Science Days 2010

Leipzig, 02. - 03. November 2010

Prof. Dr. -Ing. E. Eren
www.inf.fh-dortmund.de/eren
www.lisa-fh-dortmund.de

eren@fh-dortmund.de

Virtualisierung

Virtualisierung

- Virtualisierung von Betriebssystemen ist seit geraumer Zeit etabliert.
- Doch mit Hilfe moderner Virtualisierungssoftware bestehen heute Möglichkeiten, sogar komplexe IT-Infrastrukturen und deren Komponenten zu virtualisieren:
  - Subnetze
  - Router
  - Switches
  - Firewalls
  - DMZ
  - etc.

Projekt VISA

- Im Projekt "**VISA - Virtual IT-Security Architectures**" an der FH-Dortmund wurde auf Basis von „Open Source“-Lösungen eine komplexe IT-Sicherheitsinfrastruktur für mittelständische Unternehmen virtualisiert.
- Hierbei kamen zum Einsatz:
  - KVM als Hardwarebasierte Vollvirtualisierungslösung
  - VDE zum Aufbau von virtuellen Netzwerken
  - Vyatta als Router-Distribution

Hardwarebasierte Vollvirtualisierung

Vollvirtualisierung

- Bei der Vollvirtualisierung (auch „echte“ Virtualisierung genannt) wird die Hardware des Hostsystems in mehrere VMs aufgeteilt.
- Die Gastsysteme innerhalb der VMs entsprechen weitestgehend der Architektur des Hostsystems und laufen getrennt voneinander – jeweils mit ihrem eigenen Betriebssystem.
- Jedem Gastsystem wird eine standardisierte Hardware bereitgestellt. Die VMs greifen somit nicht direkt auf die physikalische Hardware des Hosts zu.

- Die Vollvirtualisierung lässt sich noch unterteilen in die Verfahren
  - Paravirtualisierung
  - Hardwarebasierte Virtualisierung sowie
  - Emulatoren

Hardwarebasierte Virtualisierung

- Hardwarebasierte Virtualisierung kombiniert Techniken der Voll- und Paravirtualisierung wobei die Virtualisierungsfunktionalität in die Prozessorhardware integriert wird.

- Der allgemeine Trend zur Virtualisierung brachte Hardwarehersteller dazu, ihre neuen x86-Prozessoren um sog. Virtualisierungsfunktionen zu erweitern.

- Die Grundfunktionalität dieser Erweiterungen besteht in der Erkennung und gesonderten Behandlung von kritischen Operationen.
- Diese modernen Prozessoren unterstützen die Interaktion zwischen den VMs und dem Hypervisor.

Hardwarebasierte Virtualisierung

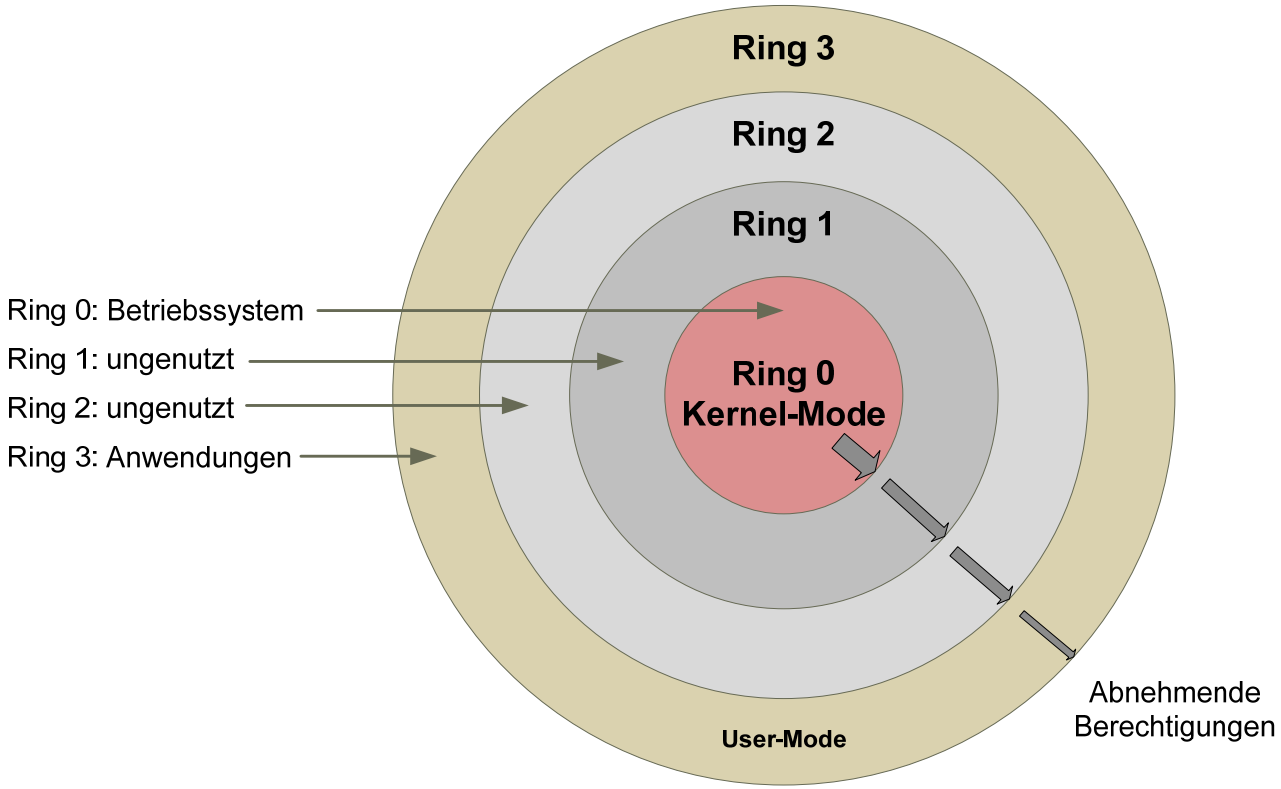
- Hardwarebasierte Virtualisierung ist prinzipiell identisch zu Vollvirtualisierung. Jedoch liegt ein entscheidender Unterschied darin, dass das Privilegiensystem der neuen Prozessoren erweitert wurde.

- Die Gastbetriebssysteme müssen nicht mehr in unprivilegierten Ringen betrieben werden, sondern deren Kernel kann direkt in Ring 0 laufen. Sie können somit unangepasst bleiben, da sie sich in ihrer gewohnten Umgebung befinden.

- Die Gastbetriebssysteme können den Prozessor direkt nutzen.

- Nach wie vor ist der Hypervisor für die Verwaltung und Kontrolle der VMs zuständig, verhält sich jedoch (im Gegensatz zur Vollvirtualisierung) rein passiv.

Vollvirtualisierung



Klassisches Ring Modell des x86-Prozessors
(Quelle: in Anlehnung an [Thorns, 2008] S.23**)**

Klassisches Ring-Modell:

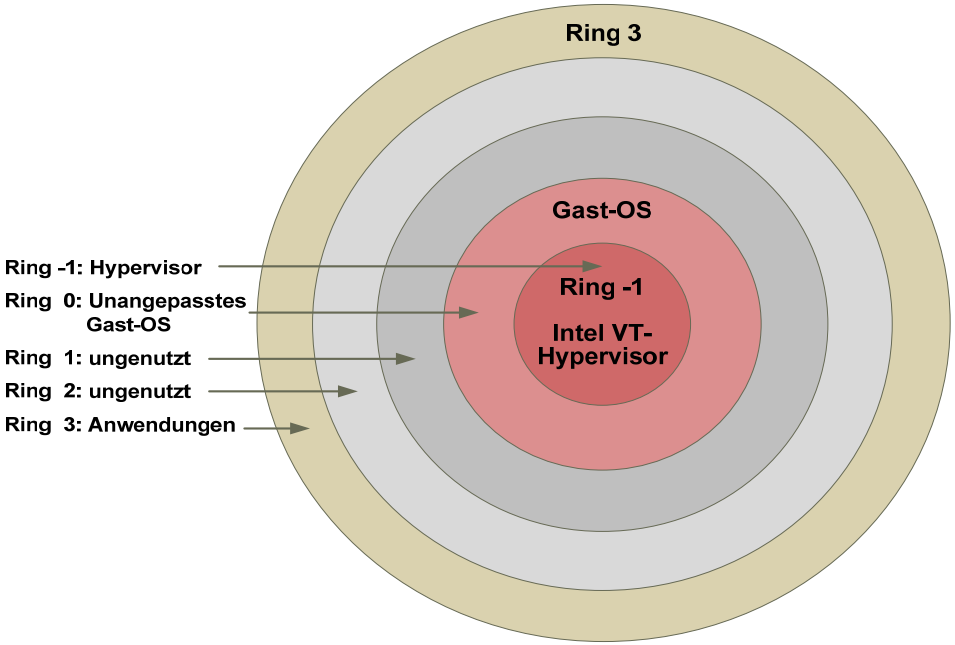
Das Betriebssystem läuft in Ring 0 und hat vollen Zugriff auf die Hardware. Es läuft im sog. Kernel-Mode.

Anwendungen laufen in Ring 3. Hierbei spricht man dann vom sog. „User-Mode“, der nur eingeschränkte Rechte besitzt und bspw. nur auf Speicherbereiche zugreifen darf, die der Anwendung vorher explizit zugewiesen wurden.

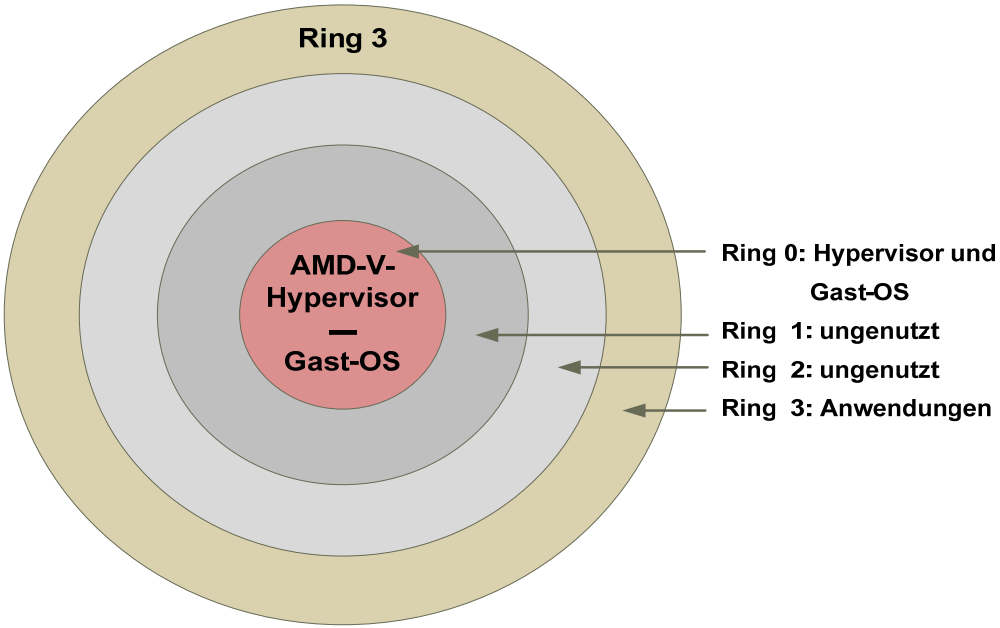
Zugriffsberechtigungen nehmen von Ring 0 nach 3 ab.

Ring 1 und 2 werden in modernen Betriebssystemen nur sehr selten genutzt.

Hardwarebasierte Virtualisierung



Ring-Modell bei Intel VT



Ring-Modell bei AMD-V

Hardwarebasierte Virtualisierung

- Durch die neuen zur Verfügung stehenden Befehlssätze können privilegierte Instruktionen regulär und somit sicher ablaufen.
- Ein Kontextwechsel zum Hypervisor ist nicht jedes Mal notwendig.
- Der Virtualisierungsprozess wird nicht gestört und damit der Virtualisierungsoverhead sehr stark reduziert.
- Bei der hardwarebasierten Virtualisierung laufen VMs aufgrund der erwähnten Vorteile des Verfahrens äußerst performant.

Hardwarebasierte Virtualisierung

Vor- und Nachteile der hardwarebasierten Virtualisierung

Vorteile	Nachteile
• Geringer Virtualisierungs-overhead • Hohe Performance der VMs • Durch die Hardware-unterstützung sind die darauf aufbauenden Virtualisierungs-lösungen relativ schlank und es wird eine hohe Stabilität erreicht • Gastsysteme müssen nicht angepasst werden (auch proprietäre Betriebssysteme können als VM installiert werden) • Bessere Trennung der VMs auf Prozessorebene	• Es werden moderne Prozessoren mit Virtuali-sierungsunterstützung benötigt

KVM
(Kernel-based Virtual Machine)

KVM

- KVM ist eine Open Source Virtualisierungslösung (GPL) für Linux, welche Vollvirtualisierung auf x86-Hardware ermöglicht.
- KVM wurde 2006 von der israelischen Firma Qumranet veröffentlicht und wurde bereits ein halbes Jahr später in den Linux-Kernel (> 2.6.20) aufgenommen. Qumranet wurde im September 2008 von Red Hat gekauft.
- Seit der Linux-Version 2.6.20 ist KVM fester Bestandteil des Mainstream Linux-Kernels und ist inzwischen in den meisten Distributionen als Standardvirtualiserungslösung integriert.
- Auch in der Industrie erfährt KVM eine breite Unterstützung. AMD, IBM, Intel, Suse und RedHat arbeiten bei der Weiterentwicklung zusammen.

KVM

- KVM ist eine Abspaltung vom Emulator „QEMU“. QEMU kann verschiedene Prozessor-Architekturen (PowerPC, ARM, Alpha, m68k, MIPS und Sparc) emulieren und stellt den VMs die virtuelle Hardware (wie bspw. virtuelle Netzwerkinterfaces) zur Verfügung.

- KVM nutzt die Befehlssatzerweiterungen der neueren Intel- und AMD-Prozessoren (Intel VT und AMD-V).

- KVM bietet Unterstützung für paravirtualisierte Gerätetreiber für Netzwerk- und Blockgeräte (Massenspeicher). Hierdurch können die performancekritischen I/O-Zugriffe weiter optimiert werden. Hierbei baut KVM auf dem offenen Standard VirtIO auf.

- Die Gastsysteme laufen nahezu mit nativer Geschwindigkeit. CPU-Instruktionen werden direkt auf der Hardware des Hostsystems ausgeführt.

KVM

- KVM besteht aus folgenden Modulen:
  - *Kernel-Modul:* Arbeitet als Hypervisor für die VMs und nutzt die Virtualisierungserweiterungen der CPUs. Er ist vollständig in den Linux-Kernel integriert. Somit übernimmt der Kernel selbst die Ressourcenverwaltung und Steuerung der VMs. Der Hypervisor ist sehr schlank und profitiert direkt von allen Weiterentwicklungen am Linux-Kernel.
  - *Modifizierter QEMU im User-Space:* Emuliert die Hardware für die VMs.
  - *KVM-Modul:* Übernimmt die Virtualisierung der CPU.
- Im Gegensatz zu anderen Virtualisierungslösungen fungiert KVM allein als Hypervisor für die Virtualisierung.

KVM

- Für die Netzwerkanbindung der VMs als auch Verbindung untereinander bietet KVM mehrere Netzwerkoptionen.
- Eine vollständige TCP/IP-Anbindung wird meist auf Basis von TUN/TAP-Interfaces realisiert. Diese virtuellen Netzwerk-Kerneltreiber simulieren Netzwerkgeräte: TUN simuliert ein Point-to-Point-Netzwerkgerät und TAP stellt ein Ethernet-Gerät dar.
- Für jedes virtuelle Netzwerkinterface einer VM wird jeweils ein TAP-Interface im Hostsystem erstellt. Ein TAP-Interface kann dann über eine Netzwerk-Bridge (IEEE 802.1d = MAC-Bridge) des Hostsystems verbunden werden.
- An eine Netzwerk-Bridge können wiederum auch physikalische Netzwerkschnittstellen des Hostsystems angeschlossen sein. So kann eine VM mit dem realen Netzwerk des Hostsystems kommunizieren.

KVM

Steckbrief KVM

Produkt	Kernel-based Virtual Machine
Hersteller	Qumranet, Inc. RedHat
Website	http://www.linux-kvm.org/page/Main_Page
Preis	Kostenlos (GNU General Public License)
Host-Plattform	Linux
Gäste	x86, x86_64, Linux, Windows, Haiku OS, AROS, ReactOS, FreeDOS, Solaris, BSD-Derivate
Merkmale	Hypervisor-System für Virtualisierungs-CPUs (Intel VT, AMD-V)

Vergleich der Virtualisierungslösungen

Lösung	Hostsystem	Hostplattform	Gastsysteme
Xen 3.2.0	Linux, openSolaris, NetBSD	x86, x86_64	Linux, NetBSD, Solaris, Windows (nur mit VT-x)
VMware Server 2.0 RC 1	Windows, Linux	x86, x86_64	Windows, Linux, BSD, Netware, Solaris u.a.
VMware ESX 3.5 und ESXi 3.5	Eigenes Betriebssystem	x86, x86_64	Windows, Linux, Netware, Solaris u.a.
Parallels Server / Server for Mac 3.0	Windows, Linux, OS X	x86_64	OS/2, Linux, Windows, BSD u.a.
Microsoft Virtual Server 2005 R2 SP1	Windows Server 2003	x86_64	Windows, Linux
Microsoft Hyper-V	Windows Server 2008 (64-Bit)	x86_64	Windows ab 2000 SP4, Linux 2.4, BSD, Solaris
UML (Linux 2.6.25)	Linux 2.2 ab 2.2.15, 2.4, 2.6	x86, x86_64	Linux 2.4 (mit Patch), Linux 2.6
KVM	Linux ab 2.4	x86, x86_64	Linux, Windows, OS, , ReactOS, FreeDOS, Solaris, BSD-Derivate u.a.

VDE
(Virtual Distributed Ethernet)

VDE

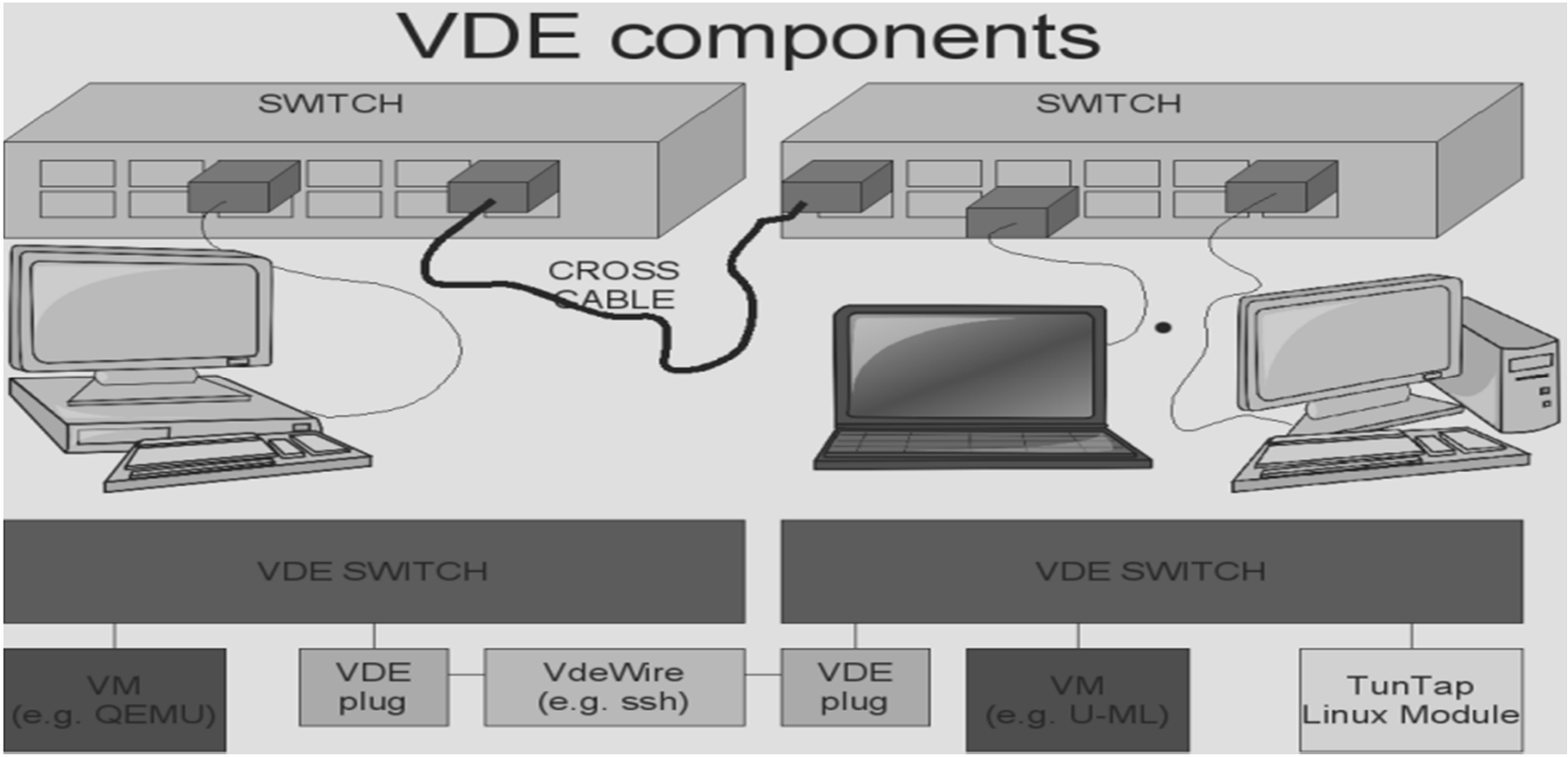
- Die Software VDE ist als weitere Netzwerkoption zu sehen.
- VDE ist Ethernet-konform und stellt virtuellen Infrastrukturen virtuelle Switches und virtuelle Kabel zur Verfügung.
- VDE-Netzwerke bestehen aus den Hauptkomponenten
  - VDE-Switch
  - VDE-Plug
  - VDE-Wire
  - VDE-Cable
- Sie stellen die Basis zum Aufbau von VDE-Netzwerken dar.

VDE

- VDE-Switches lassen sich durch VDE-Cables miteinander verbinden.
- Es besteht auch die Möglichkeit, TUN/TAP-Interfaces an einen VDE-Switch anzuschließen. So lässt sich das virtuelle Netzwerk mit dem Netzwerk des Hostsystems verbinden.
- Da sich VDE-Wire auch über Netzwerkprotokolle (z.B. netcat oder ssh) realisieren lässt, können Verbindungen zwischen VDE-Switches auch über physikalische Netzwerke hergestellt werden (z.B. virtuelle Netzwerke über einen SSH-Tunnel verbinden oder VPNs realisieren).
- KVM-VMs können über VDE-Netze direkt verbunden werden.
- VDE unterstützt auch VLANs nach IEEE 802.1Q.

VDE

Hauptkomponenten von VDE



Quelle: http://www.virtualsquare.org

VDE

- VMs verschiedener Virtualisierungslösungen, Emulatoren, reale Systeme (Betriebssysteme und Netzwerke) lassen sich miteinander verbinden.

- Auf Basis von VDE lassen sich einfach und flexibel virtuelle Netzwerke erstellen und Teilbereiche solcher Netze lassen sich sogar auf mehrere physikalische Rechner verteilen.

Vyatta

Vyatta

- Wie in der realen Welt sind Router und Firewalls auch in virtuellen Netzwerken wichtige Hauptkomponenten einer IT-Infrastruktur.
- Mittels Virtualisierungssoftware können virtuelle Router und Firewalls in Form von entsprechend konfigurierten VMs realisiert werden.

- Vyatta ist eine umfangreiche „Open-Source“ Router- und Firewall-Distribution auf Basis eines angepassten Debian Linux-Systems.
- Die Software wird von der kalifornischen Firma Vyatta Inc. unter der GNU General Public License (GPL) veröffentlicht.

- Weiterhin werden auch vorkonfigurierte Hardware-Router (sog. Hardware-Appliances) auf Basis von Vyatta angeboten.

Vyatta

- Vyatta bietet zahlreiche Netzwerkfunktionen, welche auch auf professionellen Hardware-Routern eingesetzt werden. Es werden diverse Routingprotokolle wie OSPF, BGP und RIP unterstützt.

- Jedoch bietet die Distribution nicht nur Routingfunktionen.
- Weitere Einsatzmöglichkeiten sind z.B.:

DHCP
NAT
PPPoE
VoIP QoS
WAN link load balancing
Site-to-Site Ipsec-VPN
SSL-based OpenVPN
RADIUS authentication
802.1q VLANs
Stateful Firewall
SNMP
IDS/IPS
Anti-Virus

Vyatta

- Somit stellt Vyatta eine leistungsfähige und erstzunehmende Alternative zu Hardware-Routern wie denen von Cisco oder Juniper dar.
- Die Software wurde mit der Zielsetzung entwickelt, alle wesentlichen Merkmale moderner Hardware-Router abzubilden, welches auf gängiger x86-Hardware läuft. Dies impliziert enorme Vorteile in Bezug auf Kosten und Flexibilität.

- Durch den Einsatz moderner x86-Hardware erwies sich Vyatta auch in unabhängigen Vergleichstests in Bezug auf Durchsatz und Performance als äußerst leistungsstark. Im konkreten Vergleich mit proprietären Lösungen von Cisco, wie den Cisco 2821 und 7204VXR Routern, ergaben sich sogar erheblich bessere Leistungen.

LISA
(Laboratory for IT-Security Architectures)

LISA

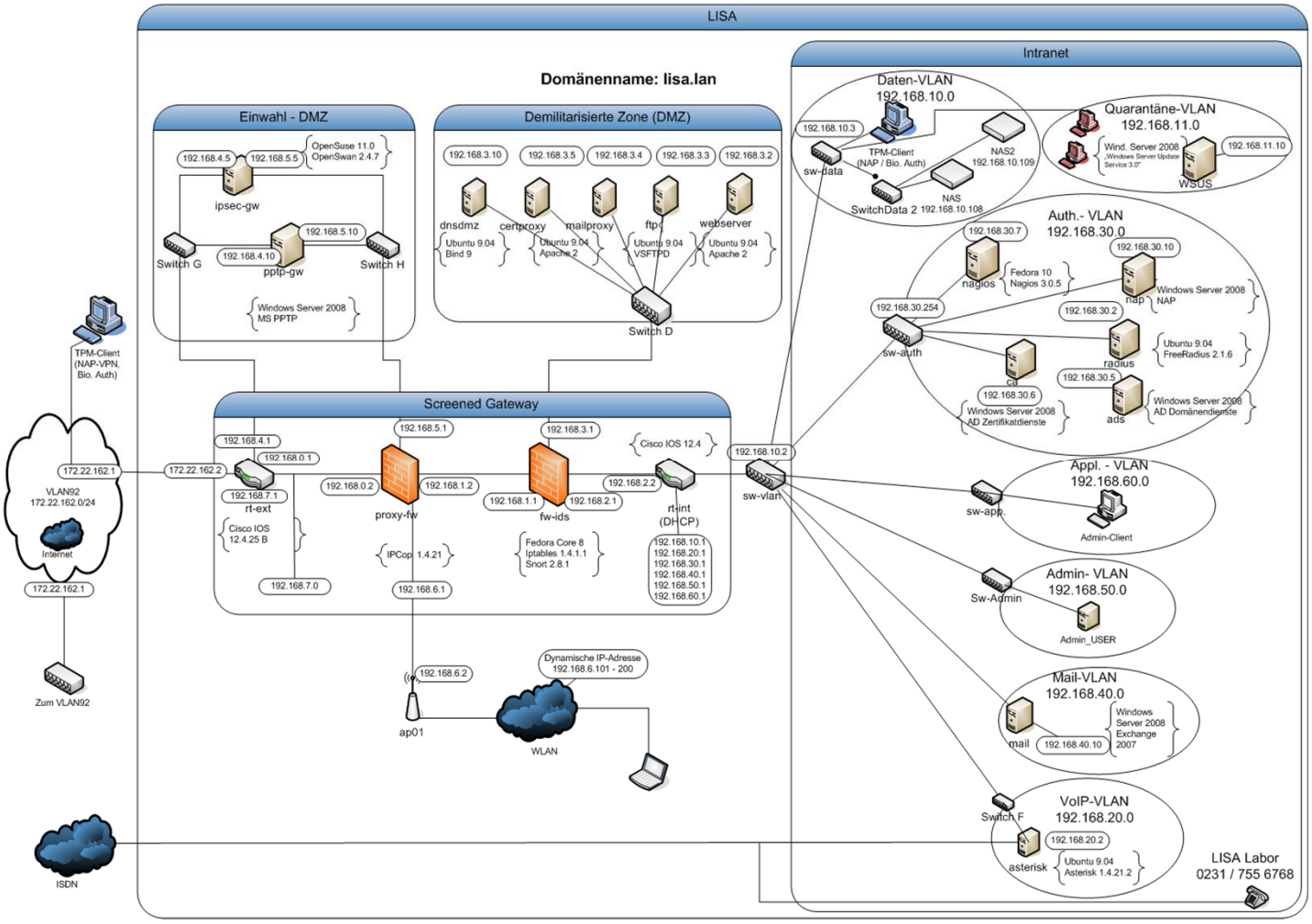
Laboratory for IT-Security Architectures

- Das „Laboratory for IT-Security Architectures – LISA” im Fachbereich Informatik der Fachhochschule Dortmund stellt eine modulare Entwicklungs- und Evaluations-Plattform für IT-Sicherheitsarchitekturen zur Verfügung.
- Hier können Sicherheitsmodelle und -architekturen exemplarisch diskutiert, erprobt und validiert werden.
- LISA wird sowohl für die praxisorientierte Lehre und Forschung, als auch als Demo-Center für Unternehmen eingesetzt.
- Dabei werden insbesondere Sicherheitsprobleme und -architekturen von KMU adressiert.

LISA

Laboratory for IT-Security Architectures

Topologiebeispiel



Virtual Security Appliances

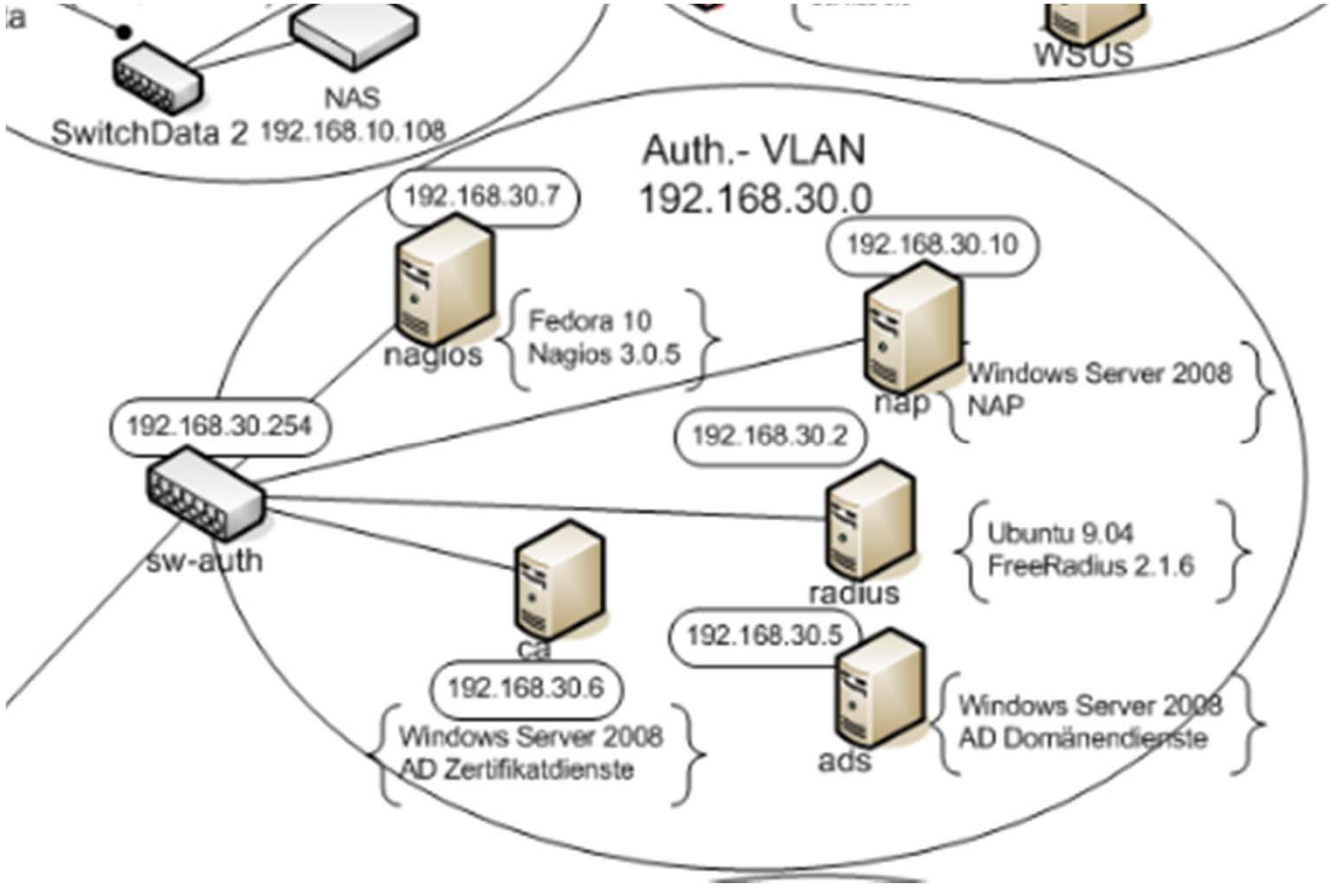
Virtual Security Applicances

- Mit Hilfe moderner Virtualisierungssoftware lassen sich komplexe IT-Infrastrukturen bis auf Layer 1 des ISO-OSI-Modells virtuell abbilden.

- Mit Hilfe von KVM, VDE und Vyatta lassen sind „Virtual Security Appliances“ modular und flexibel realisieren.

LISA
Laboratory for IT-Security Architectures

Virtual Security Applicances



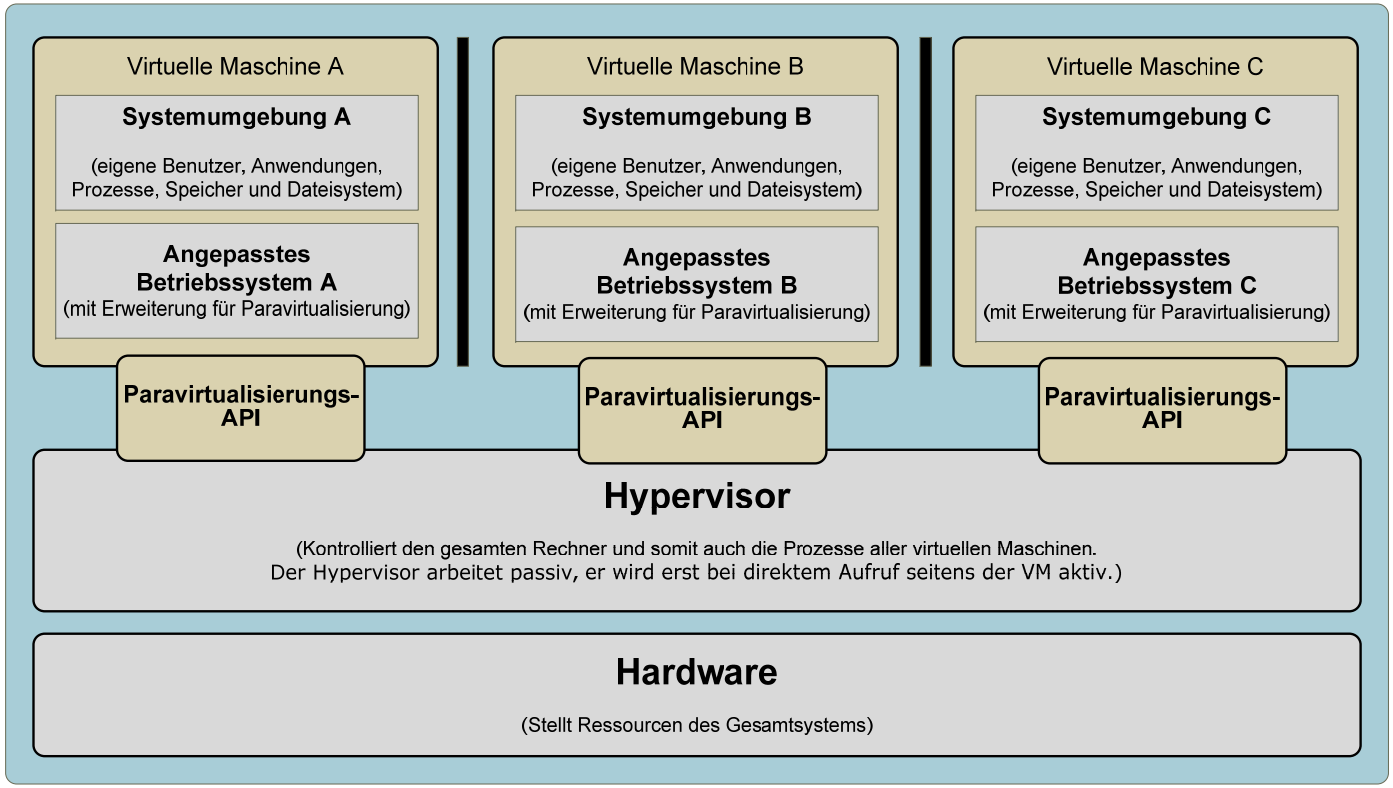
Vielen Dank

Paravirtualisierung

Paravirtualisierung

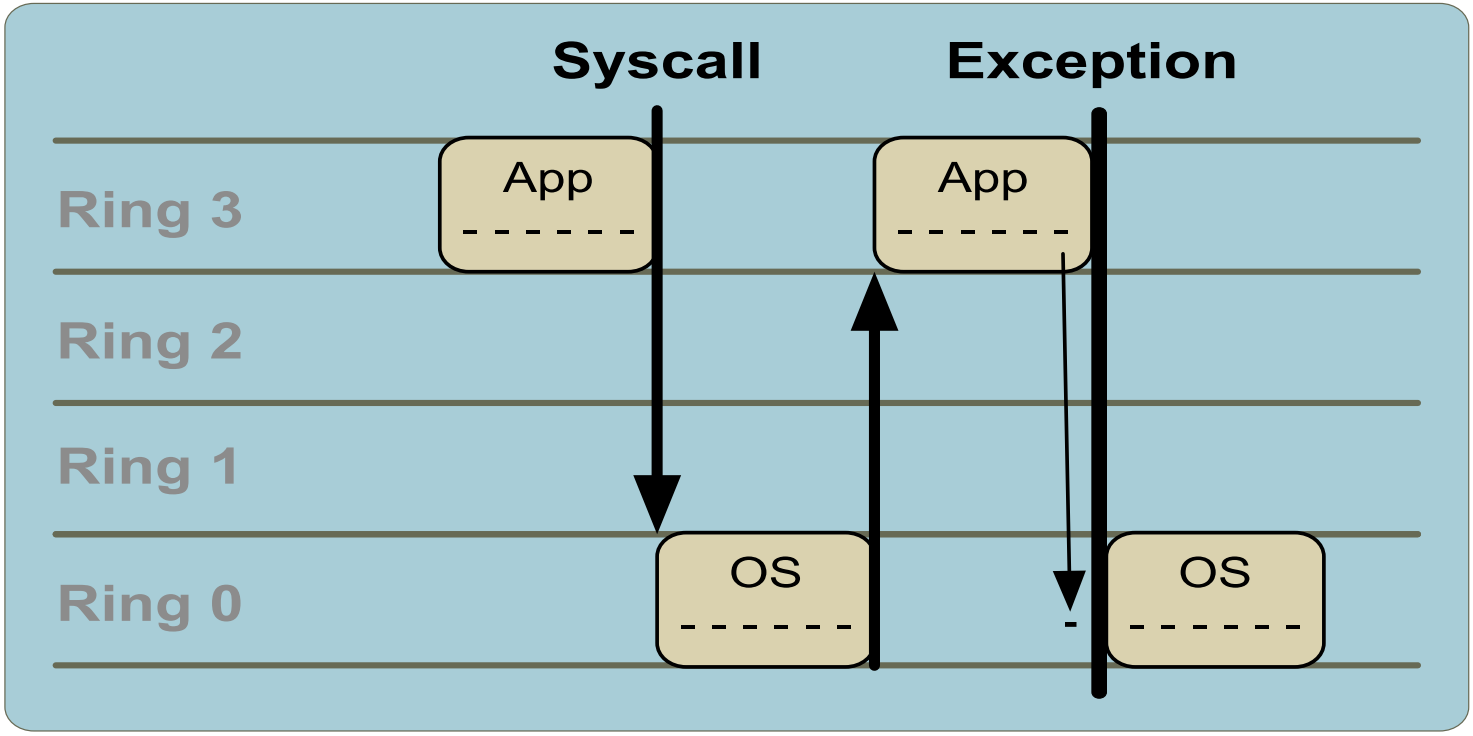
- Das Verfahren der Paravirtualisierung baut nicht auf Exception Handling auf, sondern verfolgt einen softwareseitigen Ansatz (über die Paravirtualisierungs-API).
- Hierbei initiiert das Gastbetriebssystem im richtigen Moment einen Wechsel zwischen VM und Hypervisor.
- Das Gastbetriebssystem wird hierzu angepasst und ist sich somit der Virtualisierung bewusst. Der Befehlssatz der VMs wird um einen sog. „Hypervisor-Call“ erweitert.
- Sobald eine kritische Operation bevorsteht, löst das Gastbetriebssystem aktiv einen Kontextwechsel von der VM zum Hypervisor aus, indem es selbstständig ein Signal bzw. einen Interrupt auslöst (Call).
- Die technische Umsetzung erfolgt durch Patches in den Gastbetriebssystemen, die den Kernel so modifizieren, dass eine Interaktion mit dem Hypervisor möglich wird.

Paravirtualisierung



Schematische Darstellung einer Paravirtualisierung mit angepassten Gastbetriebssystemen

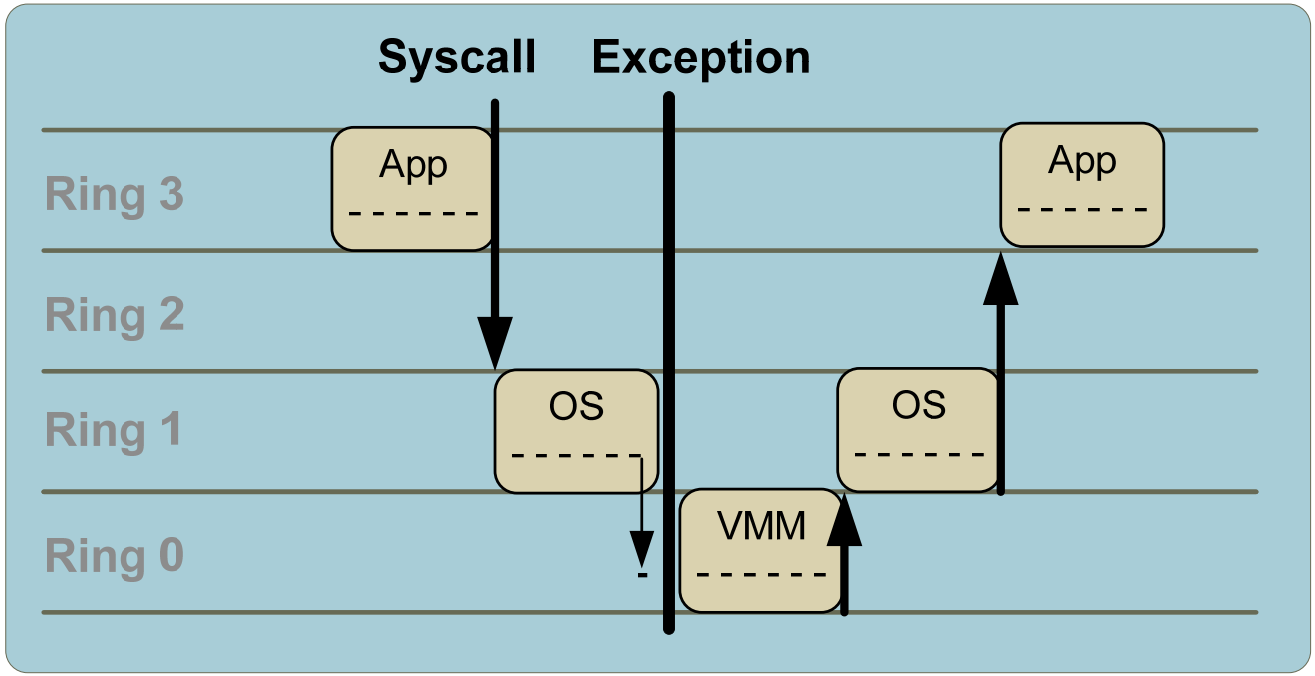
Vollständige Virtualisierung bzw. echte Virtualisierung



User-Mode-Anwendung versucht einen privilegierten Befehl auszuführen, was zur Auslösung einer Exception führt.

(Quelle: in Anlehnung an [Thorns, 2008] S.25**)**

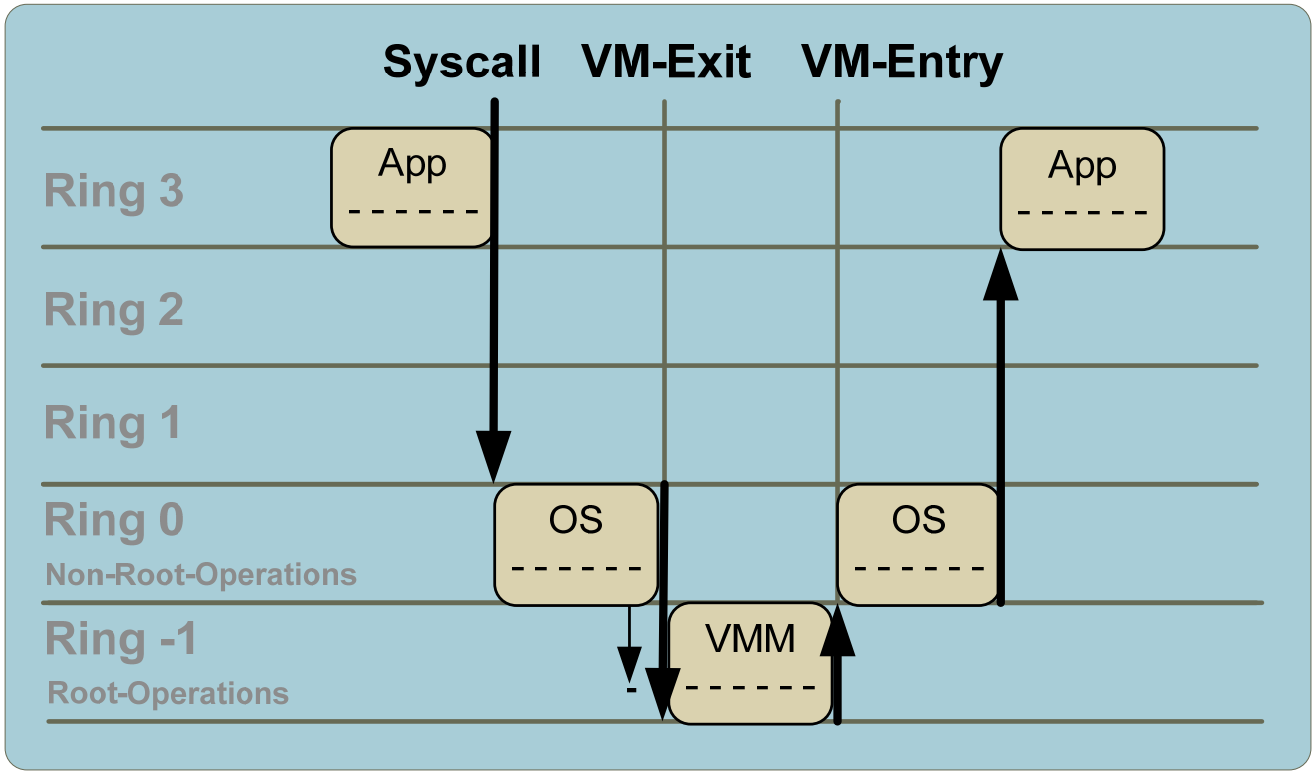
Vollständige Virtualisierung bzw. echte Virtualisierung



Trap & Emulate bei der Ausführung von Gast-Betriebssystemen in einem höheren Ring.

(Quelle: in Anlehnung an [Thorns, 2008] S.28**)**

Hardwarebasierte Virtualisierung



Intel-VT mit Root- und Non-Root-Operationen

(Quelle: in Anlehnung an [Thorns, 2008] S. 30**)**