

## Netzwerkspeicher und Dateisysteme

Systemausbildung – Grundlagen und Aspekte von Betriebssystemen  
und systemnahen Diensten

9. Juni 2021, Gregor Longariva



## Wir starten um 14:15 Uhr...



Das Mikrophon ist  
stummgeschaltet



Gerne die eigene  
Kamera aktivieren



Während des Vortrags  
gerne Fragen stellen



Dieser Vortrag wird  
nicht aufgezeichnet

# Agenda





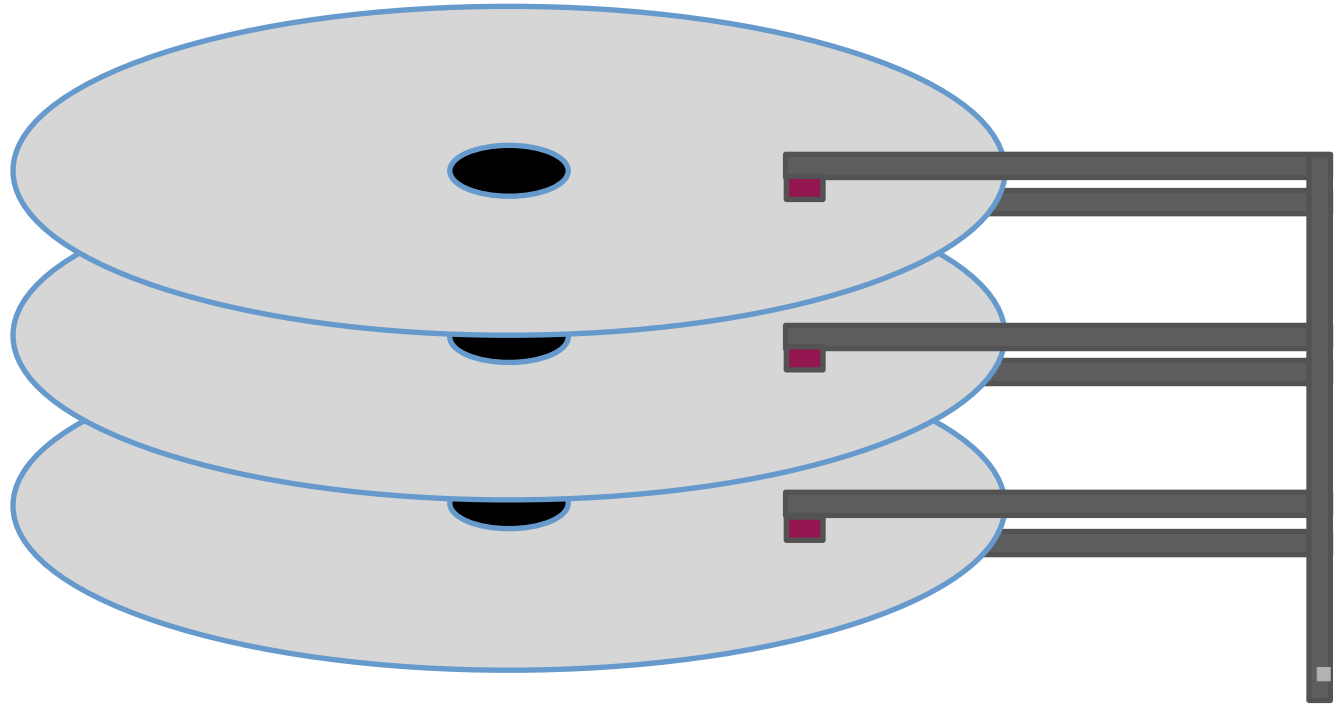
# FESTPLATTEN



## Prinzipieller Aufbau einer klassischen Festplatte

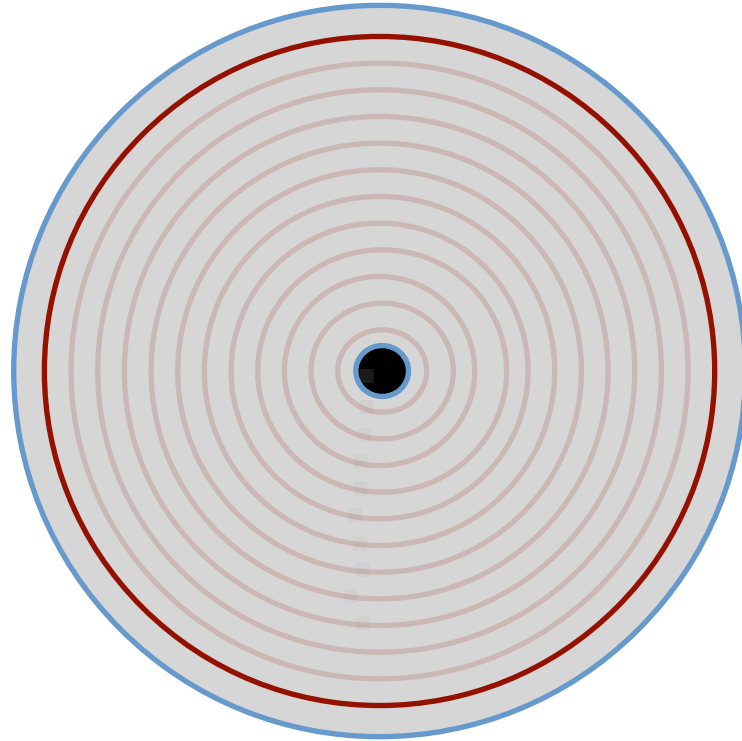


# Aufbau einer Festplatte

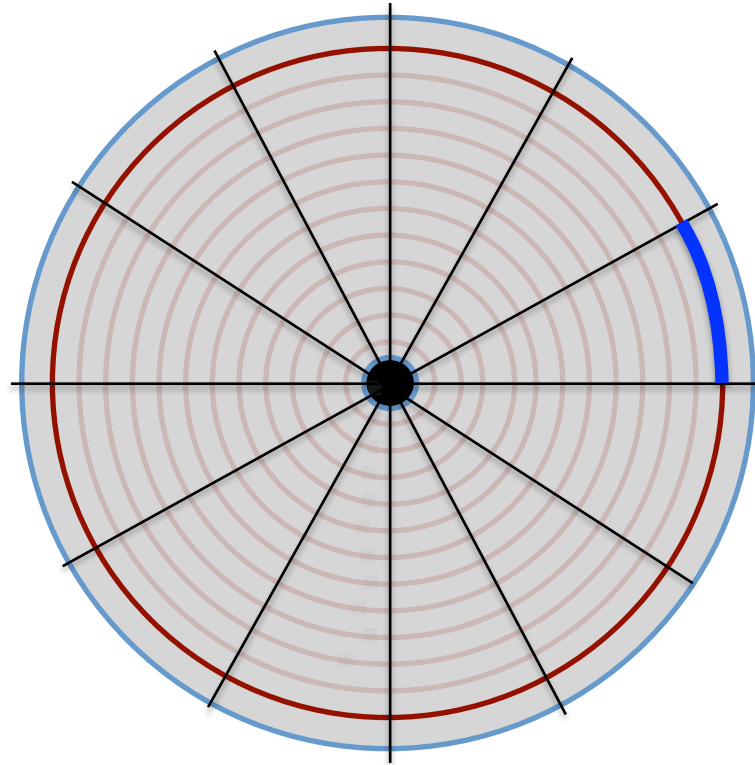


# Aufbau einer Festplatte

**Spur**



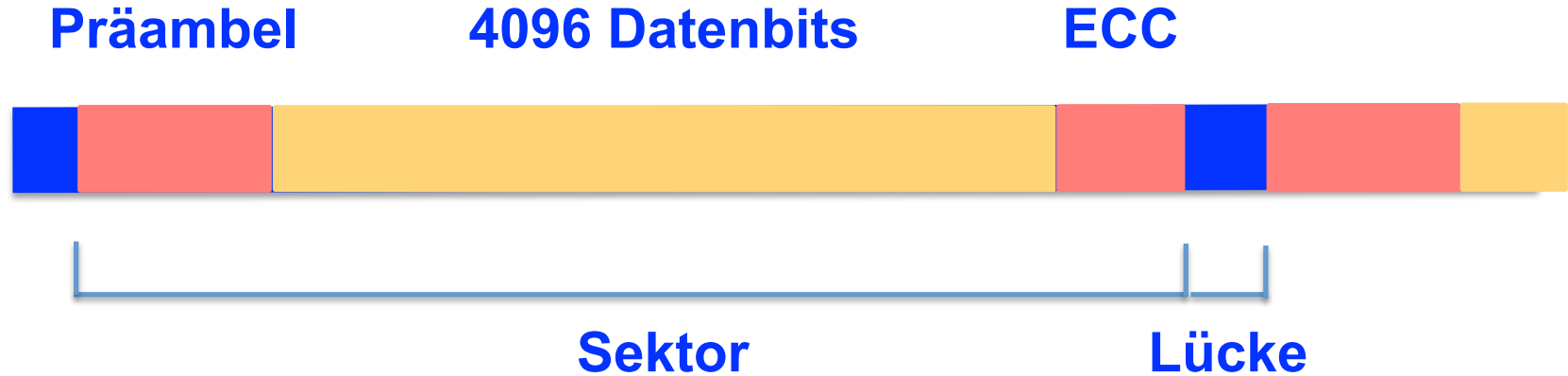
# Aufbau einer Festplatte



**Spur**

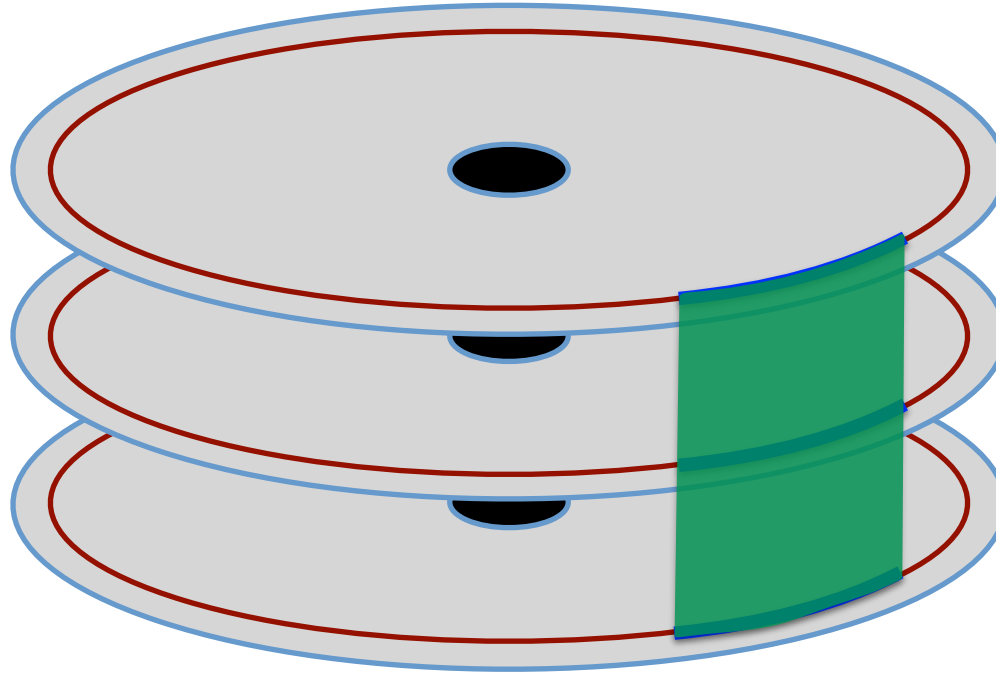
**Sektor**

# Aufbau einer Festplatte





# Aufbau einer Festplatte

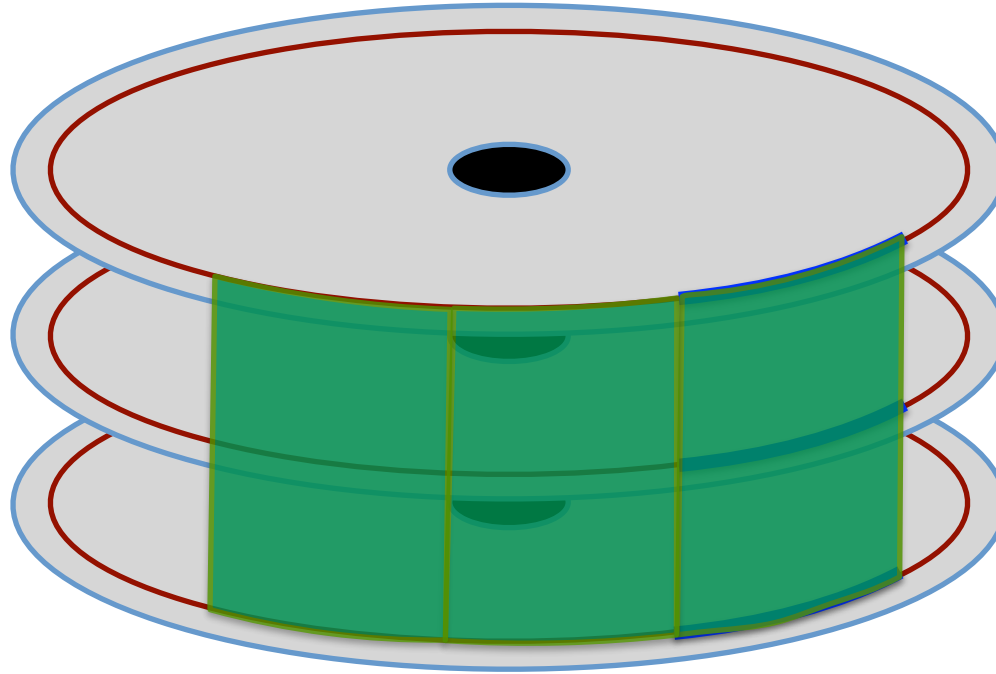


**Spur**

**Sektor**

**Zylinder**

# Aufbau einer Festplatte



**Spur**

**Sektor**

**Zylinder**



# FESTPLATTEN



Geschwindigkeit einer klassischen Festplatte

# Wie schnell ist eine Platte (worst case)?

Festplatte mit 15k (= **15.000** Umdrehungen / Min)

Latenz:  $60 \text{ sec} / 15.000 = 0,004 \text{ sec} \rightarrow 4\text{ms}$

IOPS:  $1 \text{ Operation} / 0,004 \text{ sec} = 250 \text{ Ops} / \text{sec}$

Bandbreite:  $250 \times 4096 \text{ Bytes pro Sektor} = 1.024.000 \text{ bytes} / \text{sec}$

**1MByte pro Sekunde!**



# Wie schnell ist eine Platte (best case)?

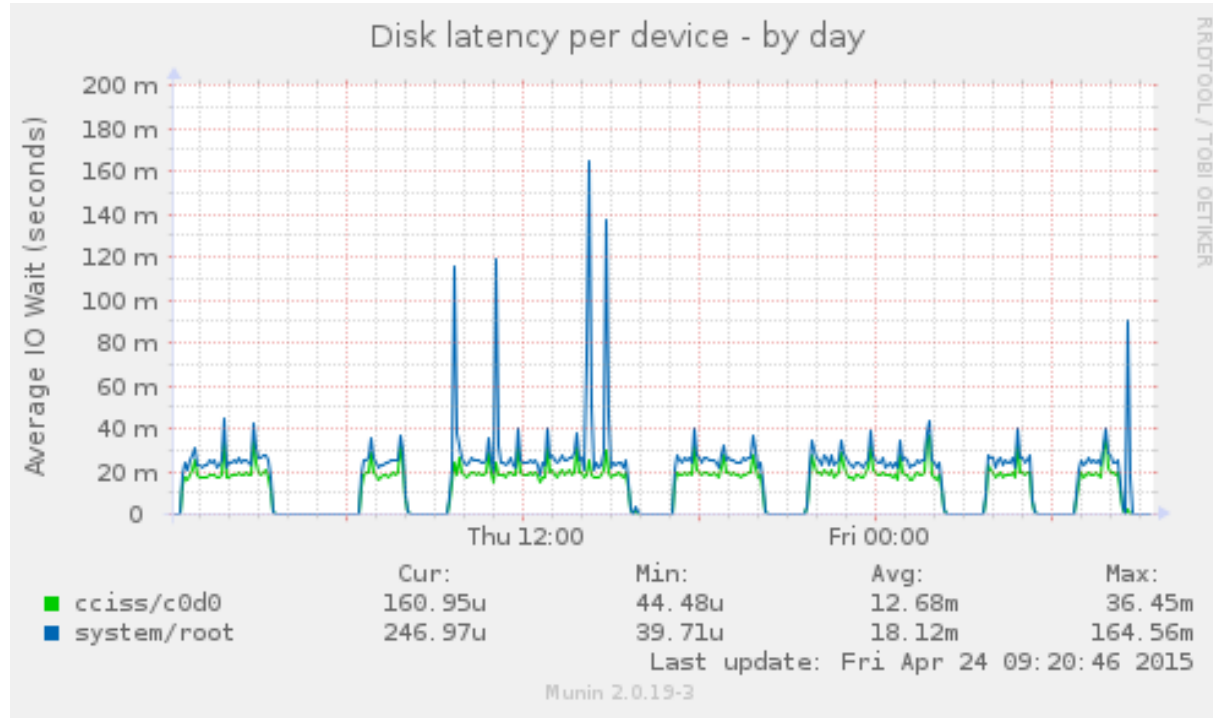
**1.024.000** bytes / sec x 6 Köpfe = **6.144.000** Bytes / sec

**6.144.000** Bytes / sec x 30 (Zylinder pro Cluster) =  
**184.320.000** Bytes / sec

**also ca. 180 MByte pro Sekunde**

(aber immer noch ohne Plattencaches)

# Plattenzugriffe beschleunigen: Beispiel mit und ohne Controller-Cache



# "elektronische" Speichermedien

- Keine mechanisch bedingten Latenzen
- Potentiell höhere Bandbreiten - Random I/O weniger „schmerzhaft“

Aber:

- Limitierte Anzahl Schreib-Zyklen pro Zelle!
- "wear-leveling" notwendig
  - › Gleichmäßige Verteilung von Schreiboperationen für max. Lebensdauer
- Verantwortlich: eigener Controller
- Problem: nicht länger genutzte Sektoren erkennen
  - › Neues Kommando: TRIM

# Vergleich Speichermedien: mechanisch / SSD / NVDIMM

Typ	Latenz+Seek (Theorie)	IOPs (Theorie)*	Bandbreite (Theorie)**	R/W IOPs (Praxis)
3,5" 15k SAS	4 ms	250	1.000 KB/s	180 / 165
2,5" 15k SAS	4 ms	250	1.000 KB/s	200 / 190
2,5" 10k SAS	6 ms	166	664 KB/s	150 / 140
2,5" 7.2k SATA	8 ms	120	480 KB/s	80 / 74
2,5" 5.4k SATA	11 ms	90	360 KB/s	52 / 50
2,5" eMLC SSD SAS	0.5 ms	-	-	~ 50.000
SSD NVMe (TOP!)	0.01 ms	-	-	~ 300.000
NVDIMM(-N) ***	0.00001 ms	-		> 1.000.000

\* 1s / (Latenz+Seek) (max. random)

\*\* bei 4k Blöcken (max. random)

\*\*\* „-N“: DRAM-based, „-F“ FLASH-based

„-P“ Mixed – Coming with DDR5?



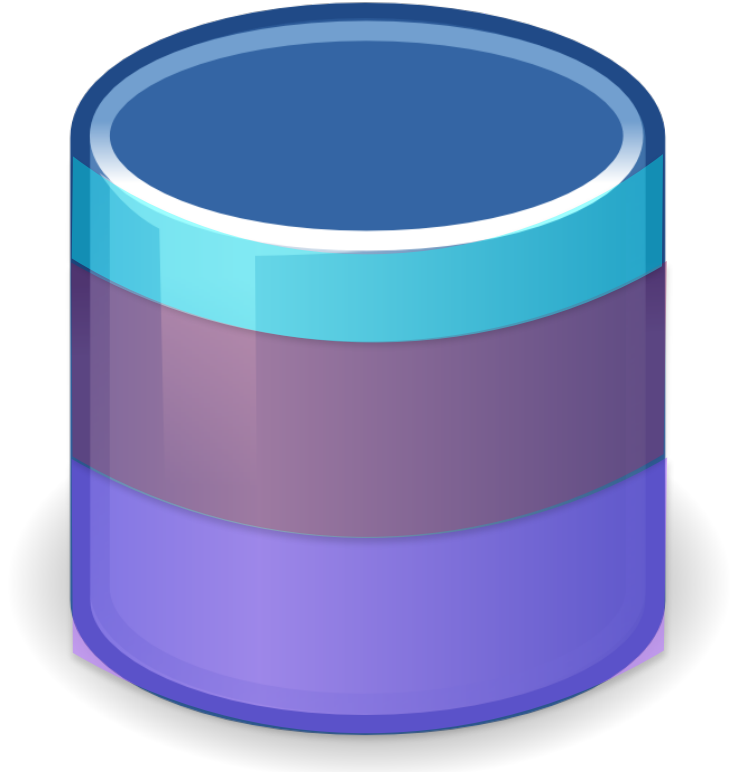


# FESTPLATTEN

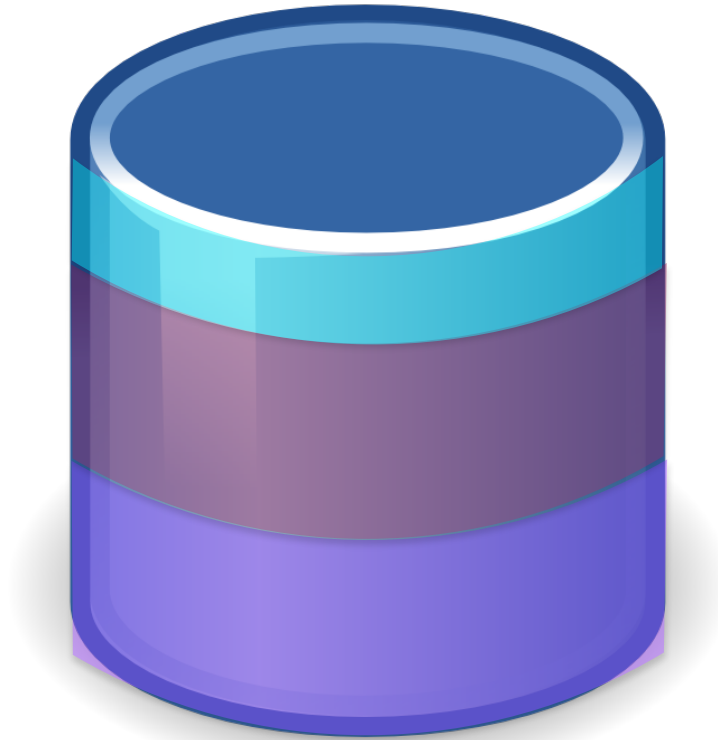


## Partitionierung

# Partitionieren



# Partitionieren - warum?



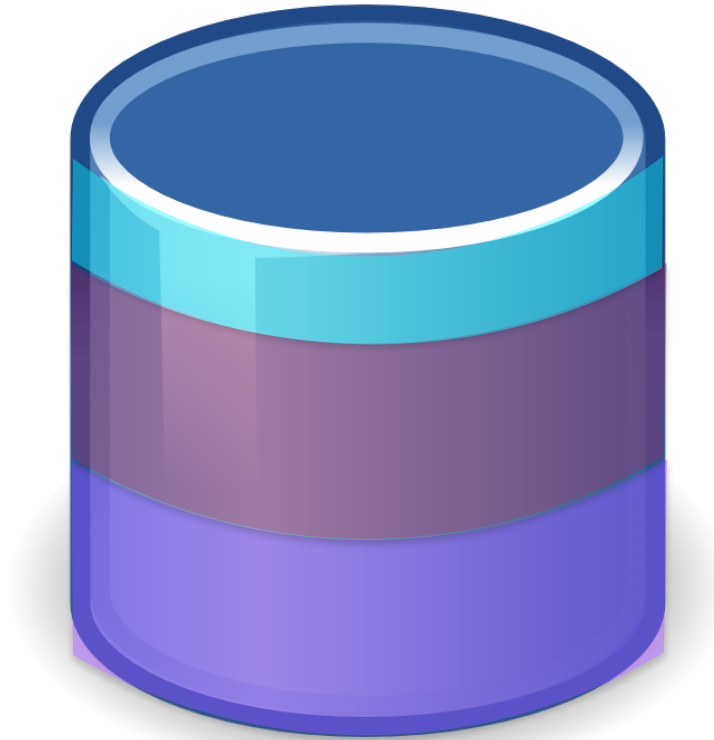
FreeBSD

Linux

Windows

verschiedene Betriebssysteme

# Partitionieren - warum?



Fotos

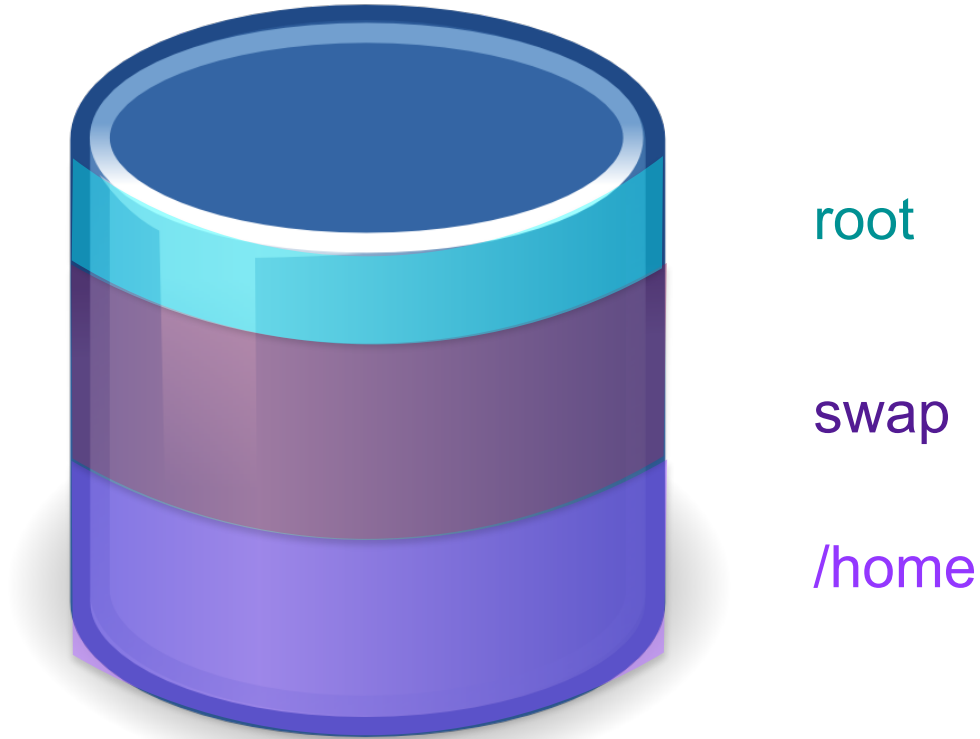
Filme

Windows

Trennung Daten und Betriebssystem

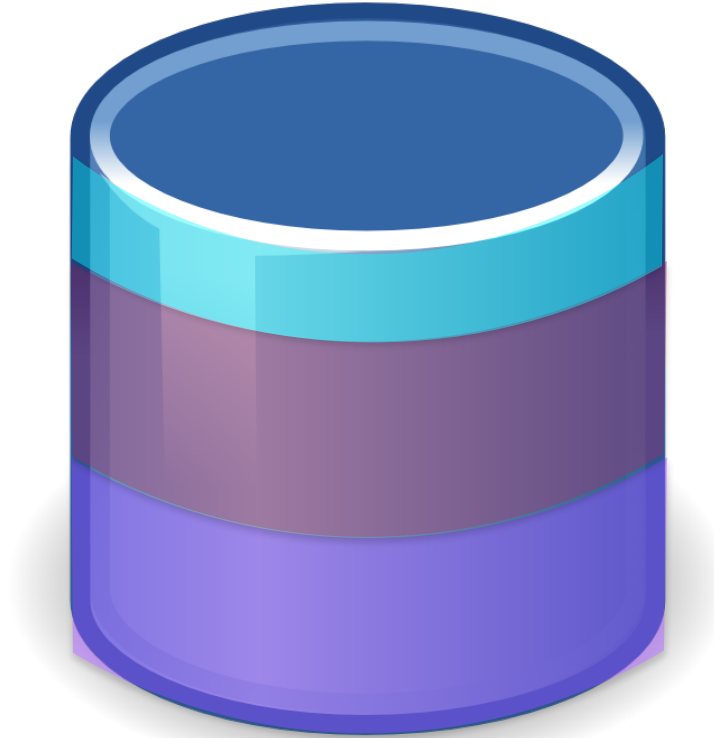


# Partitionieren - warum?



verschiedene Bereiche eines OS

# Partitionieren - warum?



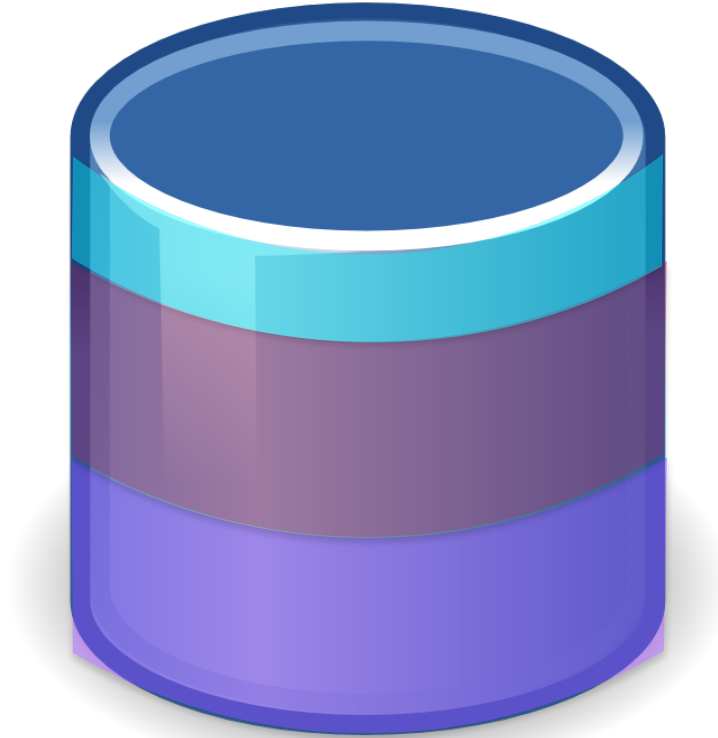
Backup

Windows 10 Devel

Windows 10

Arbeitskopien und Backups

# Partitionieren - warum?



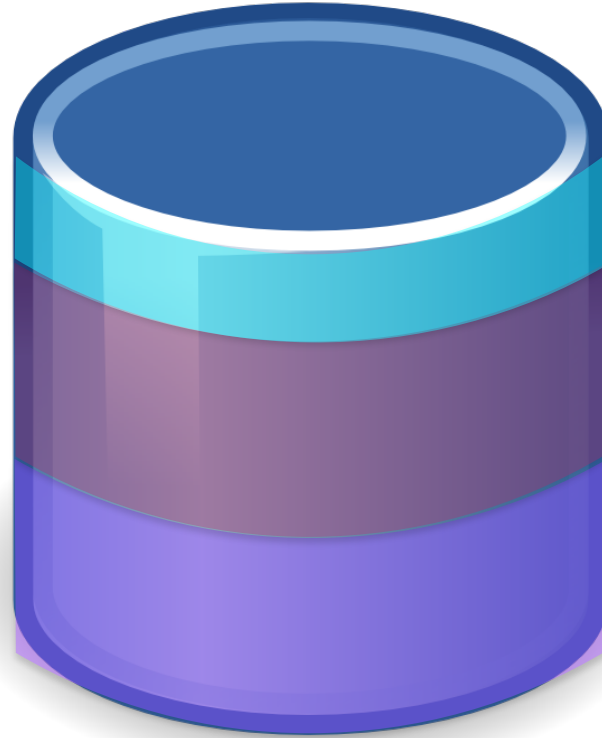
MS-DOS

Win 95a

Windows 8

Verkleinern der Platte

# Partitionieren



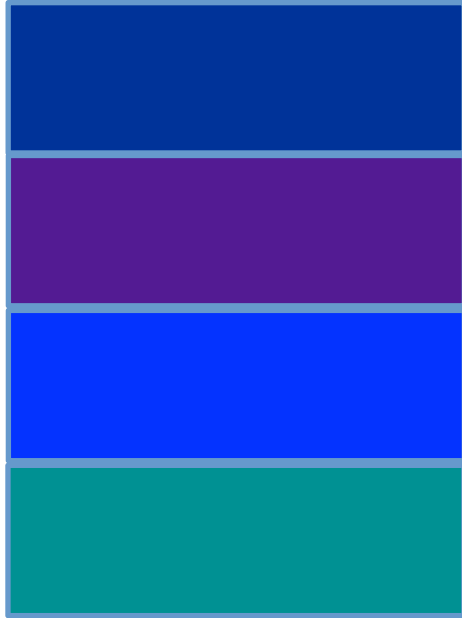
87 - NTFS

bf - Solaris

83 - Linux

System ID

# Partitionen am PC



4 Primärpartitionen

oder



3 Primärpartitionen  
beliebige erweiterte Partitionen

# Klassischer Bootsektor MBR vs. GPT

MBR

GPT

BIOS

EFI

512 Bytes

min. 16 384 Bytes

eine Partitionstabelle

Primäre Partitionstabelle

Backup Partitionstabelle

# Partitionen anderer Systeme (Solaris)

```
label - write partition map and label to the disk
!<cmd> - execute <cmd>, then return
quit
partition> p
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 14086	136.71GB	(14087/0/0) 286698624
1	unassigned	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 14086	136.71GB	(14087/0/0) 286698624
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wu	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wu	0	0	(0/0/0) 0

```
partition> █
```



# PLATTEN ZUSAMMENFASSEN



RAID -  
Redundant Array of Independent Disks

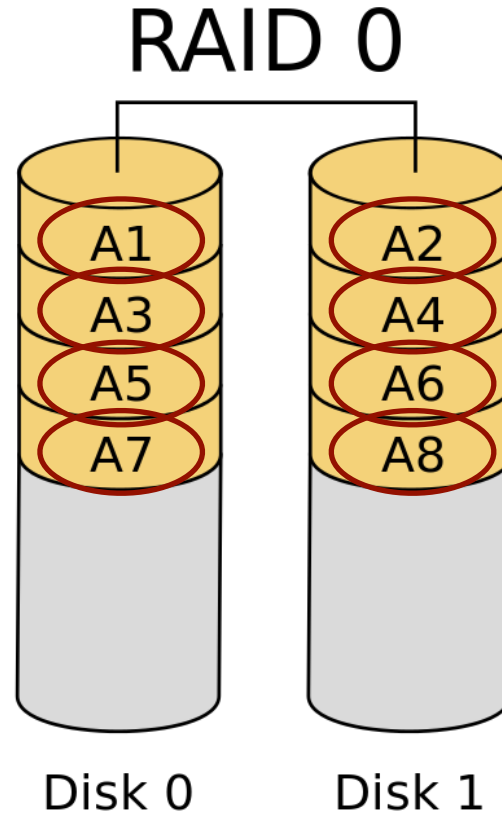


# Warum RAID

mehr Speicherplatz

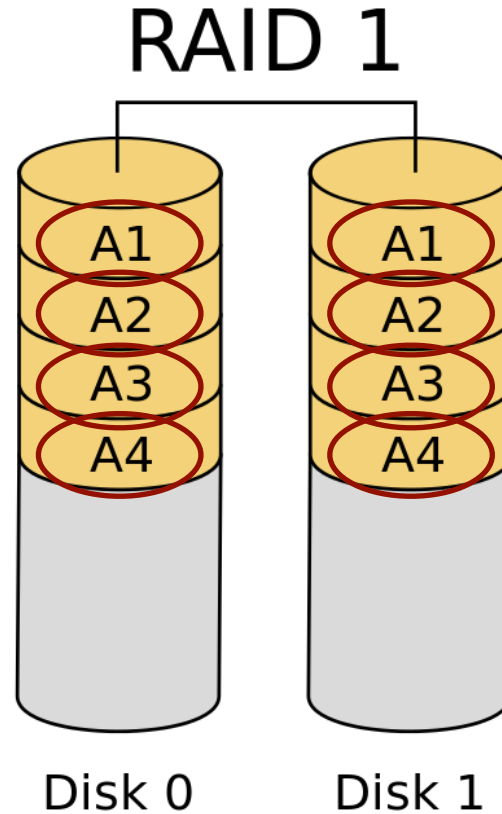
Sicherheit gegen  
Datenverlust\*

# RAID 0



Quelle: Wikimedia

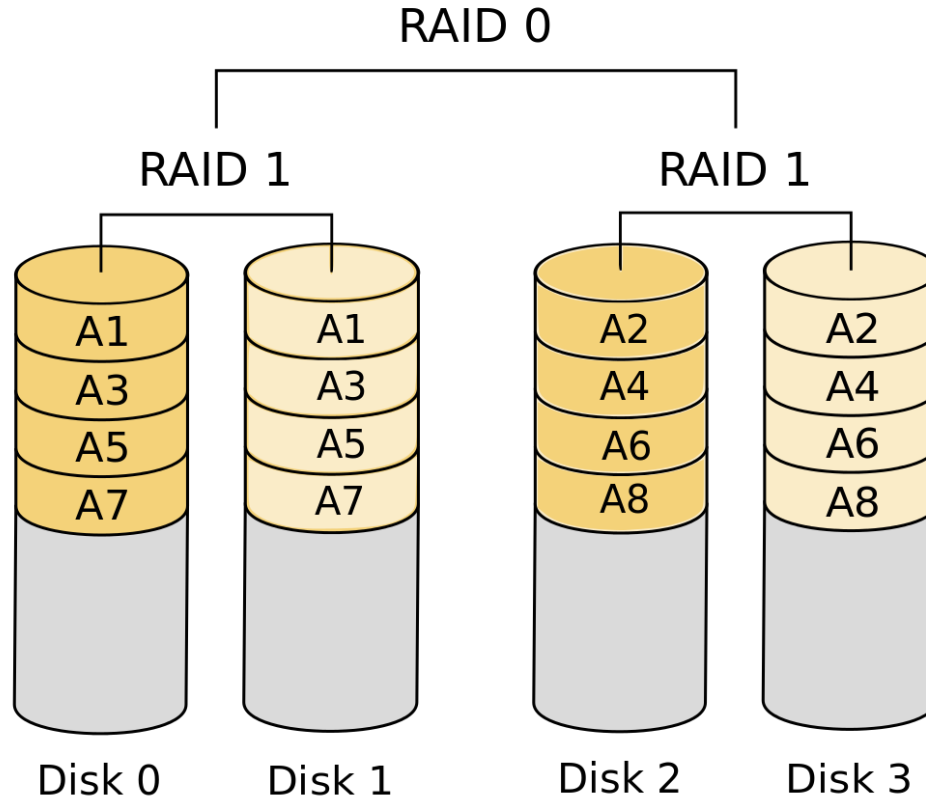
# RAID 1 - Mirror



Quelle: Wikimedia

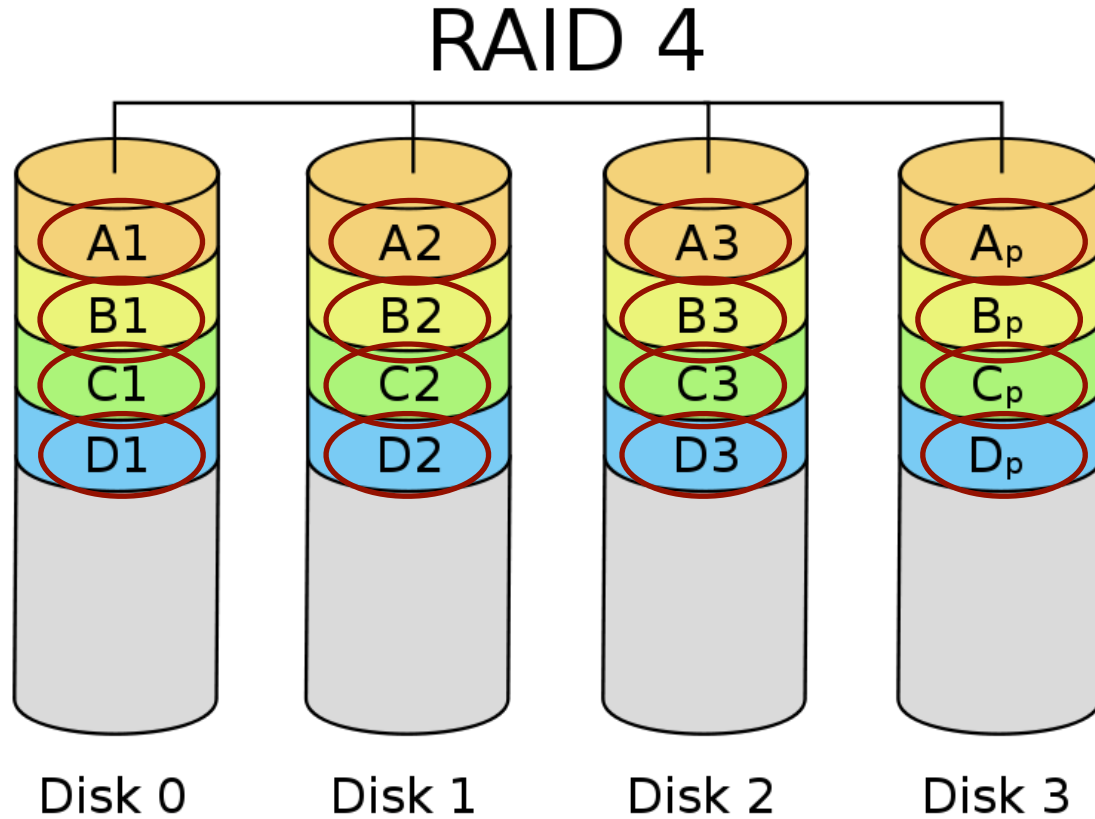
# RAID 10

## RAID 1+0



Quelle: Wikimedia

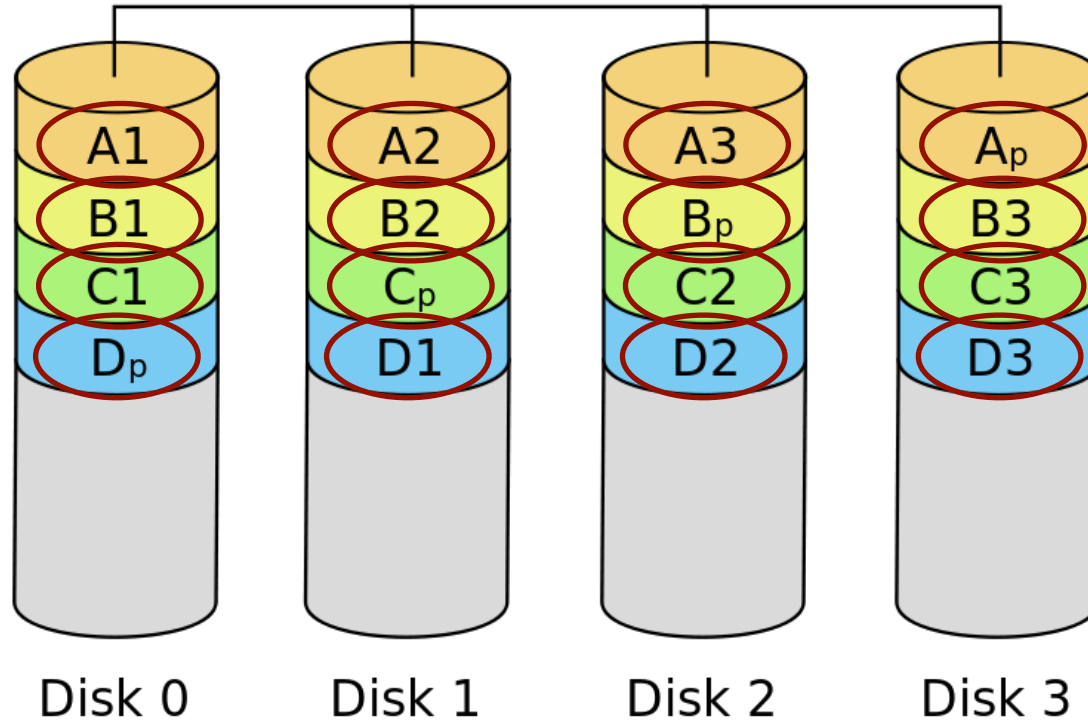
# RAID 4



Quelle: Wikimedia

# RAID 5

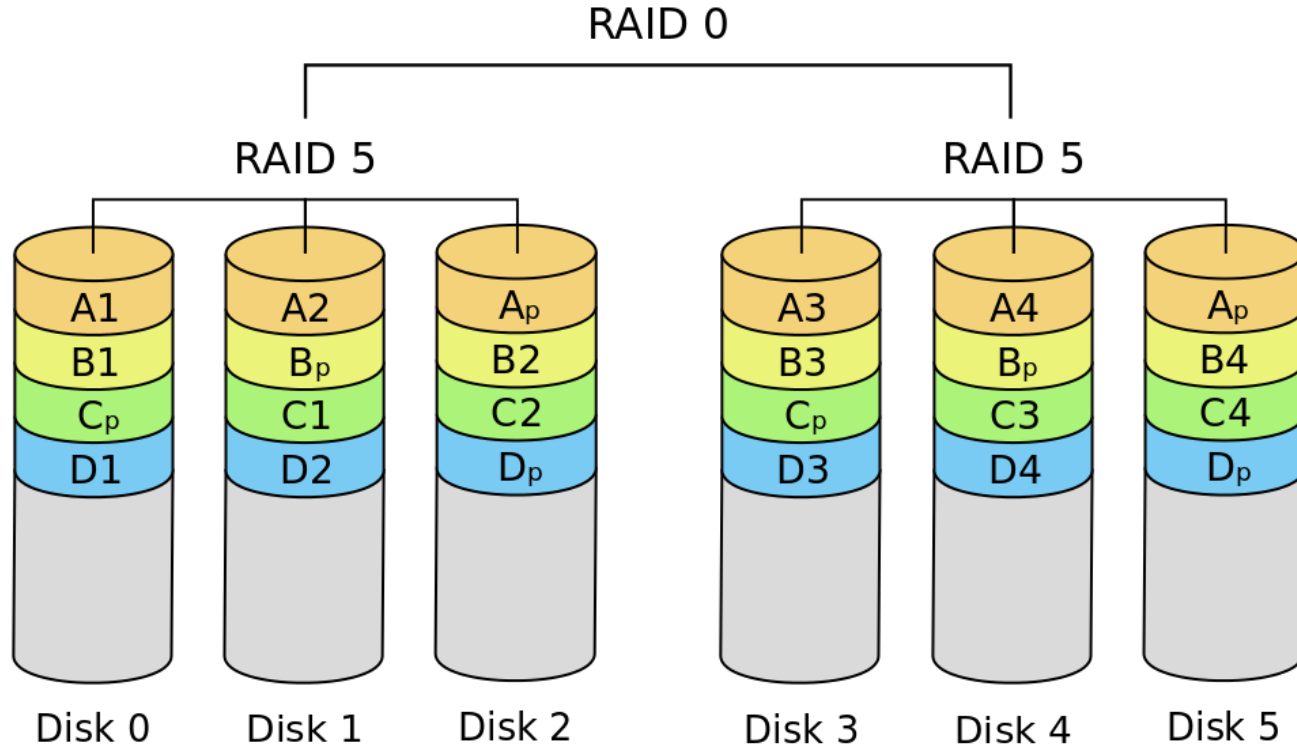
## RAID 5



Quelle: Wikimedia

# RAID 50

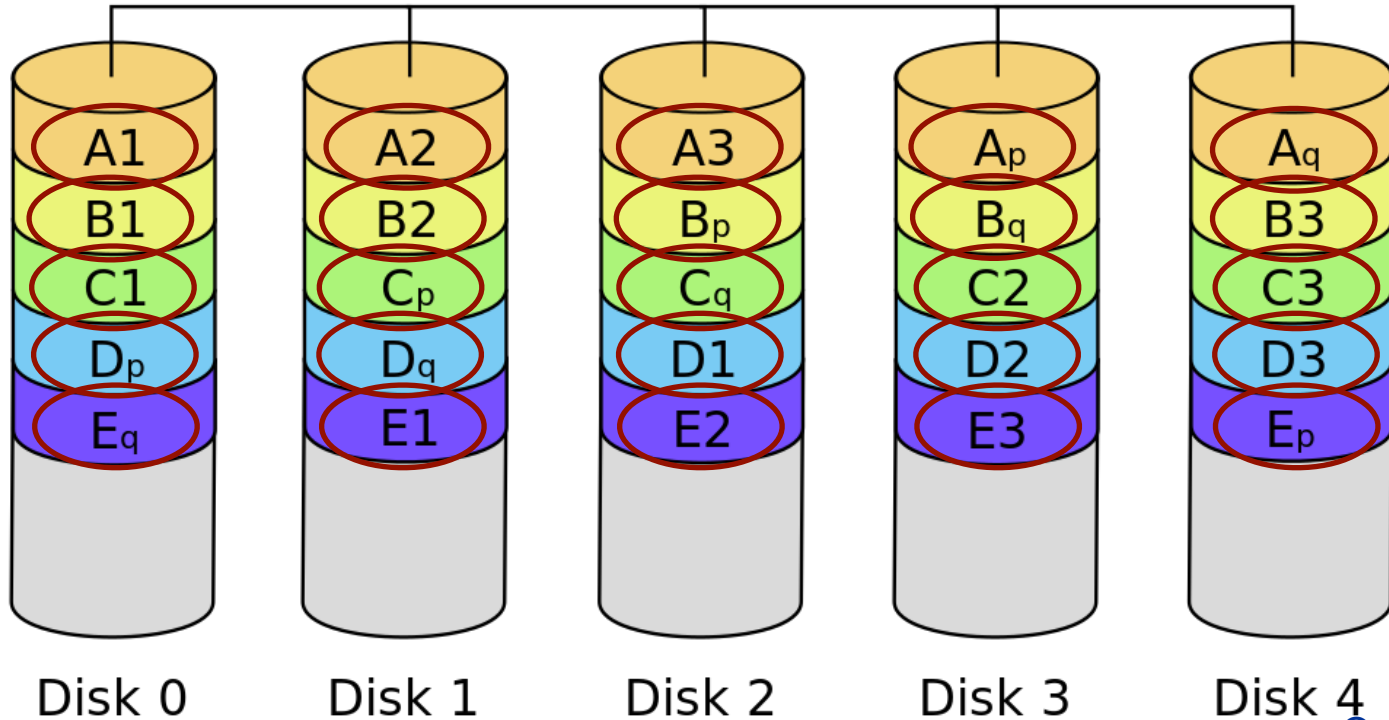
## RAID 5+0



Quelle: Wikimedia

# RAID 6

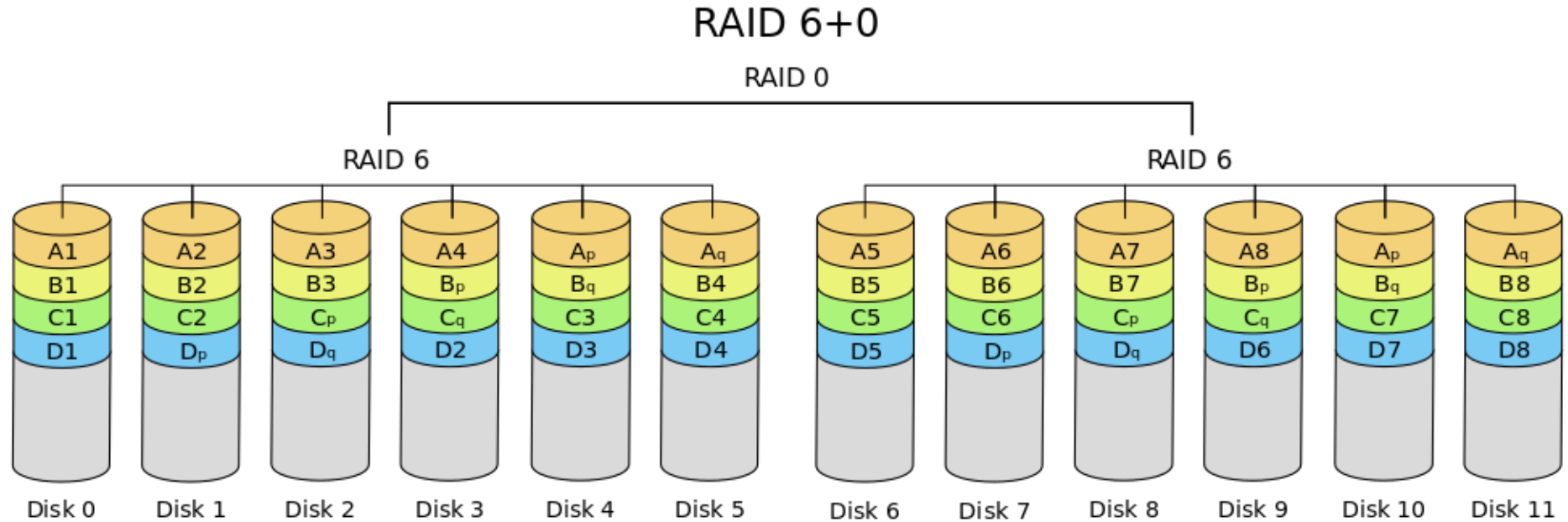
## RAID 6



Quelle: Wikimedia



# RAID 60



Quelle: Wikimedia

# RAID 5 + HotSpare oder RAID 6?

- Verschnitt an Speicherplatz ist gleich
- RAID5: Hotspare wird „geschont“

**aber:**

- Im Fall eines Plattendefekts:
  - RAID5 bietet keine Redundanz mehr (entspricht langsames Raid0)
  - Nach Einspringen der HotSpare werden alle Daten von allen verbliebenen, intakten Platte gelesen um Parity neu zu berechnen
  - Treten Lesefehler auf, ist Rebuild ohne Datenverlust unmöglich
  - Zeitfenster für Rebuild bei großen Festplatten enorm (2 TB bei 100 MB/s = 6 Stunden!)
  - Fehlerwahrscheinlichkeit durch atypisches Lesen aller Disks

# RAID 5 + HotSpare oder RAID 6?

**Fazit:**

**RAID 6 ist RAID 5 + HotSpare vorzuziehen**

# RAID Levels - Übersicht

Raid-Level	Data Disks	Parity Disks	Spare Disks	Disk Errors	Speed W / R	Usable Space	Beispiel: 8 x 1TB Platte
0	N	0	0	0	++ / ++	n	8 TB
1	N	N	0	1	0 / +	n/2	4 TB
4	N	1	0	1	- / 0	n/(n+1)	7 TB
5	N	1	0	1	0 / 0	n/(n+1)	7 TB
5 + Spare	N	1	1	1	0 / 0	n/(n+2)	6 TB
6	N	2	0	2	- / 0	n/(n+2)	6 TB
10	N	N	0	1	++ / ++	n/2	4 TB
50	N	2	0	1	0 / +	n/(n+2)	6 TB
60	N	4	0	1	0 / +	n/(n+4)	4 TB

- LVM (Logical Volume Manager)
  - zusätzliche Abstraktionsschicht zwischen HW und Dateisystem
  - mehrere Speicherentitäten werden zusammengefasst
  - können Platten, Partitionen oder auch RAID-Objekte sein
- Gibt es auch unter Windows:
  - Storage Spaces (Resilient Storage)
    - "Mirror" (→ Raid 1)
    - "Parity" (1/2 → Raid 5/6)
    - "Simple" (→ Raid 0)

# LVM (Logical Volume Manager) - Linux LVM2

Blockdevice  
(Festplatte)

sda

sdb

sdc

Physical  
Volume (PV)

PV

sda

PV

sdb

PV

sdc

Volume  
Group (VG)

vg\_disk\_abc

Logical  
Volume (LV)

vg\_disk\_abc/lv\_part1

.../lv\_part2

Filesystem /  
Blockdevice

/data (ext4)

vg\_disk\_abc/lv\_part1

.../lv\_part2



# DATEISYSTEME



Speicherung von Daten

# Probleme beim Speichern von Daten

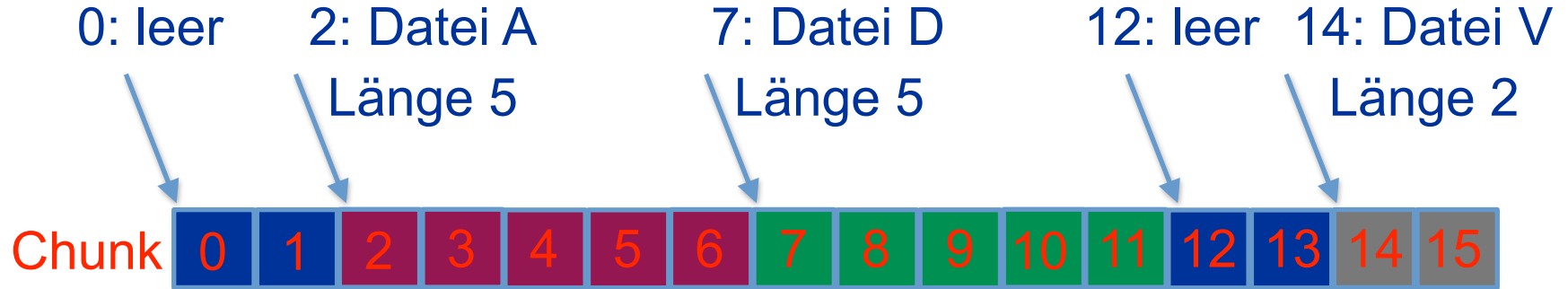
Dateisysteme verwenden Cluster

Dateien sind oft größer (oder kleiner) als ein Cluster

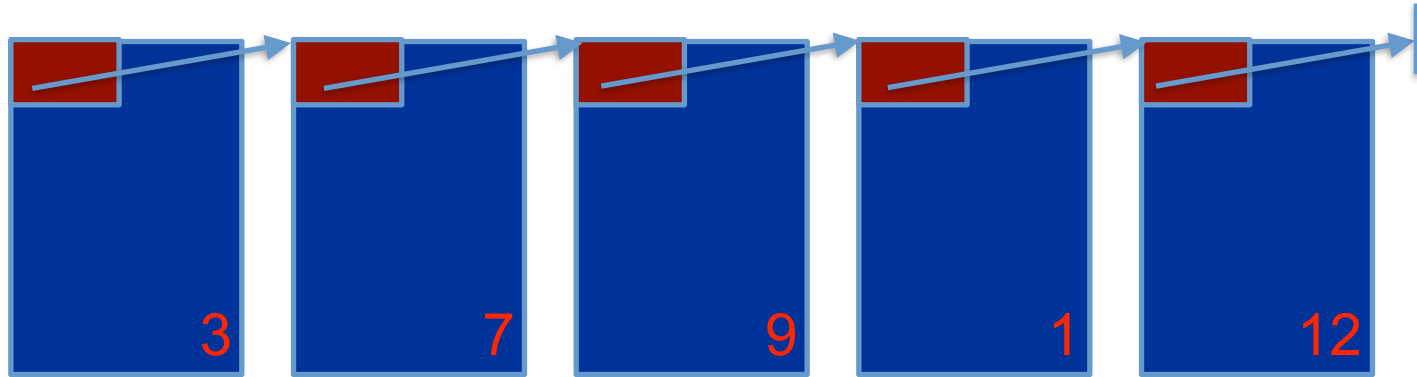
Wie kann man nun gespeicherte Daten adressieren?



# Kontinuierliche Speicherung



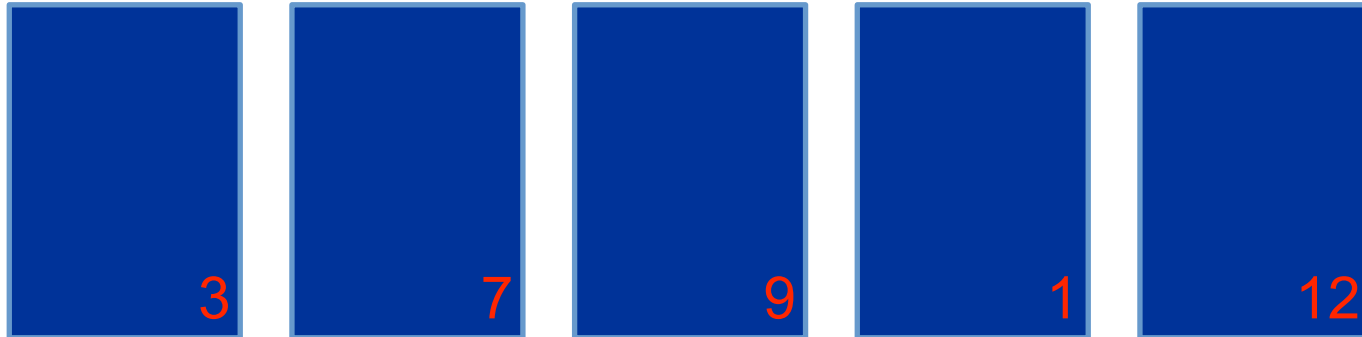
# Verkettete Speicherung



# Indizierte Speicherung



Index-Cluster

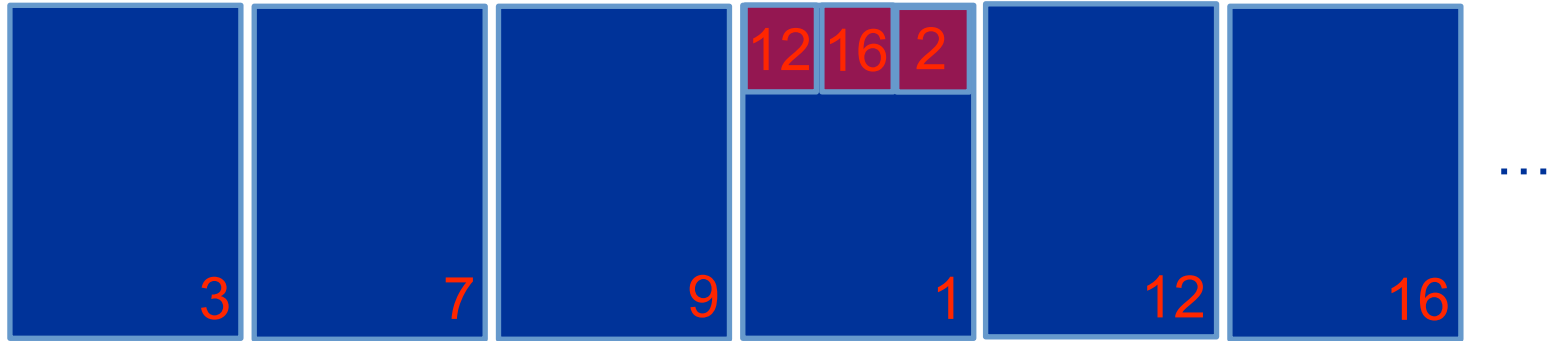


Daten-Cluster der Datei

# Indizierte Speicherung, mehrstufige Indizierung



Index-Cluster



Daten-Cluster mit einem zusätzlichen Index Cluster



# DATEISYSTEME



Beispiele anhand gängiger Dateisysteme

# FAT



# FAT

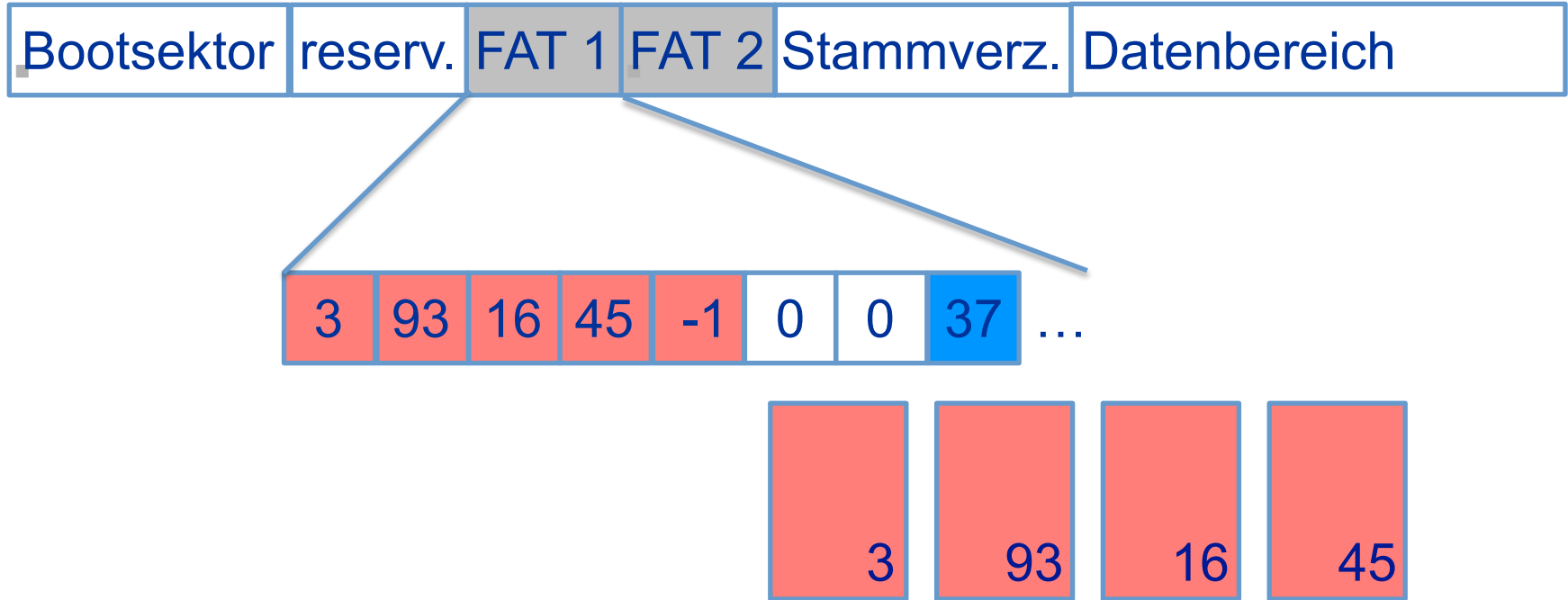


# FAT





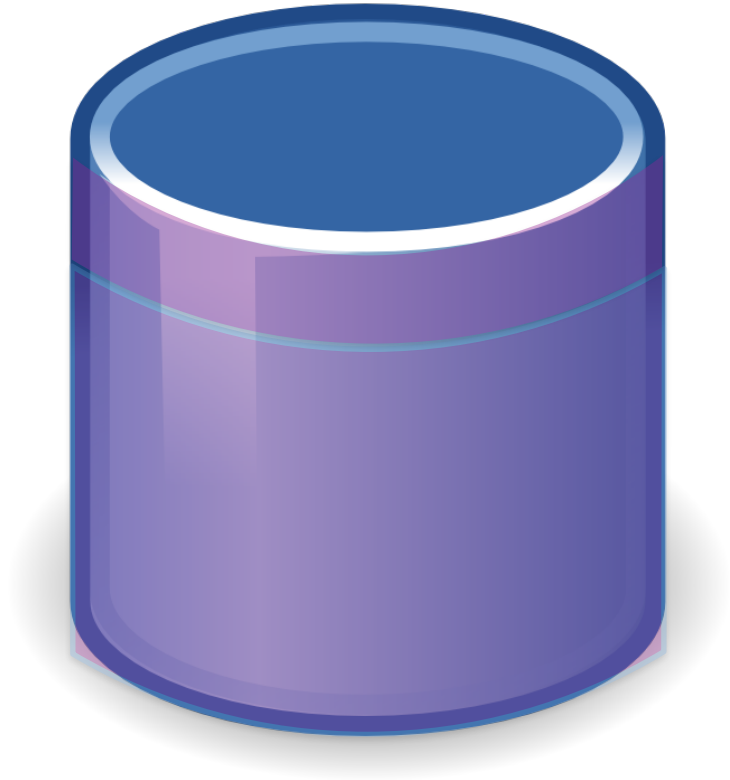
# FAT



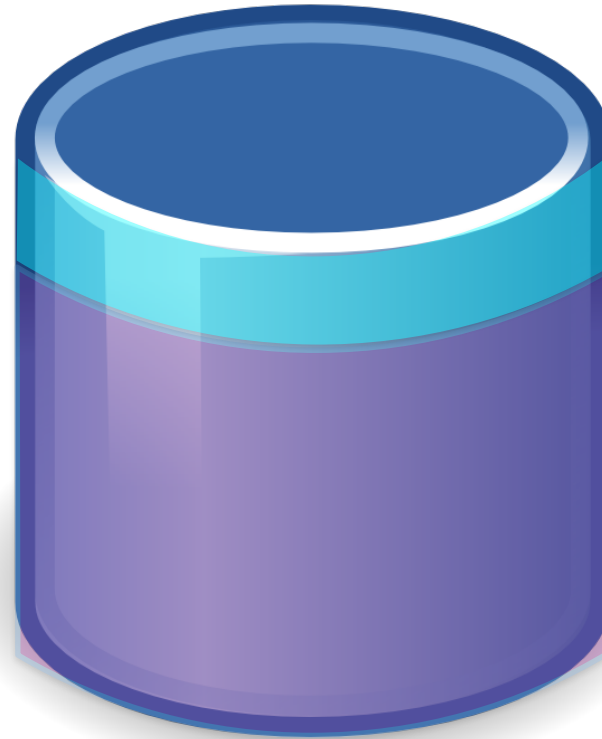
# FAT



# NTFS - Next Technology File System



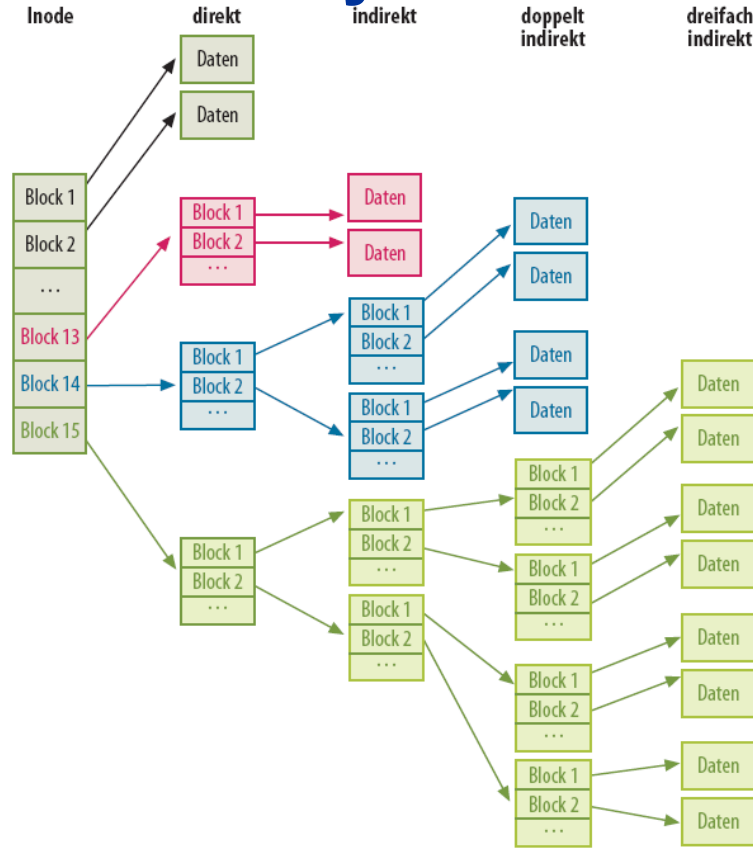
# NTFS - Next Technology File System



Master File Table (12,5%)

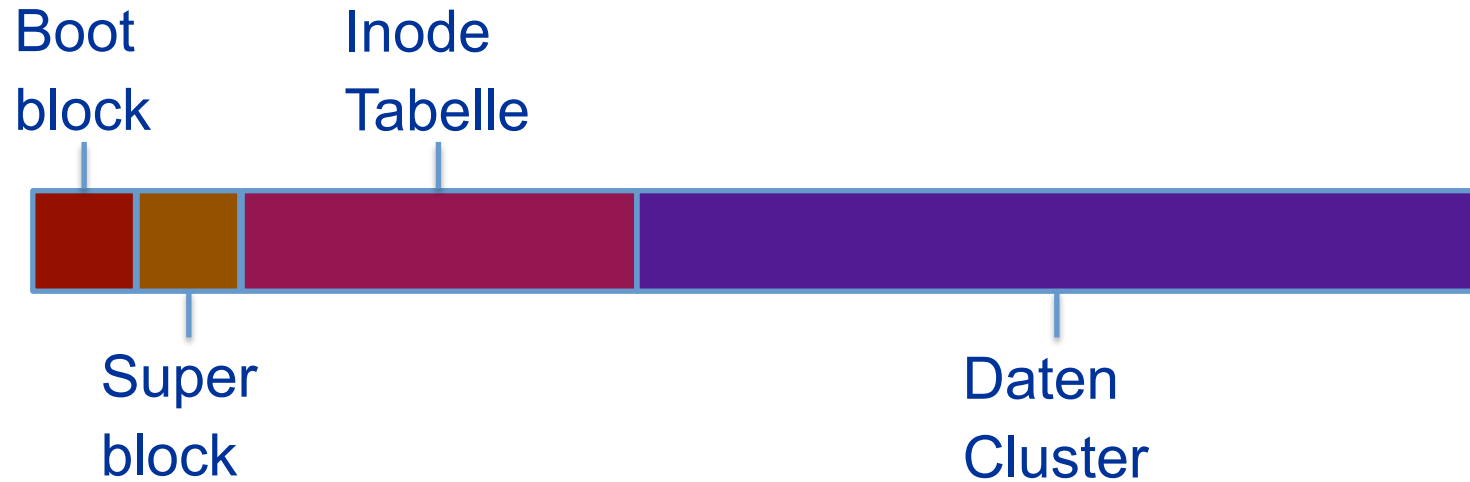
Datenbereich

# Klassische Unix Dateisysteme



Quelle: [heise.de](http://heise.de)

# System V File System





# DATEISYSTEME



Konzepte um Datenintegrität zu garantieren

# Journaling





# Metadaten - Journaling



Metadaten



Daten



# Vollständiges Journaling



Metadaten



Daten



# Ordered - Journaling



Metadaten



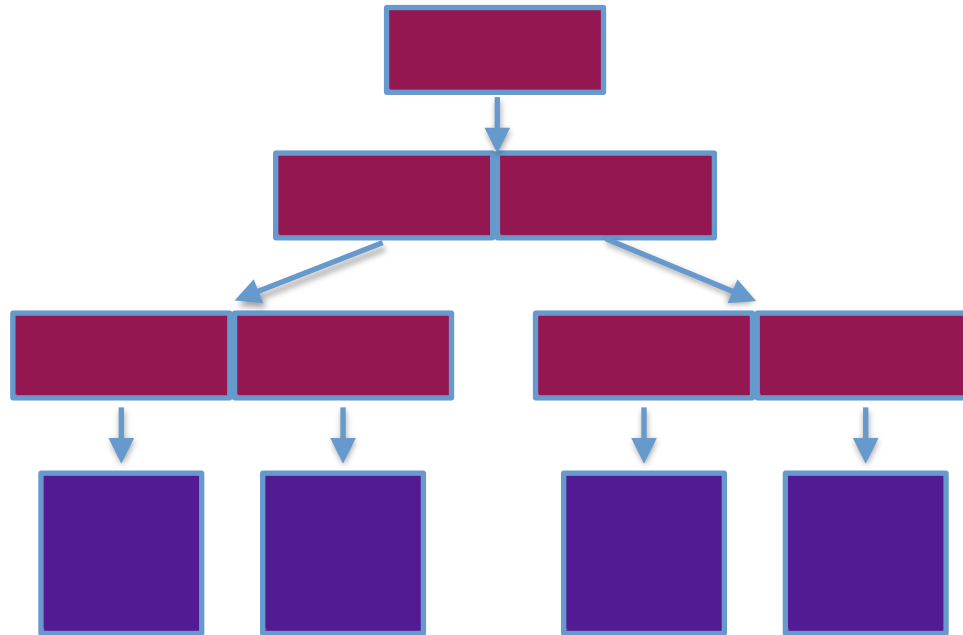
Daten



# copy on write

Daten und Metadaten werden immer in freie Blöcke geschrieben:  
es werden keine Daten überschrieben

# ZFS - Beispiel für copy on write



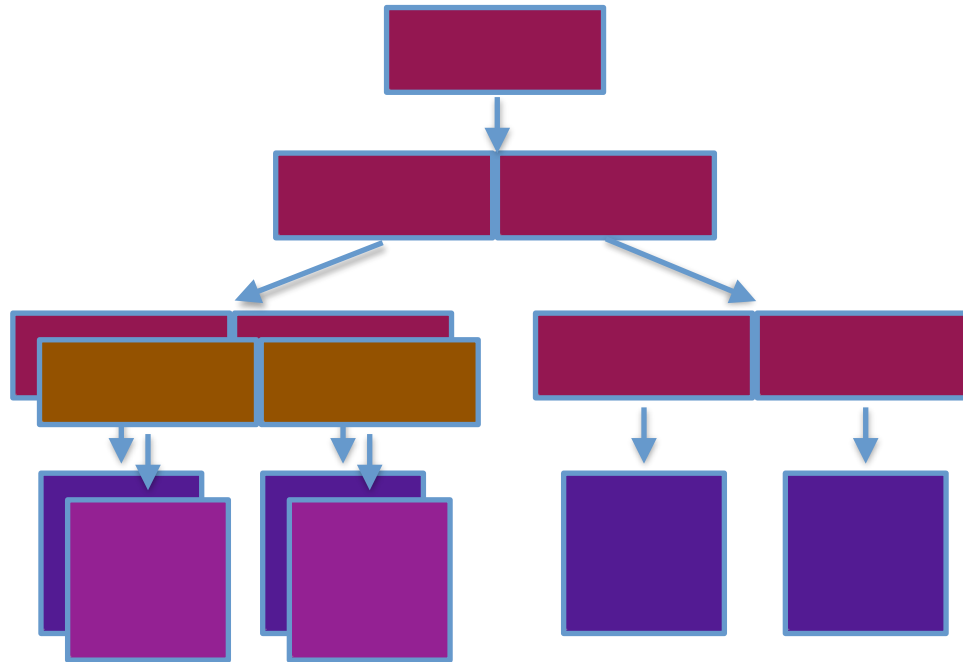
Überblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

# ZFS - Beispiel für copy on write



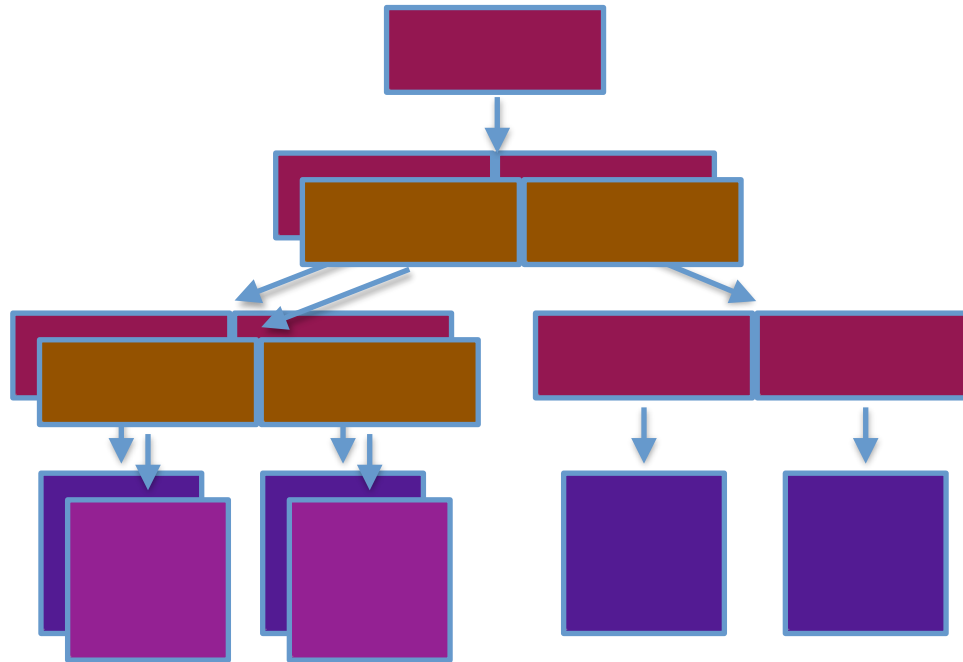
Überblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

# ZFS - Beispiel für copy on write



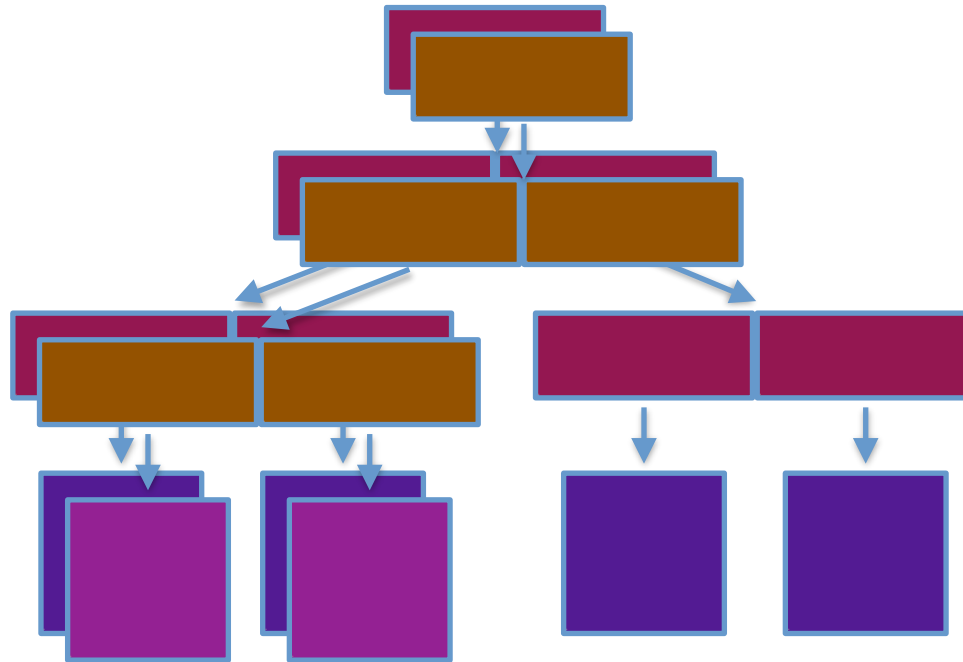
Überblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

# ZFS - Beispiel für copy on write



Überblock

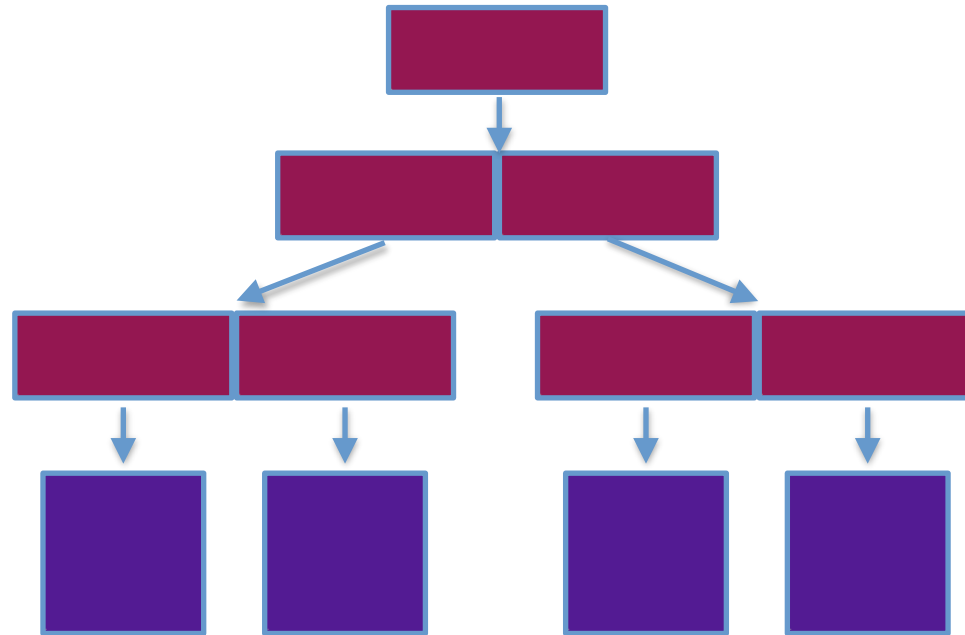
Zeiger/Metadaten

Zeiger/Metadaten

Daten



# ZFS - Beispiel für copy on write



Überblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

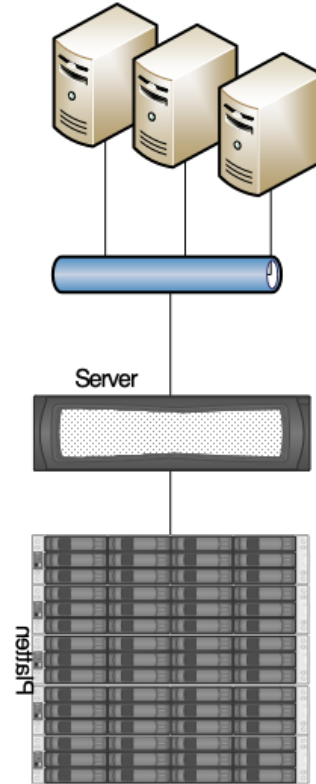


# HARDWARE

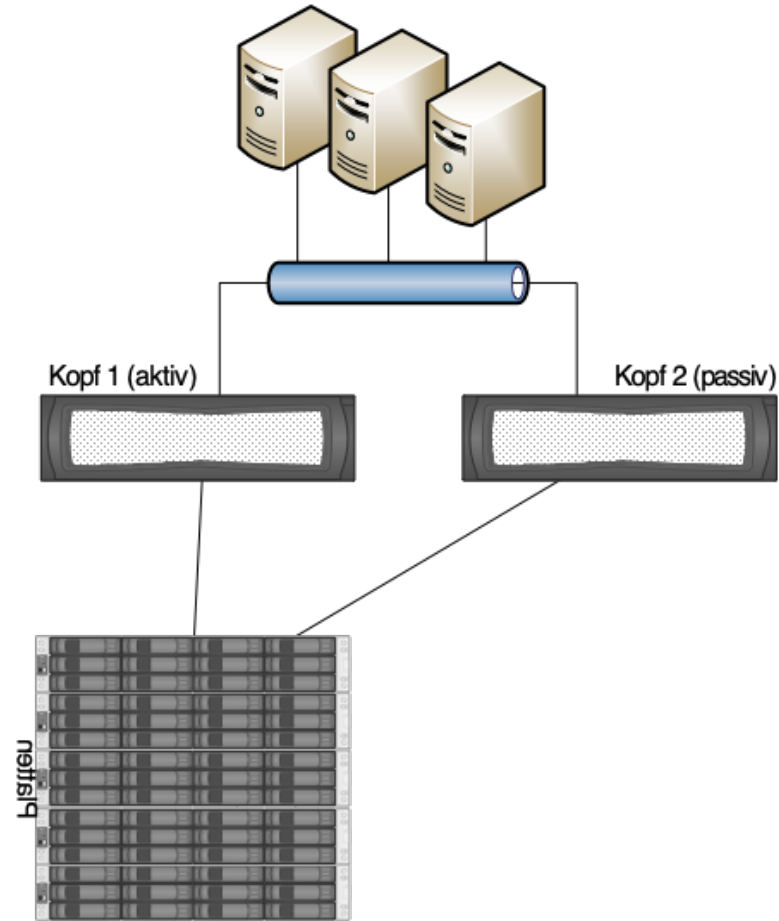


## Typischer Aufbau / Storage-Appliances

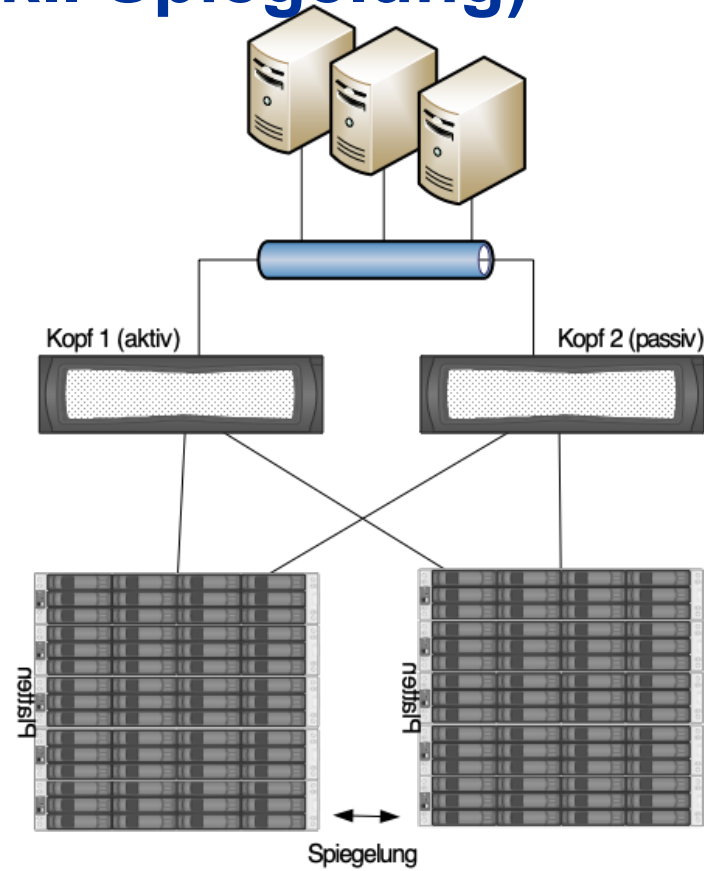
# DAS – Direct Attached Storage



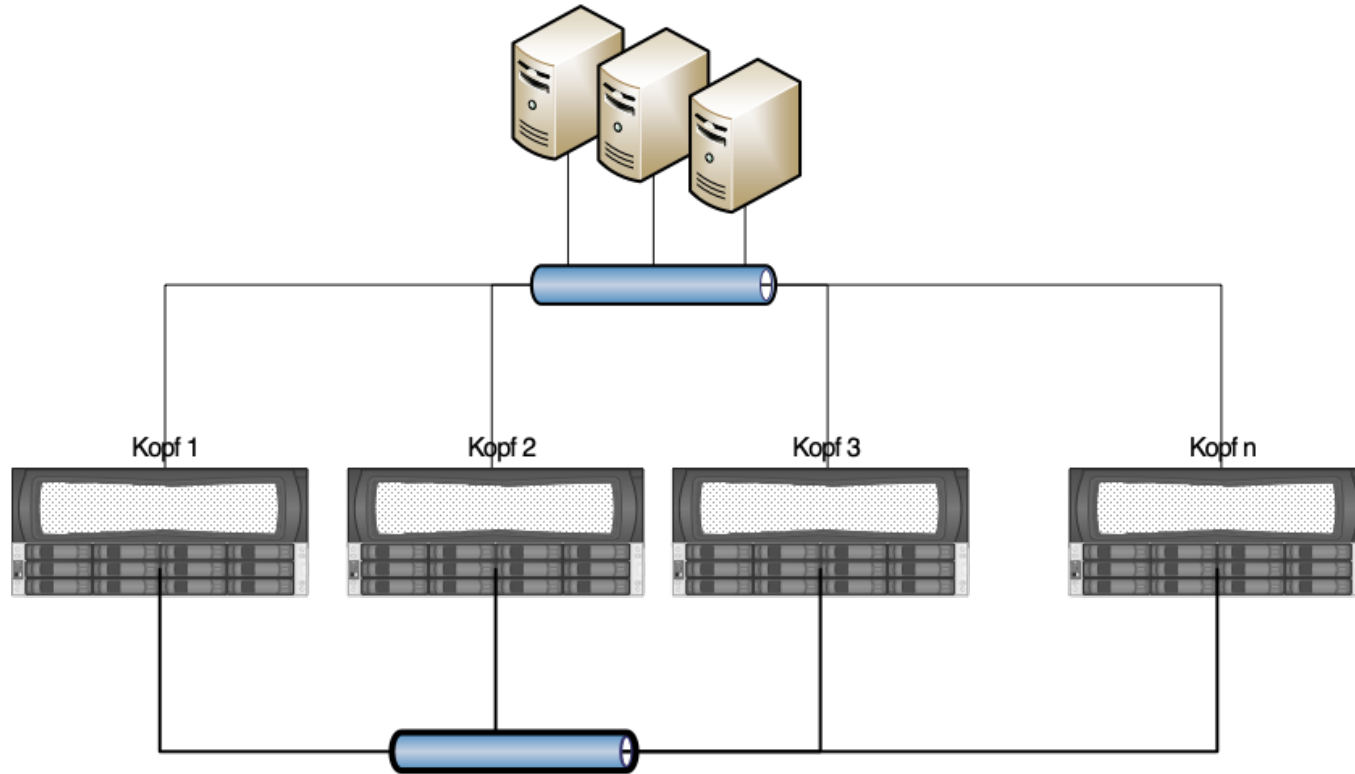
# Klassisch



# Klassisch (inkl. Spiegelung)



# Scale-Out



# Unterscheidung: SAN ↔ NAS

## SAN (Storage Area Network)

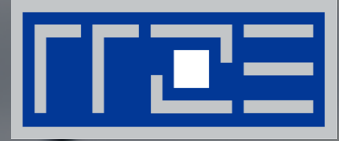
- Block-Level Zugriff
- Protokolle:
  - › Eigene Verkabelung: Fibre Channel (FC), SAS
- Basierend auf klassischem TCP/IP Netzwerk:
  - › iSCSI, FCoE, AoE

## NAS (Network Attached Storage)

- File-Level Zugriff
- Mehr dazu später ...



# NETWORK ATTACHED STORAGE



## Network Attached Storage (NAS)

- Zugriffsprotokolle



# Netzwerk-Filesystem-/NAS-Protokolle

2 „Klassiker:

## Windows-Welt: **CIFS/SMB**



- Common Internet Filesystem / System Message Block
- Ursprung: IBM / Microsoft

## Unix-Welt: **NFS**

- Network Filesystem
- Ursprung: Sun Microsystems



# Netzwerk-Filesystem-Protokolle – CIFS/SMB

## **SMB** (Server Message Block)

- Version 1.0

## **CIFS** (Common Internet FileSystem)

- **Version 2.0** (2006) ( $\geq$  Windows Vista / Server 2008)
  - › Vereinfachung (Subcommands:  $> 100 \Rightarrow 19$ )
  - › Neu: Symbolische Links, Größere Blockgröße, Unicode
- **Version 2.1** ( $\geq$  Windows 7 / Server 2008 R2)
  - › Performance



# Netzwerk-Filesystem-Protokolle – CIFS/SMB



- **Version 3.0** (ehemals 2.2, >= Windows 8 / Server 2012)
  - › SMB Direct (SMB over RDMA)
  - › SMB Multichannel
  - › End-to-End encryption
- **Version 3.0.2** (auch 3.02, >= Windows 8.1 / Server 2012R2)
  - › SMB1 abschaltbar (Sicherheit!)
- **Version 3.1.1** (>= Windows 10 / Server 2016)
  - › Secure negotiation Pflicht für SMB >= 2.x

# Netzwerk-Filesystem-Protokolle - NFS

## Version 2 (RFC 1094, 03/1989)

- Basierend auf RPC (Remote Procedure Call)
- Portmapper (Port 111):
  - › Vermittelt Dienste auf dynamischen Ports (Firewall!), UDP (später: TCP)
- 32 bit (max. 2 GB Filegröße)

## Version 3 (RFC 1813, 06/1995)

- UDP + TCP
- 64 bit Support
- Asynchrones Schreiben



# Netzwerk-Filesystem-Protokolle - NFS

## Version 4 (RFC 3010, 12/2000, Rev.: RFC 3530/7530)

- Single Standard Port 2049 (kein Portmapper!)
- NFSv4 ACLS (ähnlich Windows/CIFS ACLs)
- RPCSEC\_GSS (Kerberos)

## Version 4.1 (RFC 5661, 01/2010)

- pNFS

## Version 4.2 (RFC 7862, 11/2016)

- Sparse File Support
- Server Side Copy
- Space Reservation



# Netzwerk-Filesystem-Protokolle – Warum NFS 4.x nutzen?

Bis inkl. Version 3

- Beschränkung Host-basiert (AUTH\_SYS / AUTH\_UNIX)
- ro / rw, (no\_)root\_squash, (in)secure (NAT VMs!)
- Client-Server Mapping uid/gid-basiert (Sicherheit!)
- Posix ACLs (nur RFC, kein Standard!)

Ab Version 4.0:

- Client-Server Mapping „String“-basiert (idmap!)
- Starke Verschlüsselung / Authentifizierung
  - › krb5 (Authentication Only), krb5i (Integrity), krb5p (Privacy)

# Weitere Vorträge zur „Systemausbildung“

12.05.2021 – Backup / Archiv

19.05.2021 – Einblicke in das Management von Apple-Geräten

09.06.2021 – Storage & Filesysteme

16.06.2021 – IT-Sicherheit

23.06.2021 – Windows-Dienste

30.06.2021 – Virtualisierung

07.07.2021 – Benutzerverwaltung: MS Active Directory

14.07.2021 – Kerberos

Immer mittwochs  
(ab 14:15 Uhr)  
- online -

**Details:** [www.rrze.fau.de/veranstaltungen/veranstaltungskalender/](http://www.rrze.fau.de/veranstaltungen/veranstaltungskalender/)

# Andere Vortragsreihen des RRZE

## Campustreffen „IT-Dienste des RRZE und der FAU“

- immer donnerstags ab 15 Uhr c.t.
- vermittelt Informationen zu den Dienstleistungen des RRZE
- befasst sich mit neuer Hard- & Software, Update-Verfahren sowie Lizenzfragen
- ermöglicht den Erfahrungsaustausch mit Spezialisten

## Netzwerkausbildung „Praxis der Datenkommunikation“

- immer mittwochs in den Wintersemestern, ab 14 Uhr c.t.
- Vorlesungsreihe, die in die Grundlagen der Netztechnik einführt
- stellt die zahlreichen aktuellen Entwicklungen auf dem Gebiet der (universitären) Kommunikationssysteme dar



# RRZE-Veranstaltungskalender und Mailinglisten

- Anleitung Kalender abonnieren oder bookmarken
  - [www.rrze.fau.de/veranstaltungen/veranstaltungskalender/](http://www.rrze.fau.de/veranstaltungen/veranstaltungskalender/)
- Mailingliste abonnieren
  - Wöchentliche Terminhinweise werden zusätzlich an die Mailingliste [RRZE-Aktuelles](http://www.rrze.fau.de/rrze-aktuelles) gesendet.
  - Auch diese Liste kann man abonnieren:  
<https://lists.fau.de/mailman/listinfo/rrze-aktuelles>

# Themenvorschläge und Anregungen

Themenvorschläge und Anregungen nehmen wir gerne entgegen!

Bitte schreiben Sie uns einfach eine E-Mail an:  
[rrze-zentrale@fau.de](mailto:rrze-zentrale@fau.de) (Betreff: Systemausbildung)

# REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



## **Vielen Dank für Ihre Aufmerksamkeit!**

Regionales Rechenzentrum Erlangen [RRZE]

Martensstraße 1, 91058 Erlangen

[www.rrze.fau.de](http://www.rrze.fau.de)