

# Ausgewählte Kapitel der Systemsoftware

## Virtualisierung

Betrachtung aktueller Hypervisor wie Xen, KVM und Hyper-V

Guilherme Bufolo

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)  
[www4.informatik.uni-erlangen.de](http://www4.informatik.uni-erlangen.de)

Sommersemester 2010



# Motivation

---

- Die Cloud wächst
- Was ist die Cloud?
- Wozu Virtualisierung in der Cloud?
- Was macht Virtualisierung?
- Sind alle Virtualisierungslösungen miteinander kompatibel?



# Agenda

---

- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- Hypervisors
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- Fazit



# Agenda

---

- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- Hypervisors
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- Fazit



# Erwartungen an die Cloud

---

- Hochverfügbarkeit
  - „Unendliche“ Ressourcen
  - Standortunabhängiger Zugriff
  - Kosten senken
- 
- Wie hilft Virtualisierung hier?



# Beispiel: Kosten senken

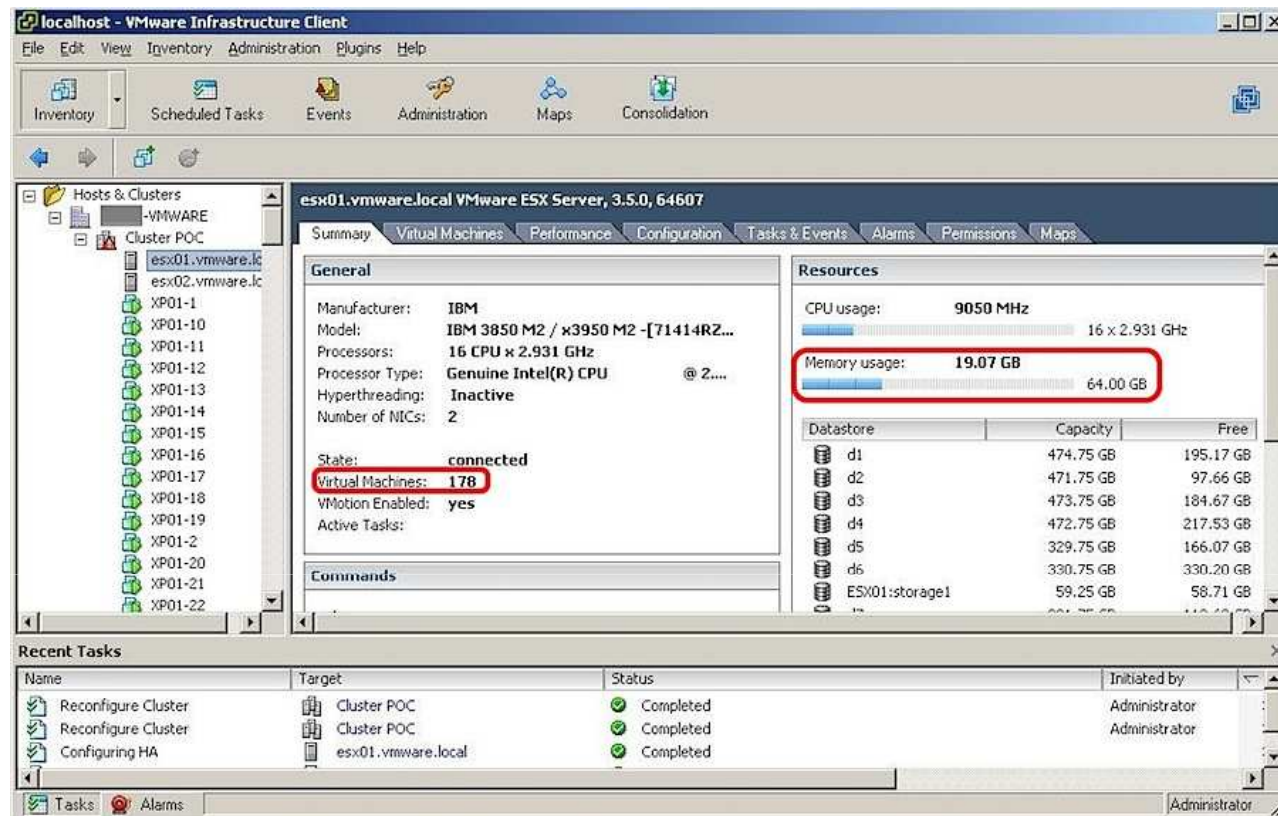
---

- Gastsystem
  - Windows XP
  - Eine virtuelle CPU, 512 MB RAM
  - Benutzt für Office Anwendungen und Customer Support
- Reales System
  - IBM x3850 M2 Quad-Socket
  - Quad-Core-System (16 Cores gesamt)
  - 64 GB RAM (physisch)
- Host-Rechner
  - 178 virtuelle Gastinstanzen
  - 178 x 512 MB = 89 GB RAM
- Einsparung von 35 GB RAM → 11.800 €
- Noch nicht optimal!

Beispiel aus Memory Overcommitment in the Real World [1]



# Beispiel: Kosten senken



- 20 GB tatsächlicher Konsum
- Einsparungspotential von 44 GB RAM!

Bild aus Memory Overcommitment in the Real World [1]



# Agenda

---

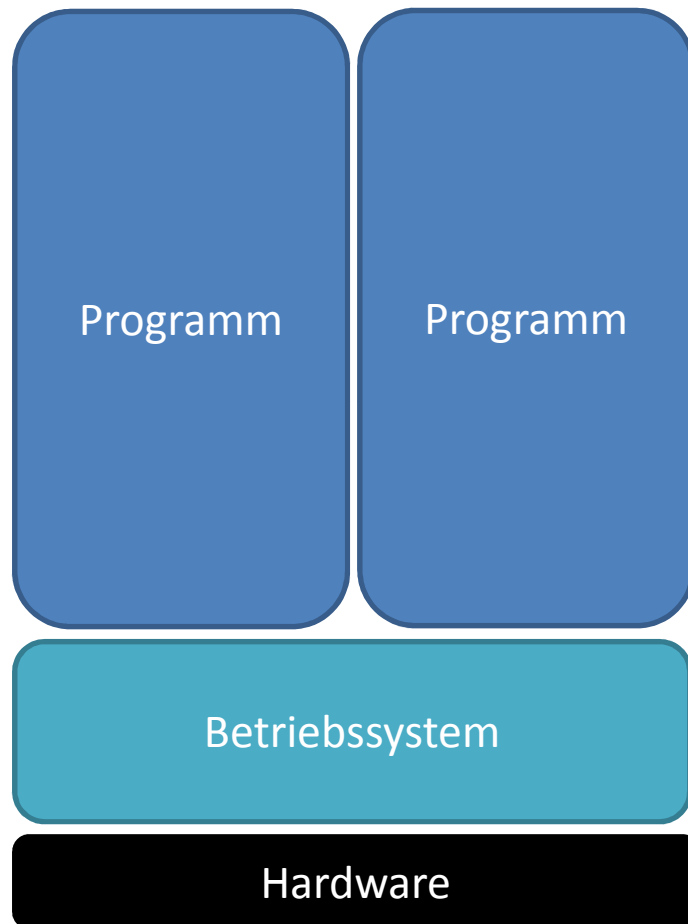
- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- Hypervisors
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- Fazit





# Virtualisierung

Verwendung eines Rechners **ohne** Rechnervirtualisierung

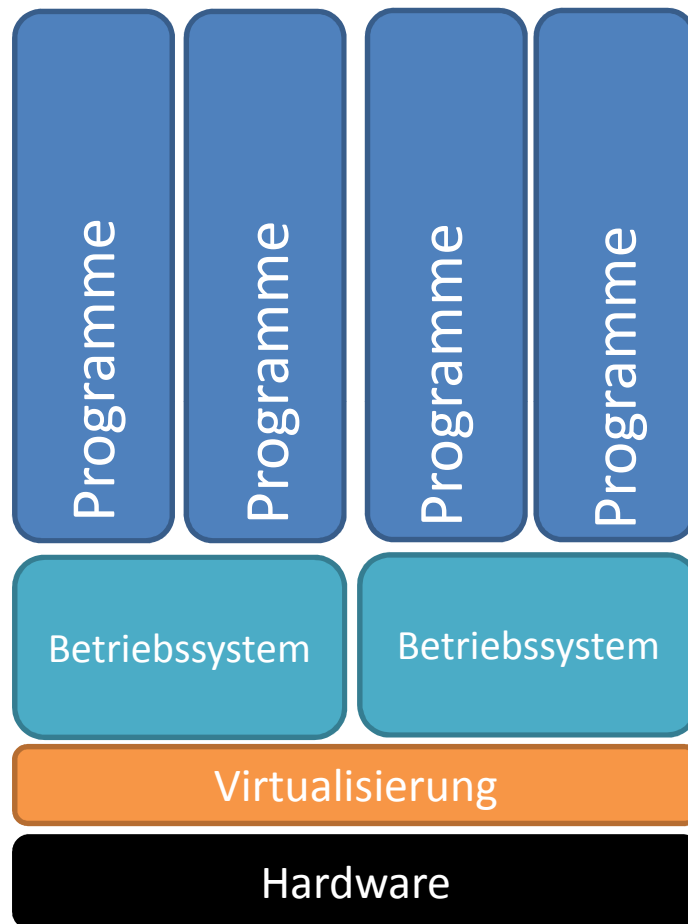


- Klassische Rechnernutzung
- Ein Betriebssystem auf einer Hardware
- Keine Möglichkeit BS-Zustand „einzufrieren“
- Mehrere parallel laufende Programme



# Virtualisierung

Verwendung eines Rechners **mit** Rechnervirtualisierung



- Betriebssystem
  - Mehrere auf einer Hardware
  - Merkt nichts
  - Isoliert voneinander
  - Unabhängig start- und beendbar
  - „Einfrieren“ des Zustandes möglich
- Programme
  - Merken nichts
  - Mehrere parallel laufend



# Agenda

---

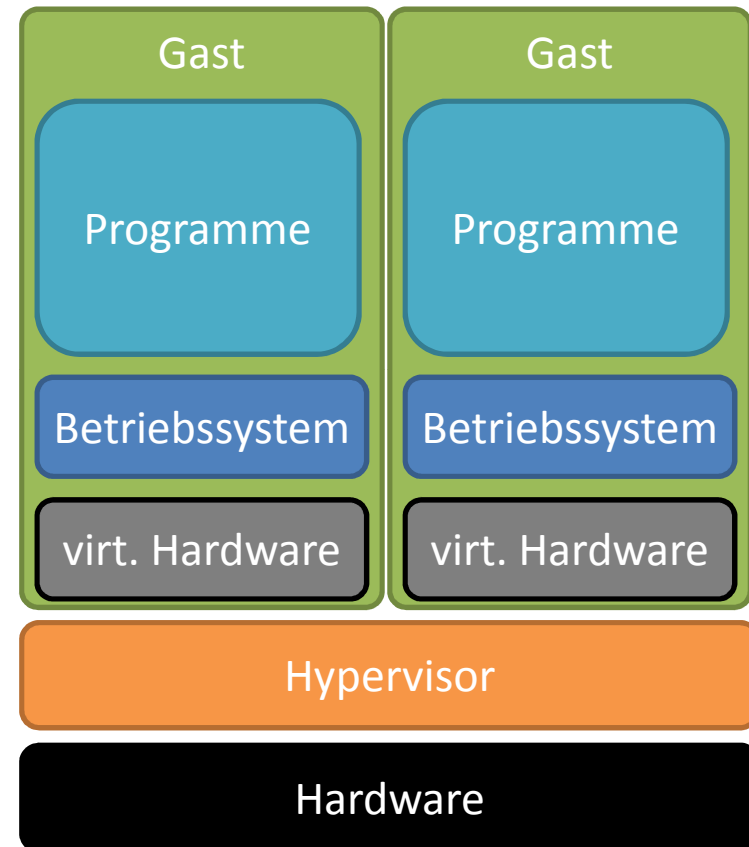
- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- **Hypervisors**
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- Fazit



# Hypervisor-Arten

- Hypervisor bootet zuerst
- Selbst zur Verfügung gestellt
  - Scheduler
  - Speicherverwaltungen
  - Treiber
- Virtuelle Hardware für Gast

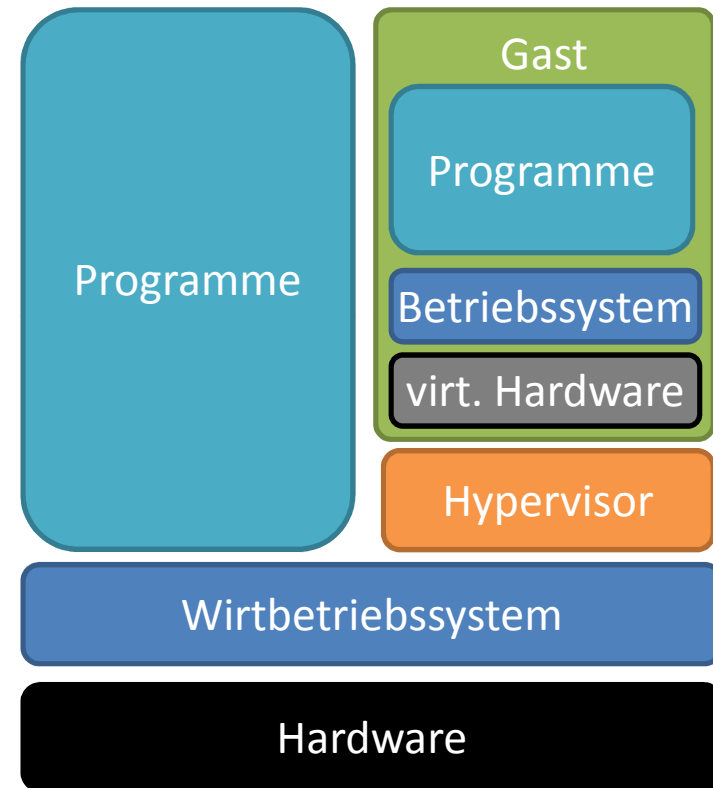
## Bare-Metal (Typ 1)



# Hypervisor-Arten (fort.)

## Hosted (Typ 2)

- Wirtbetriebssystem bootet zuerst
  - Lädt Hypervisor nach
- Hypervisor (meistens) Teil des Wirtbetriebssystems
- Von Wirtbetriebssystem mitverwendet
  - Scheduler
  - Speicherverwaltung
  - Treiber
- Virtuelle Hardware für Gast

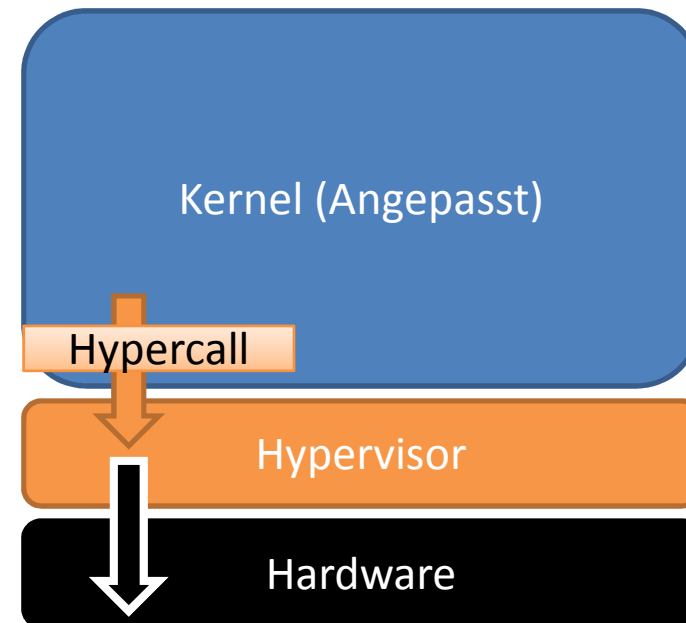


# Hypervisor-Typen

## Hypercall

- Aufruf in den Hypervisor
- Ersetzt privilegierte Operationen im modifizierten Kernel
- Ist ein Interrupt

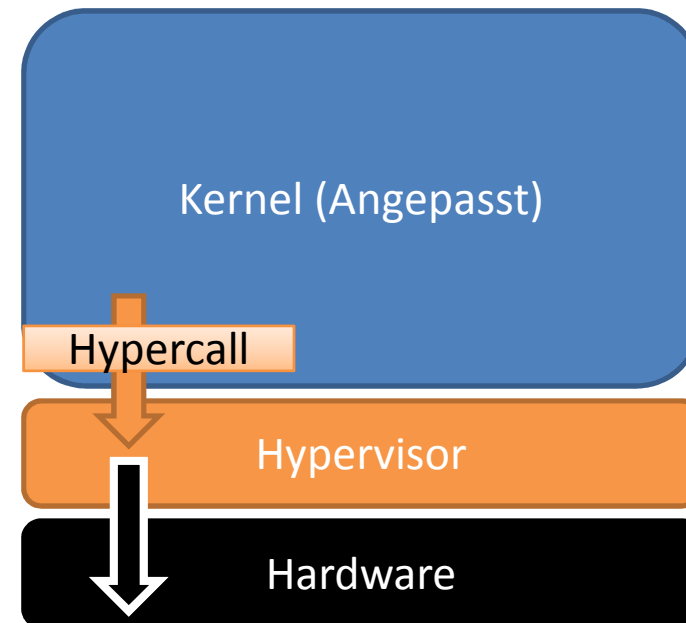
## Paravirtualisierung



# Hypervisor-Typen (fort.)

- Gastbetriebssystem wird angepasst
  - Privilegierte Funktionen benutzen Hypercalls
- Betriebssystem ist Virtualisierung bekannt
- Hardware-Zugriff nur über Hypervisor
- Nachteil
  - Closed-Source-Betriebssysteme keine Anpassung möglich
- Vorteil
  - Performant

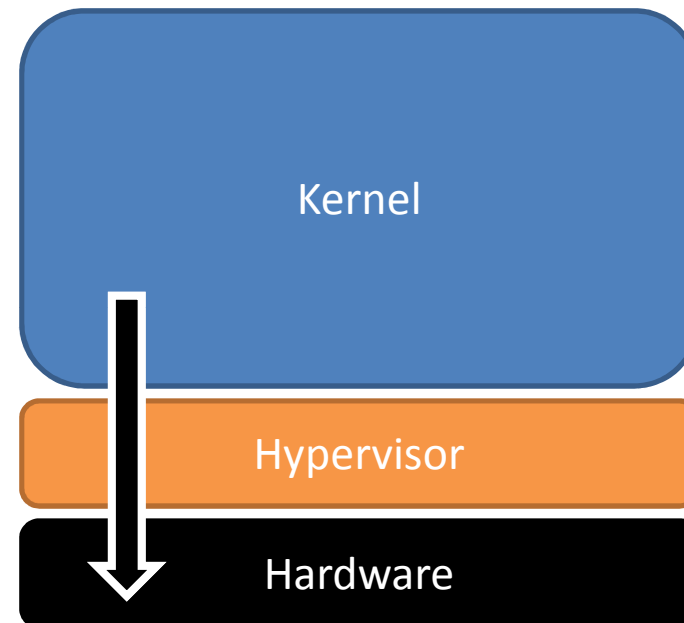
## Paravirtualisierung



# Hypervisor-Typen (fort.)

- Gastbetriebssystem bedarf keiner Anpassung
- Betriebssystem sieht keinen Unterschied zu reeller Hardware
- Kommuniziert direkt mit Hardware
- Nachteil
  - Benötigt kompatible Hardware
- Vorteil
  - Keine Modifikation des Gastbetriebssystems

## Vollvirtualisierung





# Agenda

---

- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- **Hypervisors**
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- Fazit



# Funktionalitäten

---

- Live-Migration
- Memory-Overcommit
- Memory-Ballooning
- VM-Vorlagen
- Linked-VM



# Live-Migration

---

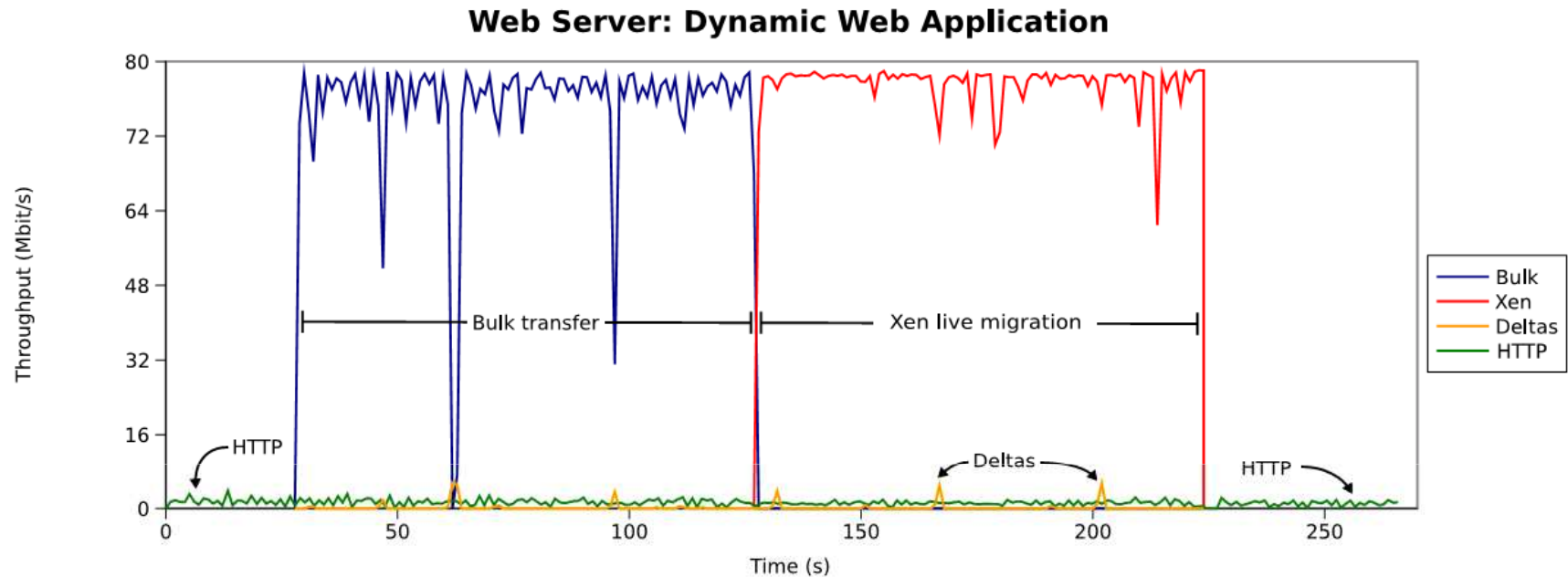
- Gäste werden im Hintergrund übertragen (VM-Image, RAM)
  - Änderungen an RAM und VM-Image werden protokolliert
- Sobald Hintergrundübertragung beendet
  - VM angehalten
  - Änderungen werden auf Zielsystem wiederholt
- VM wird auf Ziel fortgesetzt

} < 4 Sek \*

\* Aus Live Wide-Area Migration of Virtual Machines Including Local Persistent State [2]



# Live-Migration: Beispiel



phpBB Bulletin Board

„Spider“ hat 250 Anfragen/sek gestellt (Posts gemacht, Bulletin Board durchsucht)

→ **Down-time von nur 3,09 Sekunden**

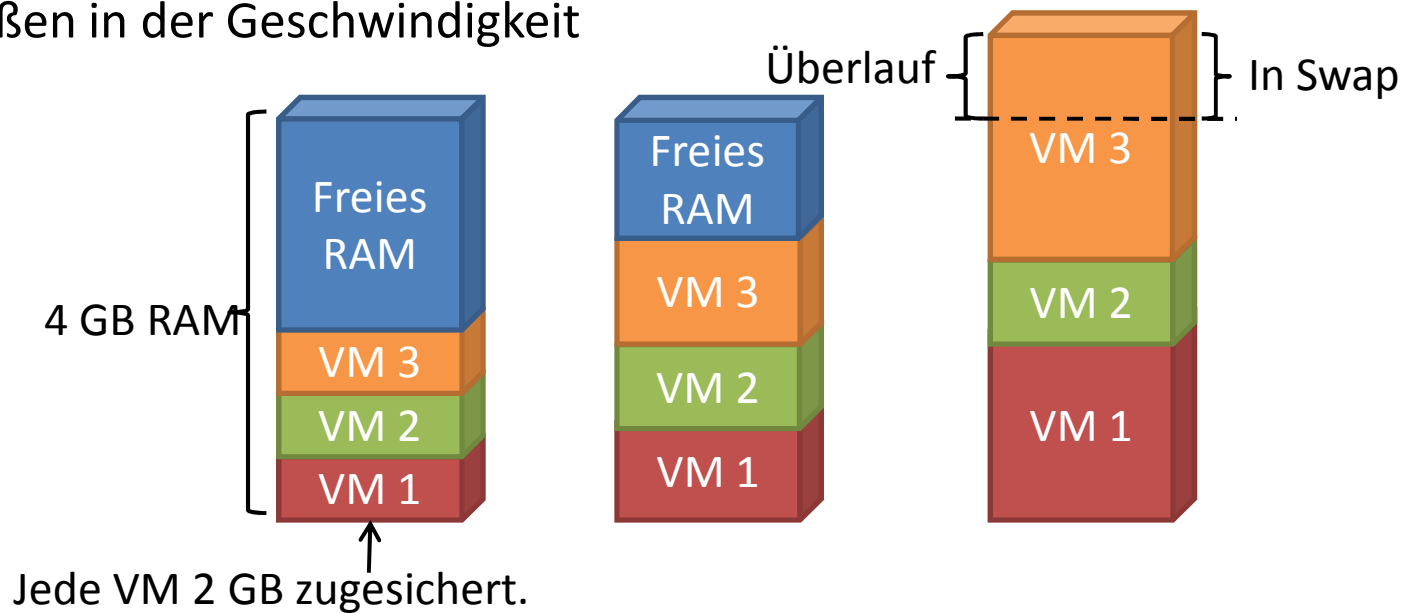
Beispiel aus

Live Wide-Area Migration of Virtual Machines Including Local Persistent State [2]



# Memory-Overcommit

- Der Hypervisor verspricht RAM pro Gast
  - In der Summe mehr als im System installiert
- $\sum(\text{RAM-Verbrauch Gäste}) \leq \text{Installiertes RAM}$ 
  - Nichts zu tun (Idealfall)
- $\sum(\text{RAM-Verbrauch Gäste}) > \text{Installiertes RAM}$ 
  - Auslagerung auf Festplatten (ähnlich Swap bei Programmen)
  - Einbußen in der Geschwindigkeit



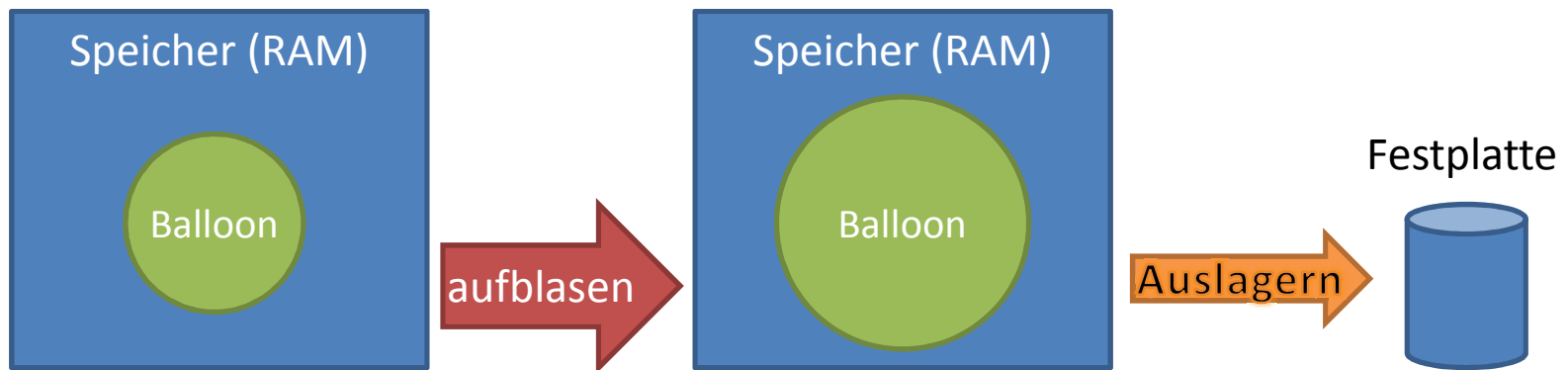
# Memory-Ballooning

---

- Betriebssystemverhalten
  - Physikalischer RAM-Kontingent variiert zur Laufzeit
- Konzept ist das dynamische Belegen von Speicherseiten durch einen Treiber
- Memory-Ballooning-Treiber sichert sich Speicherseiten
  - Verwendet diese Speicherseiten aber nicht selber
  - Hypervisor kann diese Speicherseiten wiederverwenden
- Dieser Speicher kann für andere Gäste verwendet werden



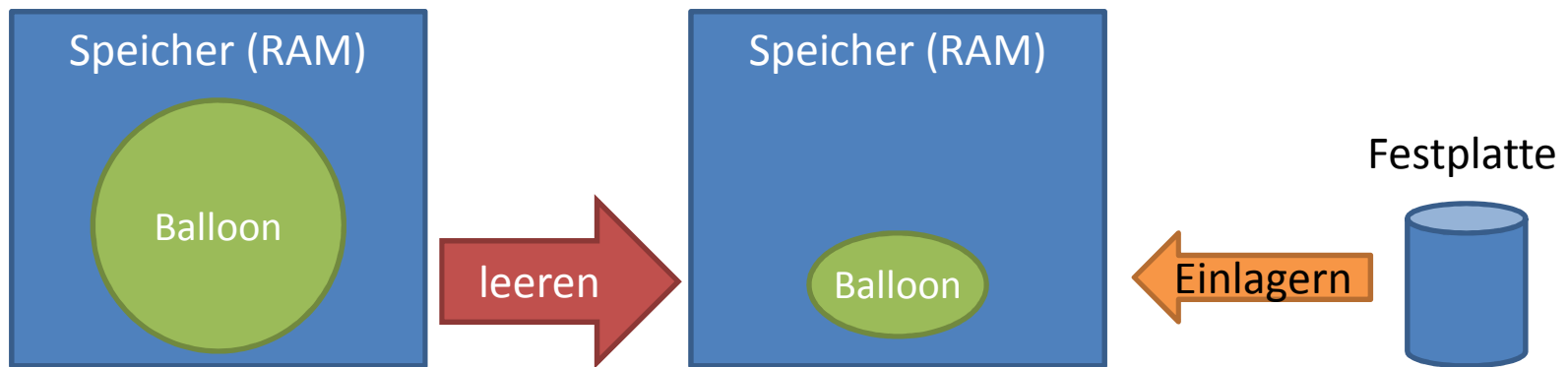
# Memory-Ballooning (fort.)



- Sobald der Balloon „aufgeblasen“ wird (Speicher belegt) ist das Gastbetriebssystem gezwungen Speicher anderer Applikationen auszulagern (Page-Out)
- Der Hypervisor kann dieses frei gewordene RAM-Segment anderen Gästen zuweisen



# Memory-Ballooning (fort.)



- Braucht der Hypervisor dieses RAM-Segment nicht mehr, kann der Balloon wieder „geleert“ werden (Speicher freigeben)
- Das Gastbetriebssystem kann dieses Segment wieder für ausgelagerte Seiten verwenden (Page-In)





# VM-Vorlagen

---

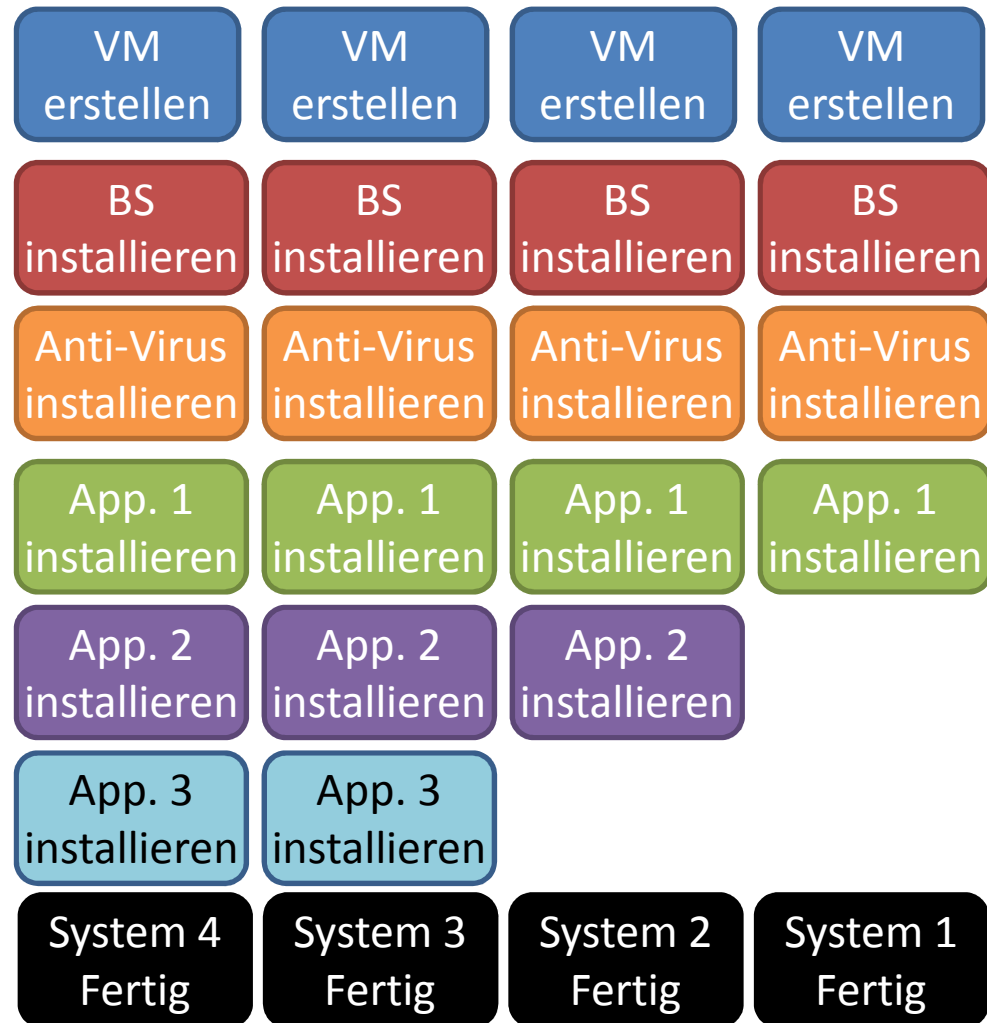
- Zur Herstellung von lauffähigen VM-Kopien verwendet
- Beinhalten Standardkonfigurationen (z.B.: Software-Bundle)
- Erstellungszeit im Vergleich sehr niedrig
  - Herkömmliche Methode ein paar Stunden
  - Eine Kopie kann sofort verwendet werden
  - Vorher vorhandene Software ist auch auf der Kopie vorhanden



# VM-Vorlagen (Beispiel)

## Ohne Vorlagen

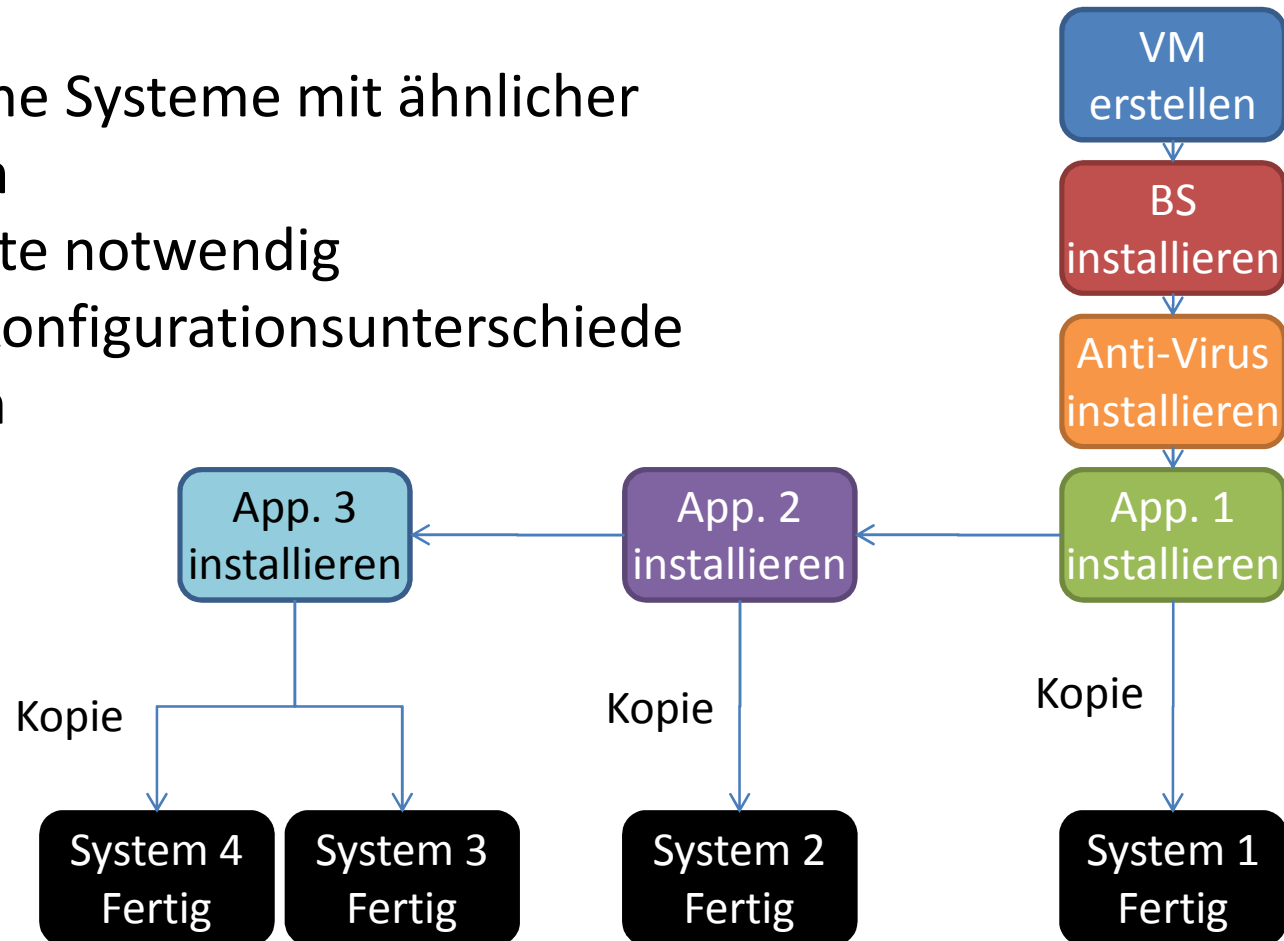
- 4 verschiedene Systeme mit ähnlicher Konfiguration
- 17 Schritte
- +4 falls in einer VM
- **Ungewollte** Konfigurationsunterschiede möglich



# VM-Vorlagen (Beispiel)

## Mit Vorlagen

- 4 verschiedene Systeme mit ähnlicher Konfiguration
- Nur 10 Schritte notwendig
- **Ungewollte** Konfigurationsunterschiede **nicht** möglich



# Linked-VM

- Basis ist gemeinsames Image
- Eigenes Image speichert nur Unterschiede zu Basis-Image
- Speicherplatznutzung wird optimiert
  - Nur benutzergenerierte Daten müssen gespeichert werden (Idealfall)
- Änderungen an Basis-Image können, im Idealfall, gleich an Linked-VM weitergegeben werden

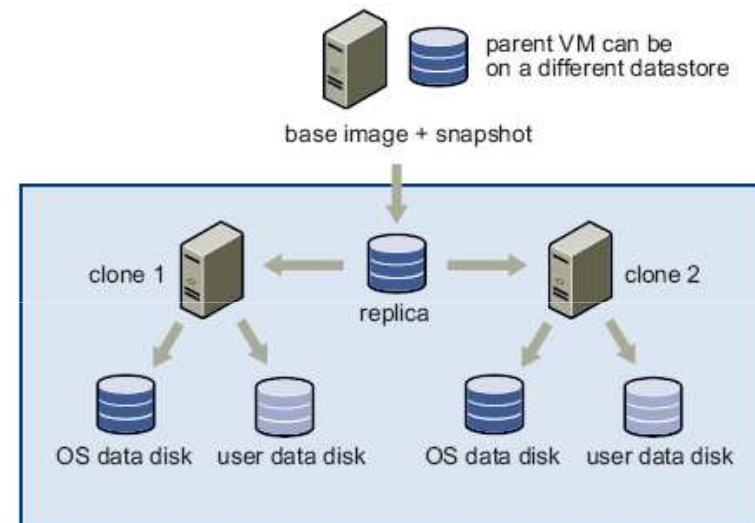


Bild aus Linked clones in VMware View 3 [3]



# Verfügbare Lösungen

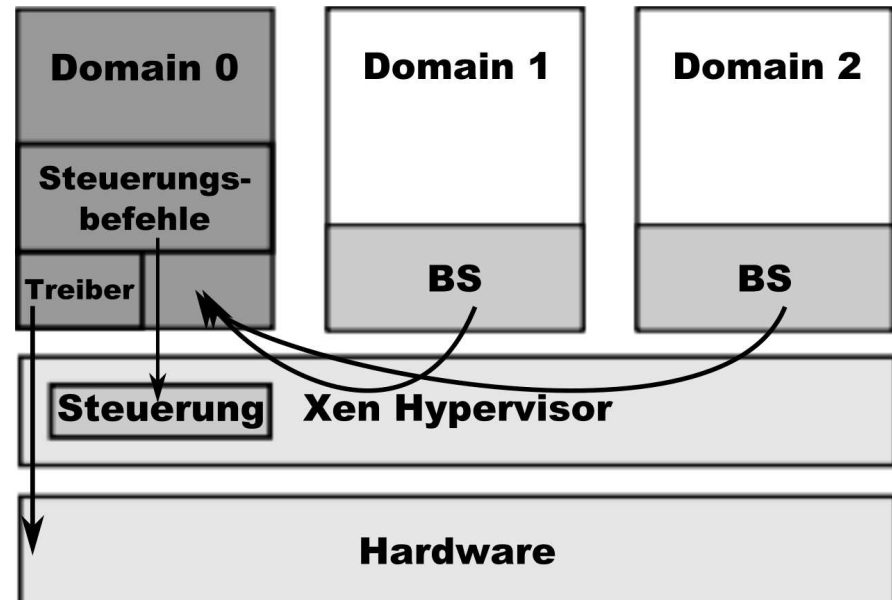
---

- Xen
- Hyper-V
- VMware
- KVM



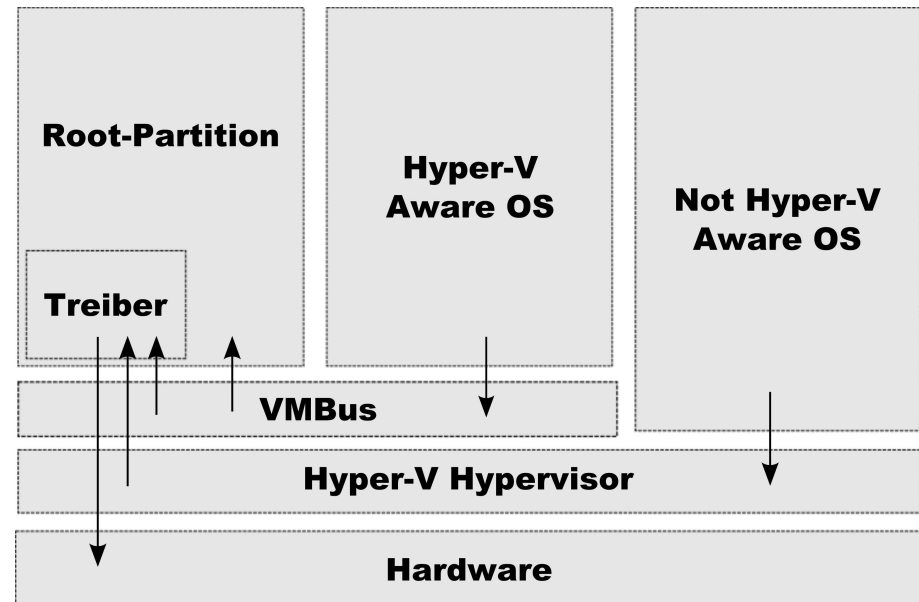
# Verfügbare Lösungen (fort.)

- Xen
  - Bare-Metal-Hypervisor
  - Paravirtualisierung



# Verfügbare Lösungen (fort.)

- Hyper-V
  - Bare-Metal-Hypervisor
  - Para- und Vollvirtualisierung
  - Vollvirtualisierung nur **mit** Hardware-Unterstützung
  - VMBus: Brücke für schnelleren Befehl- und Datenaustausch mit Root-Partition



# Verfügbare Lösungen (fort.)

- VMware
  - Bare-Metal-Hypervisor
  - Vollvirtualisierung auch ohne Hardware-Unterstützung

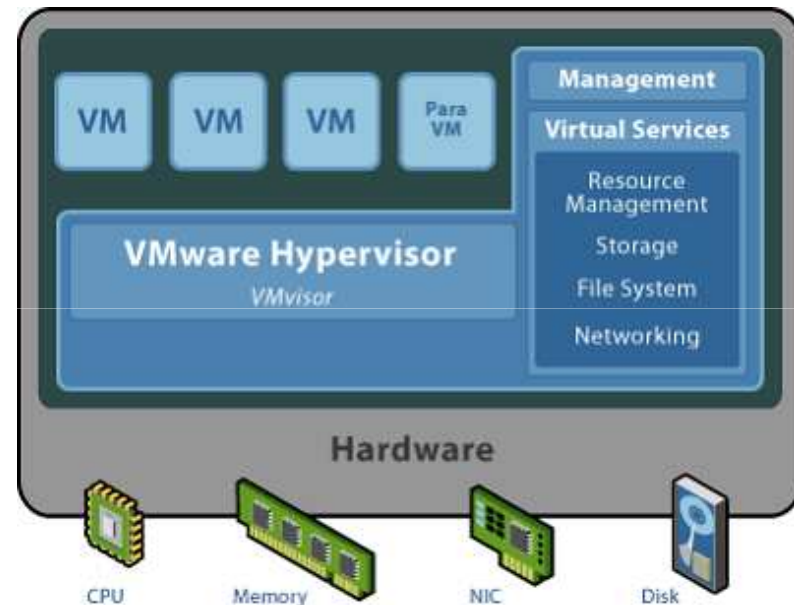


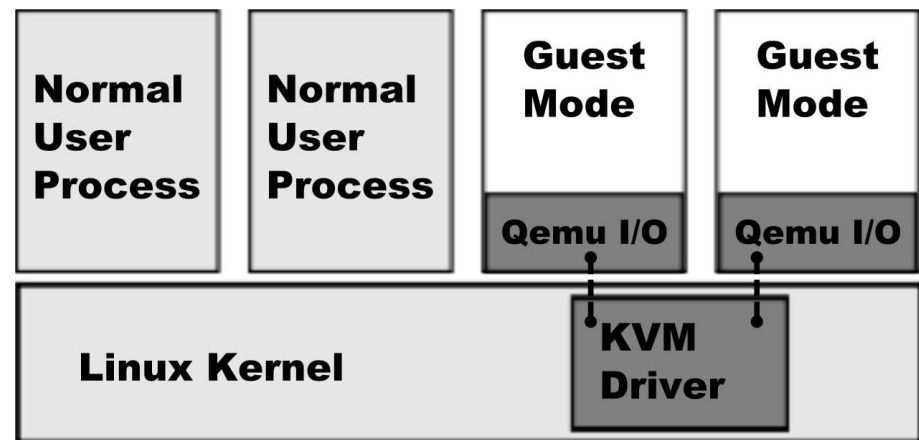
Bild aus Technology Preview for Transparent Paravirtualization [5]





# Verfügbare Lösungen (fort.)

- KVM
  - Kernelmodul (Hypervisor-Typ ?)
  - Vollvirtualisierung nur **mit** Hardware-Unterstützung



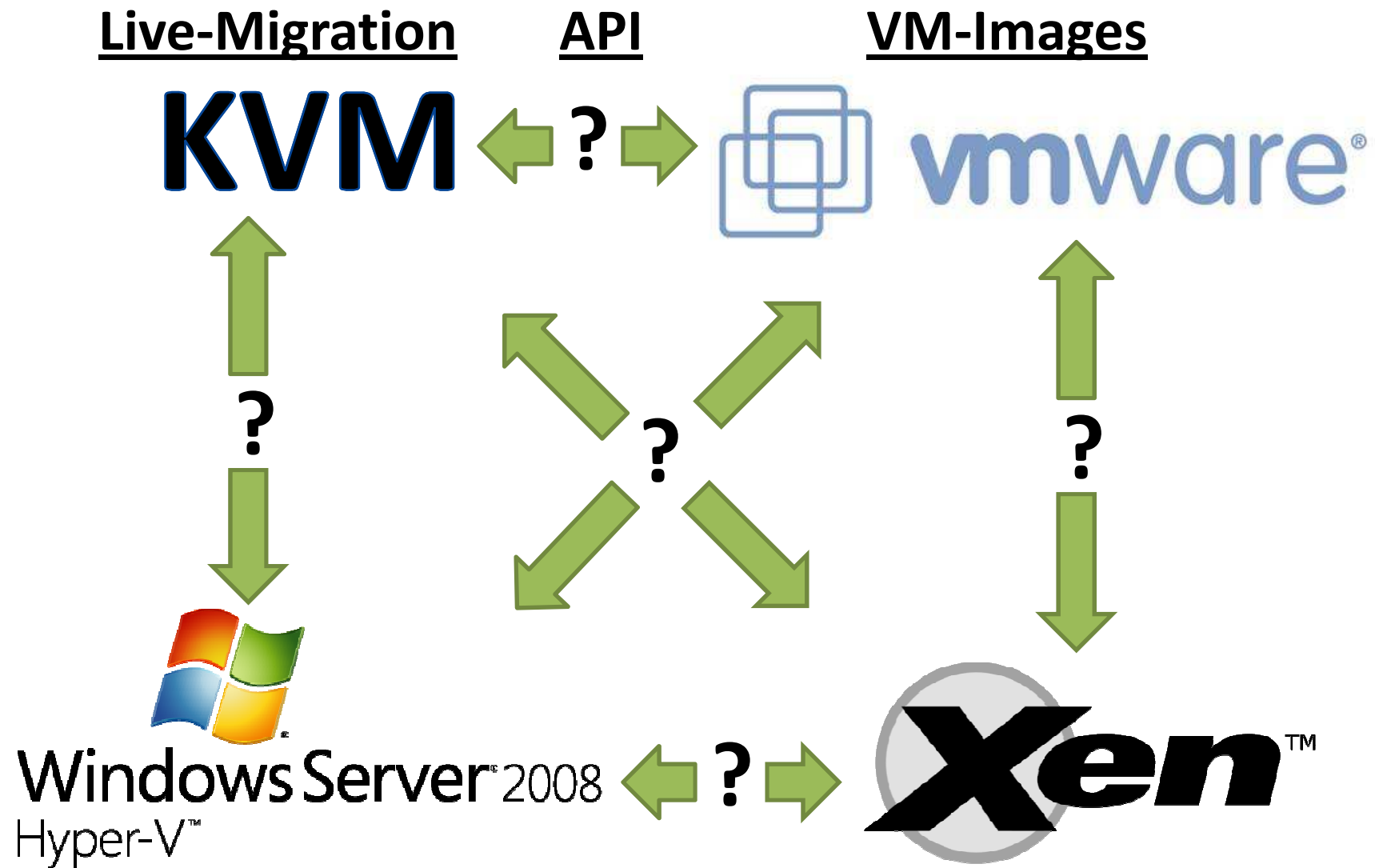
# Verfügbare Lösungen - Zusammenfassung

	Art	Typ
Xen	Bare-Metal-Hypervisor	Para
Hyper-V	Bare-Metal-Hypervisor	Para + Voll*
VMware	Bare-Metal-Hypervisor	Para + Voll
KVM	Linux-Modul	Voll*

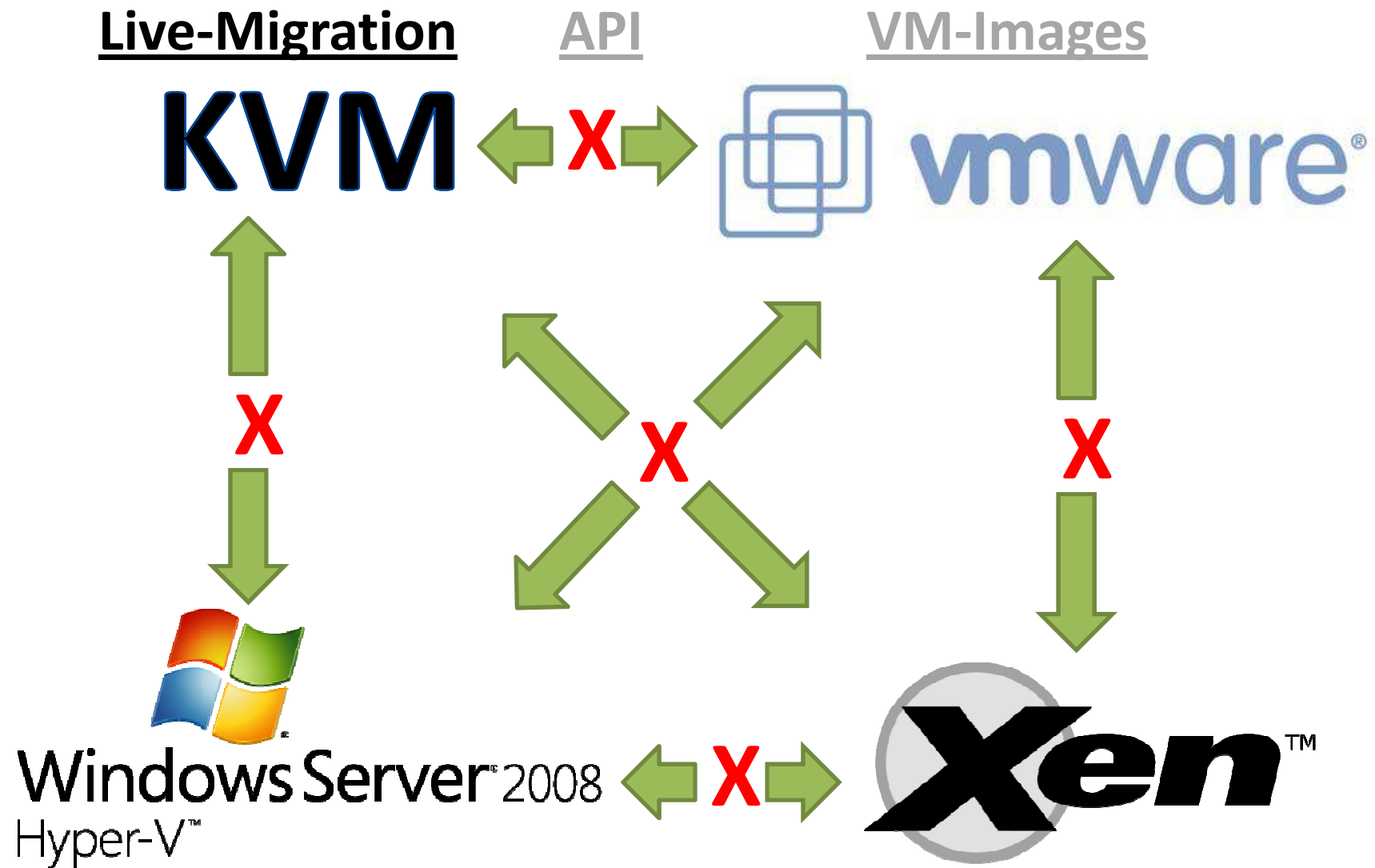
\*) Vollvirtualisierung nur mit Hardwareunterstützung



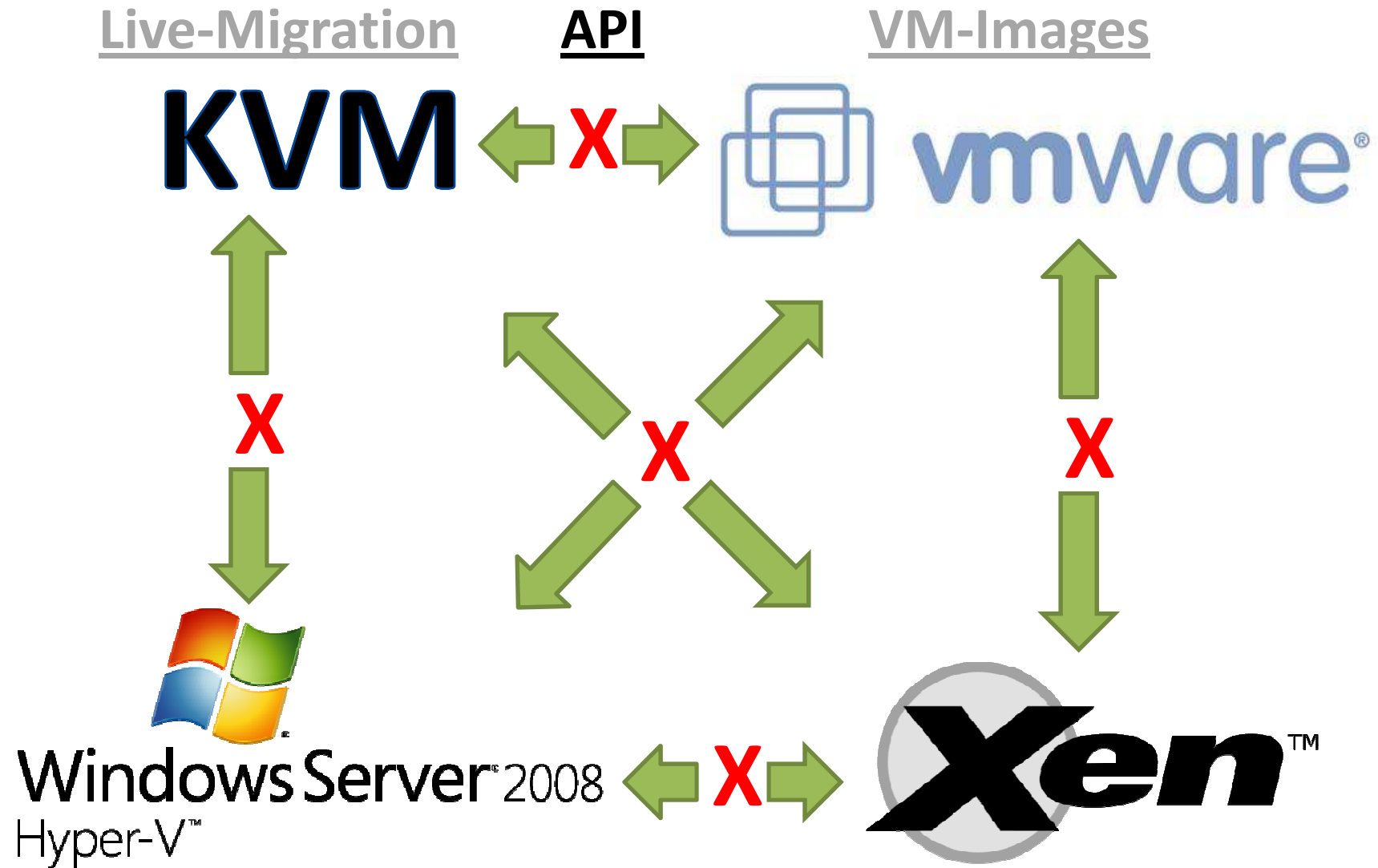
# Kompatibilität



# Kompatibilität



# Kompatibilität



# Kompatibilität

Live-Migration

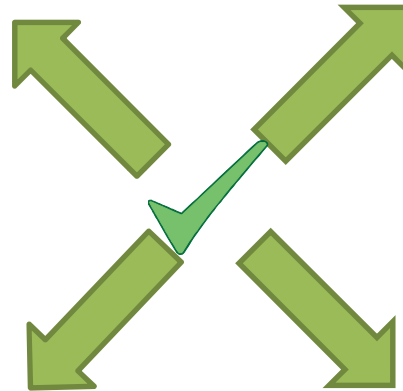
API

VM-Images

**KVM**



**vmware®**



**Windows Server® 2008**  
Hyper-V™



**Xen™**



# Agenda

---

- Cloud
  - Erwartungen
- Virtualisierung in der Cloud
  - Rechnervirtualisierung
- Hypervisors
  - Arten
  - Funktionalitäten
  - Verfügbare Hypervisors
  - Kompatibilität untereinander
- **Fazit**



# Fazit

---

- Virtualisierung für Cloud sinnvoll
  - „Unendlicher“ Ressourcen-Pool
  - Sehr niedrige Down-Time bei Hardware-Wechsel
- Technologien unterschiedlich
  - Jede Lösung speichert ihre VM in einem anderen Format
  - API unterschiedlich
- An Vereinheitlichung wird gearbeitet
  - VMware API





# Vielen Dank für die Aufmerksamkeit

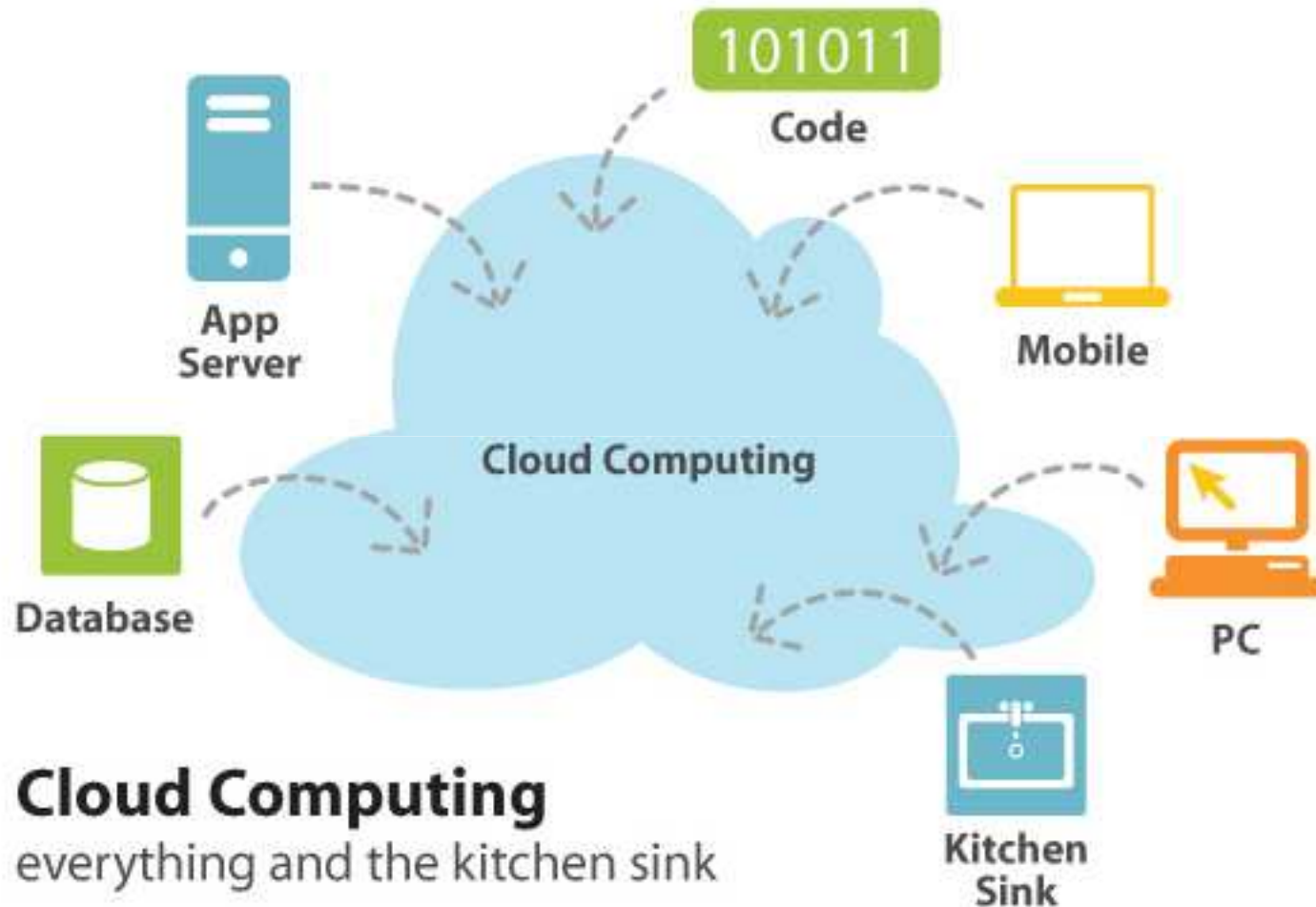


Bild aus "What Cloud Computing Really Means" [4]



# Referenzen

---

[1] Memory Overcommitment in the Real World (2008)

<http://blogs.vmware.com/virtualreality/2008/03/memory-overcomm.html>

[2] Live Wide-Area Migration of Virtual Machines Including Local Persistent State (2007)

VEE '07

[3] Linked clones in VMware View 3 (2010)

[http://www.sysdigg.com/wp-content/uploads/2008/12/linked\\_clone1.png](http://www.sysdigg.com/wp-content/uploads/2008/12/linked_clone1.png)

[4] "What Cloud Computing Really Means" (2009)

<http://lonewolf librarian.wordpress.com/2009/02/24/what-cloud-computing-really-means022409/>

[5] Technology Preview for Transparent Paravirtualization (2006)

<http://www.vmware.com/technical-resources/interfaces/techpreview.html>

