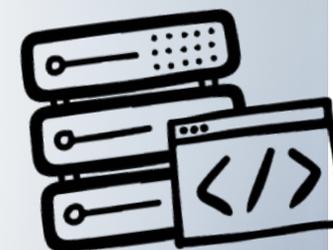




CLOUD-NATIVE

Unit:
Infrastructure as Code

(1) Grundlagen / Virtualisierungsarten



Urheberrechtshinweise

Diese Folien werden zum Zwecke einer praktikablen und pragmatischen Nutzbarkeit im Rahmen der **CCo 1.0 Lizenz** bereitgestellt.

Sie dürfen die Inhalte also kopieren, verändern, verbreiten, mit eigenen Inhalten mixen, auch zu kommerziellen Zwecken, und ohne um weitere Erlaubnis bitten zu müssen.

Eine Nennung des Autors ist nicht erforderlich (aber natürlich gern gesehen, wenn problemlos möglich).

Diese Folien sind insb. für die Lehre an Hochschulen konzipiert und machen daher vom **§51 UrhG (Zitate)** Gebrauch.

Die CCo Lizenz überträgt sich nicht auf zitierte Quellen. Hier sind bei der Nutzung natürlich die Bedingungen der entsprechenden Quellen zu beachten.

Die Quellenangaben finden sich auf den entsprechenden Folien.



KAPITEL 7

Infrastructure as Code



7 Infrastructure as Code

7.1 Virtualisierung

- Virtualisierung von Hardware-Infrastruktur
- Virtualisierung von Software-Infrastruktur

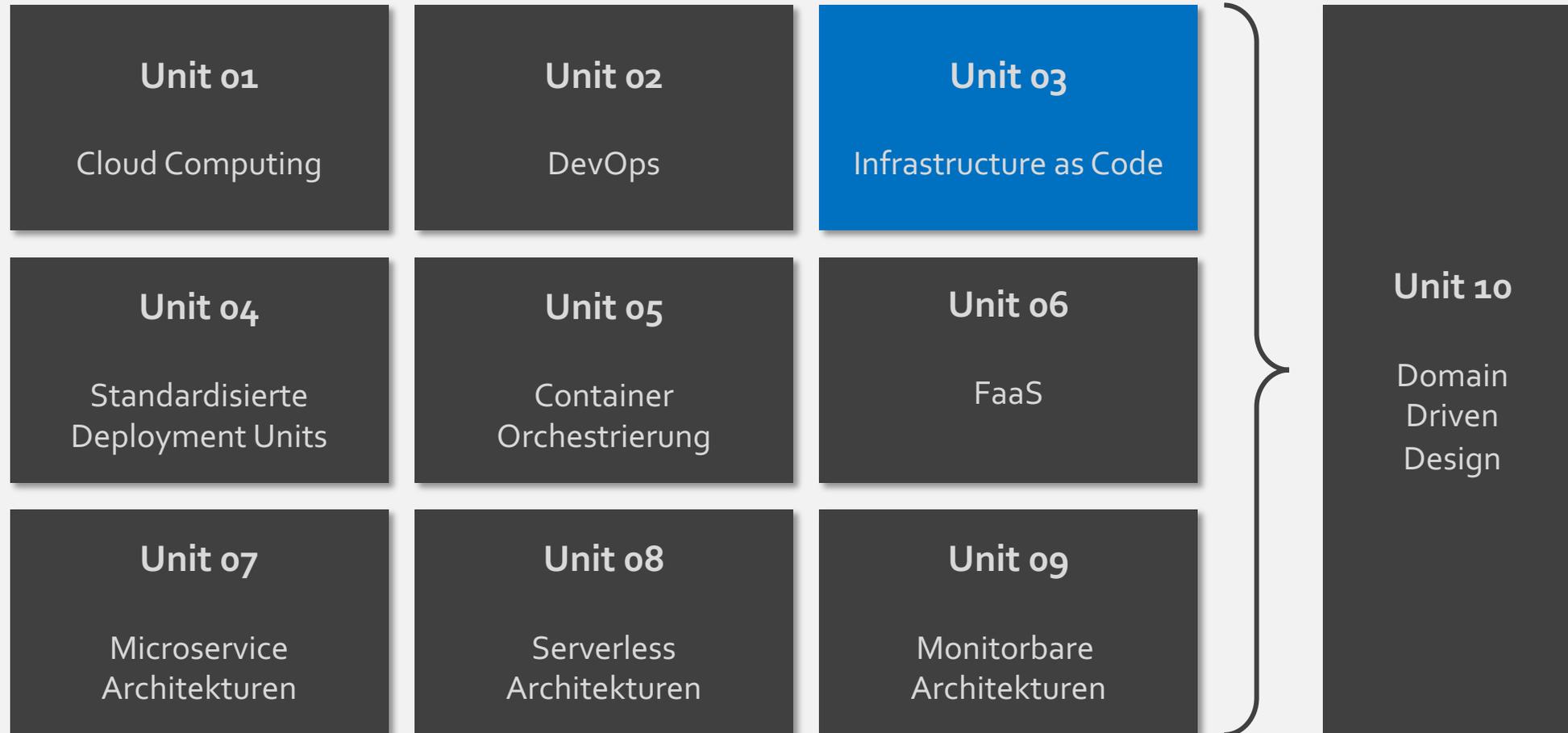
7.2 Provisionierung

- Immutable Infrastructure
- Infrastructure as Code
- Provisionierung von lokalen Umgebungen
- Provisionierung von Multi-Host Umgebungen

7.3 Zusammenfassung

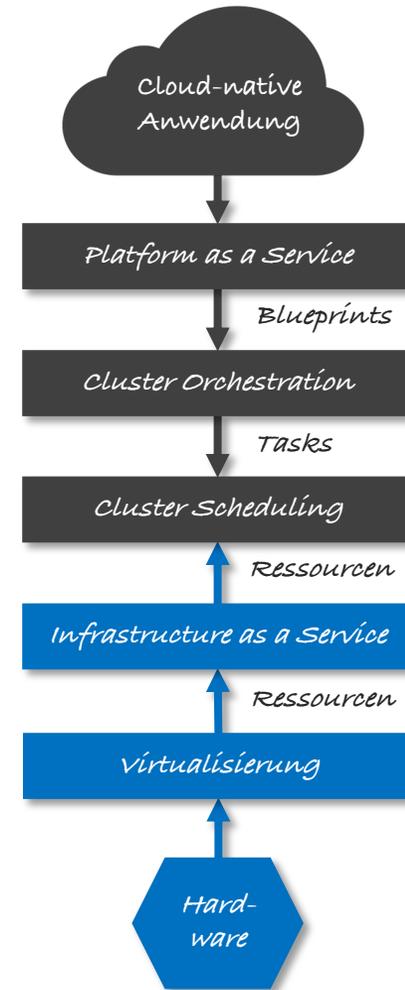
INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls



INHALTE

- **Virtualisierung**
 - Hardware-Virtualisierung (Typ-1, Typ-2)
 - Betriebssystem-Virtualisierung
 - Applikations-Virtualisierung
 - Emulation
- Infrastructure as a Service
- Provisionierung in IaaS-basierte Infrastrukturen
- Infrastructure as Code



VIRTUALISIERUNG

Unter Virtualisierung versteht man die Erzeugung von virtuellen Realitäten und deren Abbildung auf die physikalische Realität.

Zweck

- **Multiplizität:** Erzeugung mehrerer virtueller Realitäten innerhalb einer physikalischen Realität (z.B. mehrere VMs auf einem physischen Server).
- **Entkopplung:** Bindung und Abhängigkeit zur Realität auflösen (z.B. Verschiebung einer VM von einem Virtualisierungshost zu einem anderen zur Laufzeit).
- **Isolation:** Physikalische Seiteneffekte zwischen virtuellen Realitäten vermeiden (z.B. beeinträchtigt eine Endlosschleife auf einer VM nicht die Applikation auf einer anderen VM).



Wir verstehen in diesem Modul Virtualisierung immer als Virtualisierung von Hardware- oder Software-Ressourcen.

VIRTUALISIERUNGSARTEN

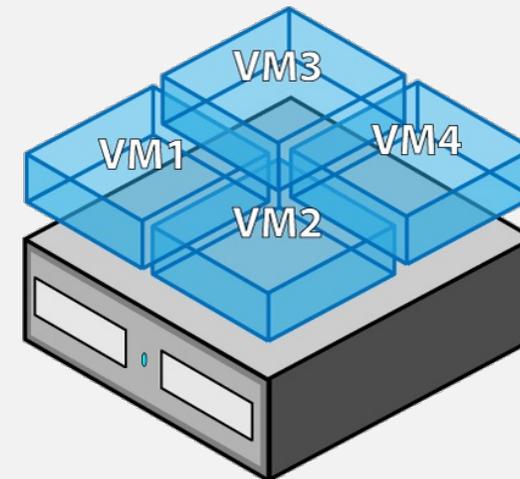
Unter Virtualisierung werden grundsätzlich verschiedene Konzepte und Technologien verstanden.

Virtualisierung von Hardware-Infrastruktur

1. Emulation
2. Voll-Virtualisierung (Typ-2 Virtualisierung)
3. Para-Virtualisierung (Typ-1 Virtualisierung)

Virtualisierung von Software-Infrastruktur

- Betriebssystem-Virtualisierung (Containerization)
- Anwendungs-Virtualisierung (Runtime, z.B. die Java Virtual Machine)



VIRTUALISIERUNG

Im Cloud Computing

Entkopplung von der (physischen) Hardware (Blech) für mehr Flexibilität und Robustheit bei Ausfällen.

Normierung von Ressourcen-Kapazitäten auf heterogener und wechselnder Hardware („S/M/L/XL-Instanzen“)

Zentrale Steuerung und Bereitstellung von Rechenressourcen über die mit Virtualisierung bereitgestellten Software-Defined-Resources.

Beispiel von Instanztypen des Public Cloud Service Providers AWS
Stand: 15.09.2020, <https://aws.amazon.com/de/ec2/instance-types/>

Instance	vCPU*	Arbeitsspeicher (GiB)	Speicher	Dedizierte EBS-Bandbreite (Mbit/s)	Netzwerkleistung
m4.large	2	8	Nur EBS	450	Mittel
m4.xlarge	4	16	Nur EBS	750	Hoch
m4.2xlarge	8	32	Nur EBS	1.000	Hoch
m4.4xlarge	16	64	Nur EBS	2.000	Hoch
m4.10xlarge	40	160	Nur EBS	4.000	10 Gigabit
m4.16xlarge	64	256	Nur EBS	10 000	25 Gigabit

HARDWARE-VIRTUALISIERUNG

Was wird virtualisiert?

Prozessor

- Virtuelle Rechenkerne (vCPU)
- Dispatching von Prozessor-Befehlen auf echte Rechnerkerne

Hauptspeicher

- Virtuelle Hauptspeicher-Partition
- Management der realen Repräsentation (im RAM, Memory-Ballooning)

Netzwerk

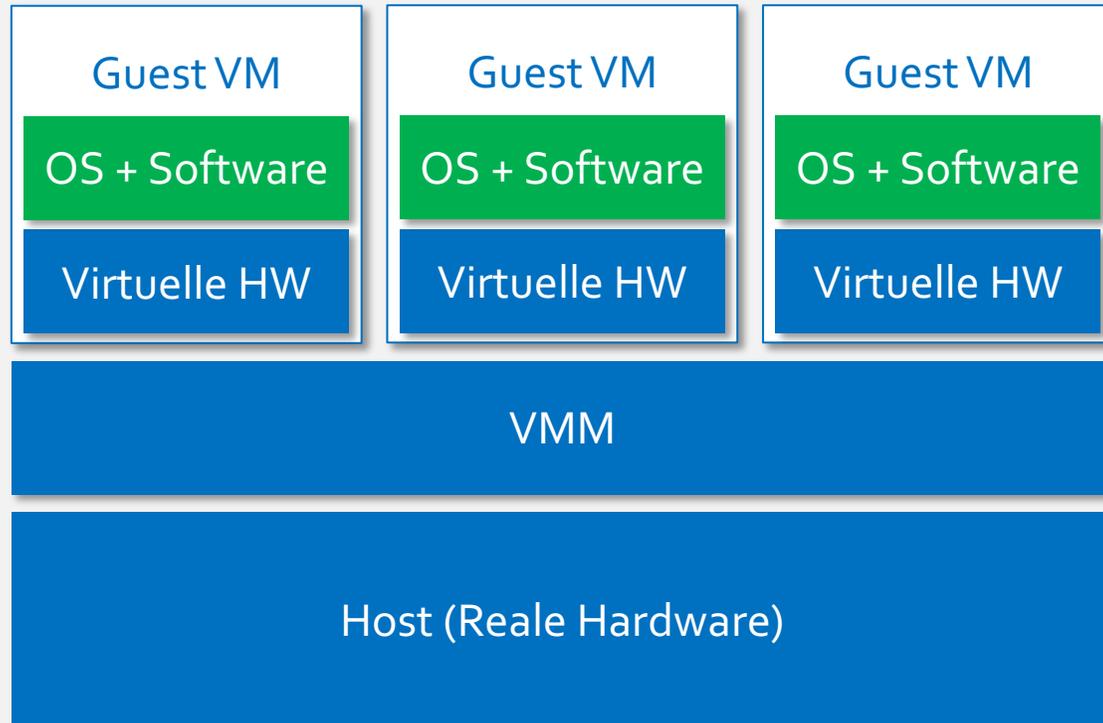
- Virtuelle Netzwerkschnittstellen und virtuelle Netzwerk-Infrastrukturen (VLAN)
- Brücken zwischen virtuellen und realen Netzen
- Firewallregeln

Storage

- Virtuelle Festplattenlaufwerke (Blockstorage).
- Abbildung auf Dateien im realen Dateisystem.
- Virtuelle Storage Area Networks (Aufteilung der Daten eines virtuellen Laufwerks auf viele Storage Einheiten).

HARDWARE-VIRTUALISIERUNG

Wie funktioniert HW-Virtualisierung?



Host: Der physikalische Rechner, der eine oder mehrere VM ausführt und entsprechende HW-Ressourcen zur Verfügung stellt.

Guest: Eine laufende (oder lauffähige) virtuelle Maschine (VM).

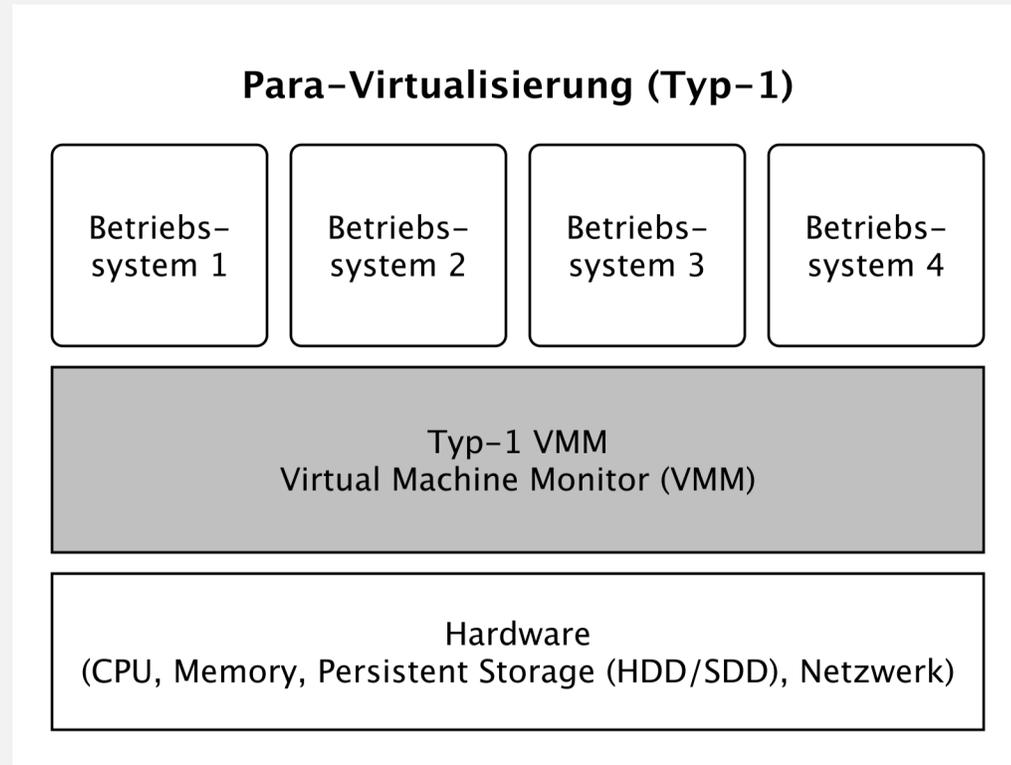
VMM (Virtual Machine Monitor): Die Steuerungssoftware zur Verwaltung von Guests und Host-Ressourcen.

- Durch Hardware-Virtualisierung werden die Ressourcen eines Rechnersystems aufgeteilt und von mehreren Betriebssystem-Instanzen genutzt.
- Anforderungen der Betriebssystem-Instanzen werden von der Virtualisierungssoftware (Virtual Machine Monitor, VMM) abgefangen und auf die reale vorhandene Hardware umgesetzt.

HARDWARE-VIRTUALISIERUNG

Para-Virtualisierung (Typ – 1)

- Der Hypervisor läuft direkt auf der verfügbaren Hardware. Er entspricht somit einem auf Virtualisierung spezialisiertem Betriebssystem.
- Das Gast-Betriebssystem muss um virtuelle Treiber ergänzt werden, um mit dem Hypervisor interagieren zu können.
- Wegen der erforderlichen Hypervisortreiber können nicht beliebige Guest OS auf Hypervisoren laufen.
- Der Hypervisor nutzt die Treiber eines Host-Betriebssystems, um auf die reale Hardware zuzugreifen. Damit muss man für den Hypervisor nicht aufwändig eigene Treiber entwickeln.



Leistungsverlust:
Ca. 2-3%



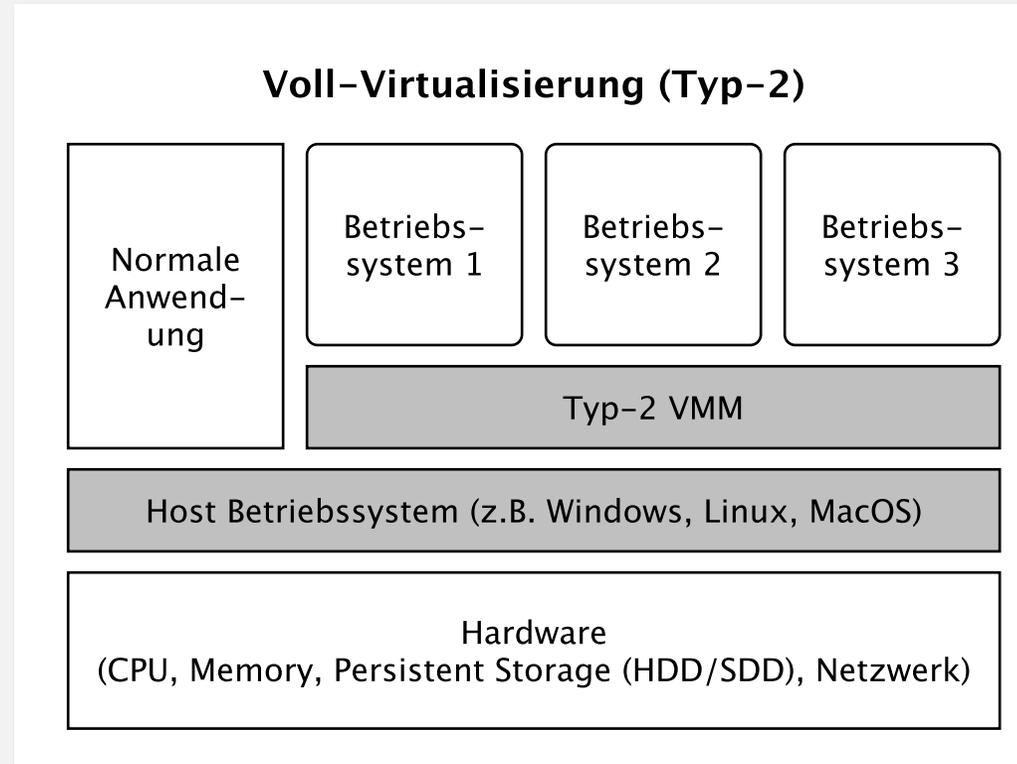
Microsoft
Hyper-V



HARDWARE-VIRTUALISIERUNG

Voll-Virtualisierung (Typ – 2)

- Jedem Gastbetriebssystem steht ein eigener virtueller Rechner mit virtualisierten Ressourcen (CPU, RAM, Disks, Netzwerkkarten, etc.) zur Verfügung.
- Der VMM läuft hosted als Anwendung unter dem Host-Betriebssystem.
- Der VMM verteilt die Hardwareressourcen des Rechners an die VMs.
- Teilw. muss die VMM Hardware emulieren, die nicht für den gleichzeitigen Zugriff mehrere Betriebssysteme ausgelegt ist (z.B. Netzwerkkarten, Grafikkarten).



Leistungsverlust:
Ca. 5-10%



VirtualBox

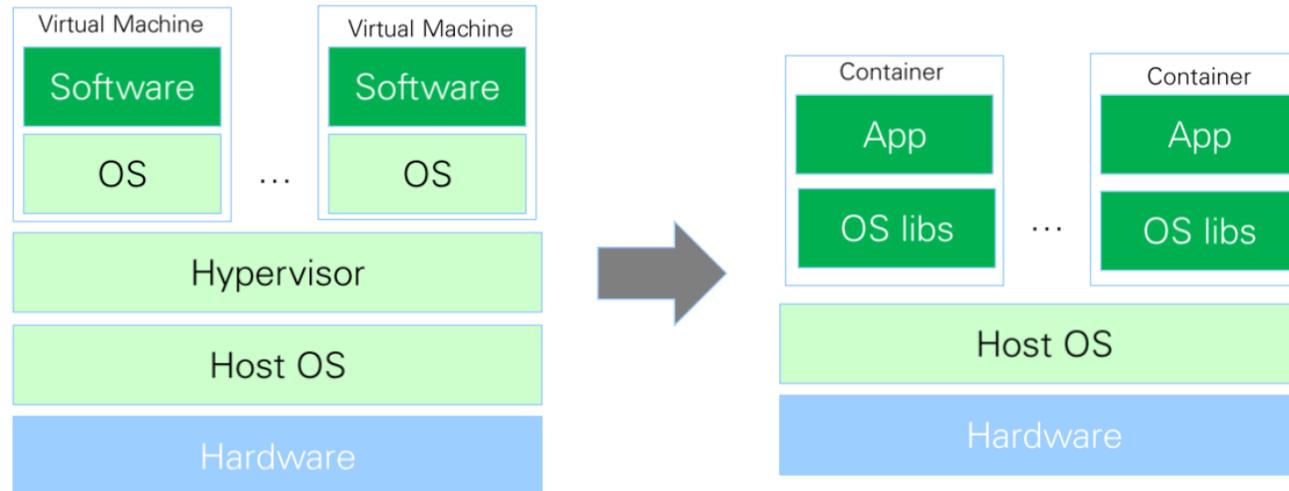


BETRIEBSSYSTEM-VIRTUALISIERUNG

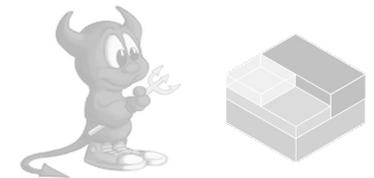
Leichtgewichtiger Virtualisierungsansatz mittels OS-Containern

Es gibt keinen Hypervisor. Jede Applikation läuft direkt als Prozess im Host-Betriebssystem.

Dieser Prozess ist jedoch mittels OS-Mechanismen isoliert.



Leistungsverlust:
CPU-/RAM-Overhead in der Regel nicht messbar (~0%)



OpenVZ

SOLARIS

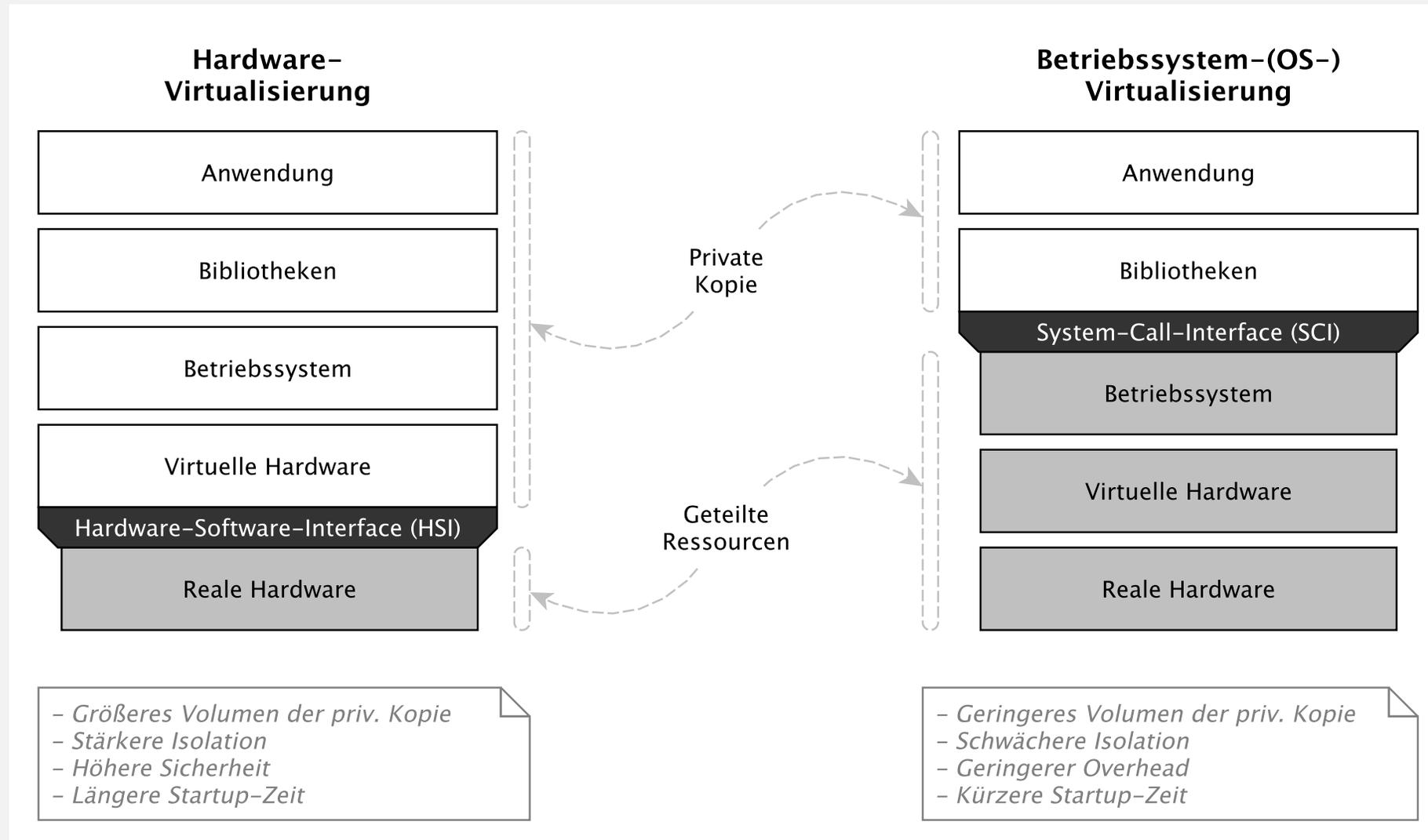
- Free BSD Jails (2000)
- Solaris Zones (2005)
- Linux OpenVZ (2005)
- Linux LXC (2008)
- und mehr ...

- Isolation des Prozesses durch Namespaces (bzgl. CPU, RAM, Disk I/O) und Containments
- Isoliertes Dateisystem
- Eigene Netzwerkschnittstelle
- Startup-Zeit = Startdauer für den ersten Prozess (kein Boot des OS erforderlich!)

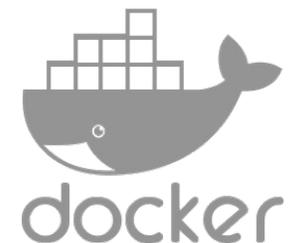
Docker v0.0.1
release: 2013

HARDWARE- VS. OS-VIRTUALISIERUNG

Trade-off zwischen Isolation (Sicherheit) und Ressourcen-Effizienz



Mit OS-Virtualisierung (Containern) werden wir uns noch intensiv in **Unit 04** und **Unit 05** befassen.



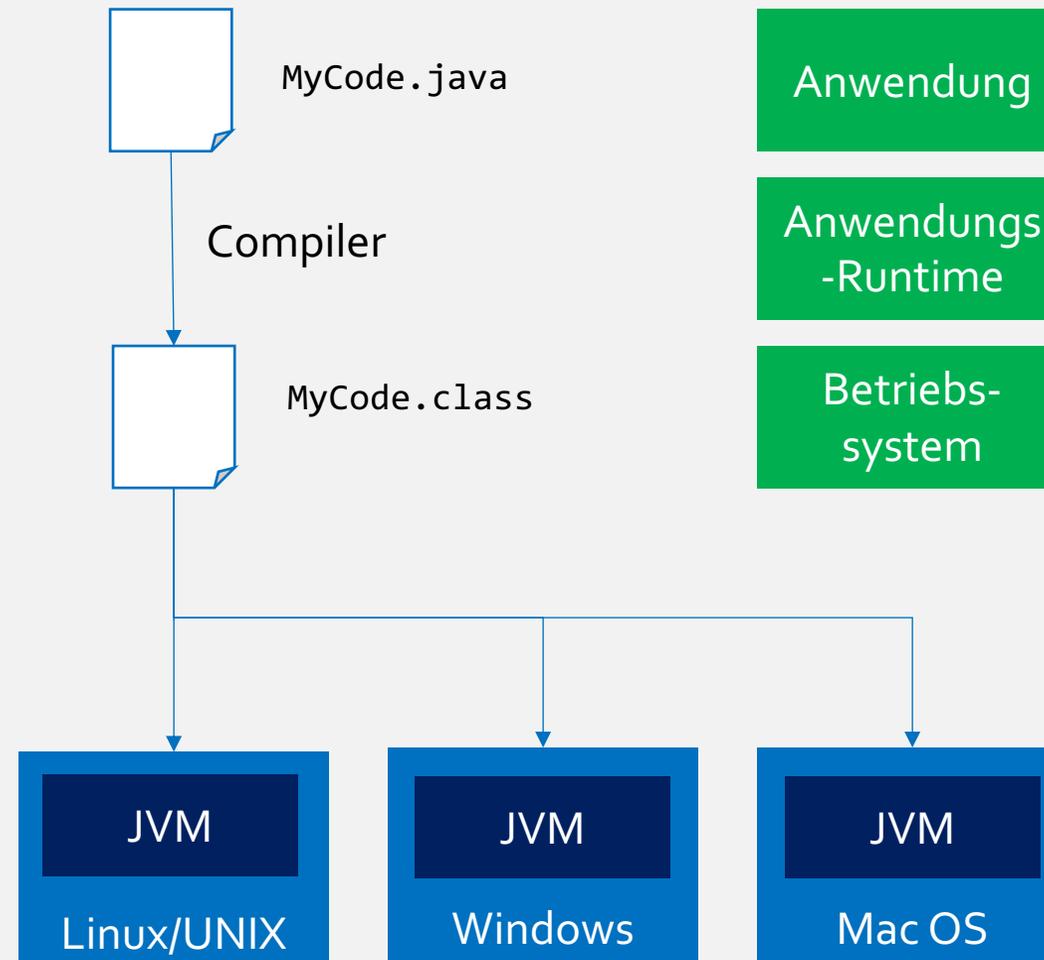
EMULATION VS ANWENDUNGS-VIRTUALISIERUNG

Anwendungs-Virtualisierung

Anwendungsvirtualisierung stellt Anwendungen eine Programmierschnittstelle und eine Laufzeitumgebung (Runtime) zur Verfügung, die (möglichst) komplett vom darunter liegenden Betriebssystem entkoppelt.

Zweck: OS-unabhängige Anwendungen.

Beispiel: Java Virtual Machine



Diese Art der Virtualisierung wird mittlerweile von vielen modernen Programmiersprachen genutzt.

Die damit einhergehenden Leistungsverluste werden billiger in Kauf genommen (siehe Java).

AUSBLICK

- Virtualisierung
- **Infrastructure as a Service**
 - Definition, Eigenschaften, Marktüberblick
 - Private Cloud Infrastrukturen
 - Public Cloud Infrastrukturen (am Bsp. des Typvertreeters AWS)
- Provisionierung in IaaS-basierte Infrastrukturen
- Infrastructure as Code

KONTAKT

Disclaimer

Nane Kratzke

📞 +49 451 300-5549

✉ nane.kratzke@th-luebeck.de

🌐 kratzke.mylab.th-luebeck.de

