

# 7 Virtualisierung und die Cloud

- 1 Einführung
- 2 Prozesse und Threads
- 3 Speicherverwaltung
- 4 Dateisysteme
- 5 Eingabe und Ausgabe
- 6 Deadlocks
- 7 ***Virtualisierung und die Cloud***
- 8 Multiprozessorsysteme
- 9 IT-Sicherheit
- 10 Fallstudie 1: Linux
- 11 Fallstudie 2: Windows
- 12 Entwurf von Betriebssystemen

# 7 Virtualisierung und die Cloud

- 7.1 Geschichte der Virtualisierung
- 7.2 Anforderungen für die Virtualisierung
- 7.3 Typ-1- und Typ-2-Hypervisoren
- 7.4 Techniken für die effiziente Virtualisierung
- 7.5 Der Hypervisor: ein idealer Mikrokern?
- 7.6 Speichervirtualisierung
- 7.7 Ein-/Ausgabevirtualisierung
- 7.8 Virtual Appliances
- 7.9 Virtuelle Maschinen bei Mehrkernprozessoren
- 7.10 Fragen bezüglich der Lizenzierung
- 7.11 Fallstudie: VMware

# 7.1 Geschichte der Virtualisierung

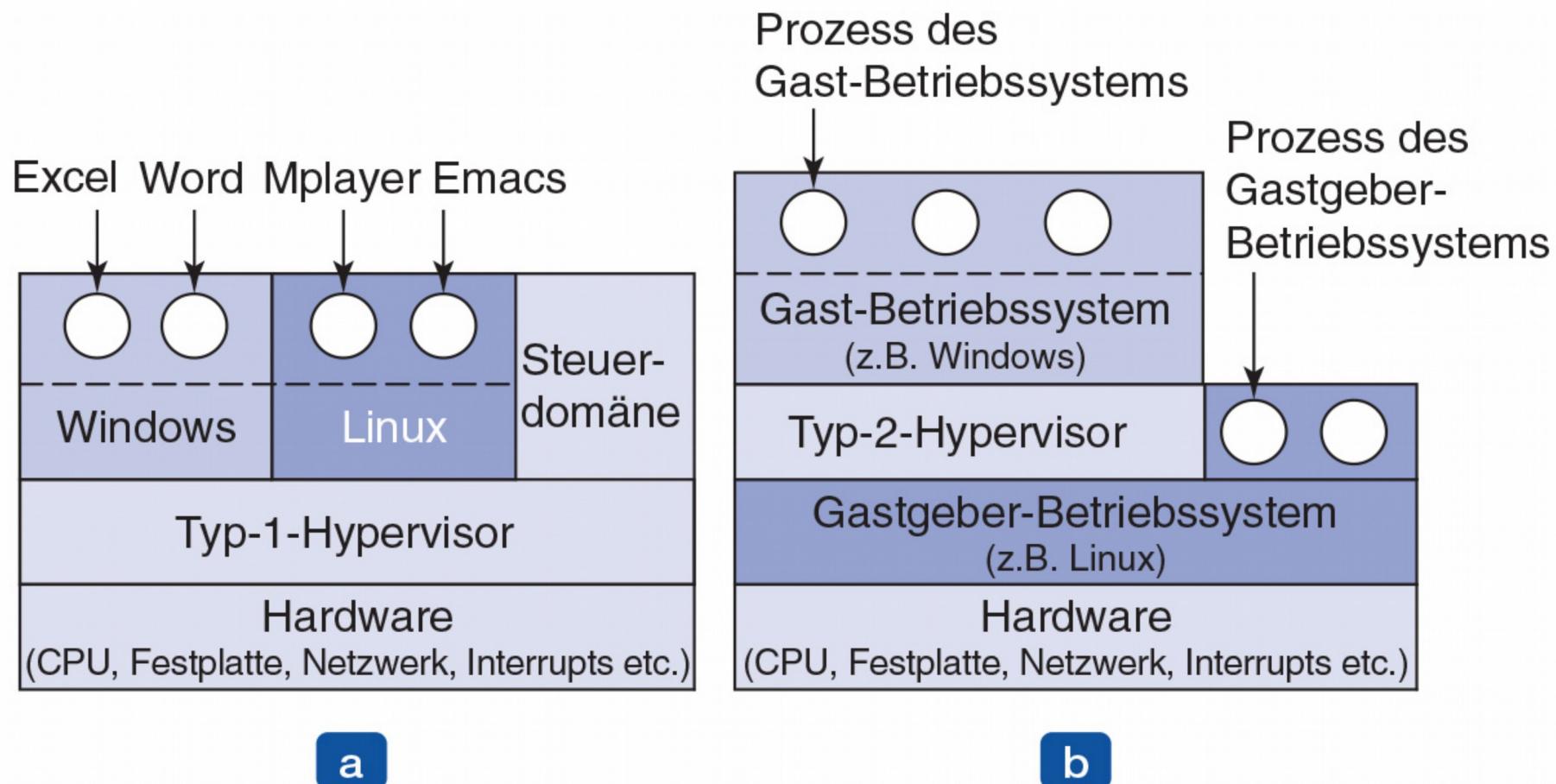
- 1966: IBM CP/CMS
  - das erste Betriebssystem mit vollständiger Virtualisierung
  - mehrere Benutzer konnten gleichzeitig auf einem physischen Rechner jeweils ein eigenes Einzelbenutzerbetriebssystem unabhängig nutzen
- Ende der 1990er: Große UNIX-Hersteller (Sun, HP, IBM und SGI) bieten Serverlösungen mit Virtualisierung an
- 2005 begannen CPU-Hersteller im x86-Bereich Virtualisierungsunterstützung zu integrieren

## 7.2 Anforderungen für die Virtualisierung

**Hypervisoren sollten in drei Dimensionen gut abschneiden:**

1. Sicherheit: der Hypervisor sollte die volle Kontrolle über virtualisierte Ressourcen haben.
2. Wiedergabetreue: Das Verhalten eines Programms auf einer virtuellen Maschine sollte mit dem gleichen Programm identisch sein, das auf reiner Hardware läuft.
3. Effizienz: Ein Großteil des Codes in der virtuellen Maschine sollte ohne Eingriff des Hypervisors ausgeführt werden

# 7.3 Typ-1- und Typ-2-Hypervisoren (1)



**Abbildung 7.1:** Platzierung der Typ-1- und Typ-2-Hypervisoren.

## 7.3 Typ-1- und Typ-2-Hypervisoren (2)

Virtualisierungsmethode	Typ-1-Hypervisor	Typ-2-Hypervisor
Virtualisierung ohne Hardwareunterstützung	ESX Server 1.0	VMware Workstation 1
Paravirtualisierung	Xen 1.0	
Virtualisierung mit Hardwareunterstützung	vSphere, Xen, Hyper-V	VMware Fusion, KVM, Parallels
Prozessvirtualisierung		Wine

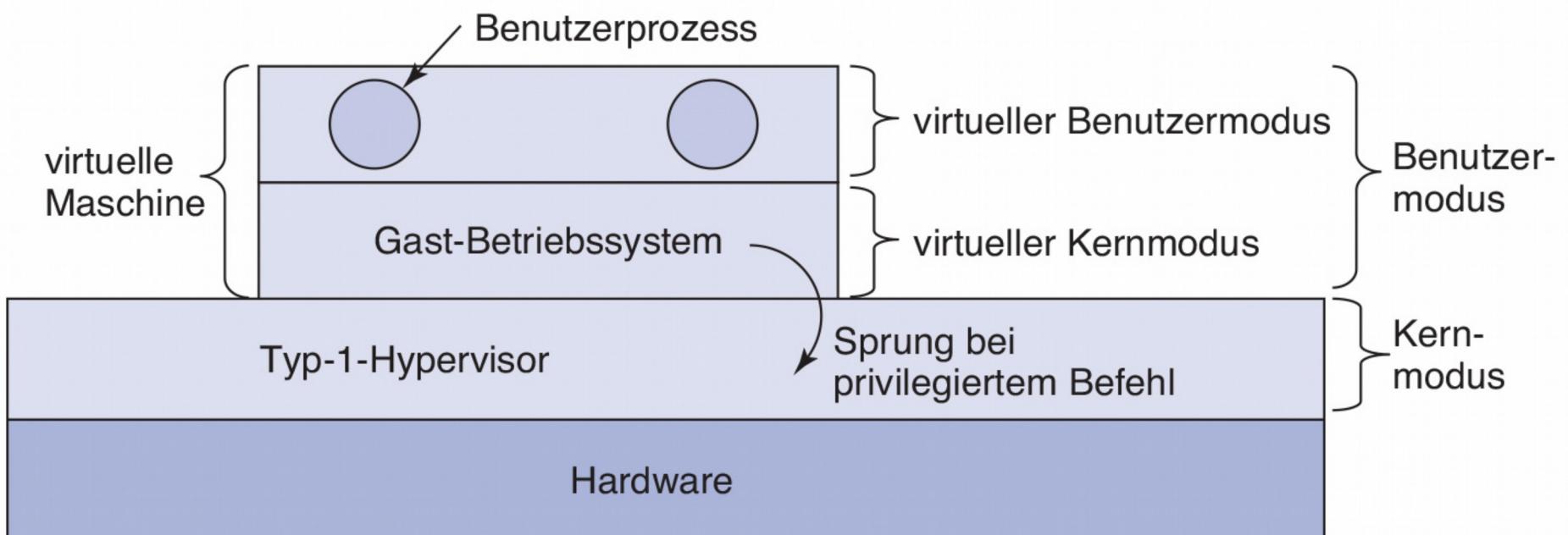
**Abbildung 7.2:** Beispiele von Hypervisoren. Typ-1-Hypervisoren laufen direkt auf der Hardware, wohingegen Typ-2-Hypervisoren die Dienste eines existierenden Gastgeber-Betriebssystems nutzen.

## 7.4 Techniken für die effiziente Virtualisierung

7.4.1 Das Nichtvirtualisierbare virtualisieren

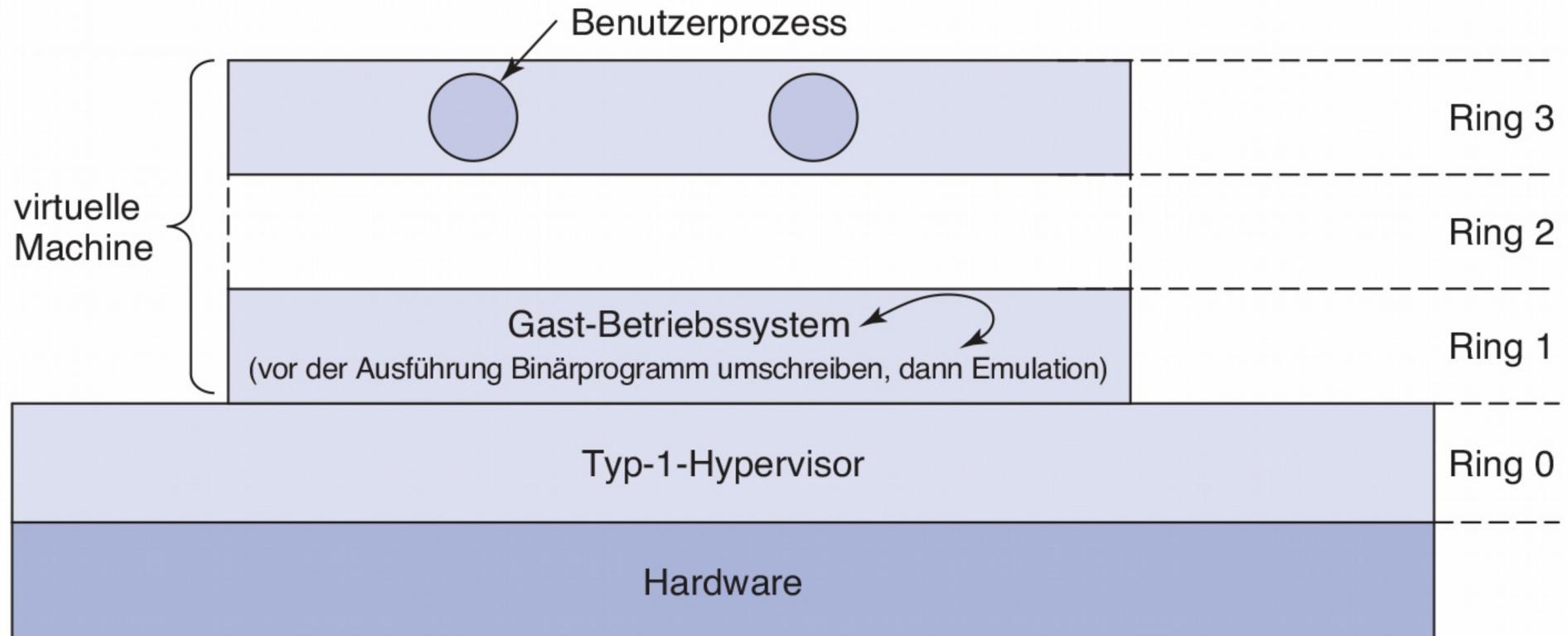
7.4.2 Kosten der Virtualisierung

# Techniken für die effiziente Virtualisierung



**Abbildung 7.3:** Wenn das Betriebssystem in einer virtuellen Maschine einen Befehl ausführt, der nur für den Kern bestimmt ist, dann springt es in den Hypervisor, falls Virtualisierungstechnologie vorhanden ist.

# Das Nichtvirtualisierbare virtualisieren

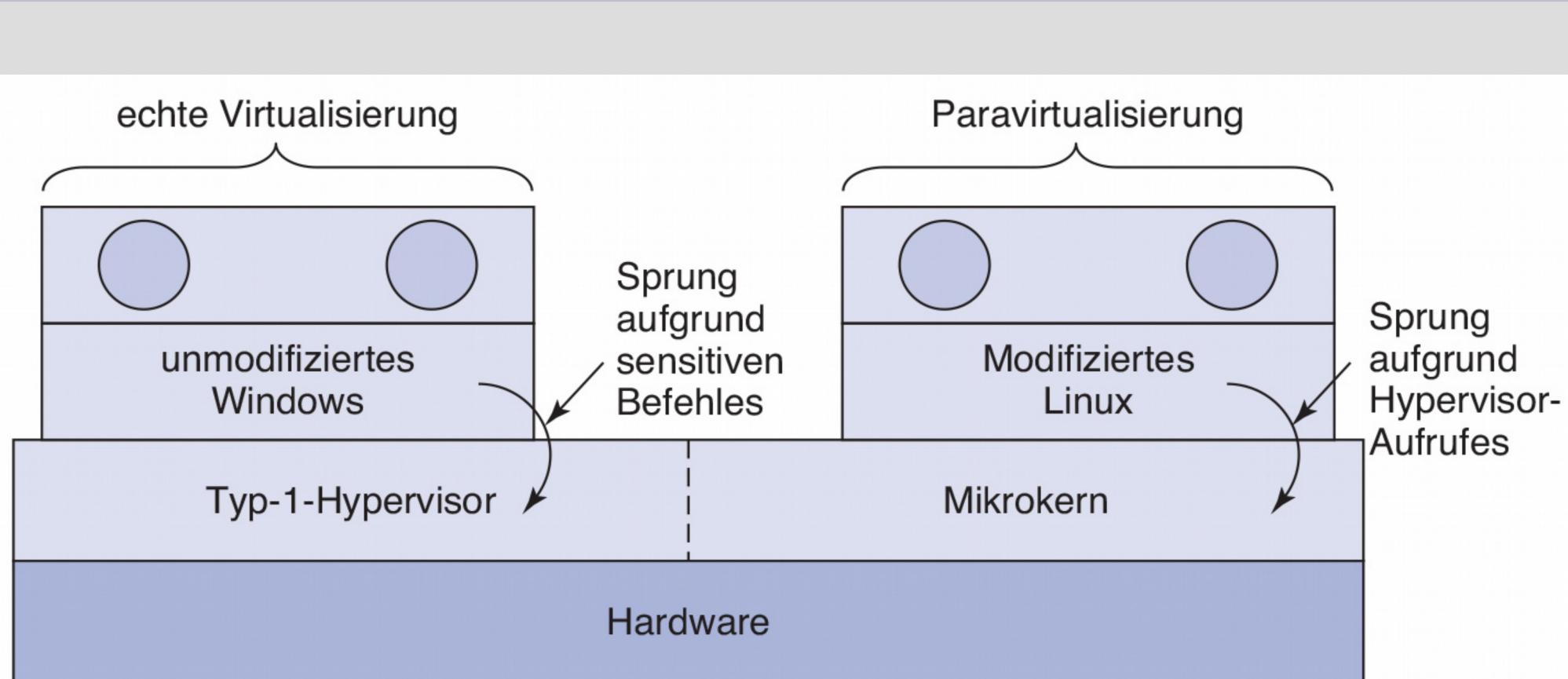


**Abbildung 7.4:** Der Binärübersetzer schreibt das Gast-Betriebssystem, das in Ring 1 läuft, um, während der Hypervisor in Ring 0 läuft.

# Kosten der Virtualisierung

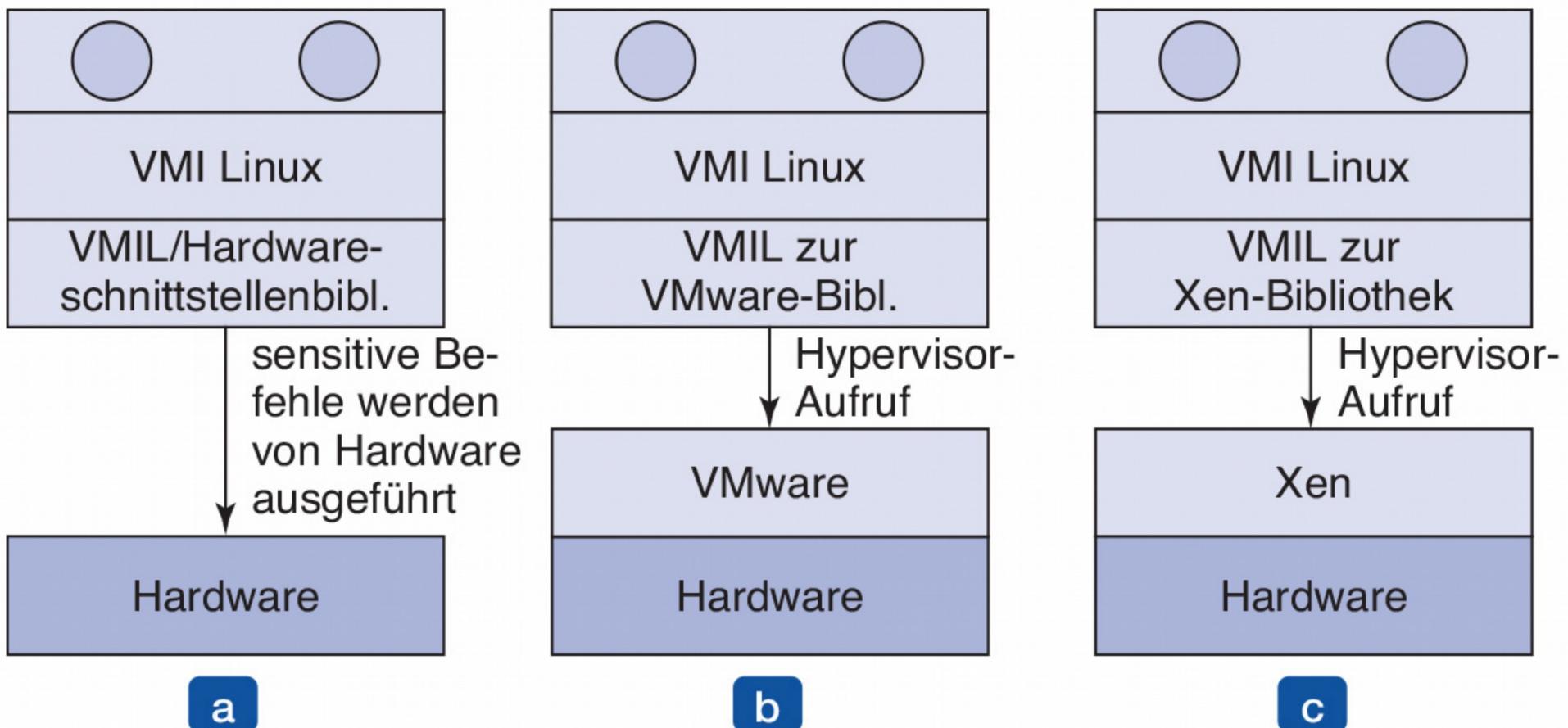
- virtuelle Maschinen auf Basis emulierter Hardware:
  - der Virtual Machine Monitor muss die Befehle vor Ausführung auf der nativen Hardware modifizieren
- Prozessorkompatibilität (AMD/Intel):
  - Befehle können ohne Änderung unmittelbar an den Prozessor geschickt werden
  - Prozessor kümmert sich selbstständig um die Abgrenzung zwischen Gastsystem- und VMM-Prozess

# 7.5 Der Hypervisor: ein idealer Mikrokern?



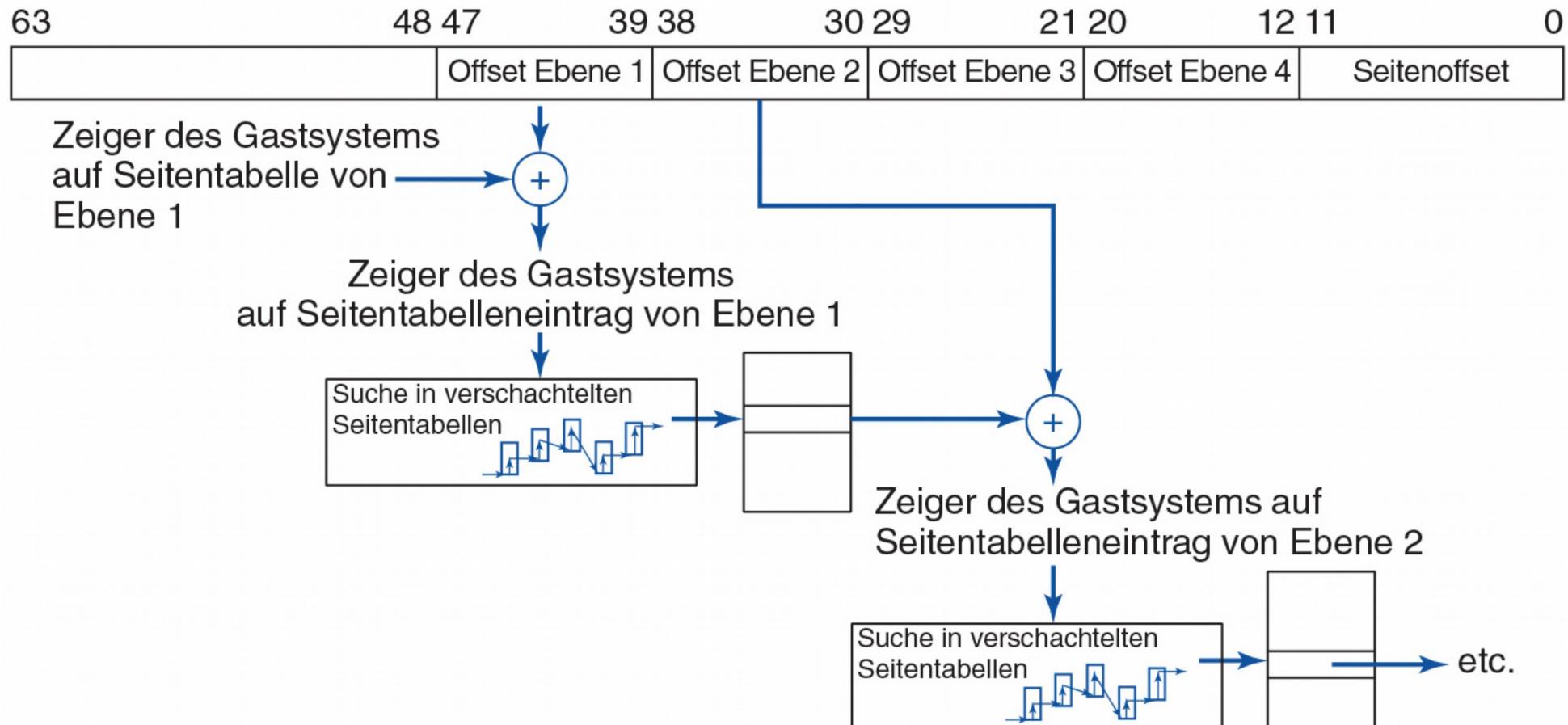
**Abbildung 7.5:** Echte Virtualisierung und Paravirtualisierung.

# Sind Hypervisors Microkernels?



**Abbildung 7.6:** VMI Linux läuft (a) auf der reinen Hardware, (b) auf VMware, (c) auf Xen.

## 7.6 Speichervirtualisierung - Hardwareunterstützung für verschachtelte Seitentabellen



**Abbildung 7.7:** Erweiterte/verschachtelte Seitentabellen werden jedes Mal durchsucht, wenn auf eine physische Gastadresse zugegriffen wird – einschließlich der Zugriffe für jede Ebene der Seitentabellen des Gast-Betriebssystems.

## 7.7 Ein-/Ausgabevirtualisierung

- I/O-Virtualisierung ist noch in der Entwicklung
- z.B. regelt der Hypervisor den Zugriff auf den Netzwerkadapter
- Nicht jedes Guest-System kann beliebig Daten die Adressbereiche des Netzwerkadapters schreiben
  - Aufspaltung eines physischen I/O-Adapters in mehrere logische
  - Emulation von I/O Endpunkten

## 7.8 Virtual Appliances

### **Virtual Appliances:**

- vorinstallierte, vorkonfigurierte und sofort einsetzbare Anwendungen und Software-Lösungen, die bereits mit einem Betriebssystem in der virtuellen Maschine zusammengestellt sind
- Virtuelle Geräte verändern grundsätzlich die Entwicklung, die Verteilung und die Verwaltung von Unternehmenssoftware:
  - Einfachere Verteilung
  - Vereinfachte Fehlerkorrektur und Aktualisierung
  - Portabilität und Betriebssystemunabhängigkeit

## 7.9 Virtuelle Maschinen bei Mehrkernprozessoren

- Mehrkernprozessoren wurden entwickelt, weil eine Erhöhung der Rechenleistung durch höhere Taktfrequenzen große technische Probleme verursachten
- Virtuellen Maschinen können ein oder mehrere Kerne exklusiv zugewiesen werden
- Damit wird die Virtualisierung effizienter

# 7.10 Clouds

7.10.1 Clouds-as-a-Service

7.10.2 Migration von virtuellen Maschinen

7.10.3 Checkpointing

# Clouds

**Definition des Nationalen Instituts für Standards und Technologie für „Cloud“:**

1. On-Demand-Selbstbedienung
2. Breitbandiger Netzwerkzugriff
3. Zusammenlegung von Ressorcen
4. Schnelle Elastizität
5. Gemessener Service

# Herausforderungen bei der Virtualisierung von x86 (1)

**Kernattribute einer virtuellen Maschine für eine x86-basierte Zielplattform:**

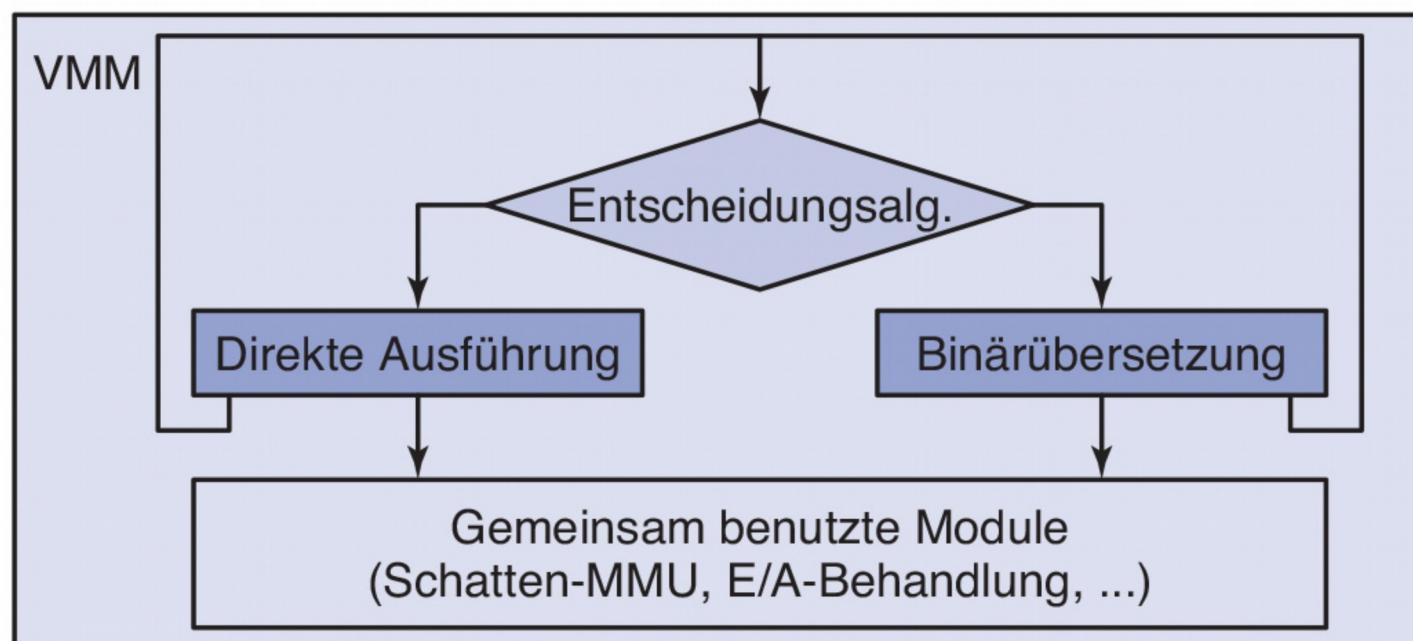
1. Kompatibilität
2. Performance
3. Isolierung

# Herausforderungen bei der Virtualisierung von x86 (2)

## Hauptherausforderungen:

1. Die x86-Architektur war nicht virtualisierbar
2. Die x86-Architektur war von überwältigender Komplexität
3. x86-Maschinen hatten diverse Peripheriegeräte
4. Die Notwendigkeit einer einfachen Benutzererfahrung

# Virtualisierung der x86-Architektur (1)



**Abbildung 7.8:** Obere Komponenten des VMM von VMware (wenn es keine Hardwareunterstützung gibt).

# Virtualisierung der x86-Architektur (2)

**Die binäre Übersetzung muss verwendet werden, wenn eine der folgenden Bedingungen zutrifft:**

1. Die Virtuelle Maschine läuft derzeit im Kernel-Modus
2. Die Virtuelle Maschine kann Interrupts deaktivieren und E / A-Anweisungen erteilen
3. Die Virtuelle Maschine läuft derzeit im Echtzeitmodus

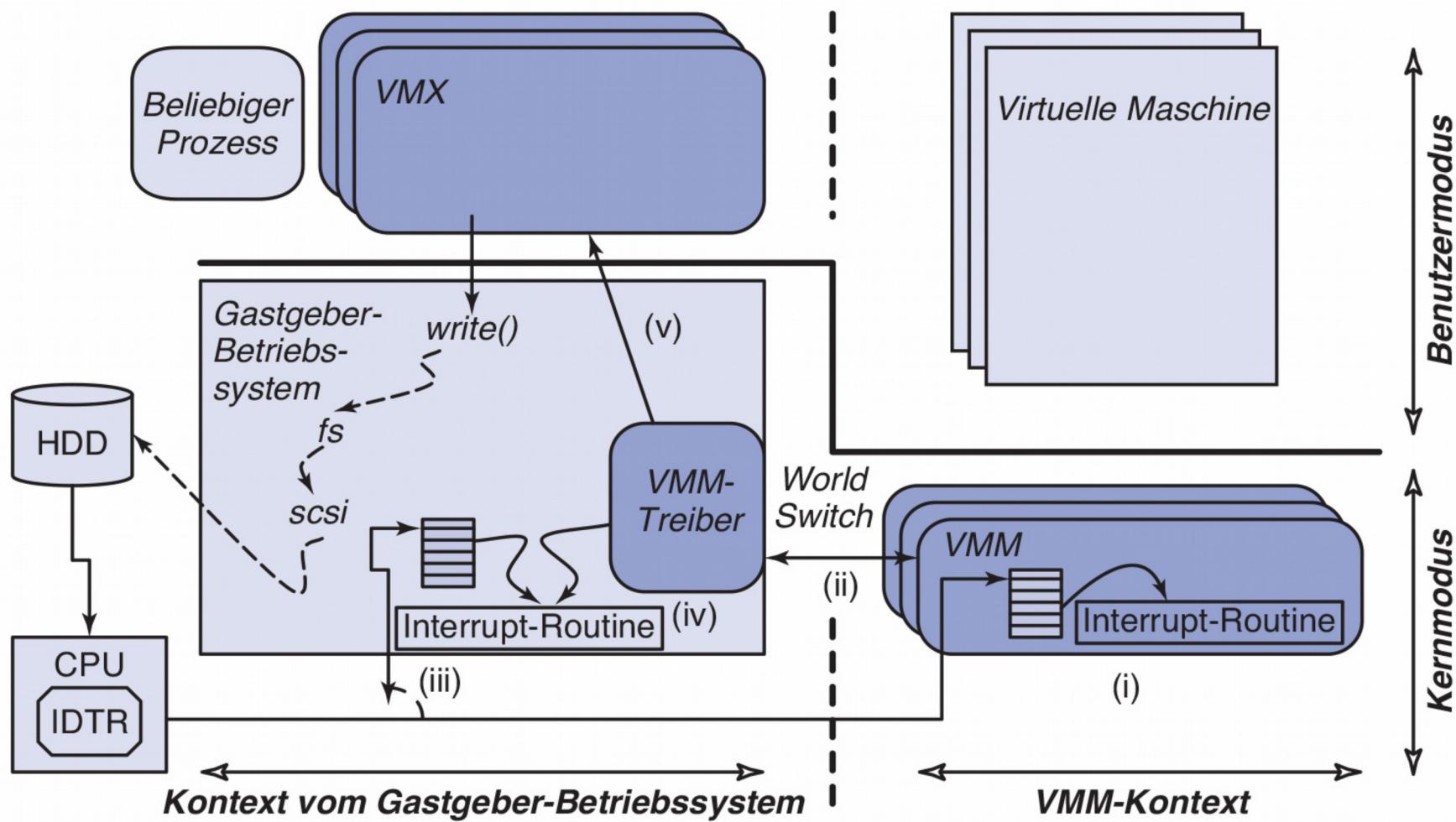
# Virtuelle Hardwareplattform

Tanenbaum, A. S.; Bos, H.: Moderne Betriebssysteme. Pearson Studium 2016

	<i>Virtuelle Hardware (Frontend)</i>	<i>Backend</i>
Multiplexing	1x virtuelle x86-CPU mit denselben Befehlssatzerweiterungen wie die zugrunde liegende Hardware-CPU	Scheduling wird vom Gastgeber-Betriebssystem durchgeführt, entweder auf einem Uniprozessor- oder Multiprozessor-Gastgeber
	Bis zu 512 MB zusammenhängendes DRAM	Belegung und Verwaltung durch das Gastgebersystem (Seite für Seite)
Emulation	PCI-Bus	Vollständig emulationskonformer PCI-Bus
	4x IDE-Festplatten 7x Buslogic-SCSI-Festplatten	Virtuelle Platten (als Dateien gespeichert) oder direkter Zugriff auf ein bestimmtes Gerät
	1x IDE-CD-ROM	ISO-Image oder emulierter Zugriff auf die reale CD-ROM
	2x 1,44-MB-Diskettenlaufwerk	Physische Diskette oder Diskettenabbild
	1x VMware Grafikkarte mit VGA- und SVGA-Unterstützung	Läuft in einem Fenster und im Vollbildmodus SVGA erfordert VMware SVGA-Gasttreiber
	2x serielle Ports COM1 und COM2	Verbindet mit seriellem Port oder Datei von Gastgeber
	1x Drucker (LPT)	Kann mit einem LPT-Port vom Gastgeber verbinden
	1x Tatstatur (104 Tasten)	Vollständig emuliert: Tastencode-Ereignisse werden erzeugt, wenn sie von der VMware-Anwendung empfangen werden
	1x PS-2-Maus	Siehe Tastatur
	3x AMD-Lance-Ethernetkarten	Bridge-Modus und Nur-Gastgeber-Modi
	1x Soundblaster	Vollständig emuliert

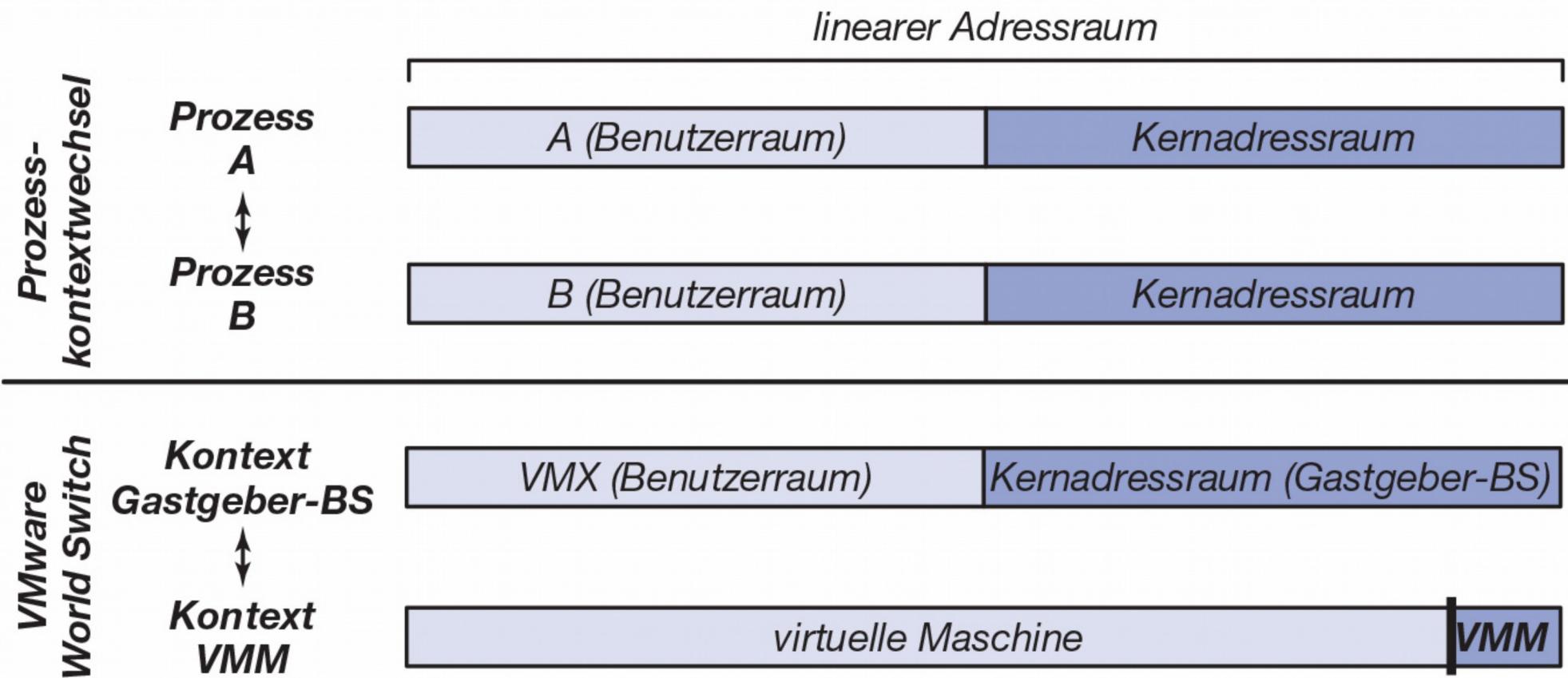
**Abbildung 7.9:** Konfigurationsoptionen der virtuellen Hardware einer frühen VMware-Workstation-Version (ca. 2000).

# Rolle des Host-Betriebssystems (1)



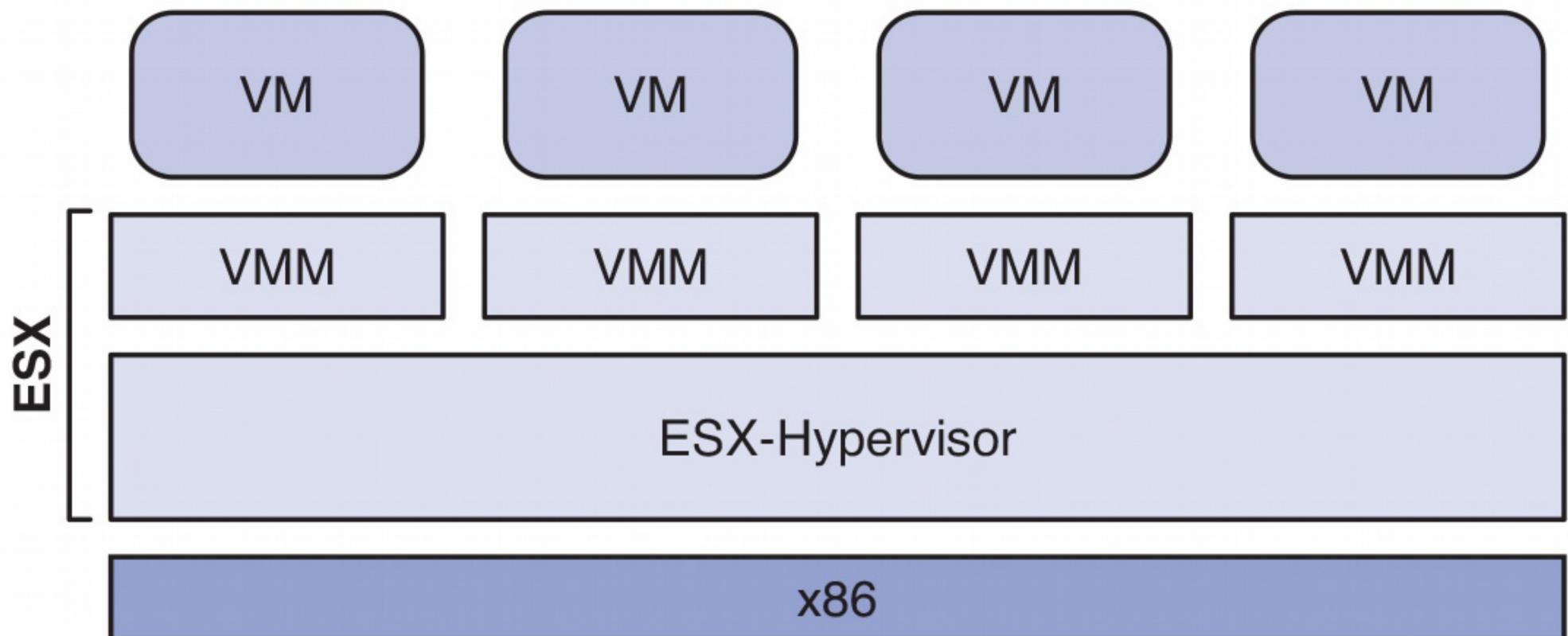
**Abbildung 7.10:** Die VMware Hosted Architecture und ihre drei Komponenten: VMX, VMM-Treiber und VMM.

# Rolle des Host-Betriebssystems (2)



**Abbildung 7.11:** Unterschiede zwischen einem normalen Kontextwechsel und einem World Switch.

# ESX Server: VMware's type-1 Hypervisor (1)



**Abbildung 7.12:** ESX Server – ein Typ-1-Hypervisor von VMware.

# ESX Server: VMware's type-1 Hypervisor (2)

Die ESX Server-Architektur bietet erhebliche Vorteile

1. Der CPU-Scheduler stellt sicher, dass jede virtuelle Maschine einen fairen Anteil der CPU erhält.
2. Der Memory Manager ist für Skalierbarkeit optimiert.
3. Das E / A-Subsystem ist auf Leistung optimiert.
4. Backends basierten typischerweise auch auf Abstraktionen, die vom Host-Betriebssystem bereitgestellt wurden.
5. ESX Server erleichtert die Einführung neuer Funktionen