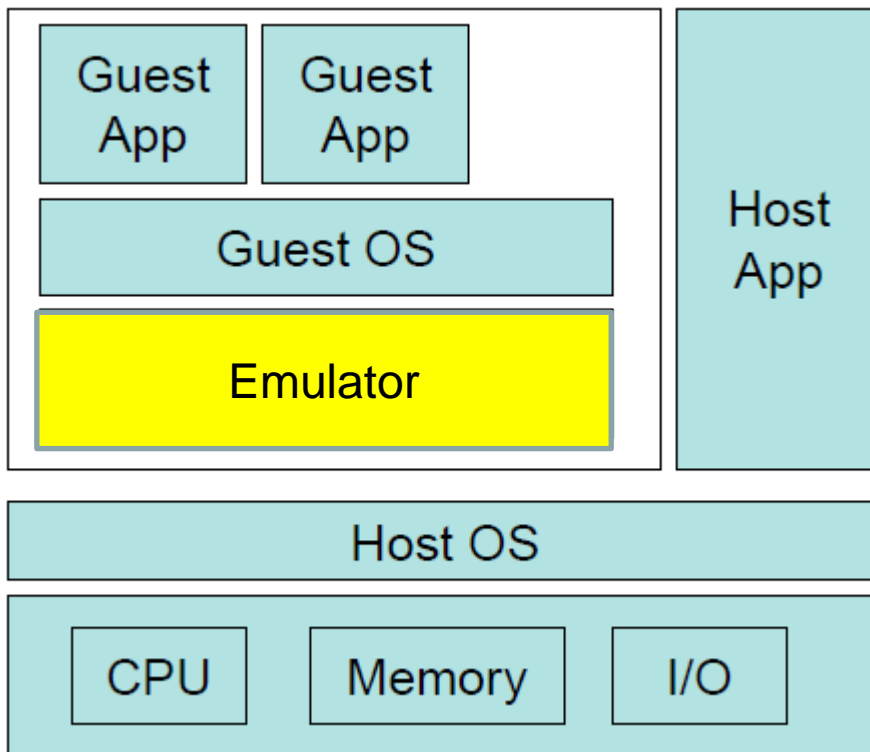
- Spiele für ältere Spielekonsolen können auf einem PC oder einer neueren Spielekonsole ablaufen.
- Bei Entwicklung eines Programmes für ein Gerät (z. B. ein Handy), das eine andere Architektur als der Entwicklungs-Rechner hat, kann dieses im Emulator getestet und korrigiert werden, ohne es jedes Mal auf das Gerät kopieren zu müssen.
Vielleicht ist das Gerät auch gar nicht vorhanden.

—

- Virtualisierungstechnik: Emulation
 - Bei der Emulation wird in den meisten Fällen versucht, die komplette Hardware eines Rechensystems funktionelle nachzubilden und so einem unveränderten Betriebssystem, das für eine andere HW-Architektur (CPU) ausgelegt ist, den Betrieb zu ermöglichen
 - Vorteile:
 - Keine Anpassungen am Betriebssystem bzw. den Anwendungen notwendig
 - Programme können auch für ganz andere CPUs als die tatsächlich zur Verfügung stehende ausgeführt werden
 - Nachteile:
 - Entwicklung von Emulationsumgebungen sehr aufwändig
 - Ausführungsgeschwindigkeit in der Regel deutlich geringer gegenüber Virtualisierungslösungen
 - Bei Hardware Virtualisierung statt Hardware Emulation wird die CPU selbst nicht emuliert. Die Instruktionen werden nur weitergereicht
→ Bei HW-Virtualisierung werden nur Systeme mit der gleichen CPU-Architektur dargestellt werden können.

Virtualisierung

- Virtualisierungstechnik: Emulation



Idee

Architektur wird unabhängig von tatsächlicher Hardware vollständig in Software abgebildet

Vorteil

- Unabhängigkeit

Nachteile

- aufwändige Entwicklung
- Performanceverluste
- Lizenzrechte

- **Virtualisierungstechnik: Auswahl von Emulatoren**
(kein Anspruch auf Vollständigkeit)

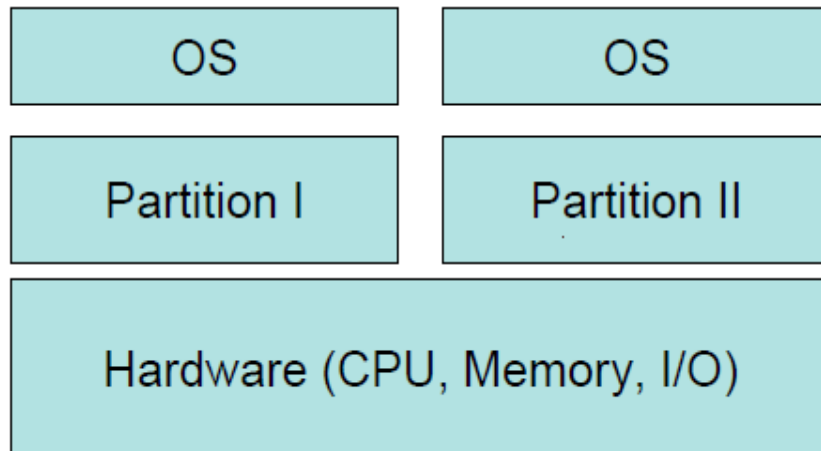
Name	Lizenz	Host	Emulierte Architektur	Gast-System
Bochs v2.3.6	LGPL	Linux, Solaris, MacOS, Windows, IRIX, BeOS	x86, AMD64	Linux, DOS, BSD, Windows, BeOS
QEMU v0.9.0	GPL	Linux, BSD, Solaris, BeOS, MacOS-X	x86, AMD64, PowerPC, ARM, MIPS, Sparc	Linux, MacOS-X, Windows, BSD
DOSBox v0.72	GPL	Linux, Windows, OS/2, BSD, BeOS, MacOS-X	x86	DOS
DOSEMU v1.4.0	GPL	Linux	x86	DOS, Windows bis 3.11
PearPC v0.4.0	GPL	Linux, MacOS-X, Windows	PowerPC	Linux, MacOS-X, BSD
Basilisk II v0.9-1	GPL	Linux, diverse UNIX, Windows NT4, BeOS, Mac OS, Amiga OS	680x0	MacOS \leq 8.1
Wabi v2.2	proprietär	Linux, Solaris	x86	Windows 3.x
MS Virtual PC v7	proprietär	MacOS-X	x86	Windows, (Linux)
M.A.M.E. v0.137	MAME-Lizenz	Linux, Windows, DOS, BeOS, BSD, OS/2	diverse Arcade	diverse Arcade
SheepShaver	GPL	Linux, MacOS-X, BSD, Windows, BeOS	PowerPC, 680x0	MacOS 7.5.2 bis MacOS 9.0.4
Hercules 3.07	QPL	Linux, MacOS-X, BSD, Solaris, Windows	IBM-Großrechner	IBM System/360, 370, 390

GPL = GNU **General Public License**

- Virtualisierungstechnik: Auswahl von Emulatoren (kein Anspruch auf Vollständigkeit)
 - Beispiel QEMU (engl. „Quick Emulator“)
 - QEMU ist **eine freie virtuelle Maschine**, die die gesamte Hardware eines Computers emuliert und durch die dynamische Übersetzung der Prozessorinstruktionen des Gastprozessors in Instruktionen für den Host-Prozessor eine sehr gute Ausführungsgeschwindigkeit erreicht.
 - QEMU emuliert Systeme mit den folgenden Prozessorarchitekturen: x86, AMD64 und x86-64, PowerPC, ARM (32 + 64 Bit), Alpha, CRIS, LatticeMico32, m68k (Coldfire), MicroBlaze, MIPS, Moxie, SH-4, S/390, Sparc32/64, TriCore, OpenRISC, Unicore und Xtensa (Stand 2015).
 - QEMU ist auf den Betriebssystemen GNU/Linux, Windows, FreeBSD, NetBSD, OpenBSD, OpenSolaris, OS/2/eComStation, DOS, Mac OS X und Haiku lauffähig. Es kann den gesamten Status einer virtuellen Maschine so speichern, dass diese auf ein anderes Host-System übertragen werden kann und dort weiterlaufen kann (Live-Migration).

- **Virtualisierungstechnik: Auswahl von Emulatoren** (kein Anspruch auf Vollständigkeit)
 - Beispiel KVM (engl. Kernel-Based Virtual Machine)
 - Die **Kernel-based Virtual Machine (KVM)** ist eine Infrastruktur des Linux-Kernels zur Virtualisierung, die auf den Hardware-Virtualisierungstechniken von Intel (VT) oder AMD (AMD-V) ausgestatteten x86-Prozessoren sowie auf der System-z-Architektur lauffähig ist.
 - KVM wurde im Oktober 2006 veröffentlicht und ist ab Version 2.6.20 des Linux-Kernels in diesem enthalten. Es wurde unter der Federführung von Avi Kivity bei dem israelischen Unternehmen Qumranet entwickelt. Qumranet wurde im September 2008 von Red Hat gekauft.
 - KVM wurde zunächst für die x86-Plattform entwickelt und besteht für diese aus dem Kernel-Modul `kvm.ko` sowie aus den hardwarespezifischen Modulen `kvm-intel.ko` (für Intel-Prozessoren) oder `kvm-amd.ko` (für AMD-Prozessoren). Inzwischen gibt es KVM auch für weitere Plattformen wie PowerPC[5] und ARM.
 - KVM selbst nimmt keine Emulation vor, sondern stellt nur die Infrastruktur dazu bereit; QEMU ist derzeit die einzige Möglichkeit, diese zu nutzen. Dazu stellt QEMU für virtualisierte Gastsysteme die notwendigen Geräte wie Festplatten, Netzwerk-, Sound- und Grafikkarten zur Verfügung.
 - Nach dem Laden des Moduls arbeitet der Linux-Kernel selbst als Hypervisor für virtuelle Maschinen. Als Gastsysteme unterstützt KVM Linux (32 und 64 Bit), Windows (32 und 64 Bit), Haiku, AROS, ReactOS, FreeDOS, Solaris und diverse BSD-Derivate. KVM läuft auch auf SMP-Hostsystemen, SMP-Gastsysteme sind ebenfalls möglich.

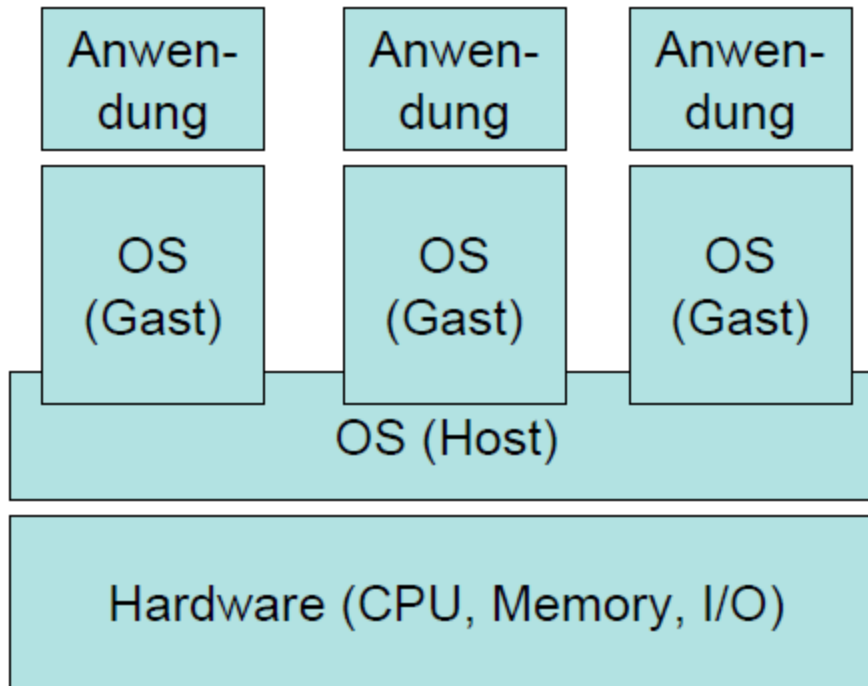
- Ansätze der Servervirtualisierung: Partitionierung



- In jeder Partition kann eine eigene Betriebssysteminstanz betrieben werden
- Aufteilung kann nur entlang von Modulgrenzen durchgeführt werden
- eine VM enthält mindestens eine CPU und ein Memory-Modul
- gute physikalische Trennung der einzelnen Systeme untereinander
- keine Performanceverluste z.B. durch Hypervisor
- keine Anpassung des Betriebssystems notwendig

Virtualisierung

- Betriebssystemvirtualisierung: Container



- Es können nur gleiche Betriebssysteme virtualisiert werden
- einige Teile des Betriebssystems werden gemeinsam genutzt
- Treiber werden vom Host-Kernel verwaltet
- Systemaufrufe werden abgefangen und vom Host ausgeführt.
- geringer Overhead (Virtualisierungsaufwand 1-3%)
- Einsatz: z.T. Internet-Hosting

Virtualisierung

- Betriebssystemvirtualisierung: Containerimplementierungen

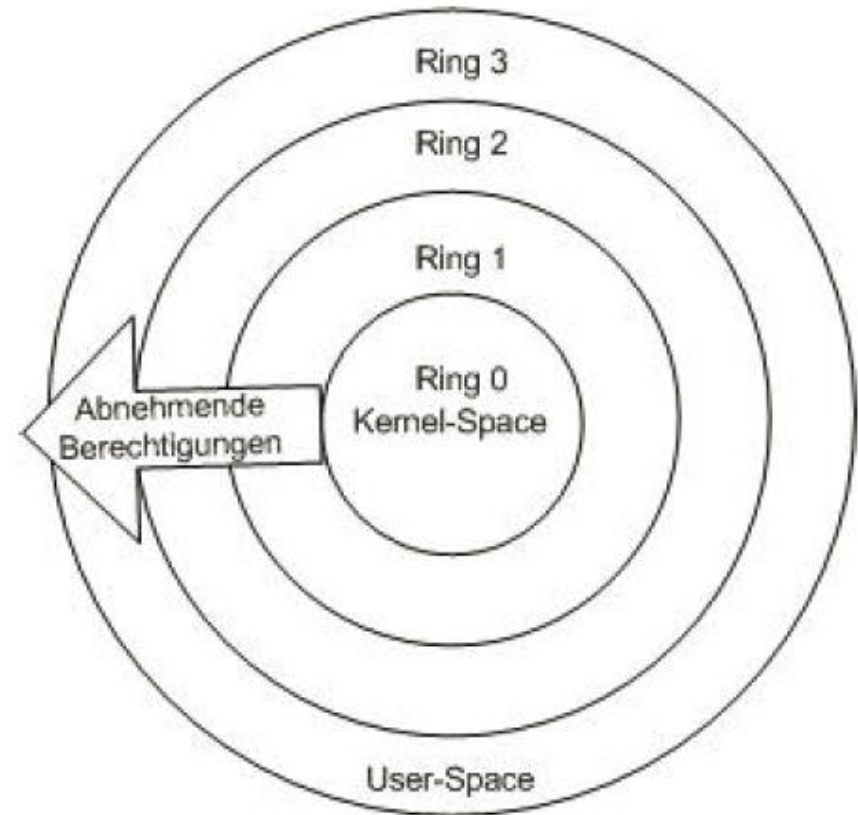
- OpenVZ
- Parallels Virtuozzo Containers
- FreeBSD jails
- Linux-Vserver
- Solaris 10 Containers/Zones
- IBM AIX6 WPARs (Workload Pa



Virtualisierung

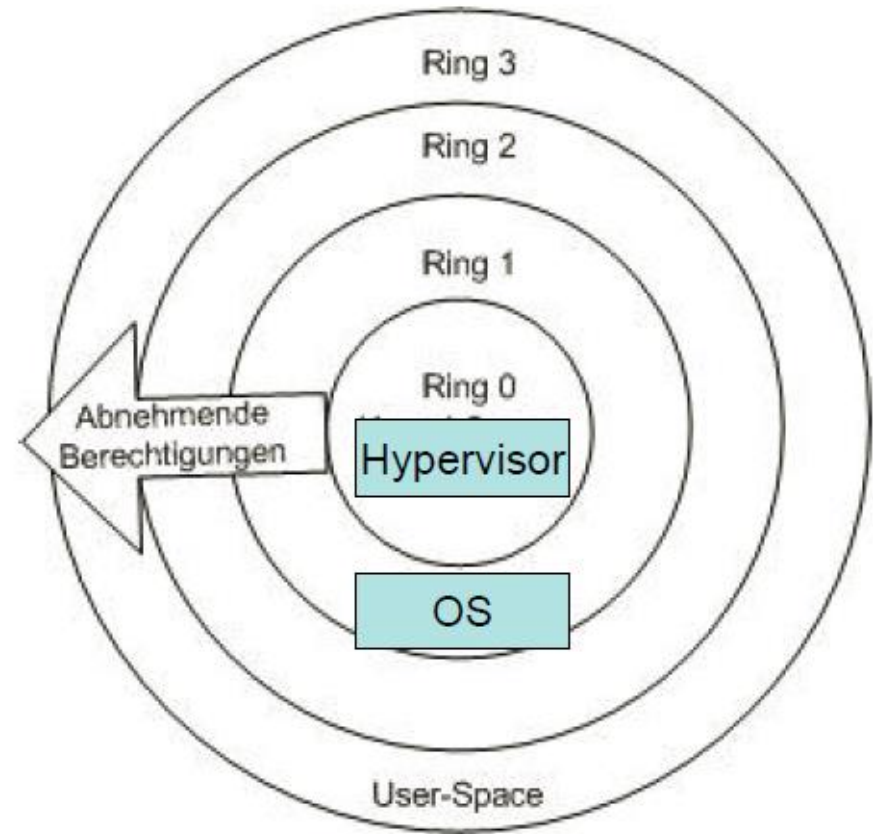
- Hierarchisches Privilegiensystem der x86-Architektur (1/2)

- Regelung des Zugriffs auf Speicher und Befehlssatz des Prozessors
- Ring 0: direkter Zugriff auf Hardware und Speicherbereiche; Kernel des Betriebssystems
- Ring 1 und 2: diese werden normalerweise nicht genutzt
- Ring 3: Benutzerapplikation



- Hierarchisches Privilegiensystem der x86-Architektur (2/2)

- x86 Betriebssysteme sind gebaut um direkt auf der Hardware zu laufen, daher gehen sie davon aus, die Hardware vollständig und exklusiv zu besitzen.
- Hypervisor soll die Kontrolle über die Hardware übernehmen.
- Betriebssystem wird in einen äußeren Ring verschoben



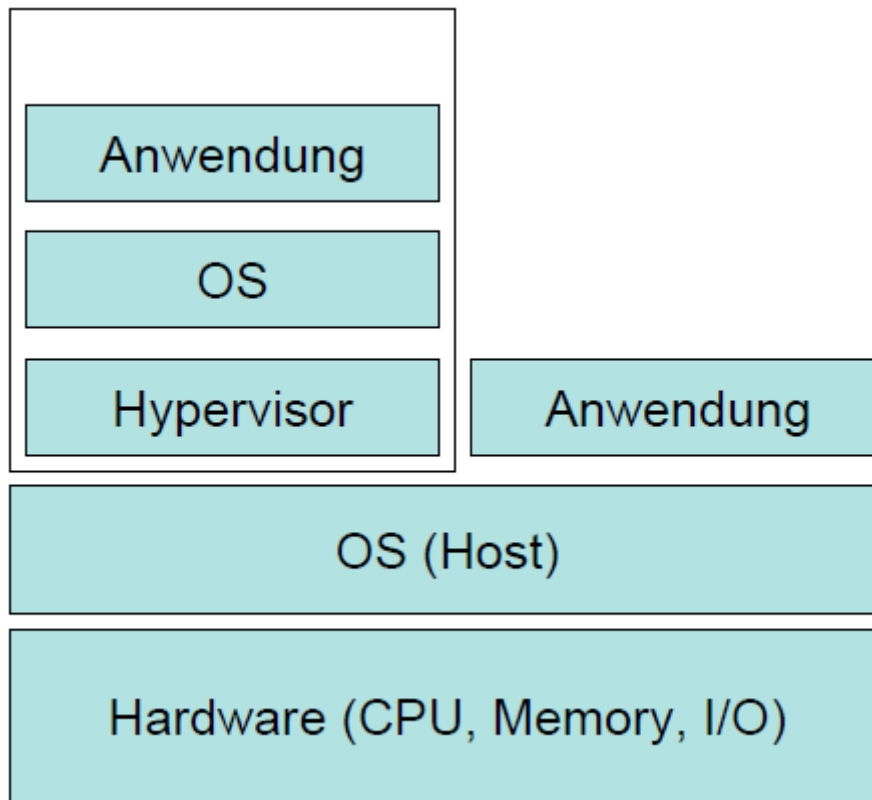
- Ansätze der Servervirtualisierung: Überblick Hypervisor

- Hypervisor

- Ein Hypervisor ist eine Software, die **virtuelle Maschinen (VMs)** erstellt und ausführt.
 - Er wird auch **VMM (Virtual Machine Monitor)** genannt
 - Der Hypervisor isoliert das Betriebssystem und die Ressourcen von den virtuellen Maschinen und ermöglicht die Erstellung und Verwaltung dieser VMs.
 - Physische Hardware wird, wenn sie als Hypervisor verwendet wird, **Host** genannt; die vielen VMs, die seine Ressourcen nutzen, sind **Guests**.
 - Die Ressourcen (wie CPU, RAM und Storage) werden vom Hypervisor als einen gemeinsamen Pool von Ressourcen behandelt. Sie können zwischen vorhandenen Guests oder neuen **VMs problemlos neu zugewiesen** werden.
 - Zur Ausführung der VMs werden einige Komponenten auf Betriebssystemebene benötigt: z.B. einen Speichermanager, Prozessplaner, Input/Output-Stack (I/O), Gerätetreiber, Sicherheitsmanager, Netzwerk-Stack und mehr.
 - Er stellt allen VMs die zugewiesenen Ressourcen zur Verfügung und verwaltet deren Zeitplanung in Abhängigkeit von den physischen Ressourcen.
 - Die physische Hardware übernimmt weiterhin die Ausführung, das heißt die CPU führt immer noch Anweisungen der VMs aus, während der Hypervisor die zeitliche Planung übernimmt.
 - Mithilfe eines Hypervisors können mehrere unterschiedliche Betriebssysteme parallel ausgeführt werden und die gleichen virtualisierten Hardware-Ressourcen nutzen. Dies ist einer der wichtigsten Vorteile der Virtualisierung. Ohne Virtualisierung könnten Sie auf der Hardware lediglich ein Betriebssystem ausführen.

Virtualisierung

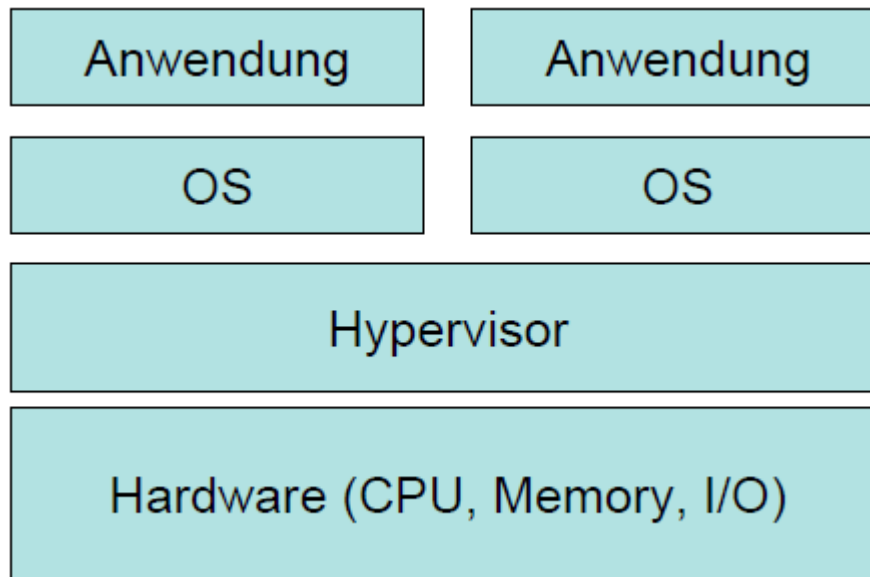
- Ansätze der Servervirtualisierung: Hosted Hypervisor (Type 2)



- **Typ 2-Hypervisor** laufen als Applikation aufgesetzt auf einem Betriebssystem (→ Software-virtualisierung)
- Virtuelle Maschine simuliert realen Rechner mit allen Komponenten
- Basiert auf herkömmlichen Betriebssystemen
- Vorteile
 - Keine Änderung an Betriebssystemen notwendig
 - Flexibilität
- Nachteile
 - „Schlechtere“ Performance

Virtualisierung

- Ansätze der Servervirtualisierung: Vollständige Virtualisierung (Type 1)



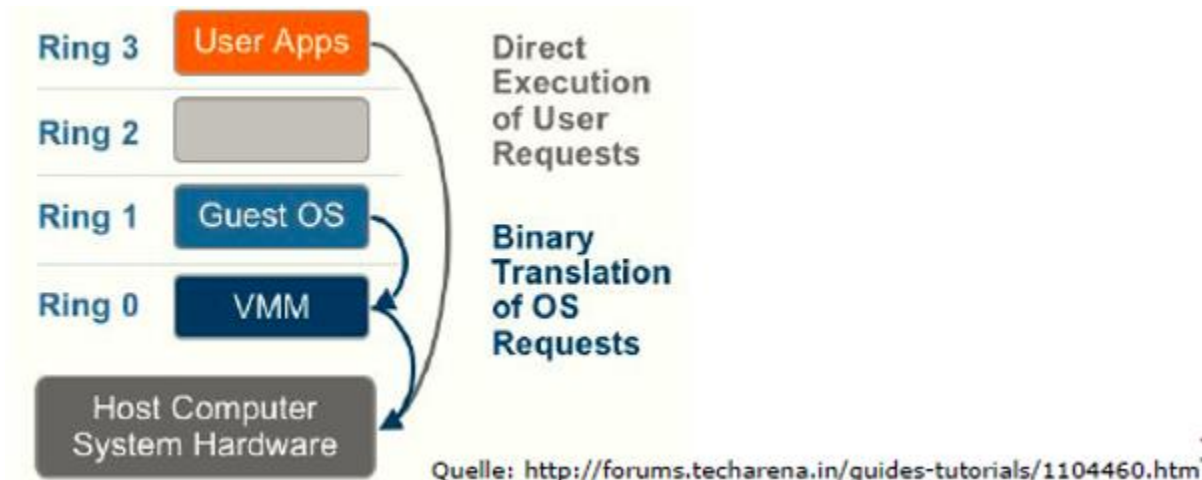
- **Typ-1-Hypervisor** laufen direkt auf der Hardware (HW-Virtualisierung)
- Virtuelle Maschine simuliert realen Rechner mit allen Komponenten
- Virtueller Maschinen-Monitor (VMM) koordiniert virtuelle Maschinen und Ressourcen
- Vorteile
 - Kaum Änderungen an Betriebssystemen
 - Flexibilität
- Nachteile
 - Hypervisor/VMM teuer
 - „Schlechtere“ Performance


- Ansätze der Servervirtualisierung: Zusammenfassung

- Zwei Typen von Hypervisoren
 - Typ 1:
 - Ein Hypervisor vom Typ 1 ist ein Bare-Metal-Hypervisor.
 - VM-Ressourcen werden vom Hypervisor direkt für die Hardware eingeplant.
 - Beispiele: KVM (KVM wurde 2007 in den Linux-Kernel integriert), Microsoft Hyper-V und VMware vSphere
 - Typ 2:
 - Ein Hypervisor vom Typ 2 ist ein gehosteter Hypervisor.
 - VM-Ressourcen werden beim Host-Betriebssystem angefordert und dann über die Hardware ausgeführt.
 - Beispiele: VMware Workstation und Oracle VirtualBox
- Erklärungen:
 - Hyper-V ist eine Hypervisor-basierte Virtualisierungstechnik von Microsoft für Computer mit x64-fähigem x86-Prozessor. Erhältlich ist Hyper-V sowohl als fester Bestandteil der Serverbetriebssysteme (z. B. Windows Server 2016 Standard und Datacenter) als auch in Form der Standalone-Ausführung (z. B. Microsoft Hyper-V Server 2016).
 - VMware ESXi ist ein Produkt, das die Virtualisierung zur Verfügung stellt
 - VMware vSphere ist der Produktname unter dem alle Komponenten zu finden sind (ESXi, vcenter und weitere Komponenten). ESXi ist eines der Produkte in vSphere

Virtualisierung

- Ansätze der Servervirtualisierung: Vollständige Virtualisierung
 - Nachbildung einer kompletten Hardwareumgebung für die virtuelle Maschine um die Zugriffe der Gastbetriebssysteme zu steuern
 - Virtuelle Maschine simuliert realen Rechner mit allen Komponenten
 - Virtueller Maschinenmonitor (VMM) koordiniert virtuelle Maschinen und Ressourcen
 - Jedes Gast-Betriebssystem hat einen eigenen virtuellen Rechner mit CPU, Hauptspeicher, Laufwerken, Netzwerkkarte, usw. zur Verfügung
 - VMM und Hostbetriebssystem in Ring 0, Gastbetriebssysteme höher

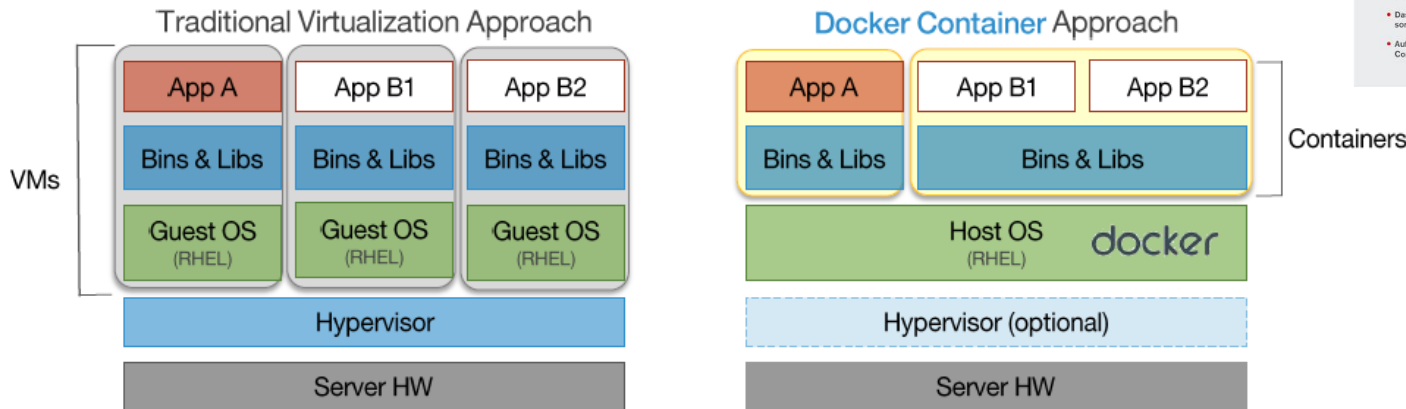


- **Ansätze der Servervirtualisierung: VMs vs. Container (1/3)**
 - Bei Container und VMs handelt es sich um paketierte Computing-Umgebungen, die verschiedene IT-Komponenten vereinen und vom Rest des Systems isolieren.
 - Der entscheidende Unterschied liegt in der **Skalierung und Portierbarkeit**.
 - VMs
 - **VMs enthalten ihr eigenes Betriebssystem**, auf dem sie mehrere ressourcenintensive Funktionen auf einmal ausführen können. Dank der zahlreichen, ihnen zur Verfügung stehenden Ressourcen können VMs ganze **Server, Betriebssysteme, Desktops, Datenbanken und Netzwerke abstrahieren**, aufteilen, duplizieren und emulieren.
 - Außerdem können auf VMs mit einem Hypervisor mehrere Betriebssysteme ausgeführt werden.
 - VMs werden gewöhnlich in Gigabyte gemessen.
-  Container werden oft als äquivalent zum Hypervisor angesehen. Das stimmt nicht ganz, denn Container und Virtualisierung erfüllen unterschiedliche Anforderungen.

- **Ansätze der Servervirtualisierung: VMs vs. Container (2/3)**
 - Bei Container und VMs handelt es sich um paketierte Computing-Umgebungen, die verschiedene IT-Komponenten vereinen und vom Rest des Systems isolieren.
 - Der entscheidende Unterschied liegt in der **Skalierung und Portierbarkeit**.
 - **Container:**
 - Ein Container ist ein **Softwarepaket**, das alles Wichtige zum Ausführen von Software enthält: Code, Laufzeit, Konfiguration und Systembibliotheken, damit das Programm auf jedem Hostsystem ausgeführt werden kann. Zur Laufzeit erhält der Container auch einen eigenen isolierten Teil der Betriebssystemressourcen wie CPU, RAM, Festplatte und Netzwerk.
 - Container tauchten erstmals vor Jahrzehnten mit Versionen wie **FreeBSD Jails** und **AIX Workload Partitions** auf. Der Beginn der modernen Container-Ära war 2013 mit der Einführung von Docker.
 - **Container virtualisieren ein Maschinenbetriebssystem auf Benutzerbereichsebene.**
 - Ein Container besteht aus einem oder mehreren Prozessen, die vom Rest des Systems isoliert sind.
 - Der Container erlaubt dem Prozess nur den Zugriff auf die Ressourcenanforderungen, die spezifiziert wurden. Mit diesen Ressourcenbeschränkungen wird sichergestellt, dass der Container auf einem Knoten mit genügend Kapazität ausgeführt werden kann.
 - Container können **nur eine Art von Betriebssystem ausführen**, z.B. kann ein Container auf einem Linux-Server nur ein Linux-Betriebssystem ausführen.
 - Container werden gewöhnlich in **Megabyte** gemessen. In einen Container werden nur die App und alle zum Ausführen erforderlichen Dateien paketierte. Häufig werden darin auch einzelne Funktionen paketierte, die bestimmte Aufgaben (sogenannte Microservices) ausführen.

• Ansätze der Servervirtualisierung: VMs vs. Container (3/3)

Containers are a **lightweight** alternative to Virtual Machines for running software in **portable** and **isolated** virtual environments



Unterschied zu Virtual Machines

Virtual Machines



- Beinhalten Applikationen und des vollständige Betriebssystem
- Ein Hypervisor wie VMware ESXi sorgt für die Virtualisierung
- Auf einem physischen Server laufen mehrere VMs isoliert voneinander

Container



- Beinhalten Applikationen und nur die notwendige Betriebssystem-Komponenten wie Libraries und Binaries
- Das Betriebssystem mit der Container Engine sorgt für die Virtualisierung
- Auf einem Betriebssystem laufen mehrere Container isoliert voneinander

Attribute	VM	Container
Start-up time & Performance	Slow (minutes) HV overhead	Fast (seconds) no HV overhead
Footprint	Large (nothing shared)	Small (OS kernel shared)
Resource Constraints	Yes	Yes (CPU, Memory)
Isolation & Security	High	High
Portability	Low	High

Aufgabe/Frage

- Wir haben gerade (A) Virtualisierungstechniken und (B) die Containerarchitektur kennengelernt.
- Welches sind die Vor- und Nachteile der Virtualisierung?
- Welches sind die Vor- und Nachteile von Containerarchitekturen?

Bedingung:

→ 5 min



5 min

Antwort zu Virtualisierung:

Vorteile

- **Bessere Ausnutzung der Hardware**
 - Server- und PC-Konsolidierung,
 - Zusammenlegen vieler virtueller Server auf möglichst wenigen physikalischen Servern (auf aktuelle Serverblades passen bis zu 40 aktive Server-Instanzen?)
 - Bessere Energie-Effizienz, Kosten-Senkung bei Hardware, Einsparung Stellplatz
- **Vereinfachte Administration**
 - Reduzierung der Anzahl der physischen Server
- **Vereinfachte Bereitstellung**
 - Sehr schnelle manuelle oder automatische Erzeugung von neuen Servern
- **Erhöhung der Verfügbarkeit**
 - Migration von Servern im laufenden Betrieb
 - Virtuelle Maschine (VM können leicht vervielfältigt und gesichert werden)
 - Snapshots vom aktuellen Zustand
- **Hohe Sicherheit**
 - VMs sind gegenüber anderen VMs und dem Host-System isoliert

Antwort zu Virtualisierung:

Nachteile

- VMs bieten geringere Performance als reale Maschine
- Nicht jede Hardware kann aus einer virtuellen Maschine emuliert werden
- Bei der Serverkonsolidierung können virtuelle Maschinen einen Single Point of Failure darstellen. Beim Ausfall eines Hosts fallen mehrere Server aus.
- Komplex; Know-How unbedingt notwendig.

Antwort zu Containerarchitekturen:

Vorteile

- Weniger Overhead – Container benötigen weniger Systemressourcen.
- Erhöhte Portabilität – In Containern ausgeführte Anwendungen können problemlos auf mehreren verschiedenen Betriebssystemen und Hardwareplattformen bereitgestellt werden.
- Grössere Effizienz – Mithilfe von Containern können Anwendungen schneller bereitgestellt, gepatcht oder skaliert werden.
- Bessere Anwendungsentwicklung – Container unterstützen die Beschleunigung von Entwicklungs-, Test- und Produktionszyklen; flexibel einsetzbar für Microservices

Nachteile:

- Nicht alle Container-Anwendungen sind miteinander kompatibel.
- Die dauerhafte Datenspeicherung ist kompliziert.
- Generelle Sicherheitsprobleme
- Komplexe Einrichtung und Verwaltung (→ Containeradministrations-SW)
- Zu hohe Erwartungen von Entscheidungsträgern und Geschäftsführern
- Wenig Fachpersonal, das sich mit Containern auskennt

Hypervisor Comparison 2019: KVM vs Hyper-V vs XenServer vs vSphere

Feature	Windows Hyper-V 2019	vSphere 6.7	XenServer 7.6	KVM
RAM/Host	24TB	12 TB	5TB	12TB
RAM/VM	12 TB for generation 2; 1 TB for generation 1	6 TB	1.5TB	6 TB
CPUs/VM	240 for generation 2; 64 for generation 1;	128	32	240
VM Disk	64 TB for VHDX format; 2040 GB for VHD format	62TB	2TB	10TB
VM Live Migration	Yes	Yes	Yes	Yes
VM Replication supports	Yes	Yes	Yes	Yes
Overcommit resources	No	Yes	No	Yes
Disk I/O Throttling	Yes	Yes	Yes	Yes
Hot plug of virtual resources	Yes	Yes	Yes	Yes

<https://www.acte.in/citrix-xenserver-vs-vmware-vsphere-article>

Virtualisierung

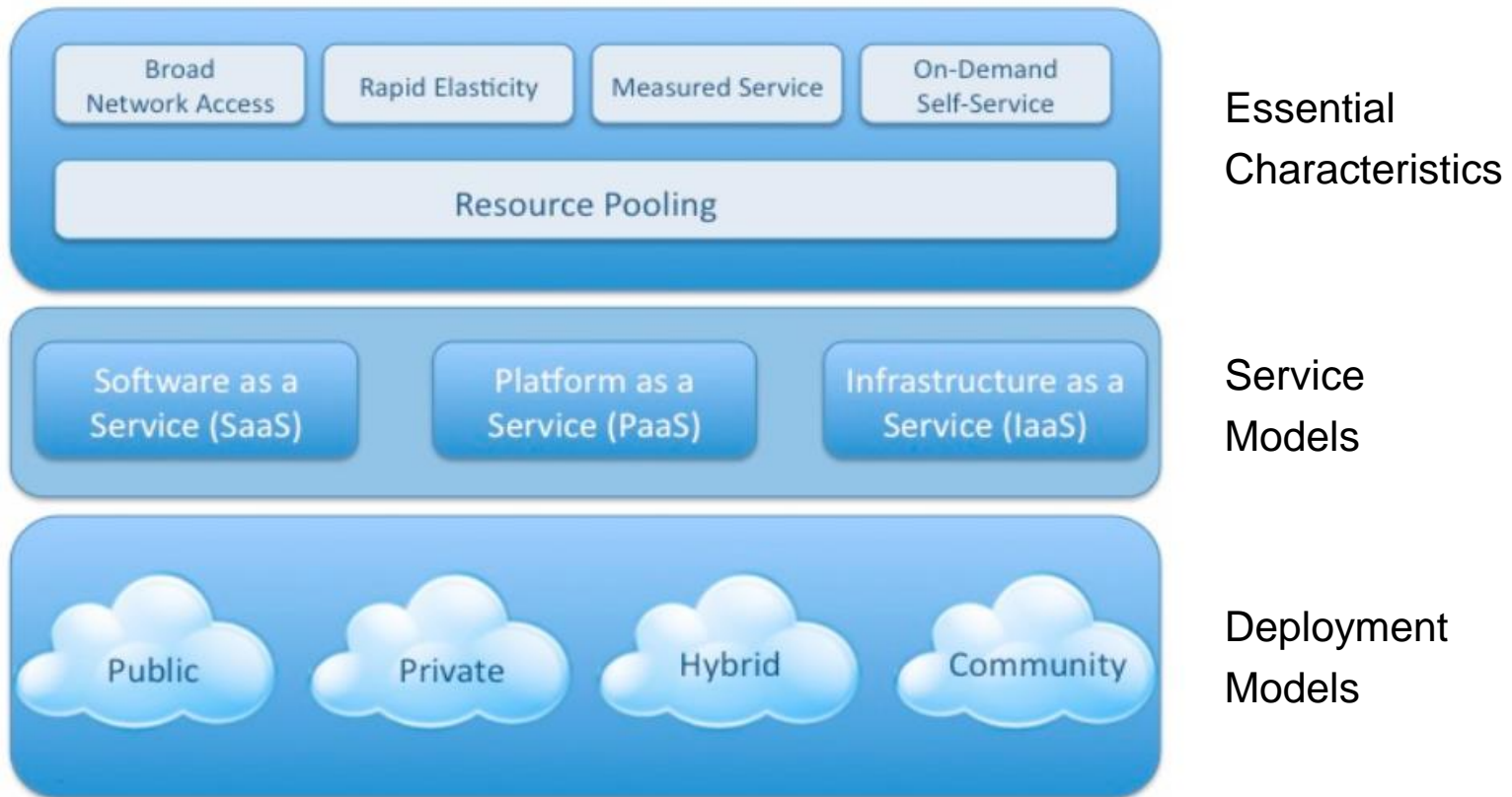
- Datenspeichervirtualisierung (SDS=Software Defined Storage) (1/2)
 - Idee: **Trenne Server und Speicher und verwende Speichernetzwerk**
 - Speichervirtualisierung (auch als **Software-definierter Speicher** oder virtuelles SAN bezeichnet) ist die Konsolidierung mehrerer physischer Speicher-Array aus SANs (SAN=Storage Area Network), sodass sie als einheitliches virtuelles Speichersystem erscheinen.
 - Bei **virtualisierten Speicherlösungen** wird die **Speichermanagement-software von der grundlegenden Hardware-Infrastruktur getrennt**, um die Flexibilität zu vergrößern und skalierbare Pools von Speichersystemen zu schaffen.
 - Die **Speichermanagementsoftware ist das Gehirn und die Intelligenzebene**. Sie bestimmt, welche Funktionen, Features, Dienste und Vorteile Sie Ihren Hosts, Anwendungen und Anwendern bereitstellen können.
 - Dieses System kann ganz **unterschiedliche Hardware unterschiedlicher Netzwerke, Anbieter oder Rechenzentren umfassen** und diese als logische Einheit abbilden, die über eine einheitliche Bedienoberfläche verwaltet wird.
 - Die Technologie kann die Speicherhardware (Arrays und Festplatten) in **virtuellen Pools abstrahieren**, so wie Computer-Virtualisierungssysteme (VMWare ESX oder Hyper-V) die Computer-Hardware (Server) als virtuelle Maschinen (VMs) abstrahieren.

- Datenspeichervirtualisierung (SDS=Software Defined Storage) (2/2)
 - Speichervirtualisierung löst folgende Herausforderungen:
 - **Abhängigkeit von Anbietern**
 - **Datenmigration über verschiedene Arrays hinweg**
 - **Skalierbarkeit**
 - **Redundanz**
 - **Performance**
 - **Hohe Kosten**
 - **Verwaltung**
 - Die Virtualisierung der Speicherumgebung trägt dazu bei, ein System flexibler zu machen, da **gängige Hardware schnell hinzugefügt** werden kann und von der Virtualisierungssoftware sofort erkannt wird. Diese Hardware-Ressourcen können dann in einem gemeinsamen virtuellen Speicher-Pool gebündelt werden, der der Virtualisierungsplattform zur Verfügung steht.
 - Durch die Virtualisierung können Daten **über ganz verschiedene physische Geräte in großen Netzwerken oder Rechenzentren hinweg verwaltet** werden. Die Daten können konsolidiert werden, um Leistungsprobleme zu isolieren, Probleme vorherzusehen und zu lösen und den zukünftigen Kapazitätsbedarf zu planen.

Virtualisierung

- Netzwerkvirtualisierung
 - **Abstraktion von in Hardware bereitgestellten Netzwerkressourcen in Software abstrahiert.**
 - **Netzwerkvirtualisierung kann mehrere physische Netzwerke in einem virtuellen, softwarebasierten Netzwerk zusammenführen** oder ein physisches Netzwerk in einzelne, unabhängige virtuelle Netzwerke unterteilen
 - VPN
 - Getrennte virtuelle Netzwerke über gemeinsame einheitliche virtuelle Infrastruktur
 - Kantenvirtualisierung
 - Mehrere unabhängige virtuelle Verbindungen werden über eine gemeinsame physikalische Verbindung (Kante) transportiert
 - Vorteile
 - Entkopplung Netzwerkservices von der zugrunde liegenden Hardware
 - Virtuelle Bereitstellung ganzer Netzwerke.
 - Physische Netzwerkressourcen wie Switches und Router werden in Pools zusammengefasst und sind für Anwender über ein zentrales Managementsystem zugänglich.
 - Hoher Automatisierungsgrad und Effizienz für administrative Aufgaben
 - Mehr Flexibilität und Personalisierbarkeit
 - Schnelleren und sicheren Zugriff auf Anwendungen und Daten

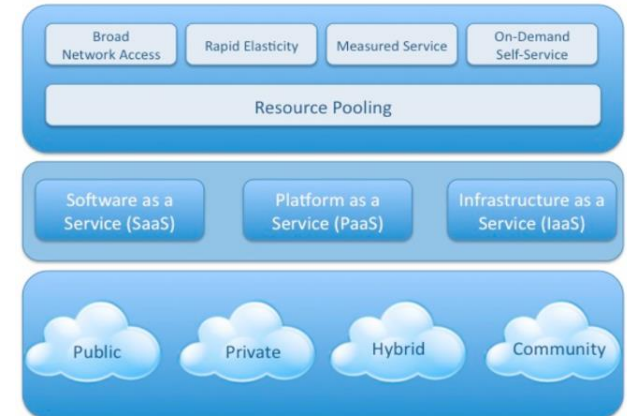
Visual Model Of NIST Working Definition Of Cloud Computing
<http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html>



NIST: National Institute of Standards and Technology
CSA: Cloud Security Alliance
<https://csrc.nist.gov/publications/detail/sp/800-145/final>

Wir sehen das Schaubild der NIST?

Bitte erklären Sie die wesentlichen
Charakteristiken des Cloud Computing?



Bedingung:

→ 5 min

5 min

Antwort: **wesentliche Charakteristiken**

- **Broad Network Access**

Sämtliche Ressourcen sind breitbandig über das Internet erreichbar.

- **Rapid Elasticity**

Ressourcen können in Echtzeit schnell und teilweise automatisiert auf die veränderten Bedürfnisse des Nutzers angepasst werden. Aus der Sicht der Nutzer stehen unbeschränkt Ressourcen zur Verfügung, die jederzeit und in jedem Umfang gekauft bzw. genutzt werden können.

- **Measured Service**

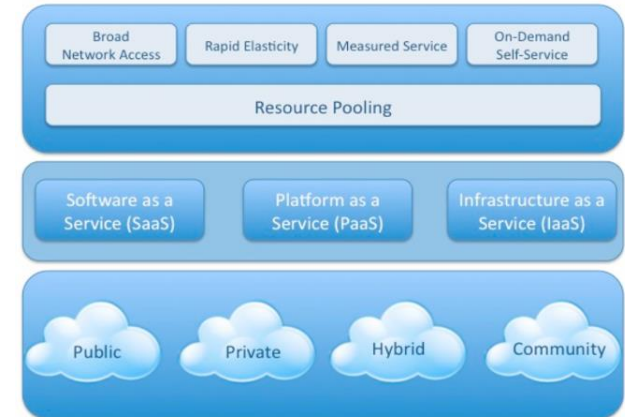
Cloud Computing Systeme kontrollieren und optimieren die Zuteilung von Ressourcen vollautomatisiert. Der Ressourcenverbrauch wird kontinuierlich gemessen, kontrolliert und berichtet, um Transparenz für den Provider und den Kunden herzustellen. Nur die genutzten Dienste und Ressourcen werden abgerechnet.

- **On-demand self-service**

Ein Kunde kann selbstständig Rechenressourcen abrufen und buchen, ohne dass hierzu eine Interaktion mit dem Service Provider notwendig ist. Die Bereitstellung von Cloud-Ressourcen (Serverzeit, Speicherressourcen) wird bei Bedarf ermöglicht, wann immer sie erforderlich sind.

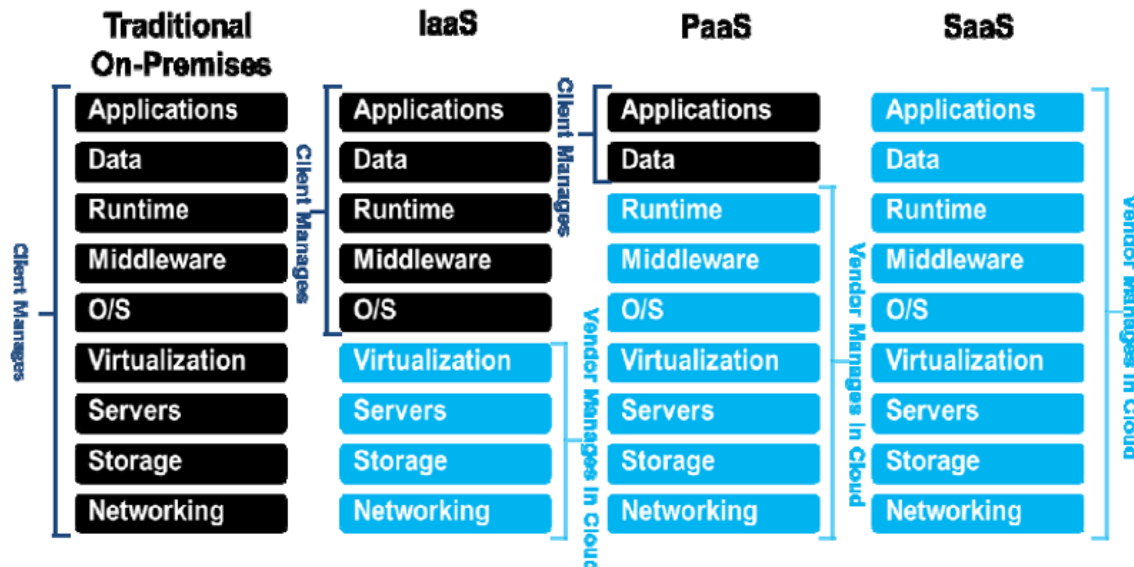
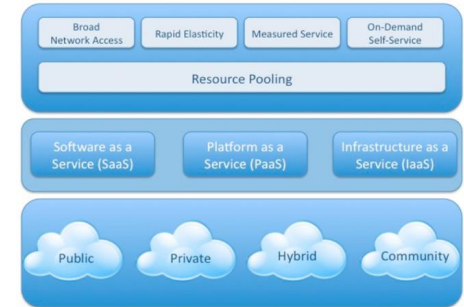
- **Resource Pooling**

Die Rechenressourcen des Providers werden an einer Stelle gebündelt und mehreren Nutzern zur Verfügung gestellt.



- **Cloud Service Model**

- SaaS
 - Bereitstellung der Funktionalität von SW via Internet
 - Webbasierter Zugriff auf Anwendungen
 - Kosten nur im Bedarfsfall
 - Immer aktuelle Version
- PaaS
 - Plattformbenutzung übers Internet (Entwicklungsumgebung)
 - Kundenanwendungen, die als Plattform dienen
- IaaS
 - ‚Mieten‘ von Ressourcen für Rechenleistung, Speicher, Netzwerk usw.
 - Kosten nur im Bedarfsfall
 - Skalierbar



Cloud Computing

- **Cloud Computing: Cloud Deployment Model**

- **Public Cloud**

Die Cloud-Infrastruktur wird zur freien Nutzung durch die allgemeine Öffentlichkeit bereitgestellt. Sie kann in Besitz sein, verwaltet und betrieben werden von einer oder mehreren Unternehmen, einer akademischen oder staatlichen Organisation oder einer Kombination davon verwaltet und betrieben werden. Es existiert in den Räumlichkeiten des Cloud-Anbieters.

- **Private Cloud**

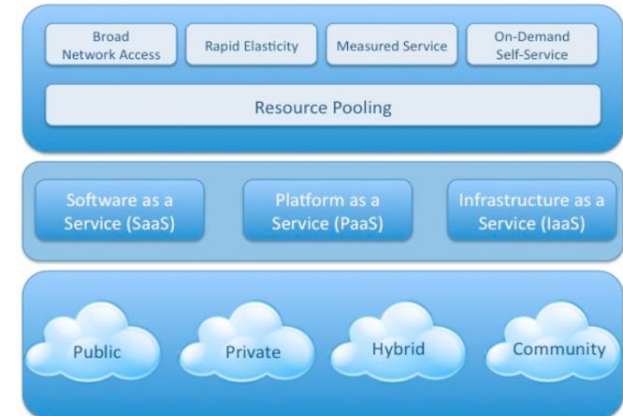
Die Cloud-Infrastruktur wird für die exklusive Nutzung durch eine einzelne Organisation bereitgestellt bestehend aus mehreren Verbrauchern (z. B. Geschäftseinheiten). Es kann Eigentum sein, verwaltet und betrieben von der Organisation, einem Dritten oder einer Kombination von ihnen, und es kann existieren auf oder außerhalb des Geländes.

- **Community Cloud**

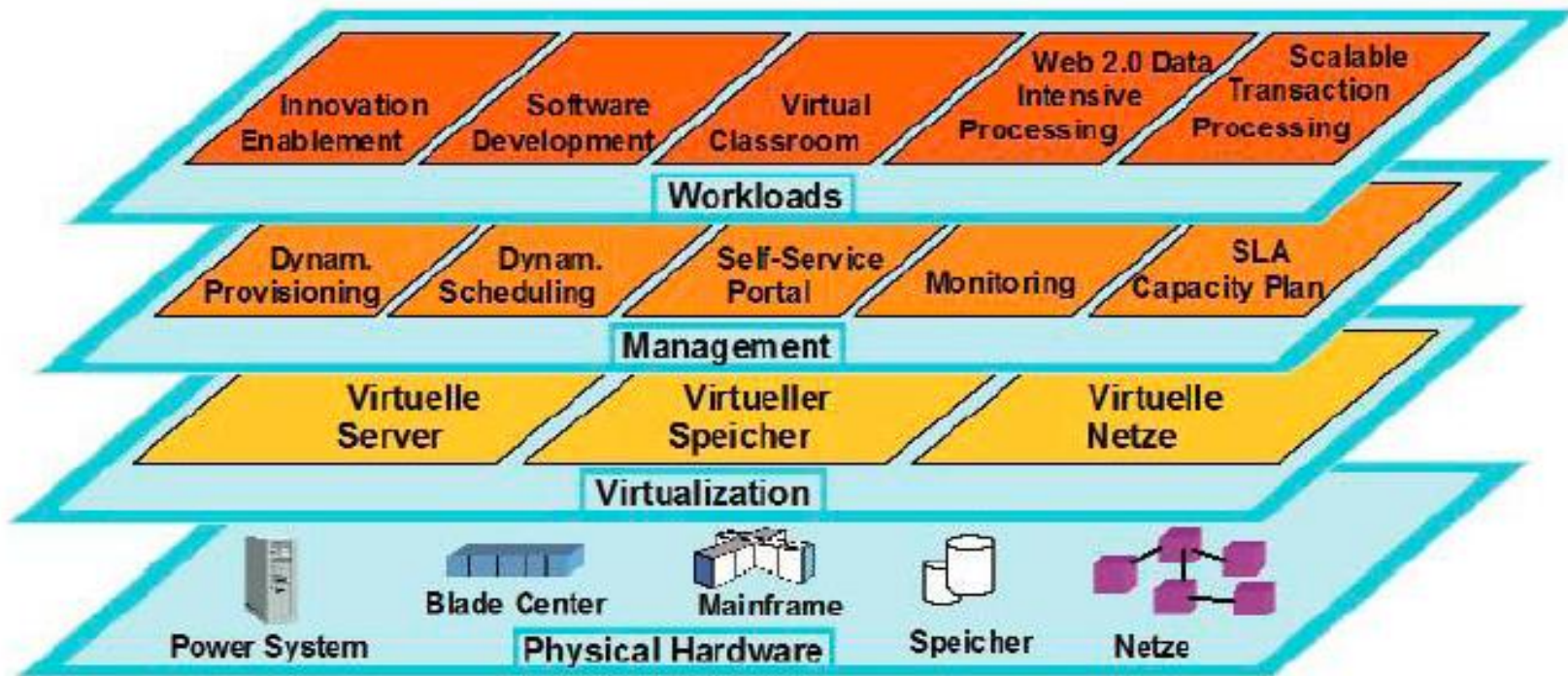
Die Cloud-Infrastruktur wird zur exklusiven Nutzung durch eine bestimmte Gemeinschaft von Verbrauchern aus Organisationen bereitgestellt, die gemeinsame Anliegen haben (z. B. Thema, Sicherheitsanforderungen, Richtlinien und Compliance-Erwägungen). Sie kann in Besitz sein, verwaltet und betrieben werden von einem oder mehreren Organisationen in der Gemeinschaft, einer dritten Partei oder einer Kombination davon. Sie kann innerhalb oder außerhalb des Betriebs- geländes existieren.

- **Hybrid Cloud**

Die Cloud-Infrastruktur ist eine Zusammensetzung aus zwei oder mehr unterschiedlichen Clouds Infrastrukturen (privat, community oder öffentlich), die einzelne Einheiten bleiben, aber zusammen verbunden sind durch standardisierte oder proprietäre Technologie, die Daten- und Anwendungs-Portabilität ermöglicht (z. B. Cloud Bursting zum Lastenausgleich zwischen Clouds)



- Generisches Architekturbild



- Cloud Anwendungen
 - Amazon Web Services



- Google Apps



- Microsoft Azure

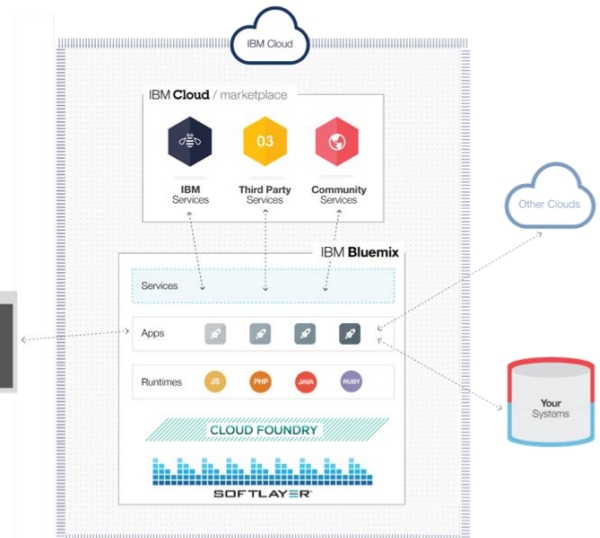


Quelle: <http://www.microsoft.com/germany/net/WindowsAzure/>



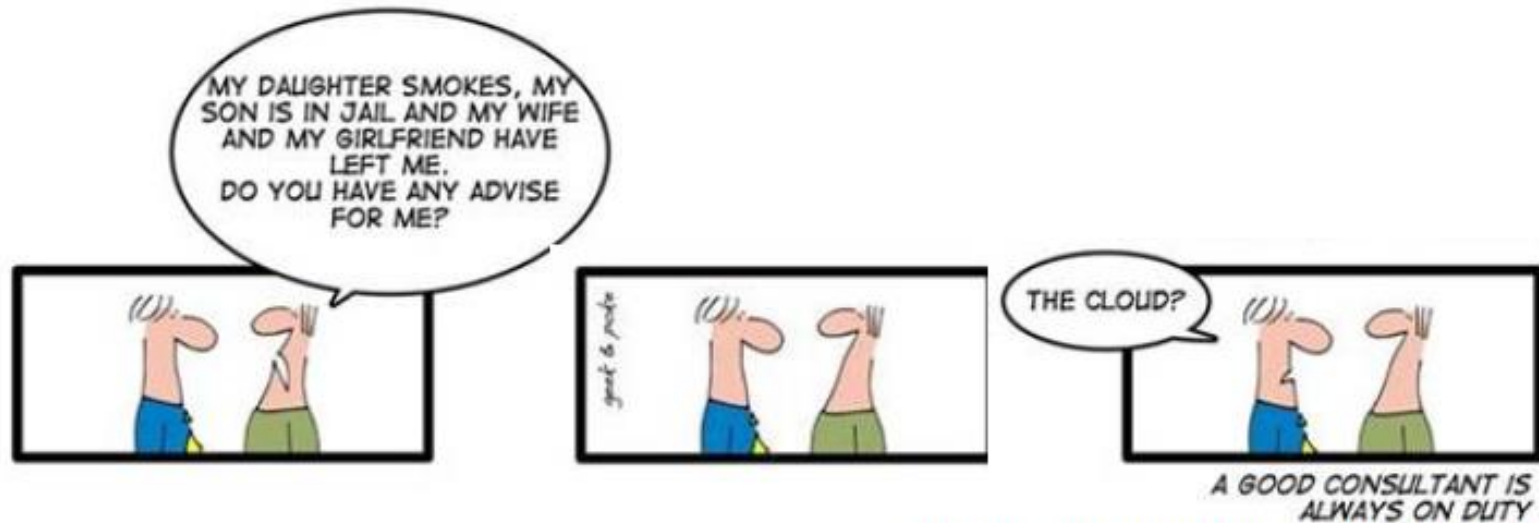
- IBM Cloud

- Alibaba



Zusammenfassung

- Nutzer werden Mieter
- Nutzer besitzen meist keine physikalische Infrastruktur
- Nutzer zahlen nur Dienste, die sie benutzt haben (Utility Computing)
- Zusätzliche Ressourcen (Speicher, Rechenleistung, Anwendungen, usw.) stehen immer zur Verfügung und können nur Bedarf angefordert werden
- Benutzer haben Anspruch auf Erbringung durch Quality of Services oder Service Level Agreements



Quelle: geek and poke , <http://geekandpoke.typepad.com>