# Parallele und verteilte Systeme

# Einführung

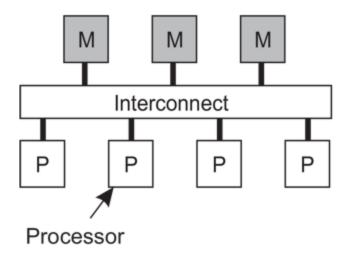
#### «Parallele Systeme»

- Eine single core CPU kann nur einen Prozess gleichzeitig ausführen
- Multi-core CPUs entsprechend mehrere gleichzeitig
- Ausser in sehr einfachen Embedded Systemen müssen jedoch immer sehr viele Prozesse «gleichzeitig» ausgeführt werden können z.B. auf einem Server oder auf einem Desktop Computer

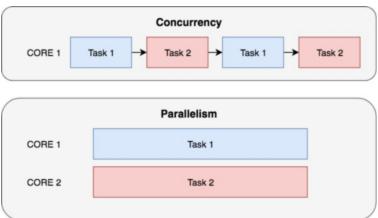
### «Parallele Systeme»

- Viele verschiedene Prozesse (tausende) werden von einem oder mehreren (bis zu dutzenden) Prozessoren ausgeführt
- Ein einzelner Prozessor kann demnach nacheinander mehrere Prozesse bearbeiten
- Die Prozessoren befinden sich auf demselben Chip oder auf dem selben Mainboard
- Sie haben geteilten sowie gemeinsamen Speicher
- Die Verbindung zwischen ihnen (Interconnect) hat geringe Latenz, hohe Bandbreite und ist zuverlässig.

Shared memory



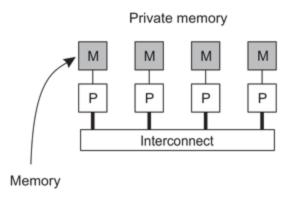
- Parallele Ausführung (parallelism): Mehr als eine Aufgabe wird gleichzeitig ausgeführt
- Nebenläufig (concurrency): Mehr als eine Aufgabe wird abgearbeitet (durch schnelles context switching)



•	Eine zentrale Aufgabe von Betriebsystemen ist es, die Prozesse auf die CPUs zu verteilen.
•	Dies wird «Scheduling» genannt.

#### **Verteilte Systeme**

«A distributed system is a collection of independent computers that appears to its users as a single coherent system.»



VanSteen, 2017, S. 26

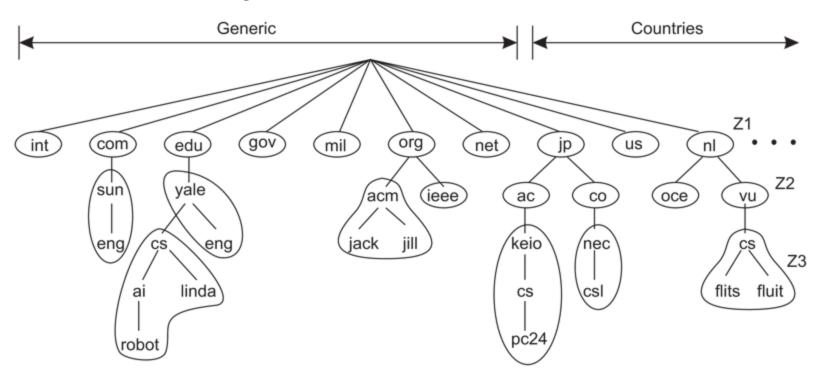
P: Prozessor,

Interconnect: Netzwerkverbindung, meistens HTTP, UDP/TCP, IP, Ethernet basiert

## **Resource Sharing**

- Ressourcen verfügbar machen: Drucker, Computing, Storage, Daten, Netzwerk
- Teure Ressourcen können besser ausgelastet werden und müssen nicht mehrfach angeschafft werden
- Zusammenarbeit

## **Domain Name System**

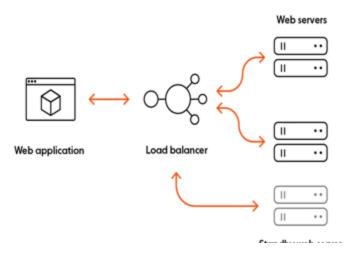


### Anforderungen an moderne Software

- Hohe Verfügbarkeit
- Skalierbarkeit
- Im Katastrophenfall sollen die Systeme schnell wiederhergestellt werden können
- Soll funktionieren, auch wenn Teile des Systems Offline sind (Resilienz)
- Kostengünstig
- Einfach
- Updates müssen einfach eingespielt werden können

## Lösungsansätze

- Verfügbarkeit, Skalierbarkeit: Mehre identische Systeme müssen verfügbar sein und bei Bedarf sollen weitere schnell gestartet werden können
- Tradeoff: Kostengünstig, Einfach



## **Decentralized vs Distributed**

#### **Decentralized**

- Matrix
- Mastodon
- Nextcloud
- ...

#### **Distributed**

- CockroachDB
- Neon
- Ably
- ...

# **Cloud Computing**

The entire history of software engineering is that of the rise in levels of abstraction.

-- Grady Booch

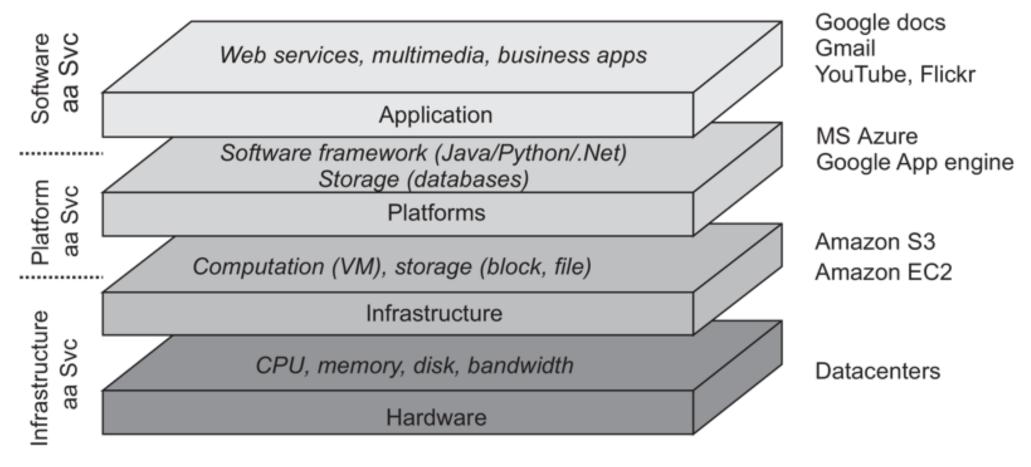
# New Pizza as a Service

Traditional Infrastructure Platform Software On-Premises as a Service as a Service as a Service Deployment (laaS) (PaaS) (SaaS) Kitchen Kitchen Kitchen Kitchen Gas Gas Gas Gas Oven Oven Oven Oven Pizza Dough Pizza Dough Pizza Dough Pizza Dough Toppings Toppings Toppings Toppings Cook the Pizza Cook the Pizza Cook the Pizza Cook the Pizza Walk-In-and-Bake Made In-House Kitchen-as-a-Service Pizza-as-a-Service

You Manage

Vendor Manages

#### **Abstractions**



(VanSteen, 2017, S. 30)

## XaaS

laaS	CaaS	PaaS	FaaS	
Functions	Functions	Functions	Functions	Customer Managed
Application	Application	Application	Application	Customer Managed Unit of Scale
Runtime	Runtime	Runtime	Runtime	Abstracted by Vendor
Containers (optional)	Containers	Containers	Containers	
Operating System	Operating System	Operating System	Operating System	
Virtualization	Virtualization	Virtualization	Virtualization	
Hardware	Hardware	Hardware	Hardware	

#### **Fallstudie**





# **Architectures**

