

IT-Systemengineering & -Operations

# **Virtualisierung im DC**

**HW Virtualisierung**

Bruno Joho



# Lernziele

- Der Studierende kennt die technischen Grundlagen wie Server Virtualisierung implementiert ist.
- Der Studierende weis Vor-/Nachteile der virtuellen Maschinen.
- Der Studierende kann sein erlerntes anwenden auf den Unterhalt und Betrieb eines Datacenters.

# Inhalt



- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung

# Virtualisierungsbereiche

- Server
- Speicher
- Applikationen
- Desktops
- Netzwerke (LAN, SAN)

„Virtualisierung ist so vielschichtig, dass es keine allgemein gültige, prägnante Definition gibt. Kern der Virtualisierung ist die Abstraktion von der darunterliegenden Gerätschaft.“

(CHIP online)

# Server Virtualisierungsarten

- **HW Partitionierung**

nur noch auf high end Computer anzutreffen, ist teuer (IBM z und p Systems, Sun/Oracle Sparc).

- **HW Virtualisierung (Hypervisor)**

Gegenstand dieser Vorlesung.

- **OS Virtualisierung**

Junge und sehr populäre sowie „leichtgewichtige“ Virtualisierungsart. Siehe Vorlesung „OS Virtualisierung“

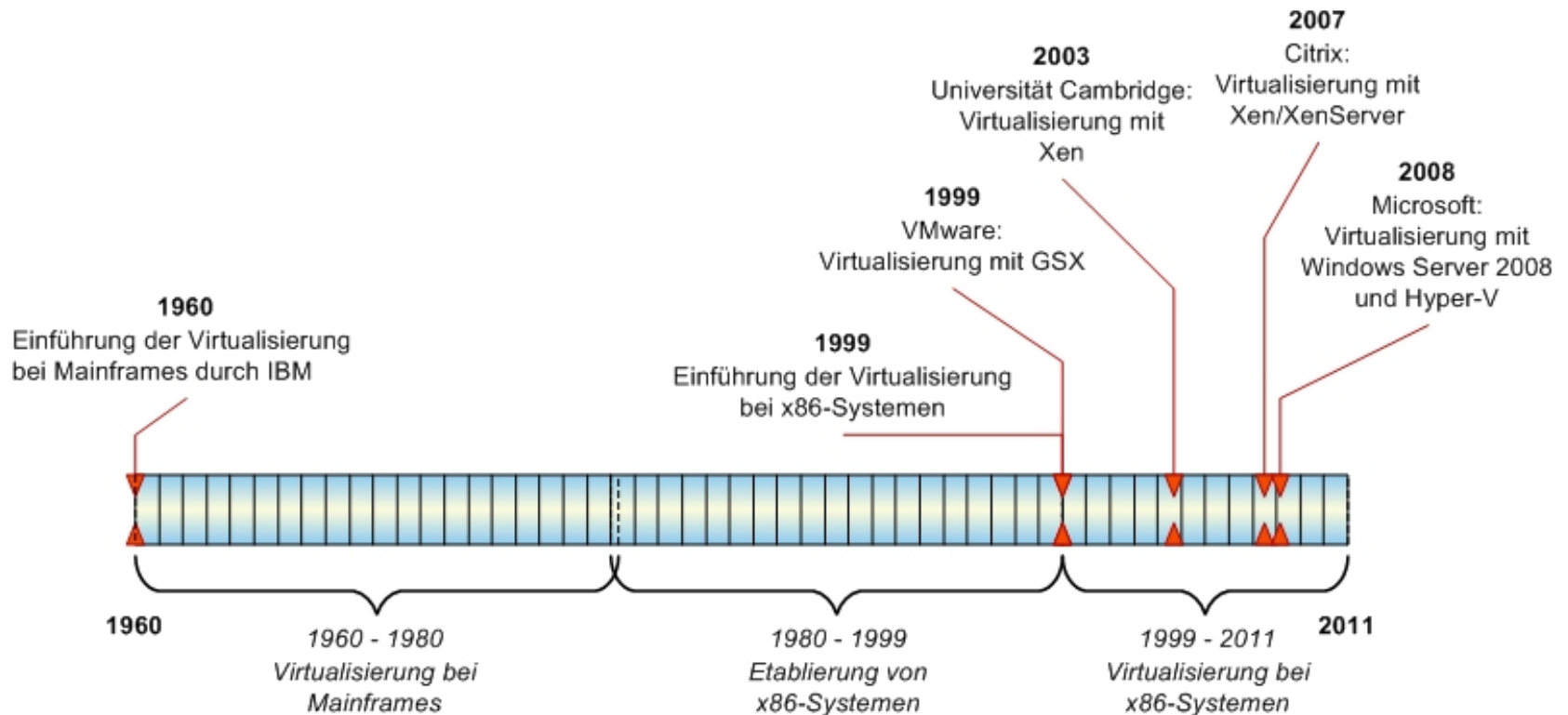
# Gründe für eine HW Virtualisierung

- Ressourcen Optimierung
- Konsolidierung
- Maximierung der Uptime
- Schutz der Applikation von Serverausfällen
- Einfache Migration wenn Anforderungen wechseln
- Schutz der Investitionen in existierende „legacy Systems“

# Inhalt

- Virtualisierung in der IT Branche
- ▪ Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung

# Zeitlinie der Virtualisierung






# Geschichte

- 1959: Christopher Strachey: Konzept über die optimale Ausnutzung von Rechenzeit.
- 1964: IBM und MIT entwickeln die erste virtuelle Maschine.
- 1999: VMware Inc. gelingt die Virtualisierung von Intel Architekturen (binary translation)

# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- ▪ Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung

SIMULATAION

EMULATION



IMITATION

VIRTUALISATION

# Emulation

- Als Emulator (von lat. aemulare, „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.

Wikipedia

# Simulation

- **Simulation** imitiert die Operationen eines real-world Prozesses oder Systems über die Zeit. Der Simulationsvorgang verlangt als erstes die Entwicklung eines Modells. Dieses Modell muss die Schlüssel-Charakteristika (Verhalten/Funktionen) des ausgesuchten abstrakten oder physikalischen Systems resp. des Prozesses repräsentieren. Das Modell repräsentiert das System selber während die Simulation die Operationen über die Zeit repräsentieren.
- Eine **Computer Simulation** ist ein Simulationsprogramm das auf einem einzelnen Computer oder einem Netzwerk von Computern läuft um das Verhalten eines Systems zu reproduzieren. Die Simulation verwendet ein abstraktes Modell (ein Computer Modell oder ein Rechenmodell) um das System zu simulieren.

# Virtualisation

**Virtualität** ist die Eigenschaft einer Sache, nicht in der Form zu existieren, in der sie zu existieren scheint, aber in ihrem Wesen oder ihrer Wirkung einer in dieser Form existierenden Sache zu gleichen.

**Virtualisierung** in Computing, bezieht sich auf den Vorgang der Erstellung einer virtuellen (und nicht tatsächlichen) Version von etwas wie zB. virtuelle Computer-Hardware-Plattform, Betriebssystem (OS), Speichergerät oder Computer-Netzwerk-Ressourcen.

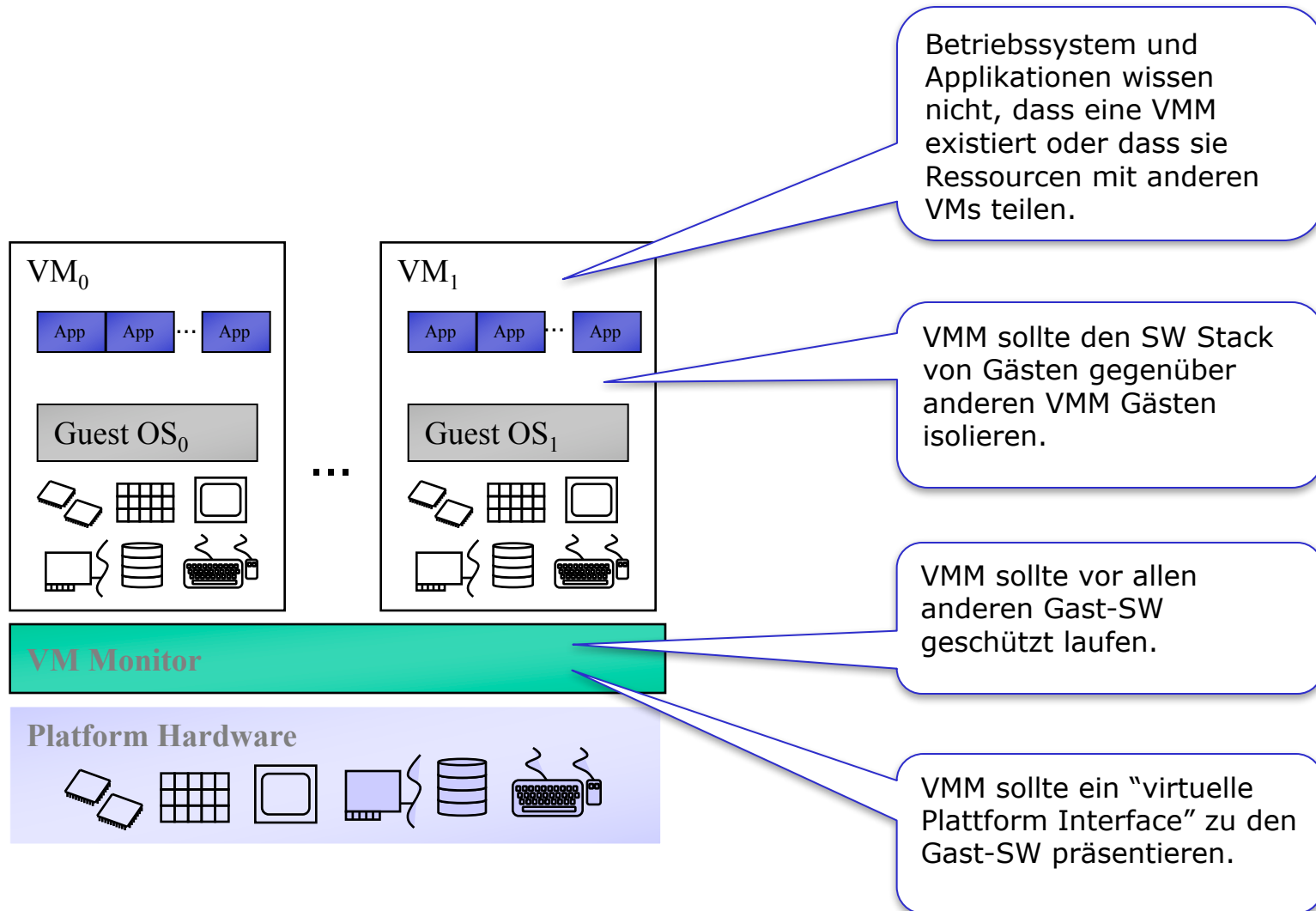
Der Begriff "Virtualisierung" hat ihre Wurzeln bei den Mainframes der 1960er Jahre, in denen die logische Unterteilung der Mainframes Ressourcen für unterschiedliche Anwendungen angewendet wurde. Seitdem hat sich die Bedeutung des Begriffs weiter entwickelt

# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung



# Herausforderungen beim Betrieb einer VMM





# Formale Definition eines Hypervisors

Popek und Goldberg haben 3 Anforderungen an ein physisches und virtuelles System gestellt:

## ▪ Gleichheit

- Jedes Programm ausgeführt in einer virtuellen Maschine verhält sich genau so wie auf der original Maschine direkt ausgeführt.

## ▪ Effektivität

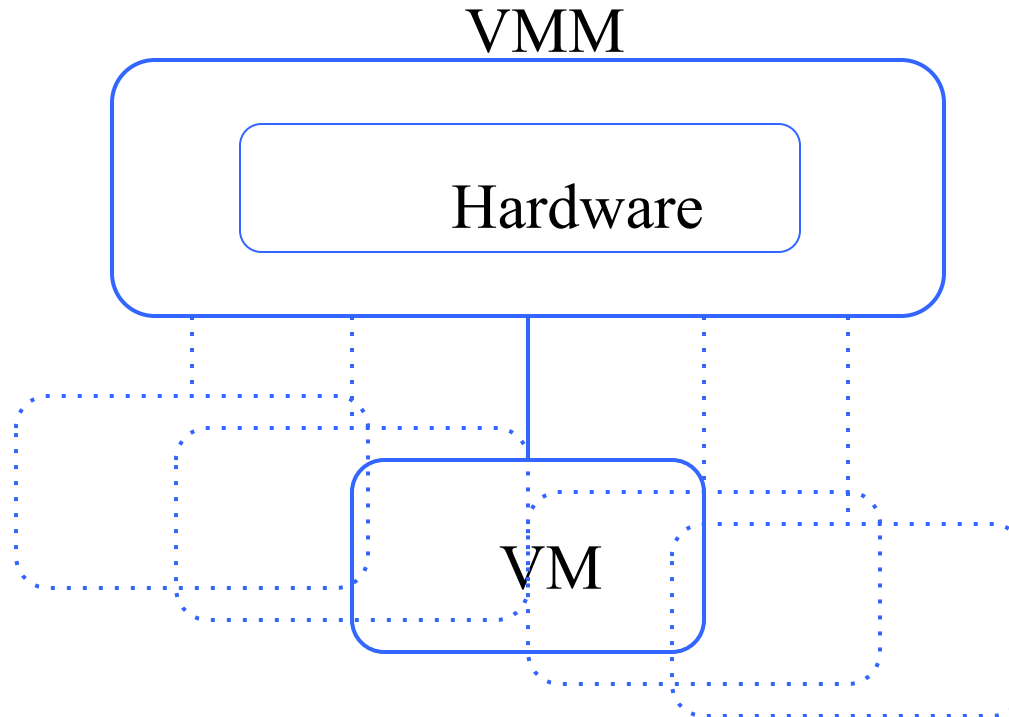
- Wann immer möglich sollten die Instruktionen nicht auf dem virtuellen Prozessor sondern direkt auf dem physischen Prozessor ausgeführt werden (ohne Intervention von der VMM).
- *harmlose\* Instruktionen* werden von der Hardware direkt ausgeführt.

## ▪ Ressourcenkontrolle

- VMM hat die komplette Kontrolle über Ressourcen wie Memory, I/O der Peripheriegeräte, aber nicht unbedingt über Prozessoraktivität.
- Es muss für jedes Programm unmöglich sein die System Ressourcen zu beeinflussen (z.B. verfügbares Memory). Der Verteiler (Allocator) des Kontroll-Programmes muss bei jedem dieser Versuche aufgerufen werden.

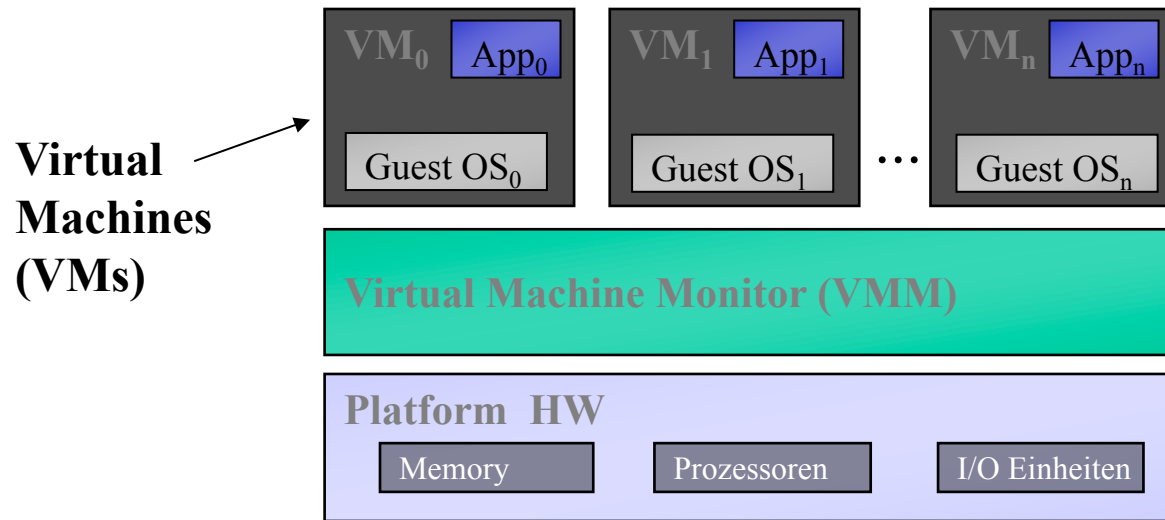
\* harmlose Instruktionen: „Instruktionen die nicht privilegiert sind“ (z.B. kein Memory allozieren)

# Virtual Machine Monitor (VMM)



1. VMM stellt Umgebung zur Verfügung die identisch ist mit jener der original Maschine.
2. Programme die in dieser Umgebung laufen zeigen im schlechtesten Falle eine geringe Geschwindigkeitseinbuße.
3. Die virtuelle Maschine ist die Umgebung erzeugt vom VMM.

# Virtual Machine Monitors (VMM)



VMM ist eine Schicht von System Software und ermöglicht VMs sich eine Hardware Plattform zu teilen. Er besteht aus einem Control Program mit 3 Teilen:

1. **Dispatcher** dessen Initial Instruction am Speicherplatz liegt wohin die HW trapped.
2. **Allocator** der entscheidet wer welche Systemressourcen bekommt. Er hat 1 oder n Member (VM)
3. **Interpreter** für alle Instruktionen die trappen, eine Interpreter Routine pro privilegierte Instruktion.

# Instruktionsklassifikation

- Existenz mindestens von 2 **Betriebsmodi**

- Uneingeschränkter (Supervisor) Modus
- Eingeschränkter (User) Modus

- **Privilegierte Instruktionen**

Trap wenn Prozessor im User Modus, kein Trap falls im System Modus (Supervisor Modus).

- **Kontrollkritische Instruktionen (control sensitive)\***

Jene die versuchen die Konfigurationen von Systemressourcen zu verändern.

- **Verhaltenskritische Instruktionen (instruction sensitive)\***

Jene Instruktionen welches Verhalten oder Resultat abhängig ist von der Konfiguration der Ressourcen (Inhalt der Relocation Register oder Prozessor Modus).

- **Harmlose Instruktionen** sind alle nicht sensitiven Instruktionen.

\* Müssen laut Popek & Goldberg eine Untermenge der Privilegierten Instruktionen sein.

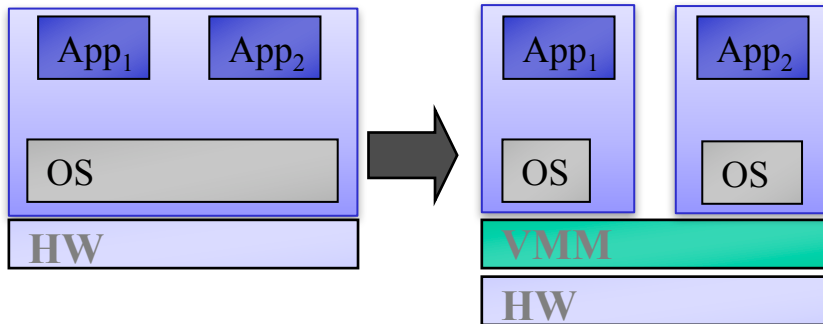
# Trap

- Wenn die Instruktion trapped wird das Memory nicht verändert, außer die Memory Stelle  $E(0)$  in welche das aktuelle PSW (vor dem Trap der Instruktion) geschrieben wird.
- Ein Trap speichert den momentanen Status der Maschine in  $E(0)$  und übergibt die Kontrolle einer Routine die „ihre“ Werte aus der Memory Stelle  $E(1)$  liest

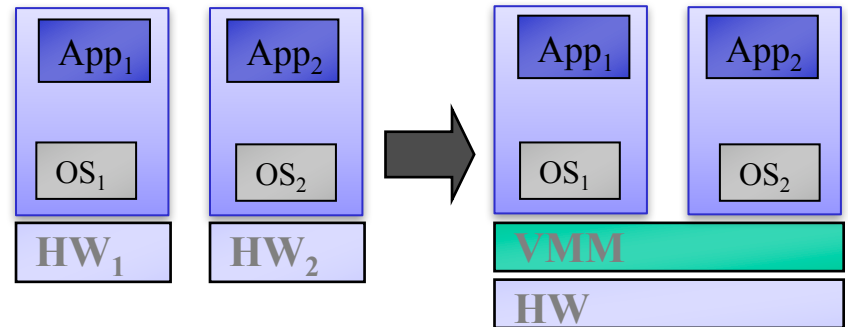


# Vorteile der (Hypervisor) Virtualisierung

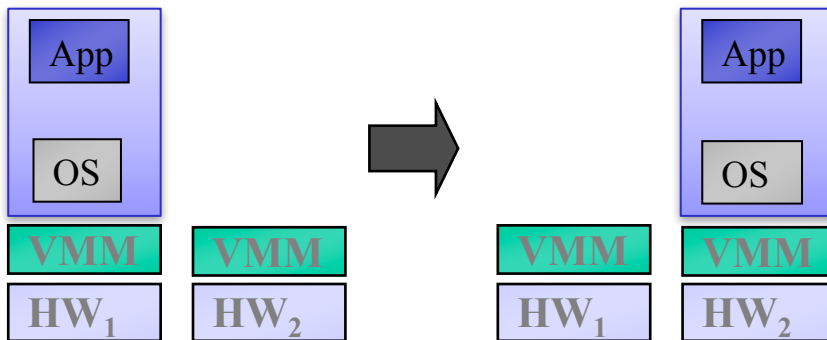
## Workload Isolation



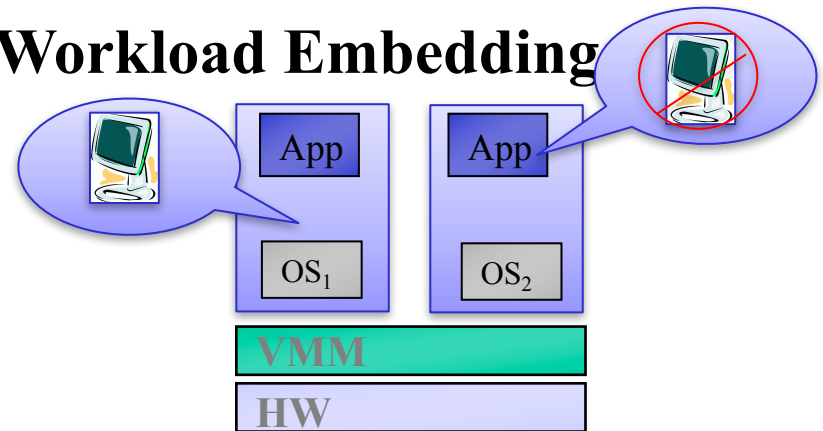
## Workload Consolidation



## Workload Migration



## Workload Embedding



**Virtualisierung hat sehr mächtige Möglichkeiten.**

# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung



# Sind x86 virtualisierbar?

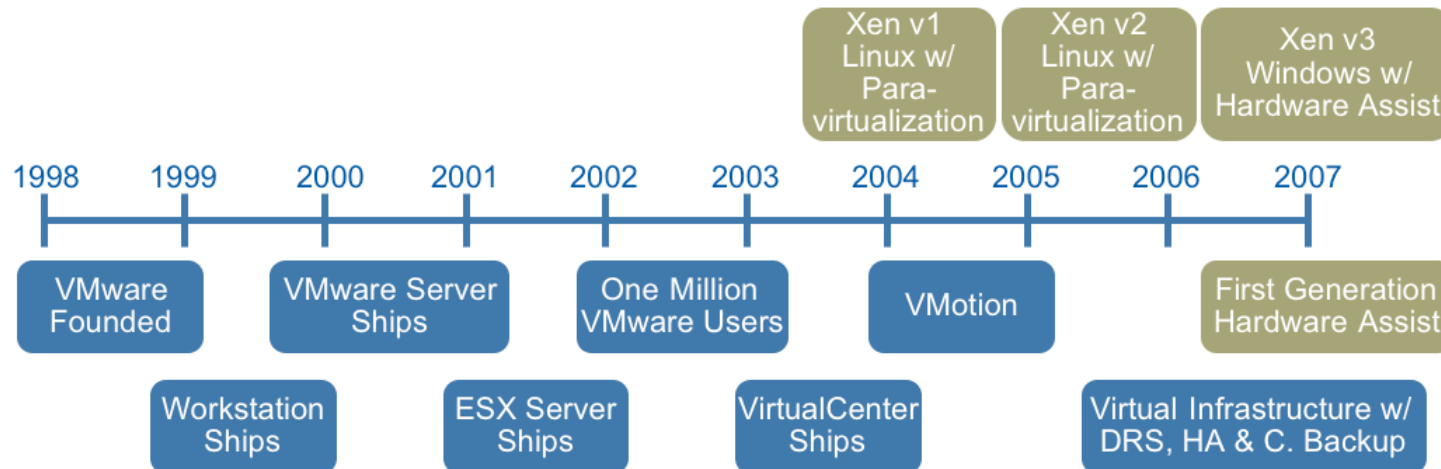
Nach Popek und Goldberg ist eine CPU (ISA) virtualisierbar, wenn alle privilegierten Instruktionen eine Exception erzeugen, wenn sie in einem unprivilegierten Prozessormodus ausgeführt werden.

Alle sensiblen Instruktionen sind privilegiert



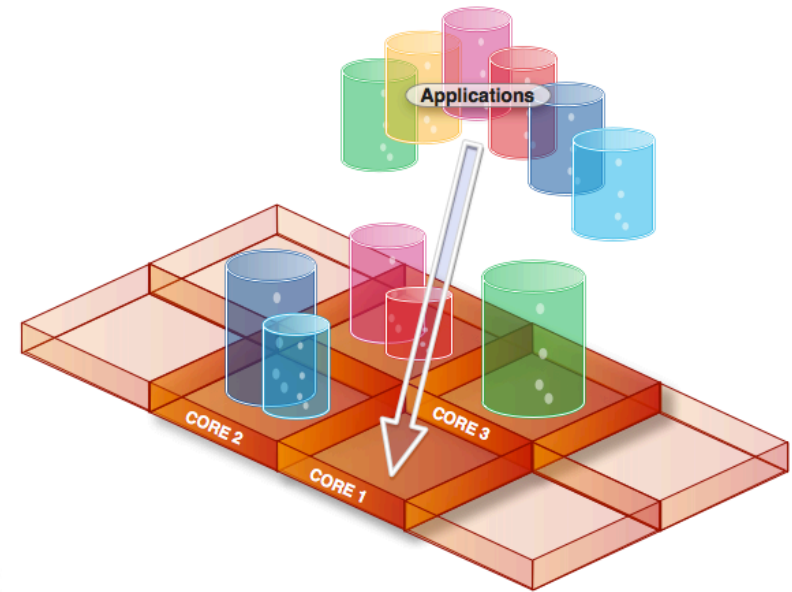
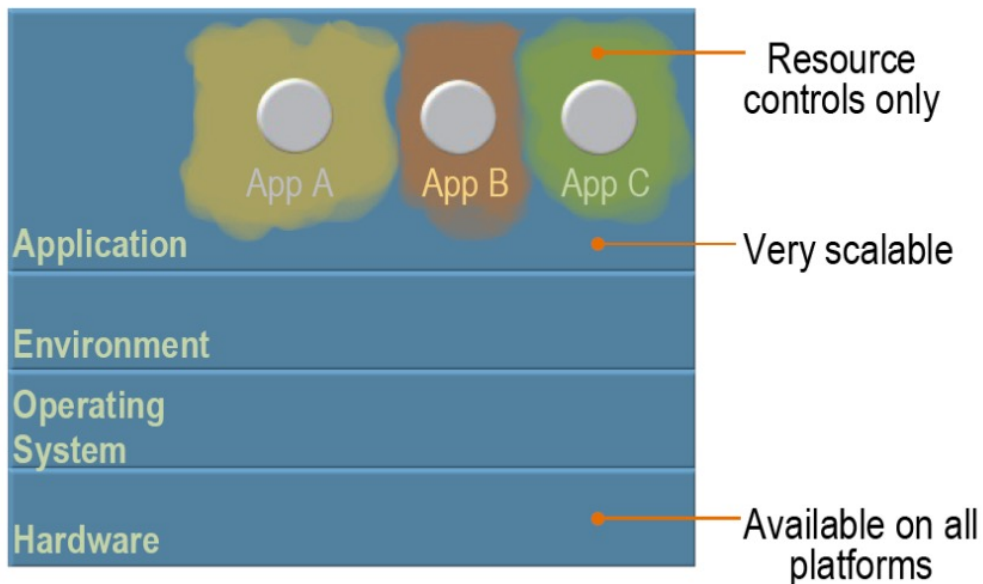
# X86 Virtualisation

- Ressourcen Management
- Binary Translation
- Memory Management
- etc



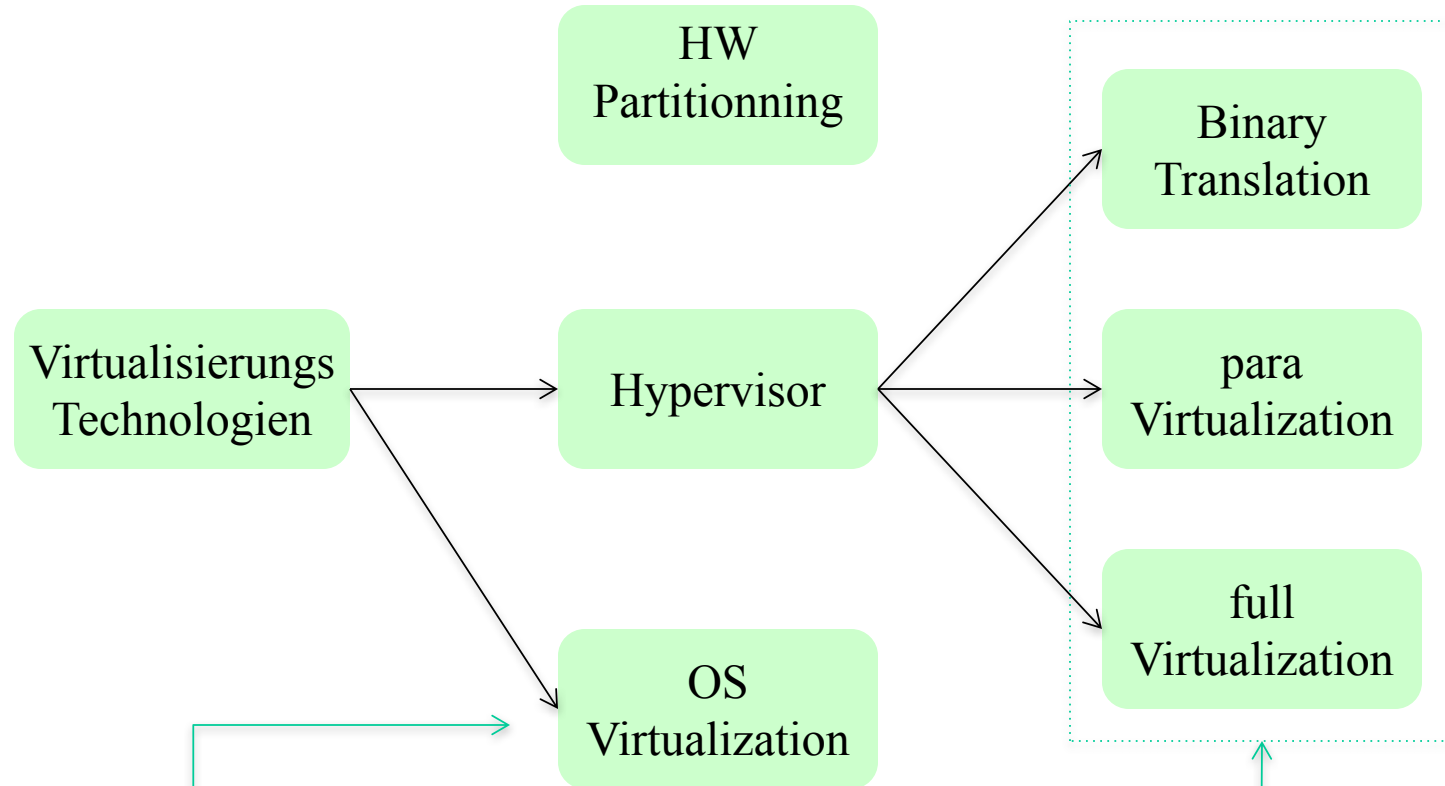
# Ziel: Ressourcen Management

Fortgeschrittene Betriebssysteme erlauben ein feingranulares Betriebsmittel Management.



Zuteilung von Ressourcen an Prozesse und OS Instanzen

# Gast OS Interaktion mit Hypervisor



Vorlesung: *OS Virtualisierung*

Diese Vorlesung

# Problem

## **Problem:**

- Mehrere Betriebssysteme gleichzeitig auf einem physikalischen Server.
- Betriebssysteme sind konstruiert dass sie die Zugriffe auf die Betriebsmittel regeln.
- Betriebssystem kann nicht abgeändert werden (Microsoft)

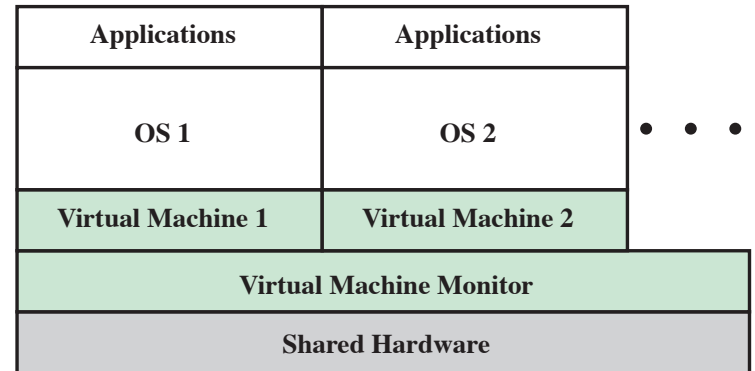
## **Lösung:**

- Hypervisor verwaltet die Betriebsmittel
- Betriebssystem ist isoliert
- Betriebssystem erfährt keine Änderung (ausser bei Paravirtualisierung)

# Hypervisors

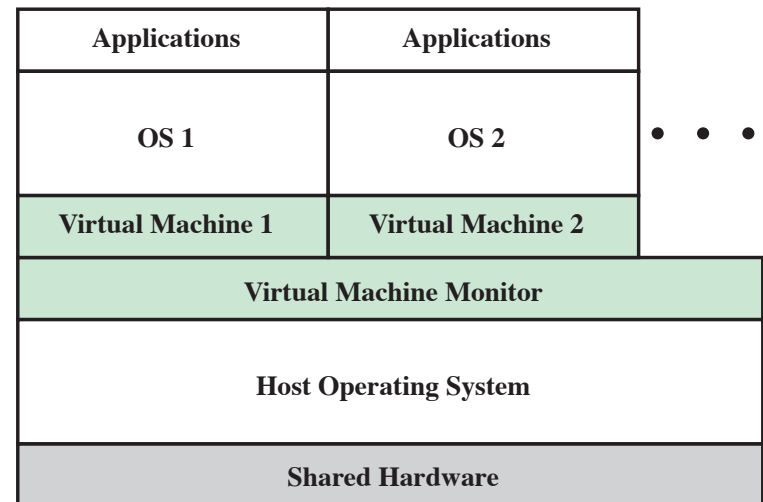
**Typ 1 Hypervisor.** Direkt auf der HW installiert.

Beispiele: VMware ESXi, XEN, Hyper-V



**Typ 2 Hypervisor.** Installiert auf einem „Host“ Betriebssystem.

Beispiele: VMware Workstation, KVM



# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung

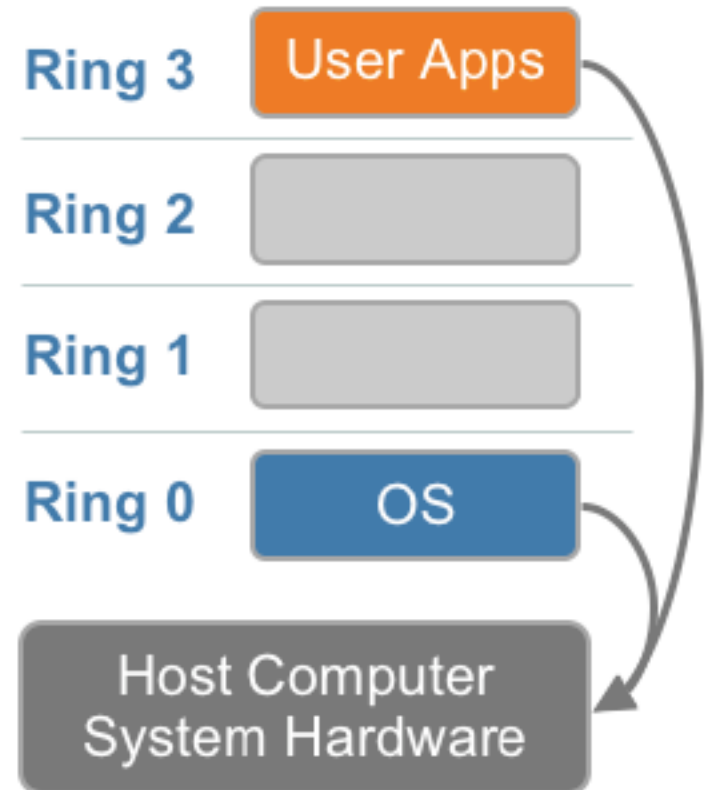


# Direct Execution

- x86 kennen 4 Levels von Privilegien.
- User Level Programme laufen nicht Privilegiert  
    ➡ Ring 3
- Das OS braucht HW Zugriff  
    ➡ Ring 0
- Ein Virtualisierungsschicht muss unter Ring 0 gelegt werden!

oder..

- Es braucht eine Schnittstelle vom Hypervisor mit speziellem Befehlssatz



# Was sind Sicherheitsringe?

CPUs hat die Möglichkeit sicherzustellen, dass die Instruktion auch das entsprechende Privileg hat.

**CPL (current privilege level)**

**2bit = 4 Privilegien = 4 Sicherheitsringe**

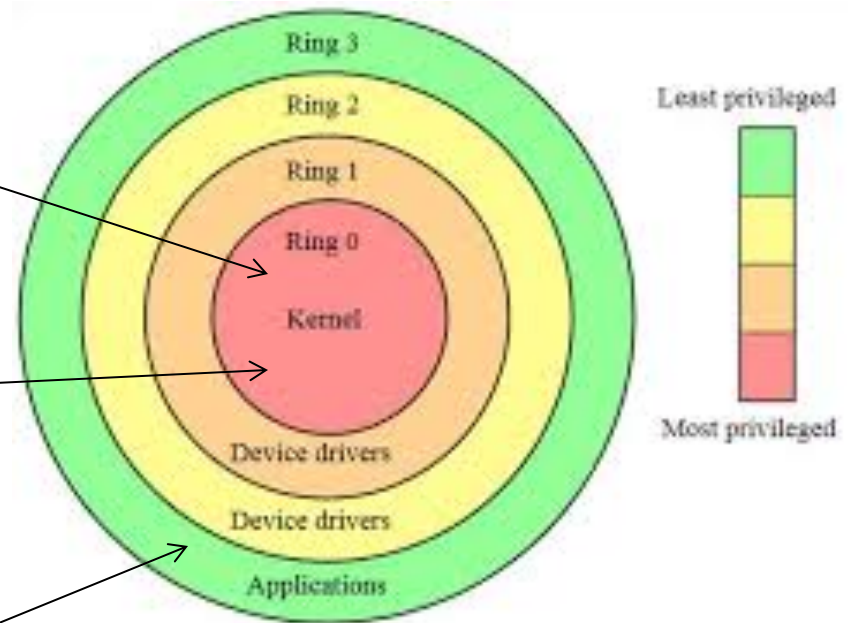


# Wo wird der Hypervisor platziert?


Hier müsste der Hypervisor platziert werden.

Hier will der Kern vom OS laufen.

Applikationen im Betriebssystem laufen im Ring 3.



# Inhalt

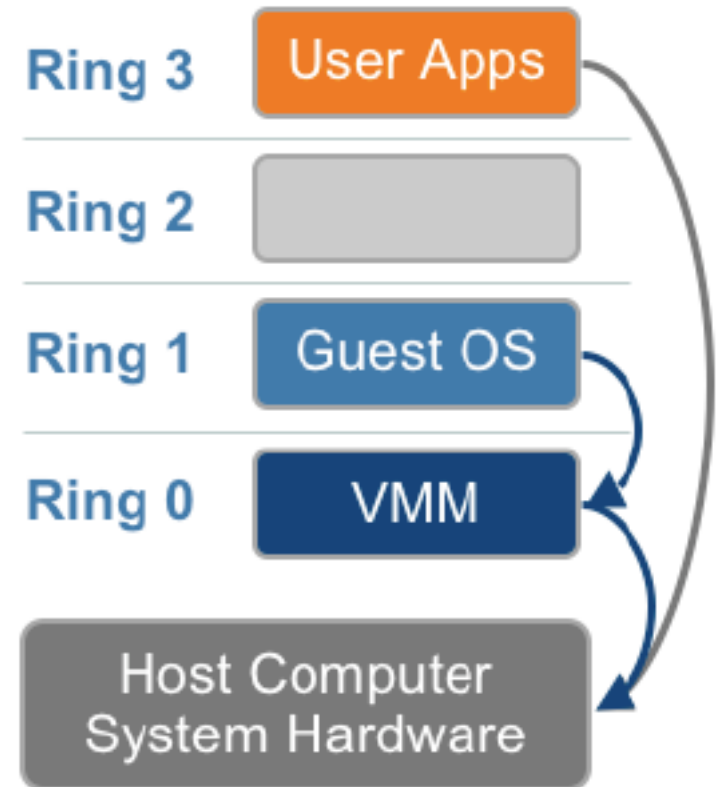
- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  -  • Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung

# x86 ISA Stealth Instructions

- Im IBM Mainframe löst jede privilegierte Instruktion einen „Trap“ aus. Z.B. bei Instruktionen die eine „Ressourcen Management“ Instruktion in einem niedrig Privilegierten Ring absetzen wollte. Die VMM fängt diese Instruktion ab und emuliert diese Instruktion ohne die anderen Gäste damit zu beeinträchtigen.
- Im x86 ISA sind nicht alle privilegierten Instruktionen Trap geschützt. Z.B. POPF (disables and enables interrupts). Wird diese Instruktion von einem Gast im Ring 1 ausgeführt interessiert das die x86 CPU überhaupt nicht und ignoriert sie.
- x86 kennt 17 solcher „nicht abfangbarer, für x86 getarnte“ Instruktionen.
- Abhilfe schaffte VMware's Patent Nr. US 6704925 B1, eingereicht am 1. Dezember 1998 „*Dynamic binary translator with a system and method for updating and maintaining coherency of a translation cache*“

# Full Virtualization mit Binary Translation

- Kombination zwischen direkter Ausführung und binary translation.
- User Level Code wird direkt ausgeführt.
- Nicht ausführbarer kontrollkritischer Code wird übersetzt (nur beim ersten mal) und im Cache abgelegt.
- Übersetzter Kernel Code ersetzt nun nicht virtualisierbare Instruktionen.
- Keine HW Unterstützung nötig.
- Gast OS ist voll abstrahiert, hat keine Ahnung dass es in einer VM läuft.



# Binary Translation Overhead

- Ein „native System Call“ braucht 242 Prozessor Zyklen
- Ein Binary Translated System Call eines 32 bit Gast OS im Ring 1 braucht 2308 Zyklen.

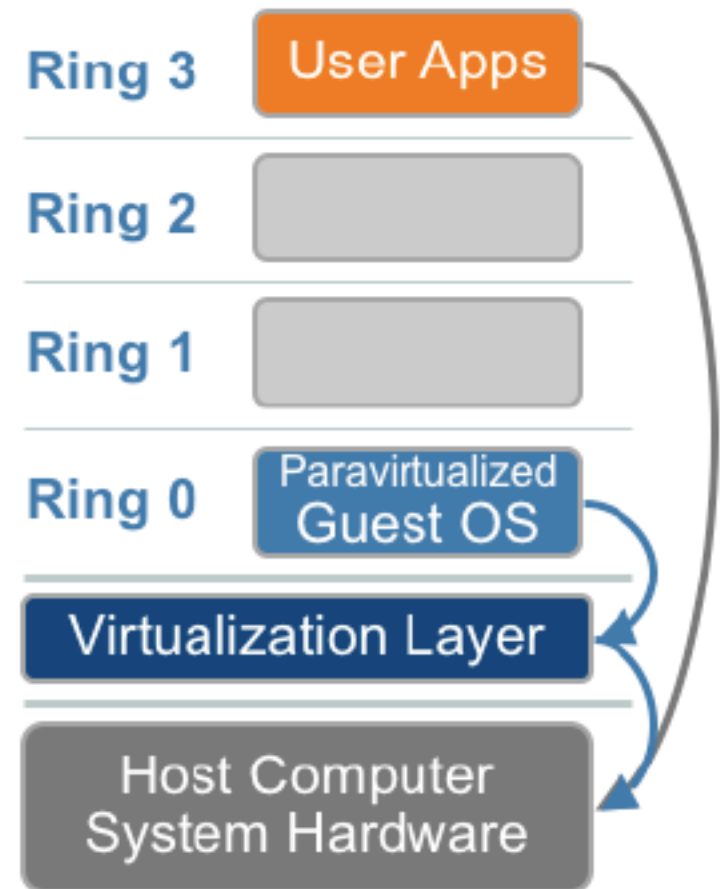
# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung



# OS Unterstützte (Para) Virtualisierung

- Para (griechisch = neben) bezieht sich auf Kommunikation zwischen Gast OS und Hypervisor um die Geschwindigkeit zu erhöhen.
- Kernel vom OS muss modifiziert werden um nicht virtualisierbare Instruktionen mit Hypercalls zu ersetzen die direkt mit dem *Virtualization Layer Hypervisor* kommunizieren.
- Der Hypervisor stellt Hypercall Interfaces zur Verfügung.
- Keine unmodifizierte Gast OS!



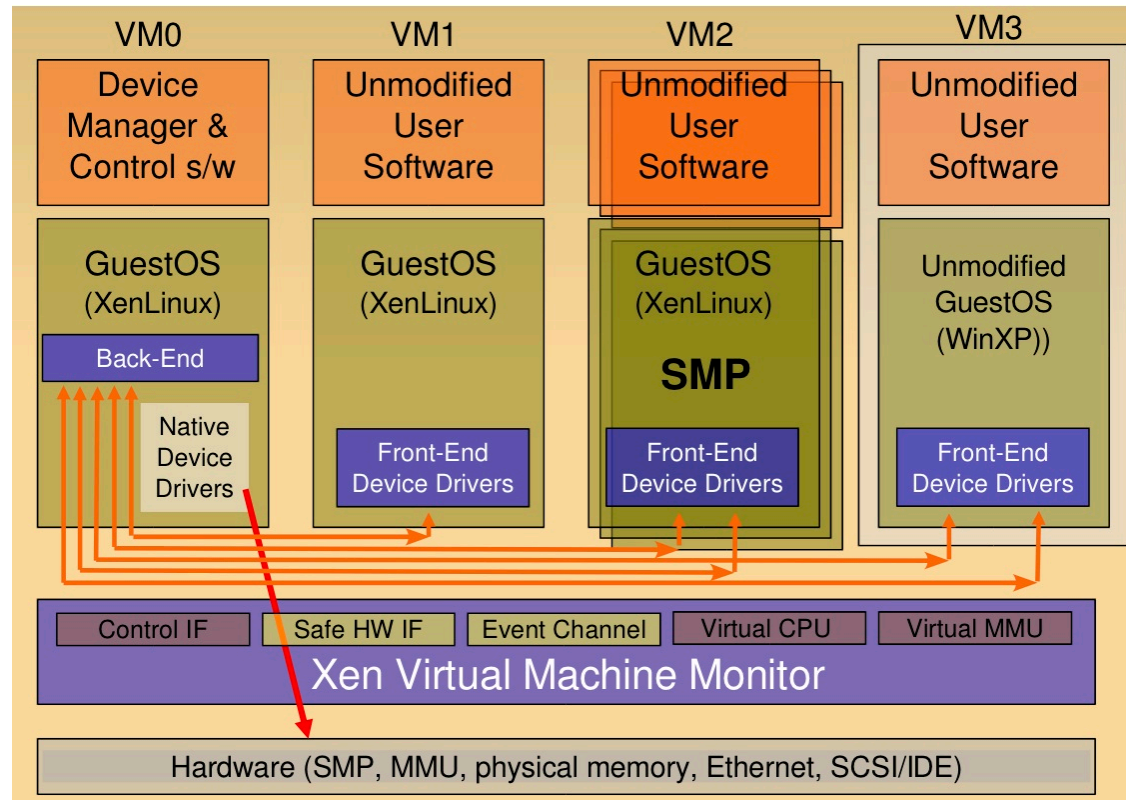
## Para Virtualisierung (cont.)

- Paravirtualisierung ist nicht so verschieden von Binary Translation.
- BT übersetzt „kritischen“ in „harmlosen“ Code.
- Paravirtualisierung macht dasselbe aber im Source Code.
- Änderungen im Source Code erlaubt grössere Flexibilität.
- Para Virtualisierung braucht keine „Laufzeit-Übersetzung“ und wird daher schneller ausgeführt.

Unveränderte Betriebssysteme können nicht ausgeführt werden.



# Vereinfachte Front-End Driver



Der Hypervisor stellt Interfaces für kritische Kern Operationen zur Verfügung. I/O Devices in der VM sind „nur“ Pointer zu reellen „native“ Drivers in einer privilegierten VM (genannt Domain 0). Keine Emulation oder Übersetzung erforderlich!

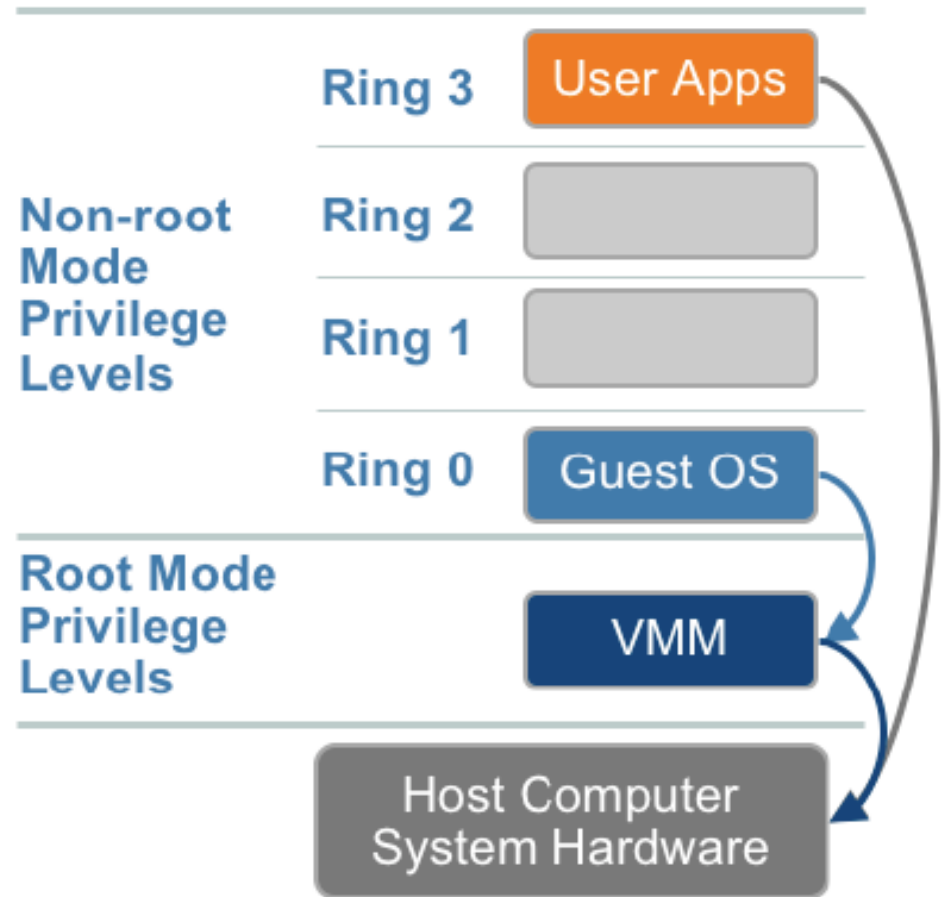
# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
- Memory Virtualisierung



# HW unterstützte Virtualisierung


- Neuer CPU Modus eingeführt.
- Erlaubt Ausführung unterhalb Ring 0.
- Privilegierte und sensitive Calls trafen automatisch zum Hypervisor ohne binary translation.
- Der Gast Status wird in *virtual machine control structures* abgespeichert.
- CPUs seit 2006 erhältlich.



# Wie löst Intel das Problem?

- VMM und VM haben getrennte Adressbereiche
- VMCS (Virtual Memory Control Structure) sind Kontroll Strukturen im Memory.
- VMCS verfolgt die Kontextwechsel Prozessorinformationen für VMs
- Neue Intel IA32 Instruktionen:
  - 2 neue Operationsmodi (die in Ring 0-3 vorhanden sind):
    - VMX root Operation:
      - Voll Privilegiert für den VM Monitor
    - VMX non-root Operation:
      - Nicht voll privilegiert für die Gast Betriebssysteme
      - Reduziert Gast SW Privilegien ohne auf die Ringe zurückgreifen zu müssen.

# Inhalt

- Virtualisierung in der IT Branche
- Geschichte der Virtualisierung
- Was bedeutet Virtualisierung?
- Formale Definition
- X86 Virtualisierung
- HW Virtualisierung (Hypervisor)
  - Direct Execution & Sicherheitsringe
  - Binary Translation
  - Para Virtualization (SW Assist)
  - Full Virtualization (HW Assist)
-  • Memory Virtualisierung

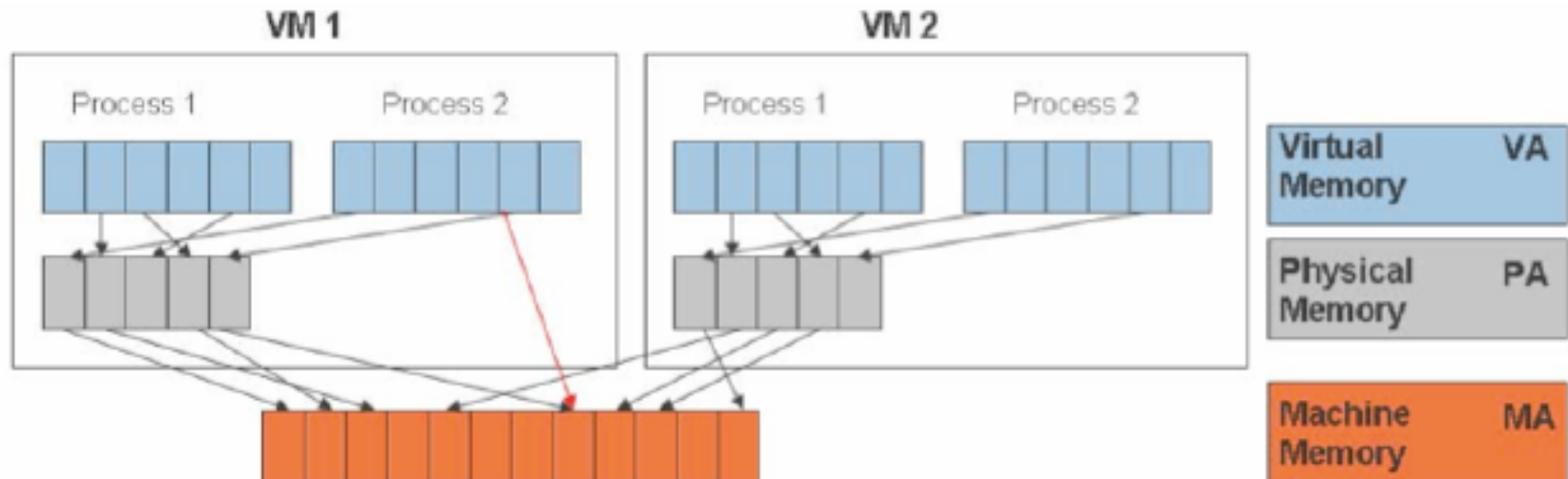
# Memory Virtualisation

## ohne Hypervisor

- Applikation sieht kontinuierlichen Adressbereich.
- OS mapped virtual Page Nummer zu physikalischen Page Nummer

## mit Hypervisor

- mehrere Gast OS auf einem Hypervisor benötigen eine weitere Memory Virtualisierung.
- Das Gast OS hat keinen direkten Zugriff mehr auf das Memory.



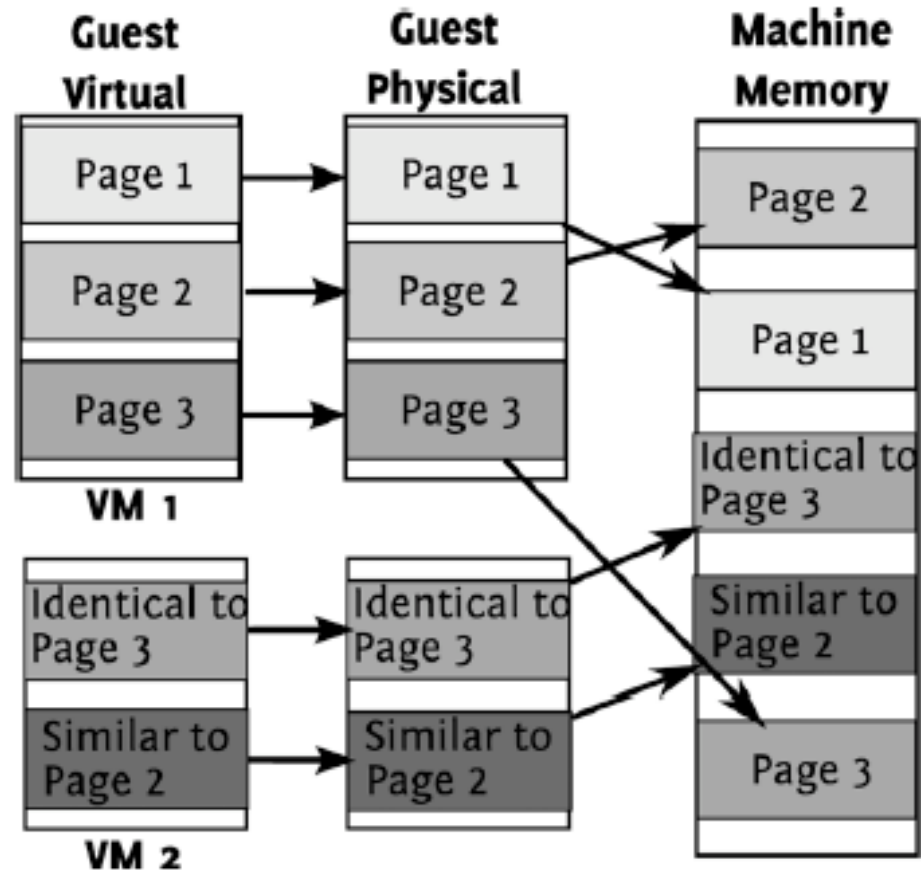
# Problem: Memory Verbrauch

3 Möglichkeiten um Memory zu sparen:

- Page Sharing
- Page Patching
- Page Compression

Eine Möglichkeit Memory freizugeben:

- Balloning



Ausgangslage: 2 VMs, 5 Pages

# Memory Rückgewinnungs-Technologien

- **Page Sharing**

In homogenen Gast Systemen finden sich viele identische Pages.

- **Page Patching**

*fast gleiche* Pages gibt es sehr viele.

- **Page Compress**

Viele Pages die in naher Zukunft nicht verwendet werden.

- **Balloning**

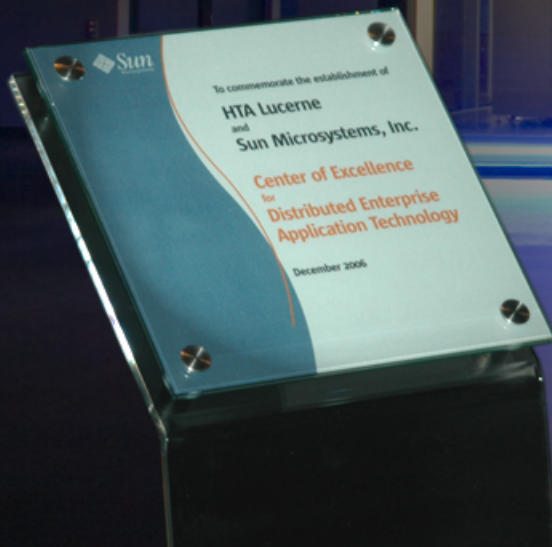
Wenn das Memory vom Gast OS knapp wird...



Lucerne University of  
Applied Sciences and Arts

**HOCHSCHULE  
LUZERN**

Engineering & Architecture



Questions?