Fachhochschule Nordwestschweiz
Hochschule für Technik

Application Performance Management

Clustering & High Availability

Michael Faes

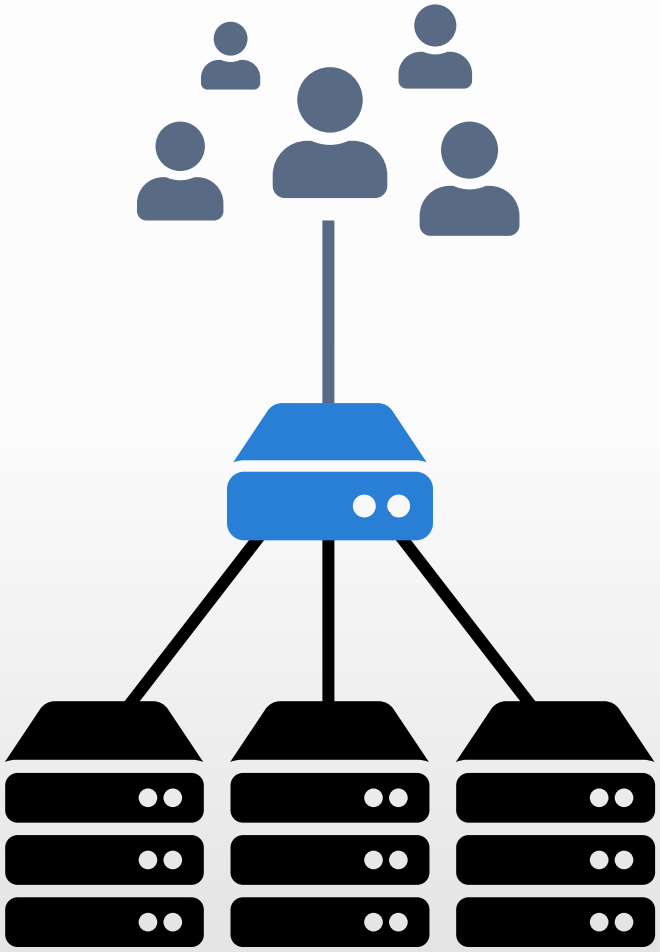
Rückblick: Load Balancing

Verteilen von Requests auf mehrere Server

Neue Komponente: *Load Balancer*

Herausforderungen:

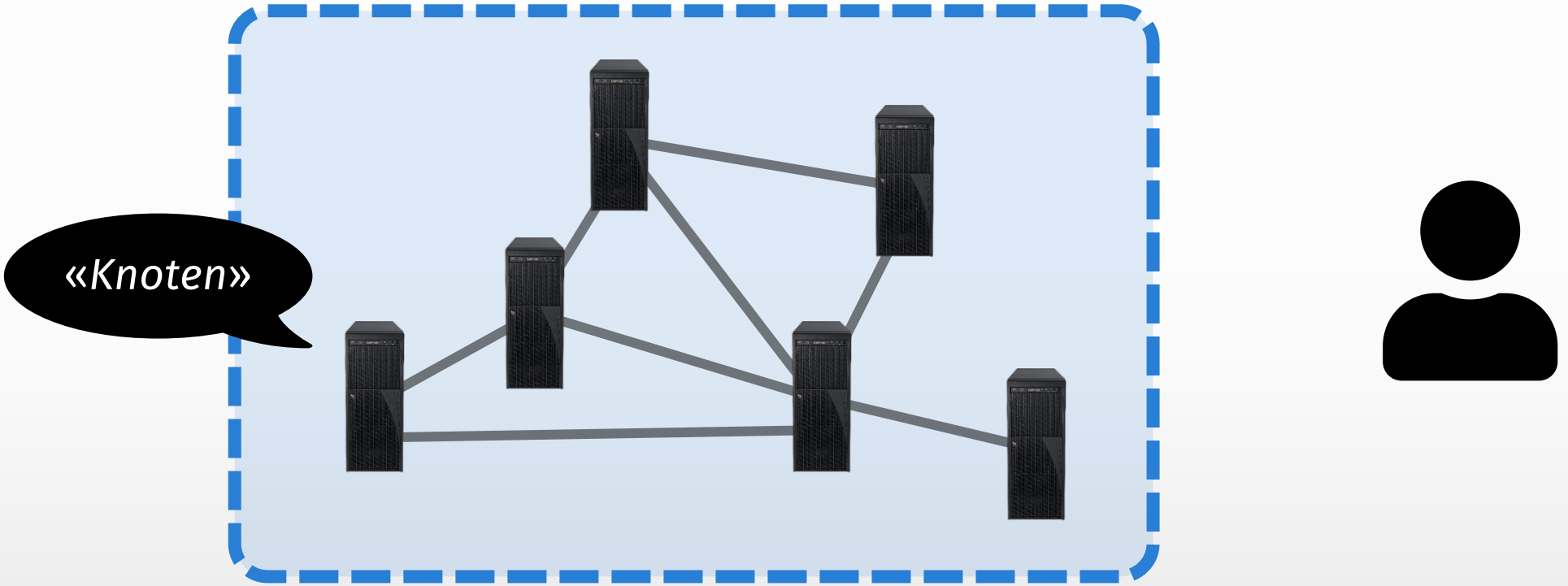
- Monitoring
- Persistenz
- Heute: **Verfügbarkeit**



Übersicht

1. Rückblick Load Balancing
2. Clustering & High Availability
   - Failover
   - Anwendungsanforderungen
   - Virtualisierung und Container
3. Übung

Was ist ein «Cluster»?



Cluster: Mehrere vernetzte Rechner (*Knoten*), die zusammen arbeiten, und die man im Prinzip als ein System ansehen kann.

Arten von Clusters

Compute Cluster

- Rechenpower für High-Performance Computing
- Applikationen werden explizit parallelisiert

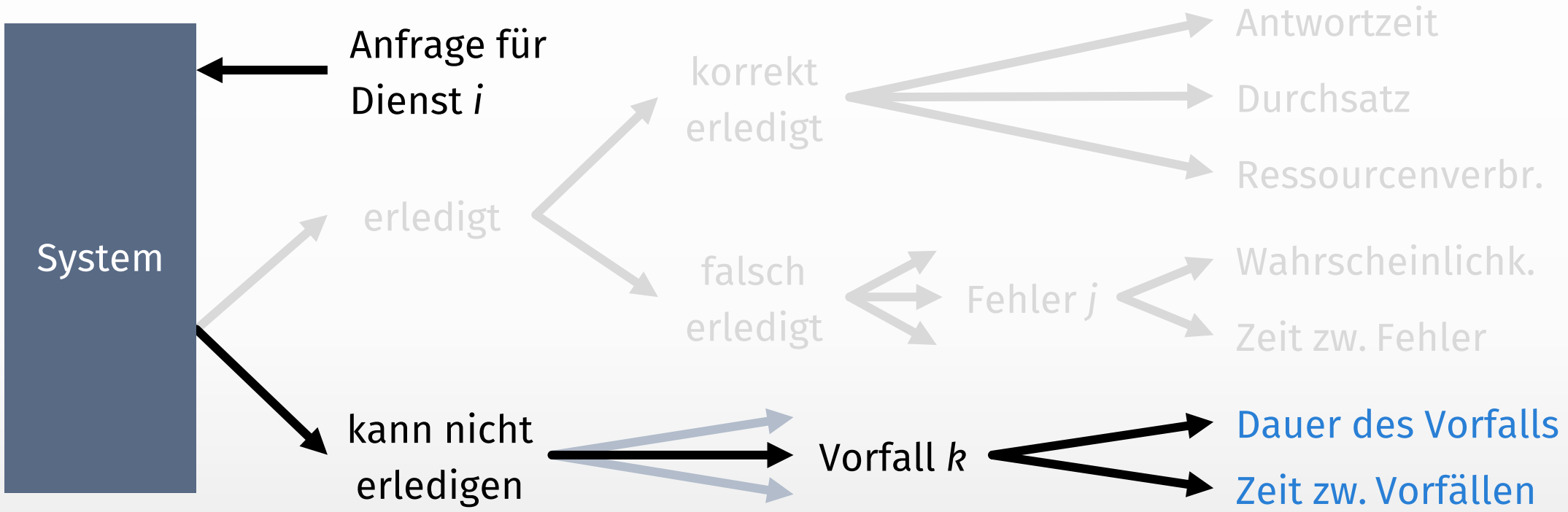
Load-Balancing Cluster

- Verteilen von «gleichartigen» Requests
- In unserem Setup: die Web-Server

High-availability (HA) Cluster

- Wenn Knoten ausfällt, übernimmt ein anderer
- In unserem Setup: noch nötig für die Load Balancers

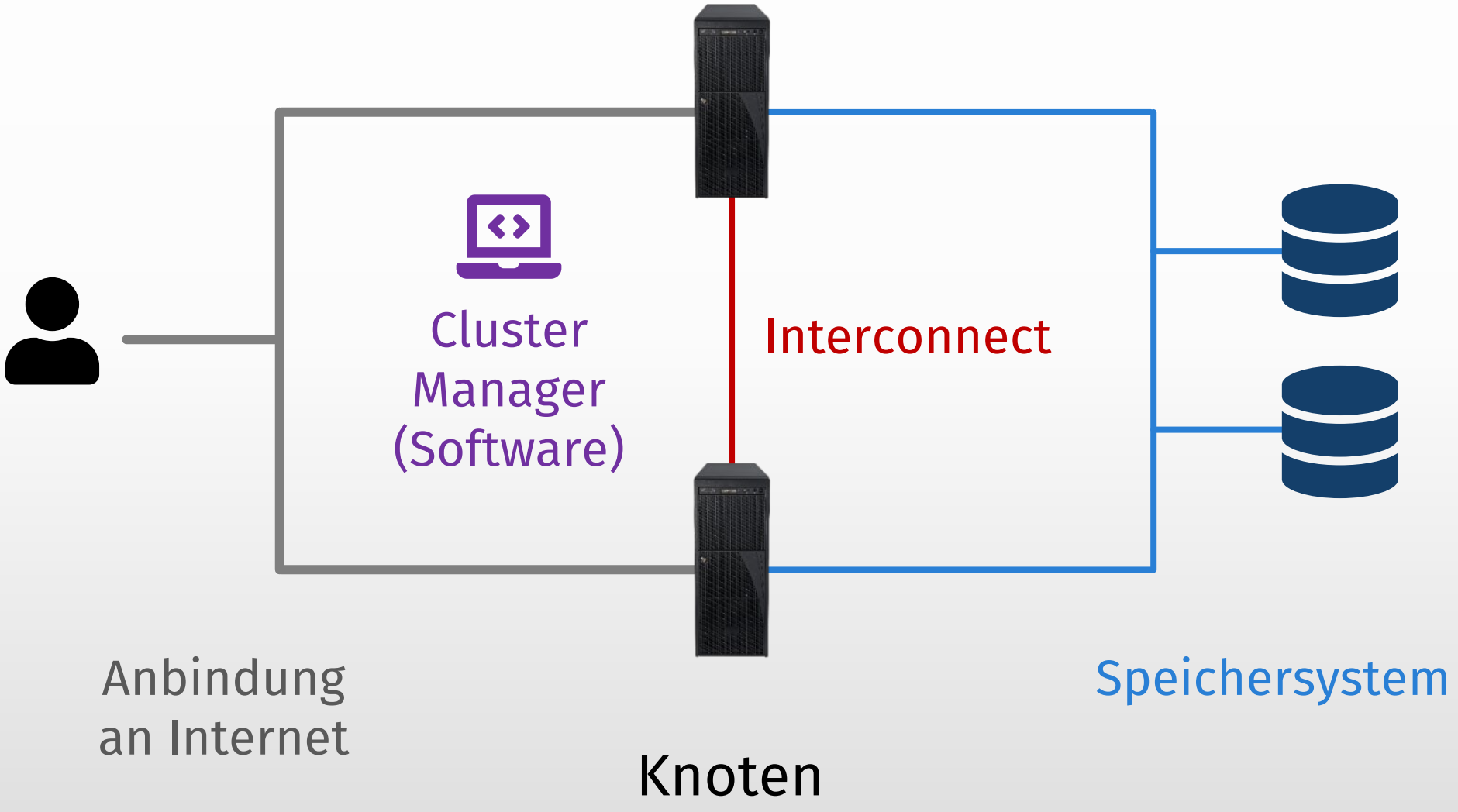
Rückblick: Performance-Metriken



Verfügbarkeit (availability)

Anteil der Zeit, während der das System Anfragen beantwortet, z. B. 99%

Komponenten eines HA-Clusters



Failover

Interconnect wird für *Heartbeat* verwendet: regelmässiges Signal, welches anderen Knoten mitteilt, dass ein Knoten noch verfügbar ist.

- Sobald ein Knoten den Heartbeat eines anderen eine bestimmten Zeitraum nicht mehr erhält, übernimmt er dessen Rolle: *Failover*.

Beispiel: Zwei Load Balancers, ein Primär-/Master- und ein Sekundär-/Backup-Load-Balancer. Sobald Master ausfällt, *übernimmt* Backup.

Herausforderungen

- Alle Knoten müssen stets aktuellen Zustand haben/synchronisieren
- Andere Komponenten müssen über Failover «informiert» werden
  Beispiel: VRRP (siehe Übung)

Anwendungsanforderung

1. Einfacher Weg, App zu starten, stoppen und Status abzufragen (automatisierbar!)
2. App muss geteilten Speicher verwenden können
3. Zustand der App muss möglichst oft und vollständig auf persistentem Speicher gesichert werden
4. Keine Datenkorruption bei Crash oder Neustart

→ **Viele der Anforderungen können durch *Virtualisierung/Containerisierung* minimiert werden**

Virtualisierung und Container

Virtualisierung: Prinzip

Applikation

Betriebssystem

Maschine

Applikation | ... | ...

Betriebssystem | ... | ... «Guest OS»

Virtuelle Maschine | ... | ... «Guest»

Hypervisor

Reale Maschine «Host»

Funktionsweise

Einfachste Technik: *Emulation*

- Gast-Instruktionen werden von Software *interpretiert*
- Kann beliebige Hardware virtualisieren (z. B. ARM auf x86-64 CPU)

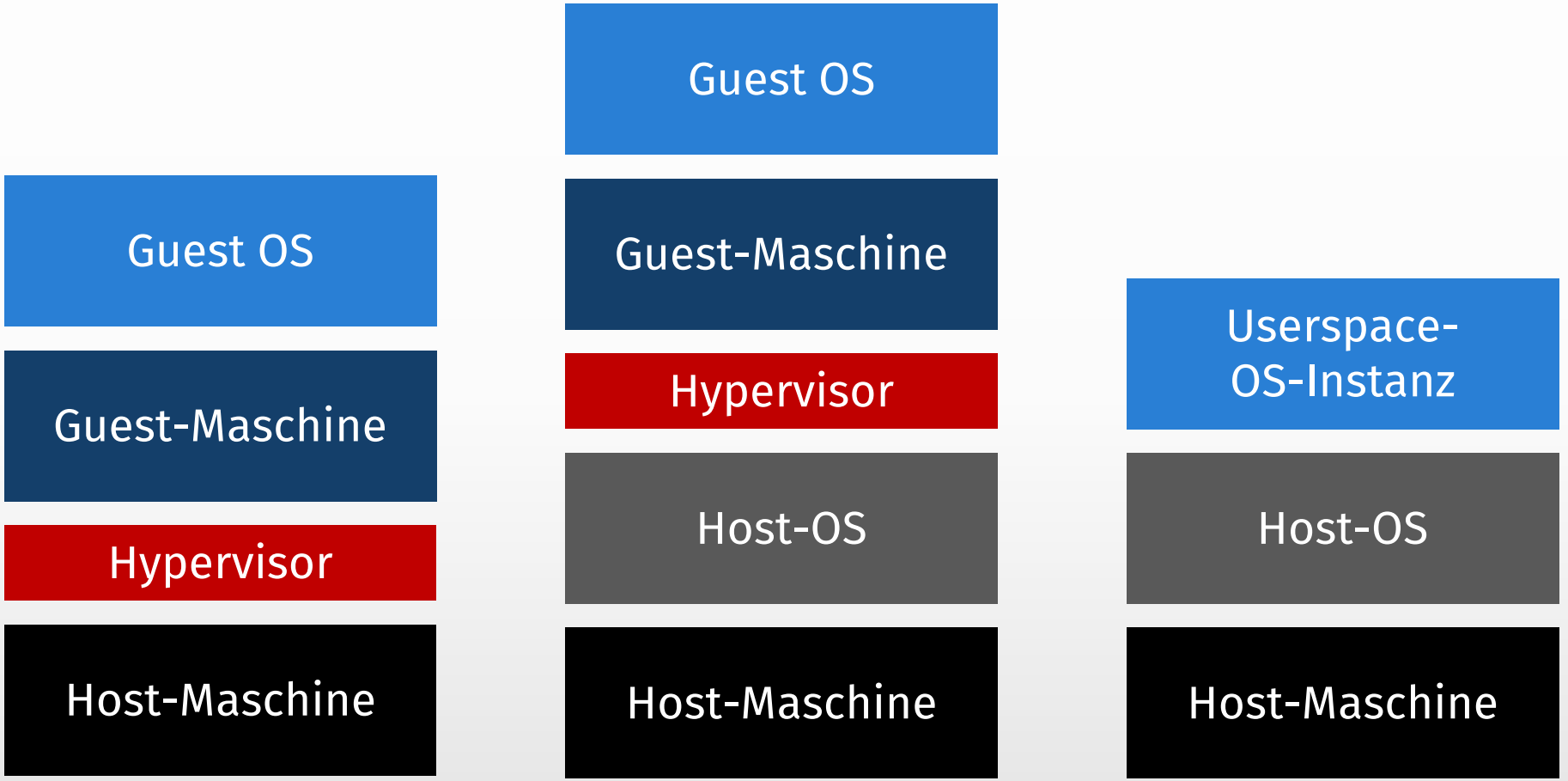
Langsam! Bei gleichem *Instruction Set* (z. B. x86-64), können Code-Stücke stattdessen direkt auf Host-CPU ausgeführt werden

- Nicht alle Instruktionen! Gewisse sind Kernel vorbehalten («Ring 0»)

Hardware-assisted Virtualisation erlaubt noch effizientere Ausführung

- CPU gaukelt dem Gast vor, er würde in Ring 0 laufen, aber schützt Host-OS von unerwünschten Änderungen
- Beispiele: Intel VT-x, AMD-V

Arten von Virtualisierung



«Typ-1»
(Bare-Metal) Hypervisor

«Typ-2»
Hypervisor

Containerisierung

Virtualisierung vs. Containerisierung

Virtualisierung

- Beliebige Host/Guest-Kombinationen möglich (z. B. x64-Windows / ARM-Mac)
- Images sind portabel
- Images sind schwergewichtig, da vollständiges OS enthalten
- Langsamer Start durch Booten
- Sicherheit durch Guest- & Host-Kernel & Hypervisor

Containerisierung

- Guest-Architektur und -Kernel müssen zu Host passen (nur x64-Linux / x64-Linux)
- Images sind Plattform-spez.
- Images sind leicht, da nur App & Userspace-Abhängigk. drin
- Schneller Start
- Sicherheit: abhängig von Kernel und Konfiguration...

Bedeutung für Clustering/Cloud

Ressourcen-Pooling

NIST: «*Computing-Ressourcen werden zusammengelegt, um mehrere Kunden mit denselben physischen Ressourcen zu bedienen.*»

- CPU & RAM: Hypervisor kann Limiten für VMs/Container festlegen
- Speicherplatz: Zentralisierter Speicher für alle Gäste, wird nach Bedarf aufgeteilt
- *Thin Provisioning*: Speicherplatz wird Guest zugeschrieben, aber erst alloziert, wenn er wirklich verwendet wird
- *Deduplication*: Identische Blöcke von VMs werden nur 1x gespeichert

Live Migration

RAM-Inhalt und Netzwerk-Verbindungen können beibehalten werden, wenn VM von Host zu Host migriert wird

Fragen?