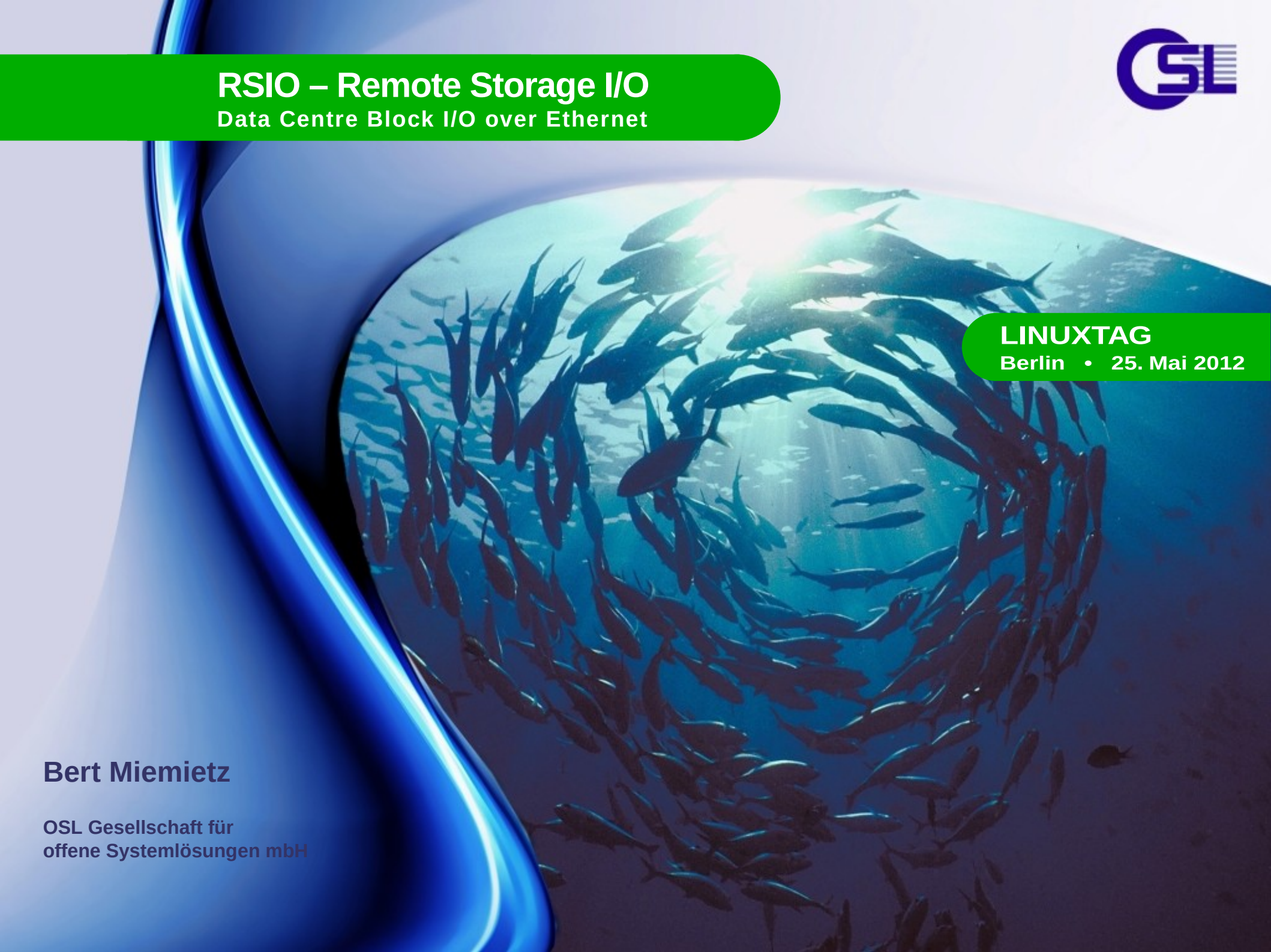
RSIO – Remote Storage I/O

Data Centre Block I/O over Ethernet



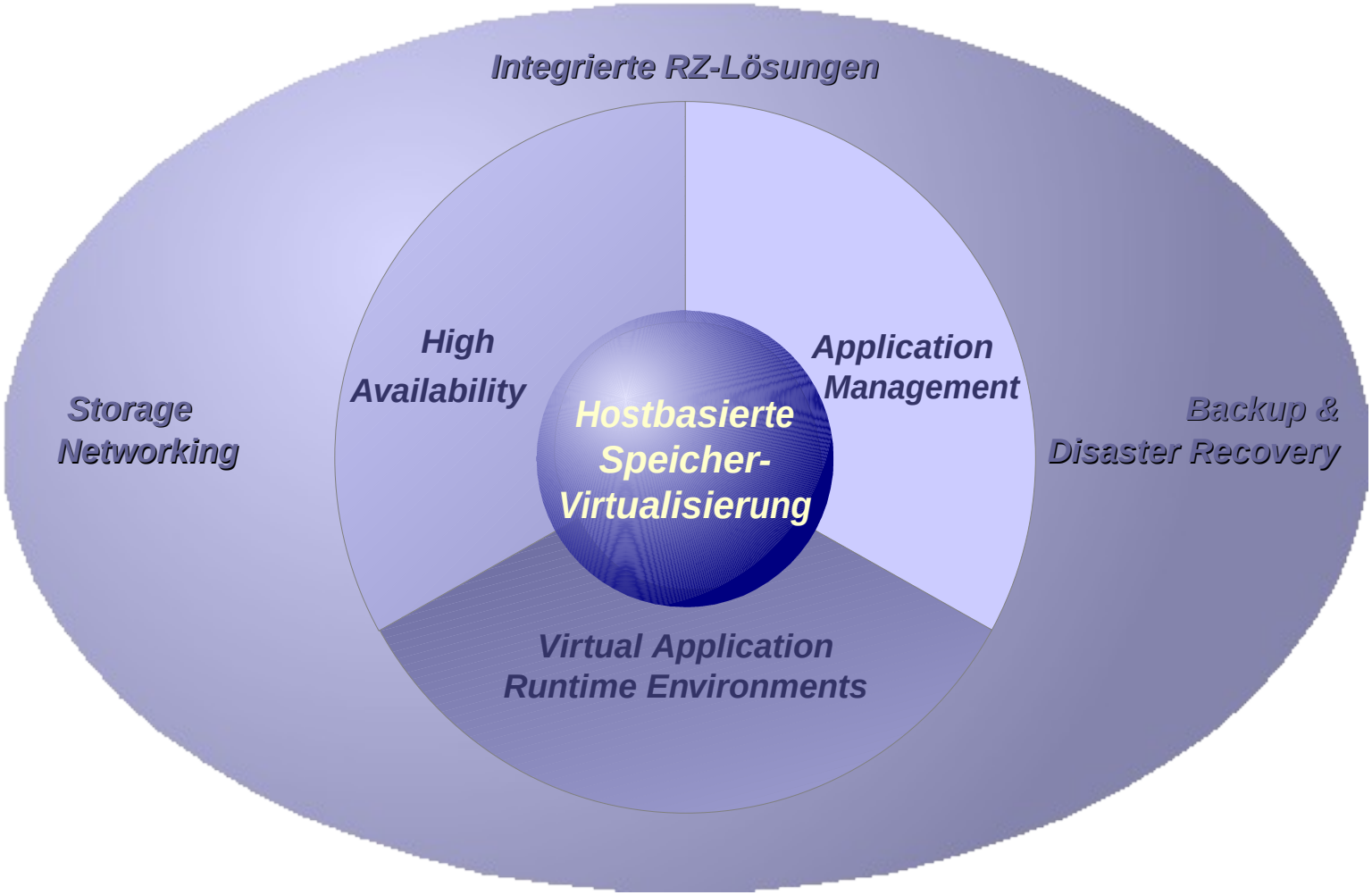
LINUXTAG
Berlin • 25. Mai 2012

Bert Miemietz

OSL Gesellschaft für
offene Systemlösungen mbH

OSL entwickelt Infrastruktur-Software

Storage Networking & Virtualization • Volume Management
Clustering • High Availability • Disaster Protection • Consulting & Services



Denken Sie selbst!

Sonst tun es andere für Sie.

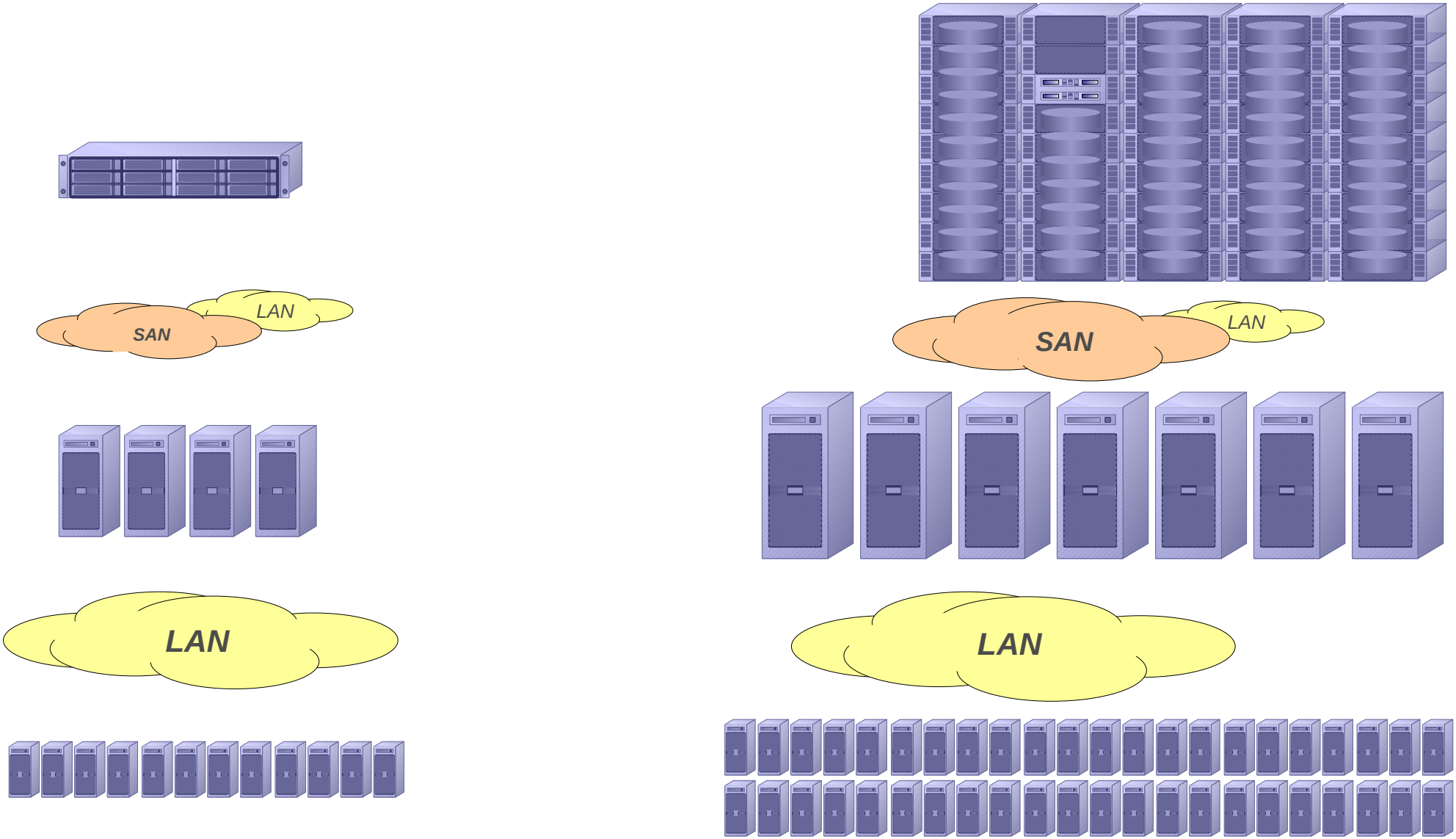
(Vince Ebert)

RSIO - Remote Storage I/O

Die Motivation

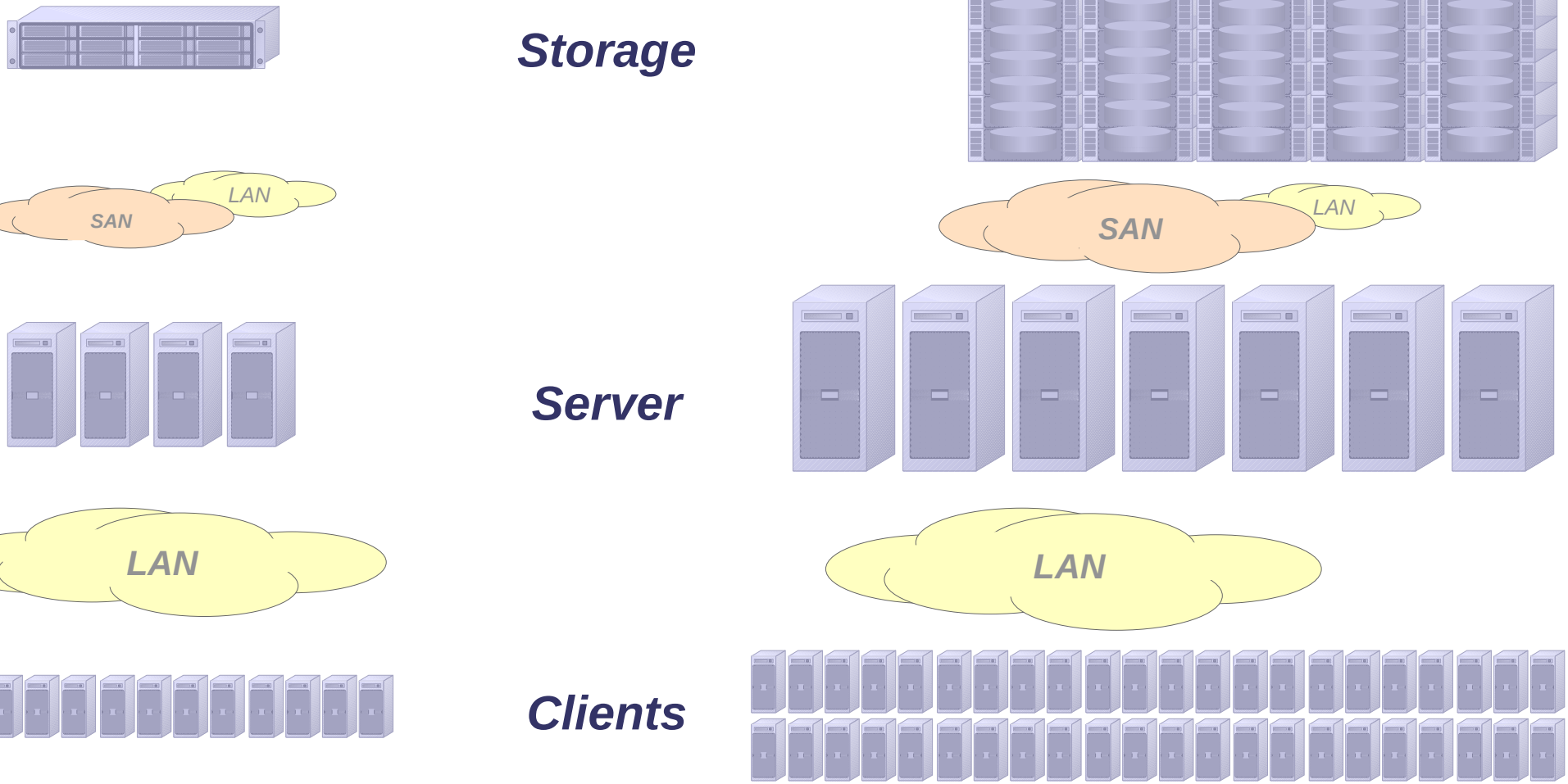
RZ-Achitektur heute

Ein immer gleiches Prinzip: Trennung von Servern und zentralisiertem Storage



RZ-Achitektur heute

Ein immer gleiches Prinzip: Trennung von Servern und zentralisiertem Storage



RZ-Achitektur heute

In den letzten 10 Jahren haben sich gewisse Trends verstärkt

- *Zahl der Serversysteme hat sich weiter vergrößert*
- *Massenspeicher heute extrem zentralisiert*
- *Vermittlung über oft komplexe Speichernetzwerke, die Anwender der Applikationen eigentlich nicht interessieren*

RZ-Achitektur heute

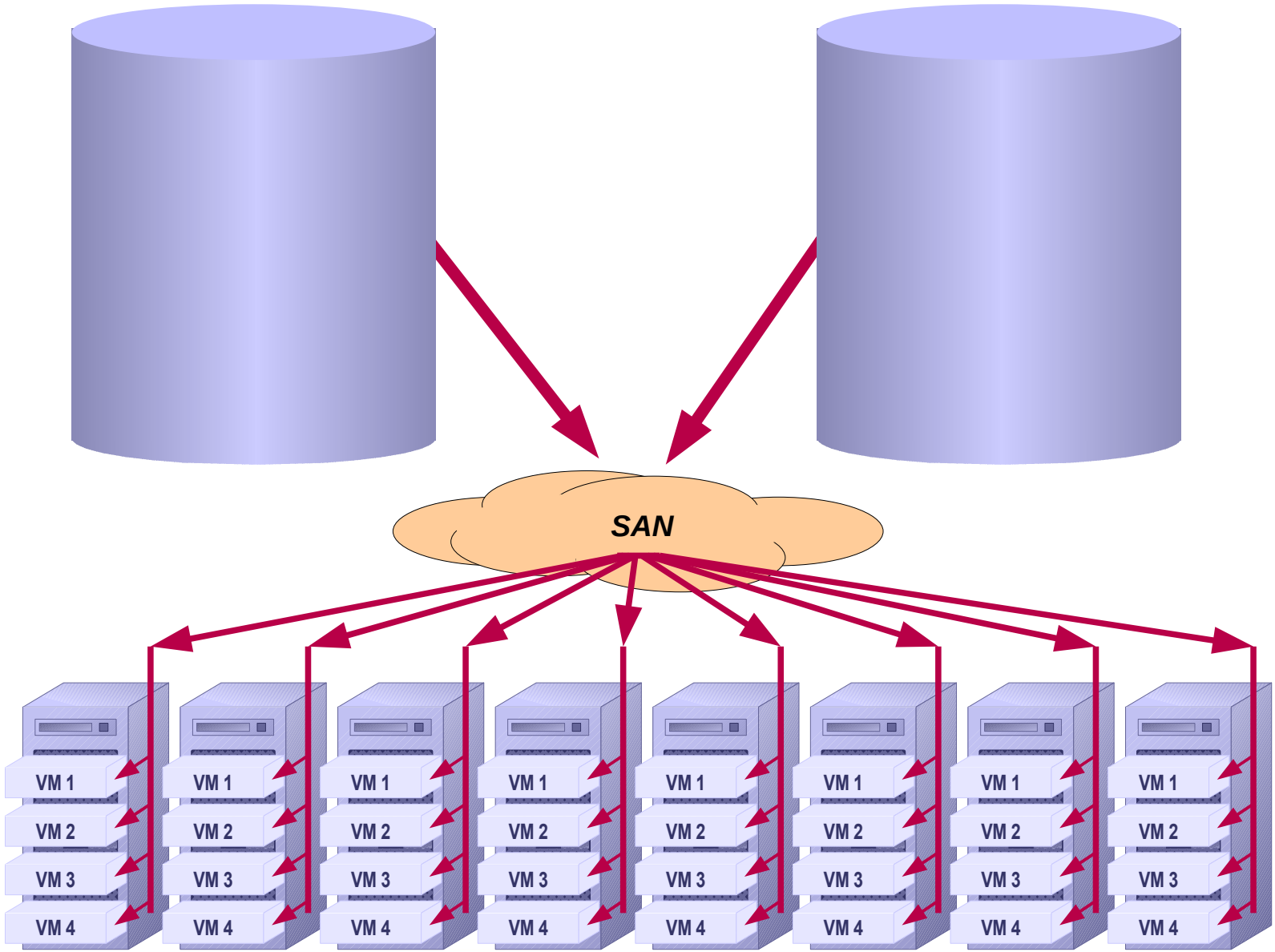
In den letzten 10 Jahren haben sich gewisse Trends verstärkt

- *Zahl der Serversysteme hat sich weiter vergrößert*
- *Massenspeicher heute extrem zentralisiert*
- *Vermittlung über oft komplexe Speichernetzwerke, die Anwender der Applikationen eigentlich nicht interessieren*

... und welchen Einfluß hat OS-Virtualisierung?

Vom Einfluß virtueller Maschinen

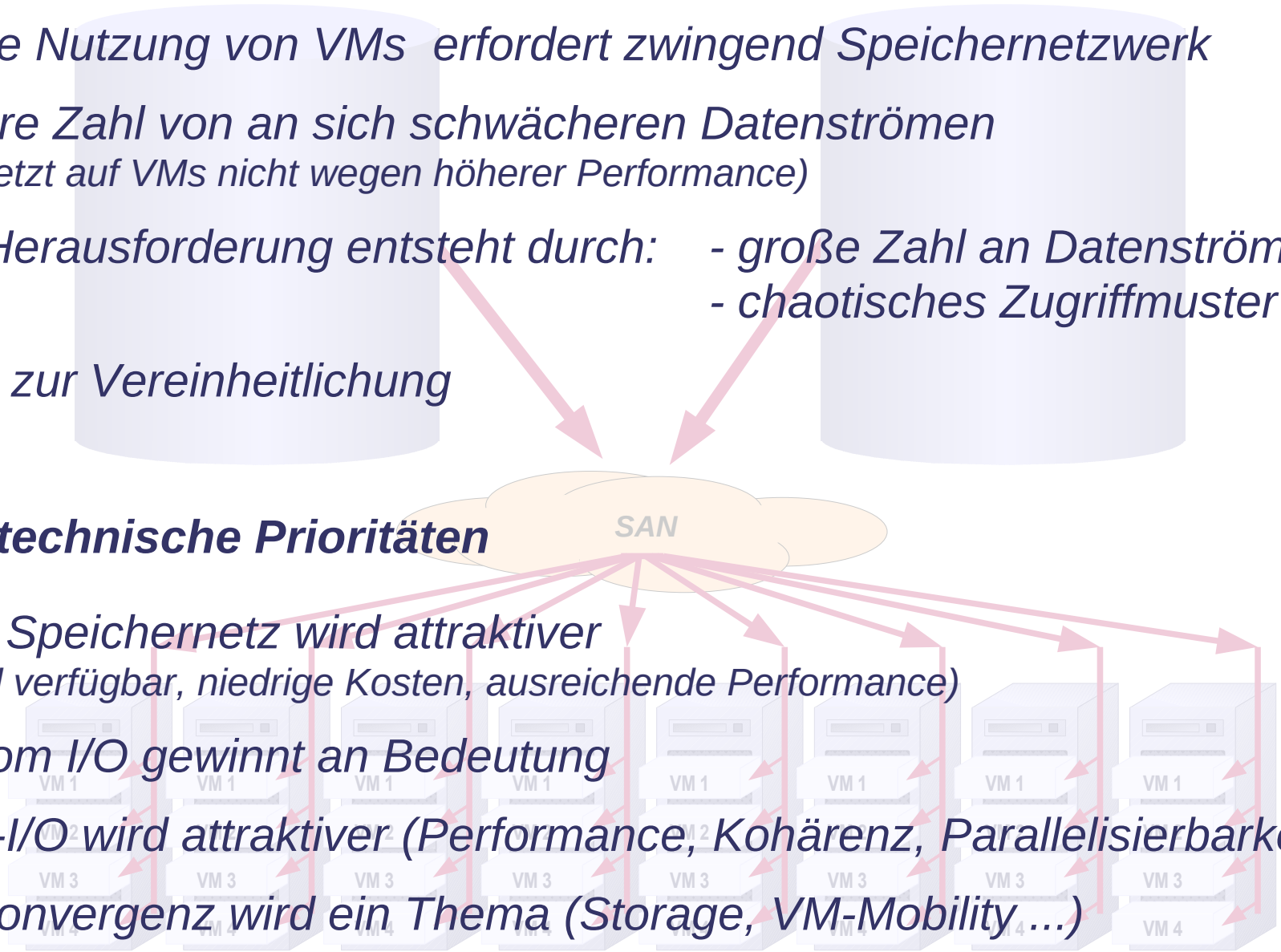
Anders strukturierte Datenströme / andere Prioritäten



Vom Einfluß virtueller Maschinen

Anders strukturierte Datenströme / andere Prioritäten

- *flexible Nutzung von VMs erfordert zwingend Speichernetzwerk*
- *größere Zahl von an sich schwächeren Datenströmen*
  (man setzt auf VMs nicht wegen höherer Performance)
- *Last-Herausforderung entsteht durch: - große Zahl an Datenströmen*
  - chaotisches Zugriffmuster
- *Druck zur Vereinheitlichung*

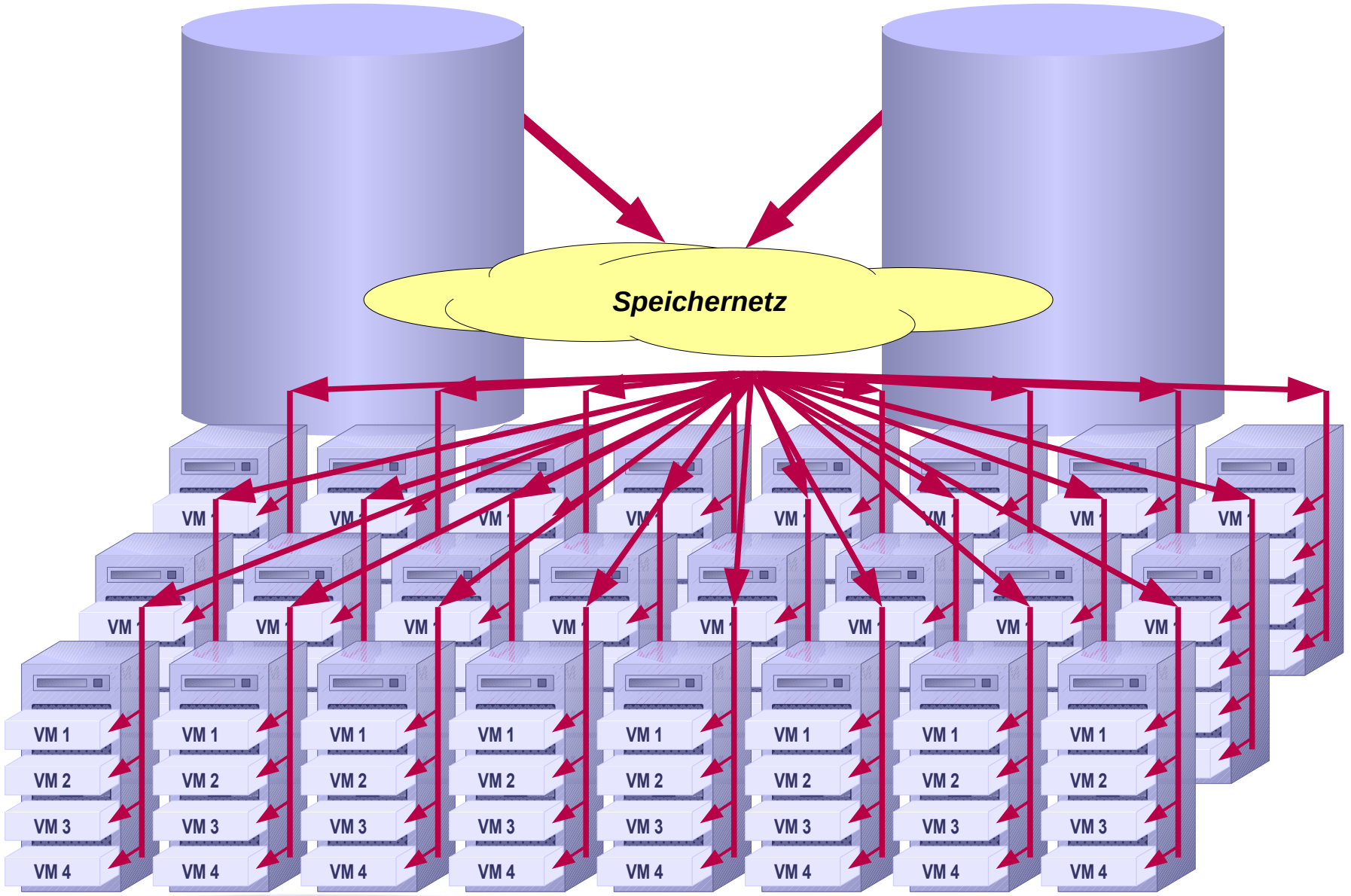


Andere technische Prioritäten

- *IP als Speichernetz wird attraktiver*
  (überall verfügbar, niedrige Kosten, ausreichende Performance)
- *Random I/O gewinnt an Bedeutung*
- *Block-I/O wird attraktiver (Performance, Kohärenz, Parallelisierbarkeit)*
- *Netzkonvergenz wird ein Thema (Storage, VM-Mobility …)*

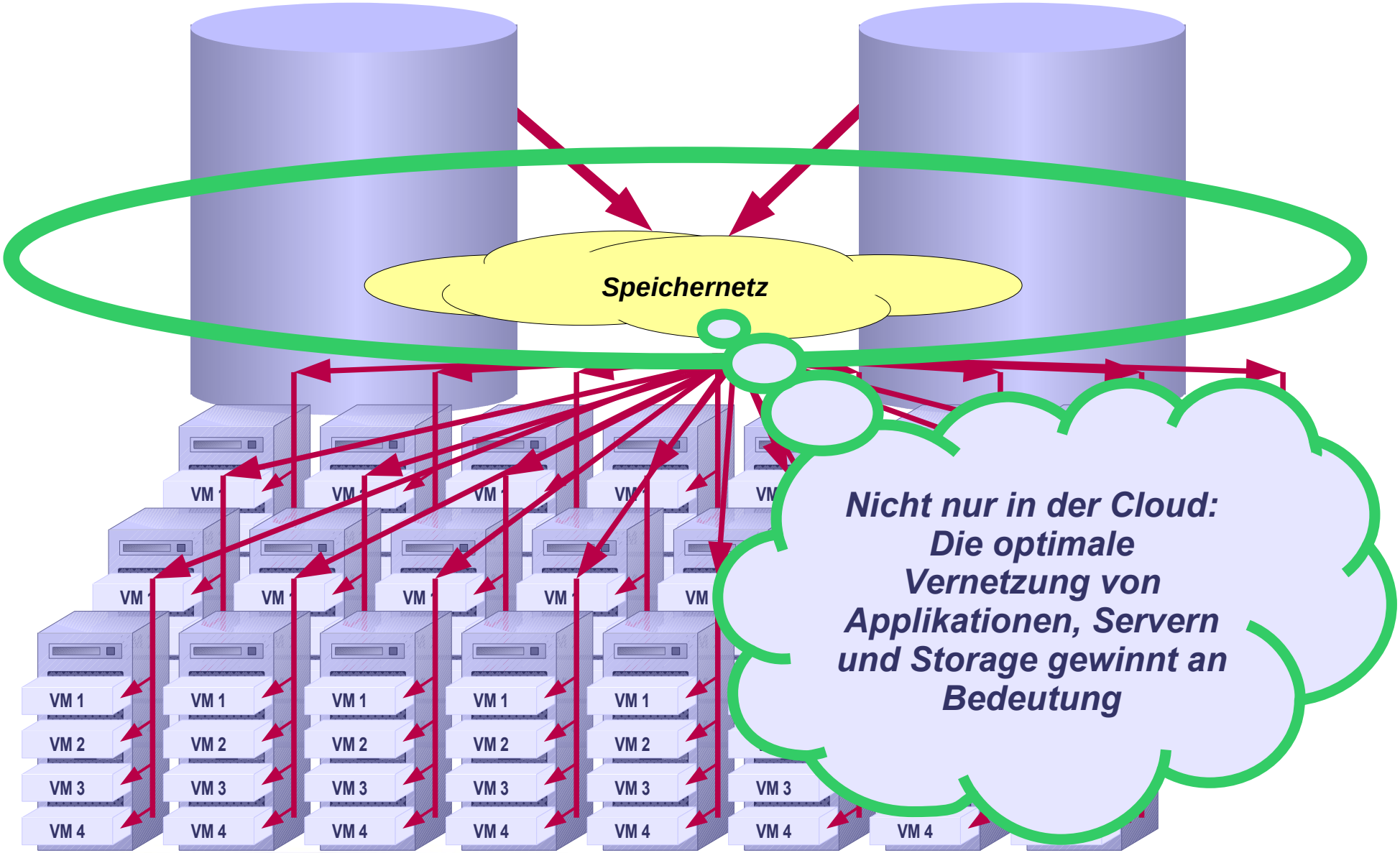
Die Steigerung: Cloud-Infrastrukturen

Spätestens hier ist eine neue Qualität erreicht



Die Steigerung: Cloud-Infrastrukturen

Spätestens hier ist eine neue Qualität erreicht



Zwei weitere Fragen

- *Erlauben die heutigen High-End-Speichersysteme Zugriff auf die modernsten Technologien im Storage-Bereich?*
- *Wenn die Grenzen zwischen Storage und Servern verschwimmen: Wie passen heutige Speichernetzparadigmen dazu?*

Zwei Hypothesen

- *Die schnellsten Komponenten stehen quasi für PC-Technologie (Standardsysteme) zur Verfügung*
- *SCSI (auch über FC oder IP) ist keine adäquate Antwort (mehr):*
  - keine oder wenig Kenntnis von Vernetzung / verteilten Funktionen
  - kaum parallelisierbar
  - in heutigen Systemen unnütze I/O-Transformationen
  - damit Performance- und Verfügbarkeitsnachteile

Zwei weitere Fragen

- *Erlauben die heutigen High-End-Speichersysteme Zugriff auf die modernsten Technologien im Storage-Bereich?*
- *Wenn die Grenzen zwischen Storage und Servern verschwimmen: Wie passen heutige Speichernetzparadigmen dazu?*

Zwei Hypothesen

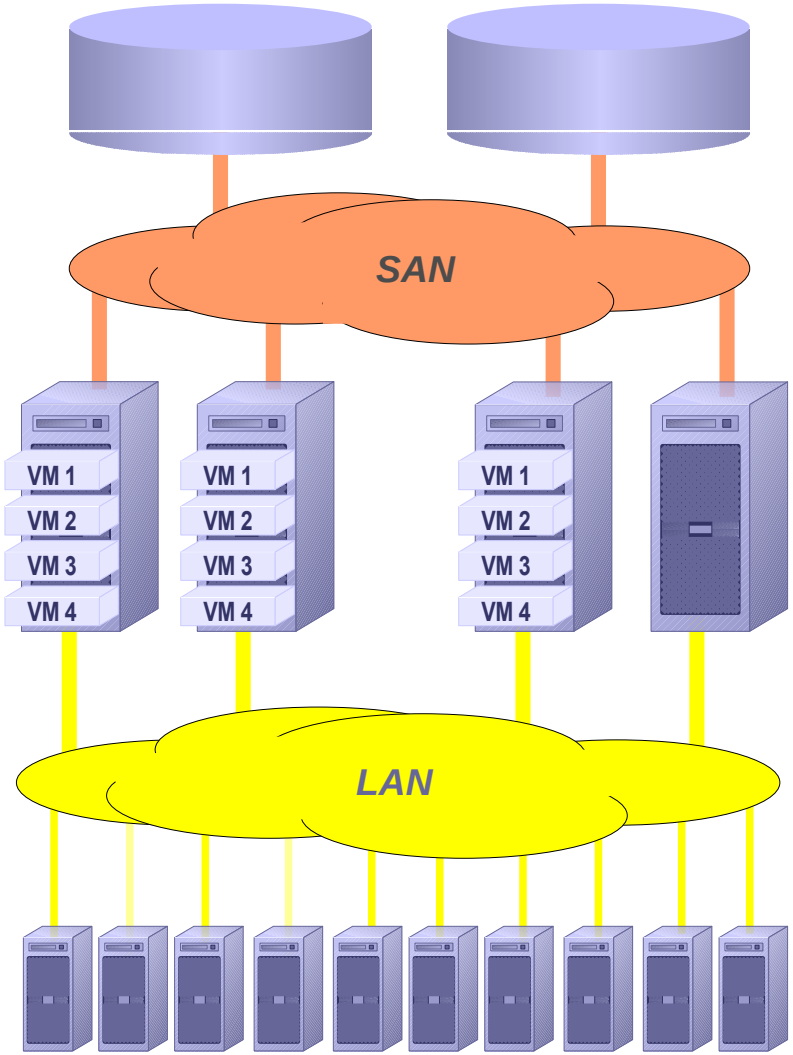
- *Die schnellsten Komponenten stehen quasi für PC-Technologie (Standardsysteme) zur Verfügung*
- *SCSI (auch über FC oder IP) ist keine adäquate Antwort (mehr):*
  - keine oder wenig Kenntnis von Vernetzung / verteilten Funktionen
  - kaum parallelisierbar
  - in heutigen Systemen unnütze I/O-Transformationen
  - damit Performance- und Verfügbarkeitsnachteile

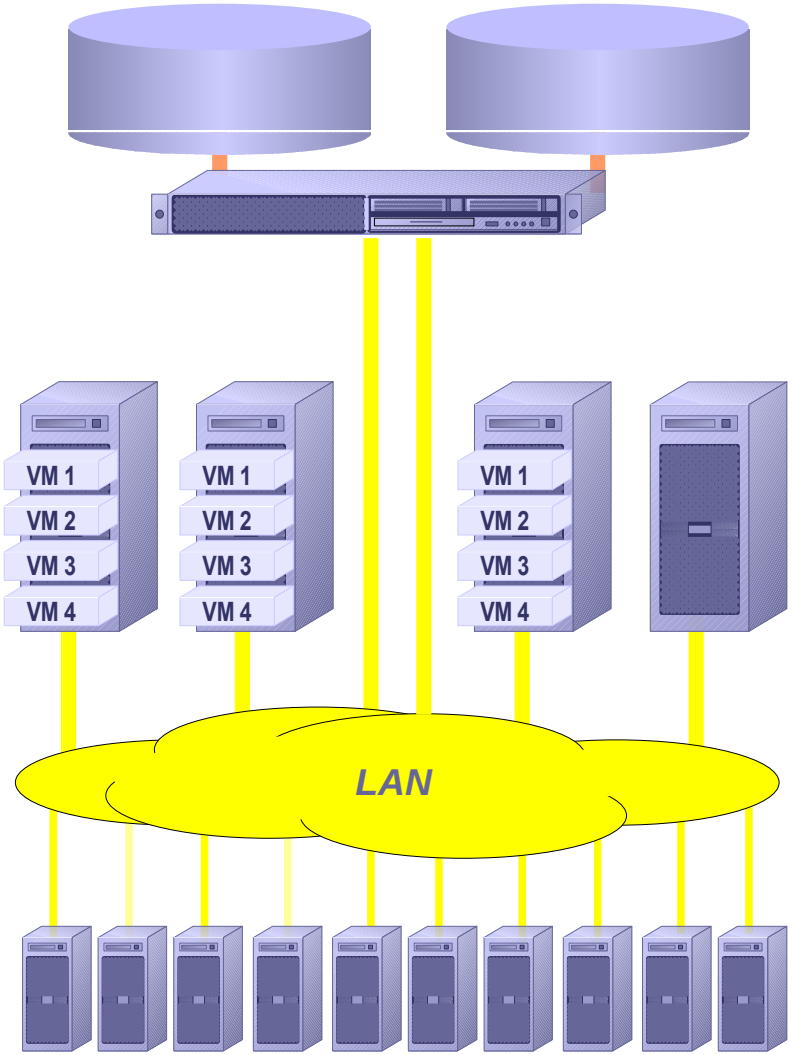
Ableitung / Schlußfolgerung

Es könnte eine Aufwertung von Standard-Servern, Mehrzweck-Netzwerken und hostbasierter Software geben!

Netzwerkkonvergenz

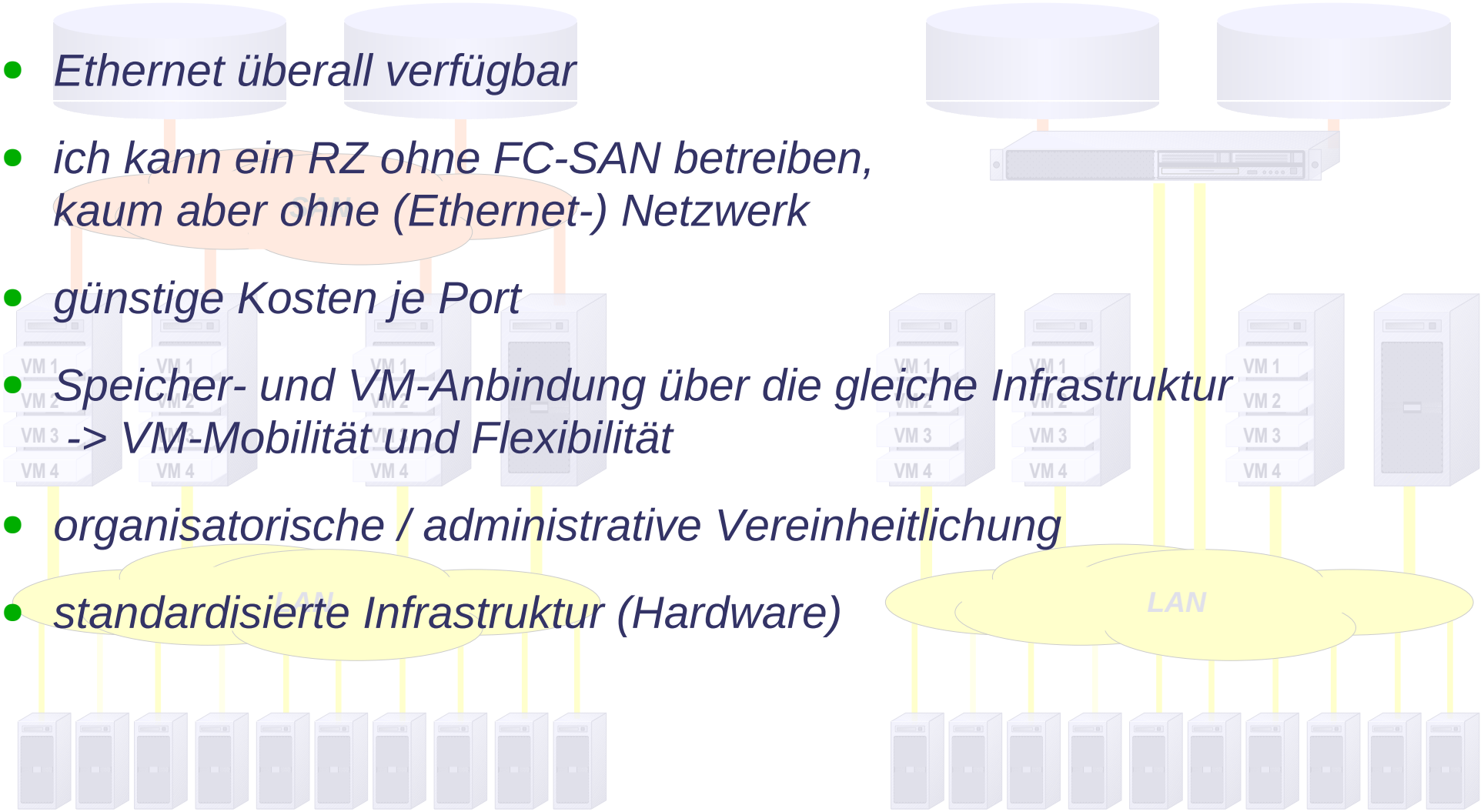
hat Vor- und Nachteile, ist aber unter vielen Aspekten interessant





Netzwerkkonvergenz

hat Vor- und Nachteile, ist aber unter vielen Aspekten interessant



- *Ethernet überall verfügbar*
- *ich kann ein RZ ohne FC-SAN betreiben, kaum aber ohne (Ethernet-) Netzwerk*
- *günstige Kosten je Port*
- *Speicher- und VM-Anbindung über die gleiche Infrastruktur -> VM-Mobilität und Flexibilität*
- *organisatorische / administrative Vereinheitlichung*
- *standardisierte Infrastruktur (Hardware)*

Und wo bleibt der Administrator?

Network Operations mit Unified I/O per FCoE

Server Operations
- *Drivers, Bindings*
- *Miscellaneous SCSI-settings*
- *LUN-Setup*
- *Multipathing*
- *Disk Partitioning*
- *Network Management*

SAN Operations
- *FC Interface Mode and Speed*
- *Virtual Fabrics*
- *Port Channels & Trunks*
- *NPV*
- *Monitoring & SAN Mgmt. Systems*
- *Device Management*

Converged Network Operations
- *Switch Management*
- *Images, Users, Virtual Interfaces*
- *Management Interfaces*
- *System Level Support*
- *Span Ports*

LAN Operations
- *Drivers, Port Configurations*
- *VLANs*
- *Port Channels*
- *Trunks*
- *Monitoring + LAN Management*
- *Device Management*

Vgl. “Design and Implementations of FCoE for the DataCenter” - SNIA Educational Paper, 2011

Und wo bleibt der Administrator?

Network Operations mit Unified I/O per FCoE

Server Operations
- *Drivers, Bindings*
- *Miscellaneous SCSI-settings*
- *LUN-Setup*
- *Multipathing*
- *Disk Partitioning*
- *Network Management*

SAN Operations
- *FC Interface Mode and Speed*
- *Virtual Fabrics*
- *Port Channels & Trunks*
- *NPV*
- *Monitoring & SAN Mgmt. Systems*
- *Device Management*

Converged Network Operations
- *Switch Management*
- *Images, Users, Virtual Interfaces*
- *Management Interfaces*
- *System Level Support*
- *Span Ports*

LAN Operations
- *Drivers, Port Configurations*
- *VLANs*
- *Port Channels*
- *Trunks*
- *Monitoring + LAN Management*
- *Device Management*

Vgl. “Design and Implementations of FCoE for the DataCenter” - SNIA Educational Paper, 2011

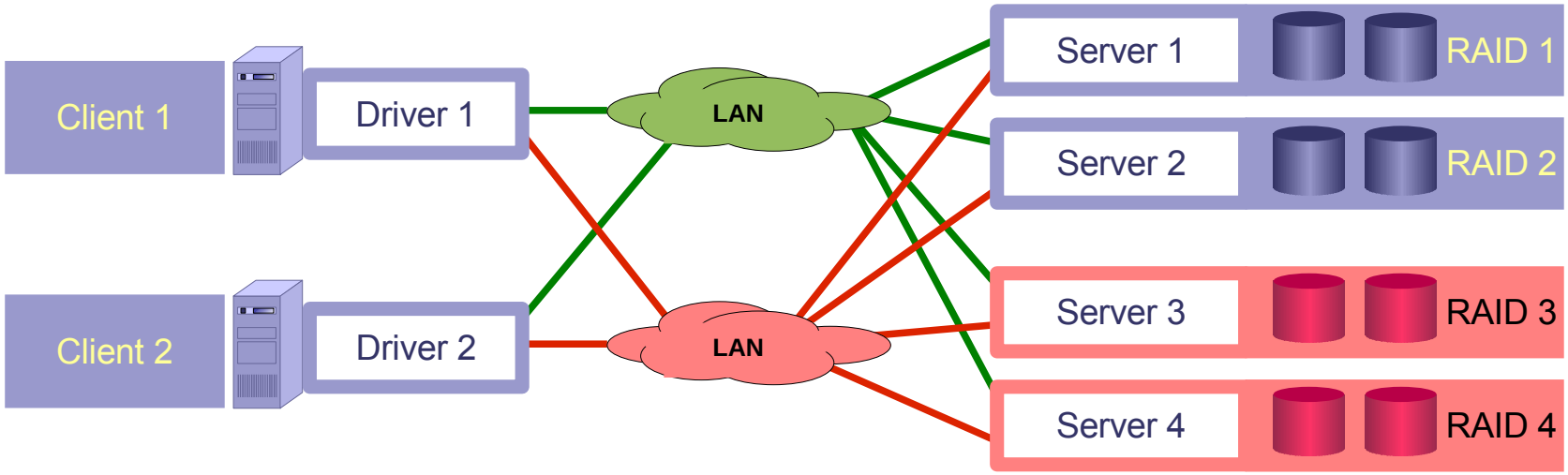
Netzwerkkonvergenz bei Hardware ist nur die halbe Miete. Genauso wichtig ist eine Vereinfachung der Administration.

RSIO - Remote Storage I/O

Idee & Konzept

Block-I/O über Ethernet – einmal anders gedacht

Für vernetzte Strukturen auch Netzwerkparadigmen anwenden

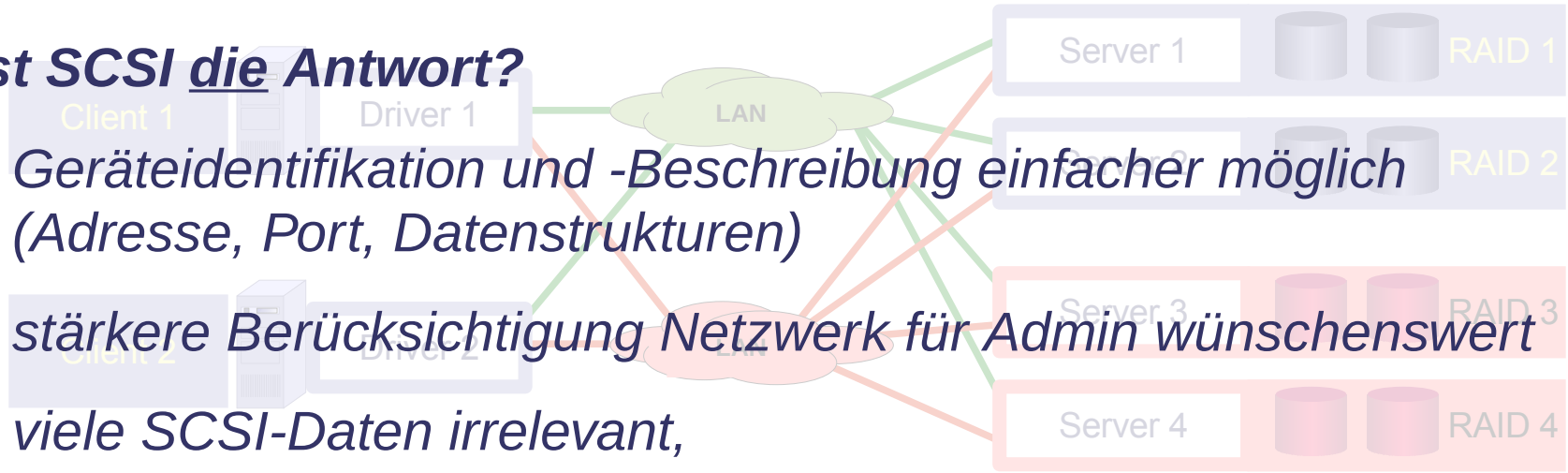


- *I/O-Requests senden read(), write(), ioctl()*
- *geeignete Kapselung*
- *Verbindungsauf- und Abbau, Überwachung*
- *Kanal-Multiplexing*

- *I/O-Requests verarbeiten read(), write(), ioctl()*
- *geeignete Kapselung*
- *Verbindungsauf- und Abbau, Überwachung*
- *Kanal-Multiplexing*

Block-I/O über Ethernet – einmal anders gedacht

Für vernetzte Strukturen auch Netzwerkparadigmen anwenden



Ist SCSI die Antwort?

- *Geräteidentifikation und -Beschreibung einfacher möglich (Adresse, Port, Datenstrukturen)*
- *stärkere Berücksichtigung Netzwerk für Admin wünschenswert*
- *viele SCSI-Daten irrelevant, dafür sind viele interessante Funktionen kaum darstellbar*
- *ohne SCSI keine Wandlung auf Low-Level-Protokoll erforderlich*
- *bestimmte SCSI-Mechanismen im Netz kontraproduktiv (z. B. Bus-Reset)*
- *reduzierter Kommunikationsaufwand möglich*

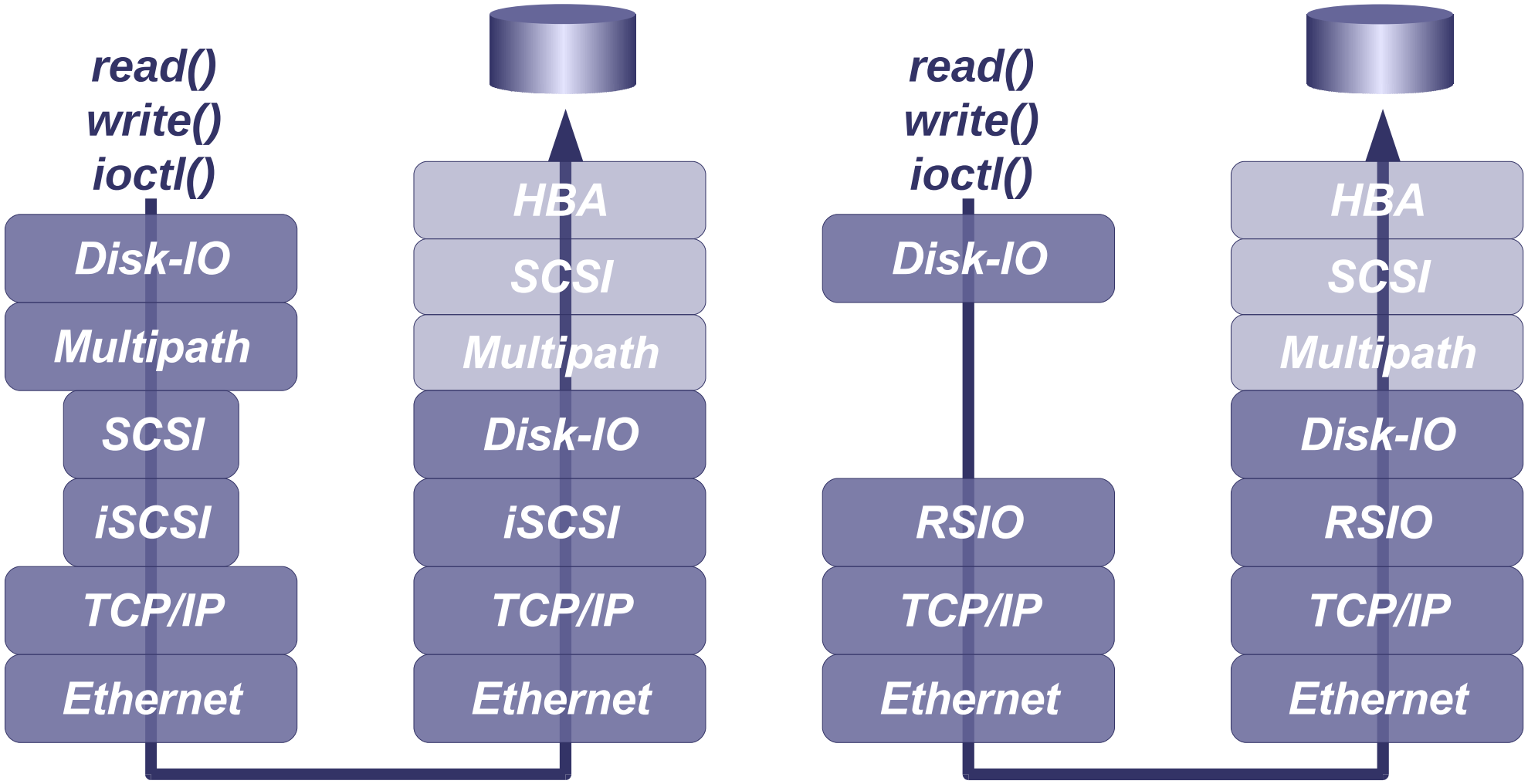
RSIO - Remote Storage I/O

Eckdaten der neuen Technologie für LAN-attached (Shared) Block Devices

- *neues, von OSL entwickeltes Protokoll*
- *direkter Transport aller relevanten IO-Aufrufe (read, write, ioctl)*
- *integriert Verbindungsaufbau, Überwachung, Path-Multiplexing, Trunking*
- *fähig zu Selbstkonfiguration und Error Recovery*
- *kann alle modernen Storage-Szenarien abbilden:*
  - *einfache Server und Clients, ggf. mit Multipathing*
  - *Cluster von Storage-Servern (Targets)*
  - *Cluster von Storage Clients (Initiators)*
  - *integrierte Cluster von Servern und Clients*
  - *Storage Server Farms*
  - *Cloud-Konzepte*
- *besondere Eignung für Kombination mit Speichervirtualisierung*
  - *eingängige Namen*
  - *fdisk (Partitionierung) auf Clientseite entfällt*
  - *On-Demand-Allokation und Online-Rekonfiguration*
  - *viele weitere Sonderfunktionen*
  - *ermöglicht Administration vom Client aus*

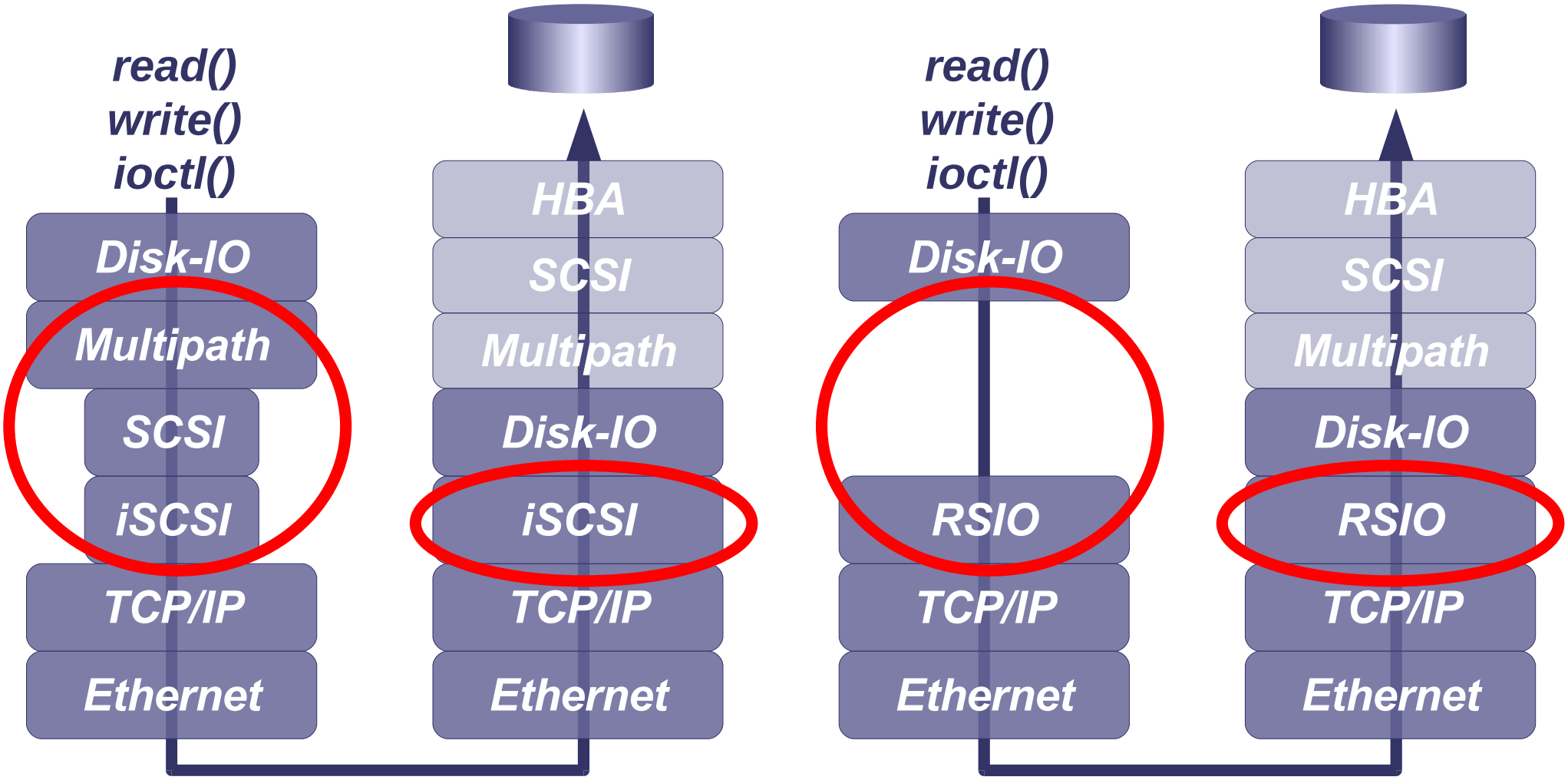
RSIO - Remote Storage I/O

Vergleich der Protokollstacks



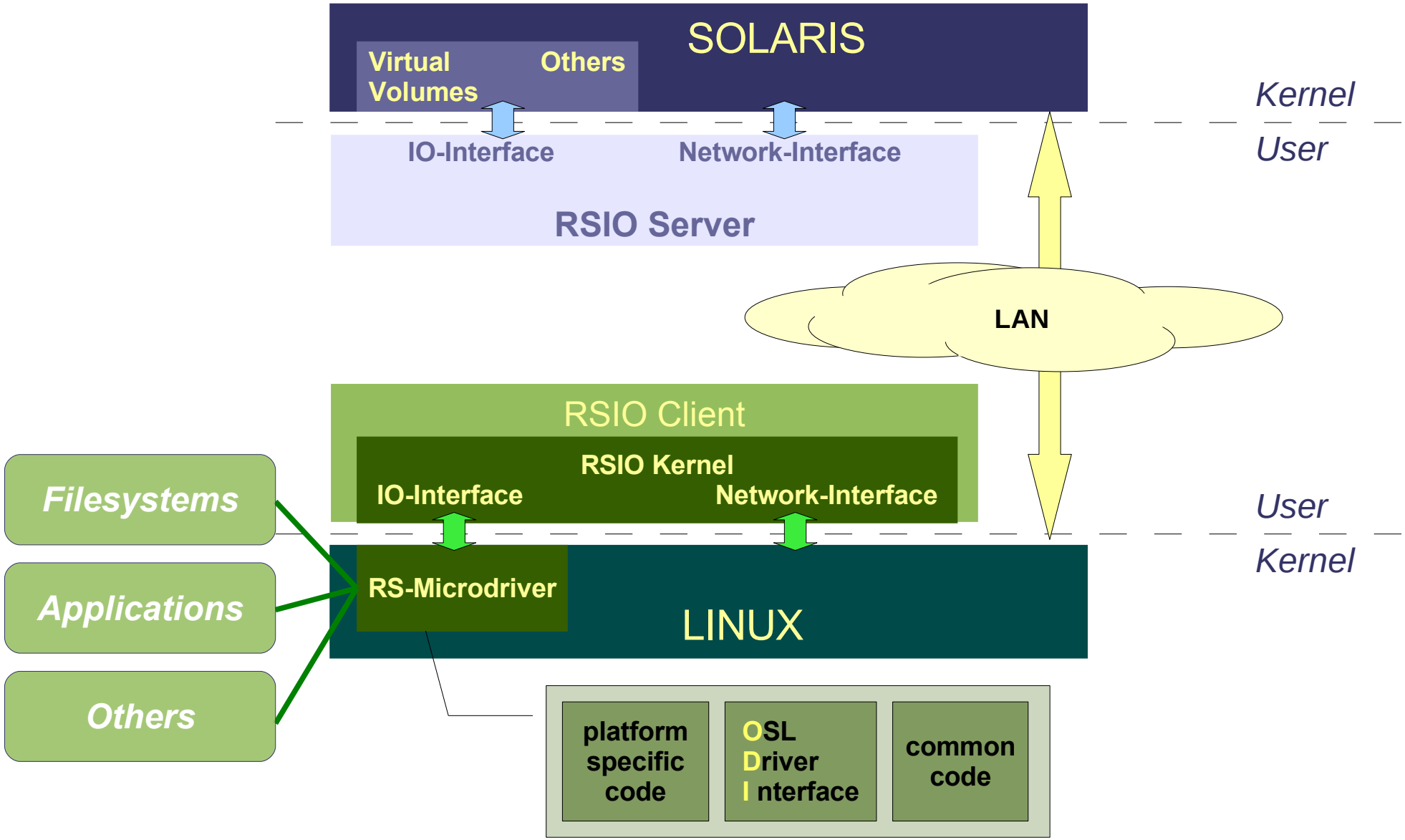
RSIO - Remote Storage I/O

Vergleich der Protokollstacks



RSIO – Portables Treiberdesign

Der Kernel im Userspace



RSIO – Portables Treiberdesign

Der Kernel im Userspace

Vorteile:

- *Größter Teil läuft im Userspace*
  - *-> Portabilität*
  - *-> Systemstabilität*
  - *-> Fehlerbehandlung + Debugging*
  - *-> Handling (z. B. Clusterumgebungen)*
  - *-> Wahlfreiheit bei Entwicklungsplattform*
- *Schlanker Kernel*
- *Common Kernel Code*

Und was ist mit der Performance?

Protokoll erlaubt hohe Performance und beeindruckende Skalierbarkeit

Server-Performance bei Cache Read / 8k

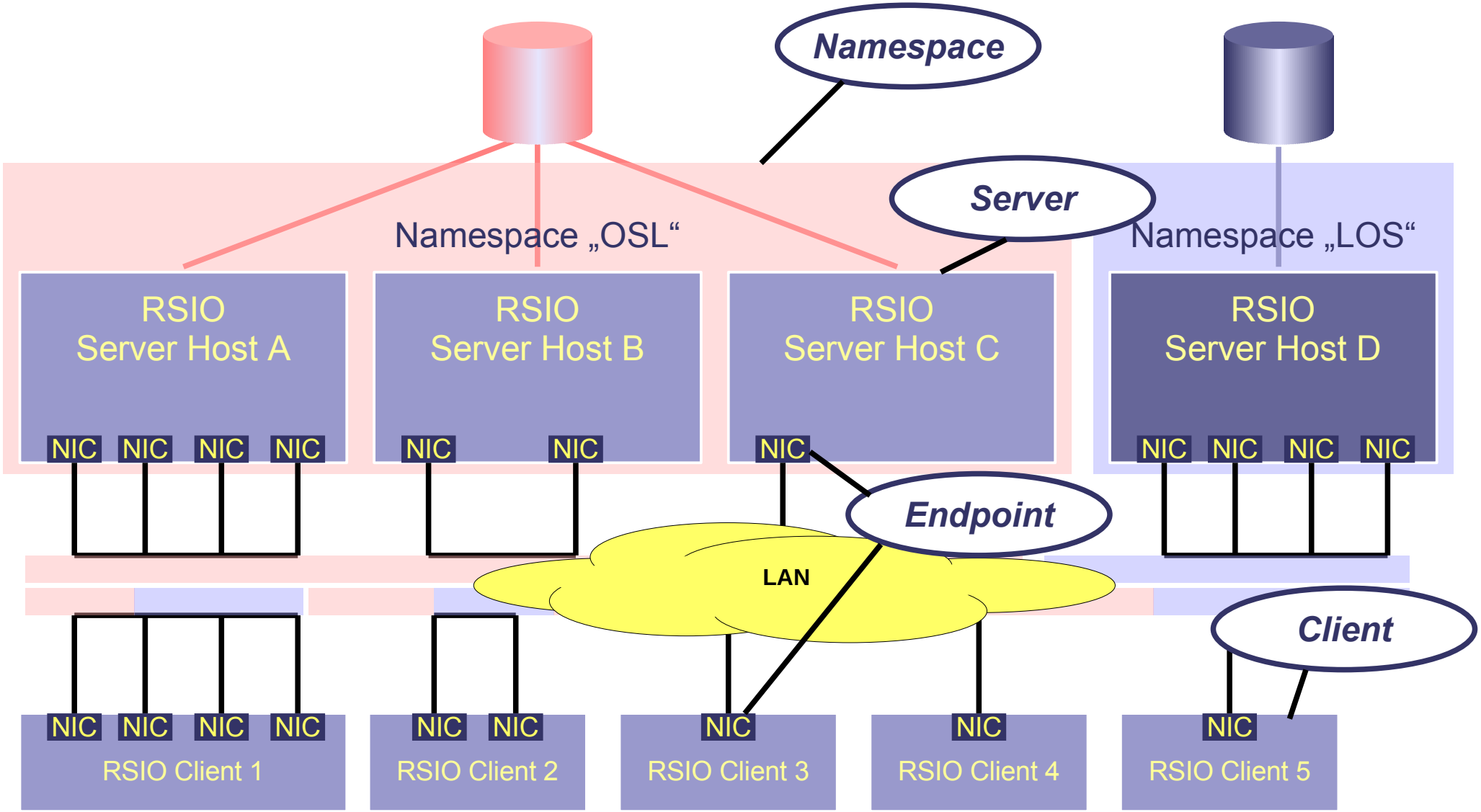
iSCSI	*10 Clients*	*100 Threads*	*7,6 Cores*	***31.000 IOPS***
iSCSI / comstar	*10 Clients*	*100 Threads*	*10,0 Cores*	***85.000 IOPS***
RSIO	*4 Clients*	*64 Threads*	*5,6 Cores*	***98.000 IOPS***
RSIO	*4 Clients*	*128 Threads*	*6,3 Cores*	***102.000 IOPS***

Client-Performance Throughput

RSIO	*1 x 1 GBit*	*ca. 0,5 Cores*	***> 110 MByte/s***
RSIO	*2 x 1 GBit*	*ca. 1,0 Cores*	***> 220 MByte/s***
RSIO	*4 x 1 GBit*	*ca. 2,0 Cores*	***> 440 MByte/s***
RSIO	*8 x 1 GBit*	*> 4,0 Cores*	***bis > 900 MByte/s***

RSIO – Architektur im Überblick

Klar gegliedertes und flexibles administratives Konzept



RSIO - Remote Storage I/O

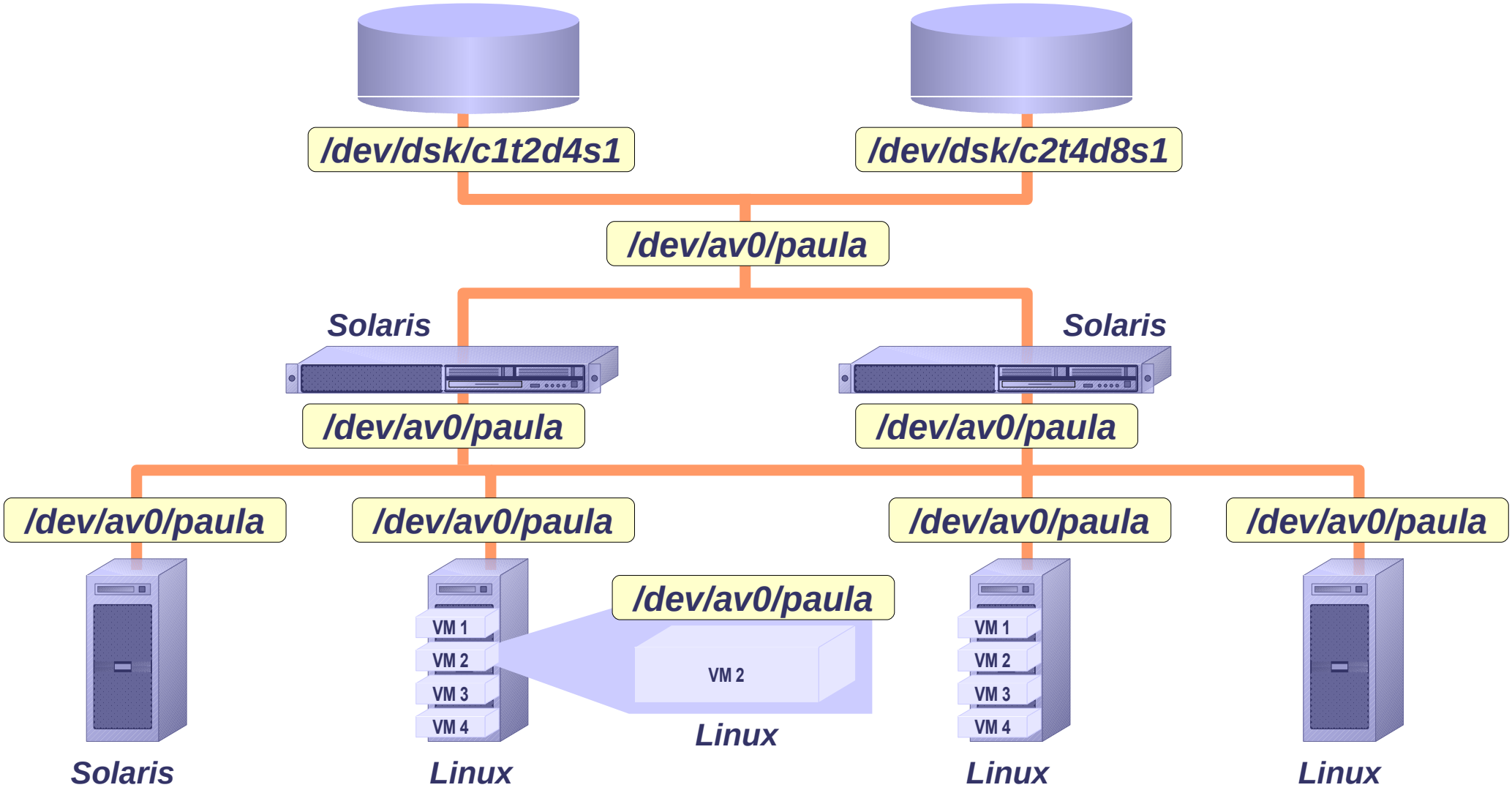
Die Funktionen

- *Server(prozeß)*
  - *optionaler Info-Service*
  - *erwartet Verbindungsanfragen und arbeitet diese ab*
  - *unterhält Client-Listen*
  - *Access Control Lists (welcher Client hat Zugriff auf welche Volumes)*
  - *Abarbeitung der I/O-Anfragen*
  - *Multiplexing über vorhandene Interfaces*
  - *Management-Interface (rsadmin)*

- *Client*
  - *Zugriff auf Server-Informationen (Info-Service, Liste verfügbarer Volumes)*
  - *Initiieren der Verbindungen, Steuerung Path-Recovery und Server-Failover*
  - *Bereitstellung der Devices*
  - *Initiierung I/O-Anfragen / Umsetzung Antworten*
  - *Multiplexing über vorhandene Interfaces*
  - *Management-Interface (rsconfig)*

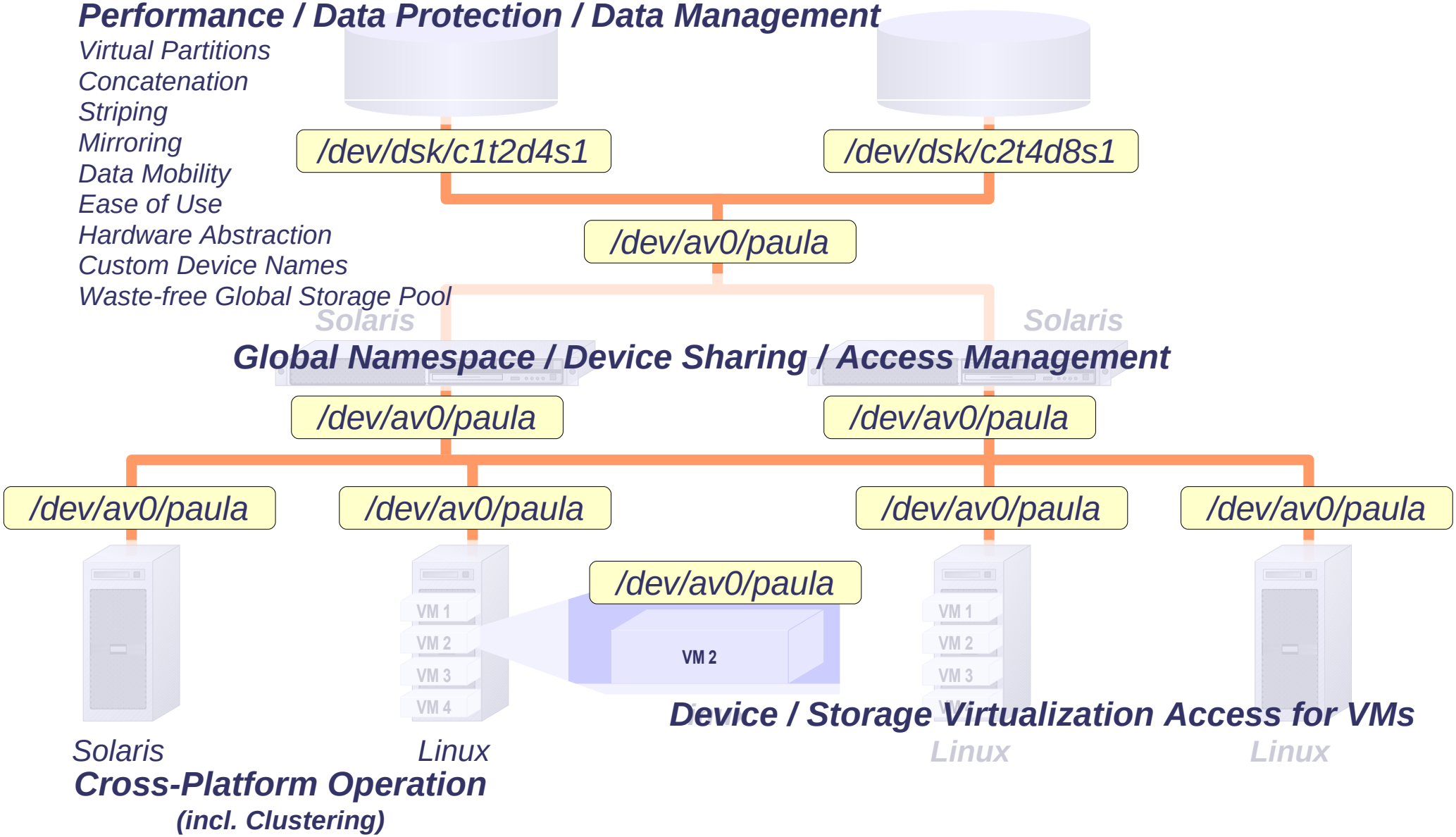
RSIO und Speichervirtualisierung

Was bringt die Verknüpfung mit einer clusterfähigen Speichervirtualisierung?



RSIO und Speichervirtualisierung

Was bringt die Verknüpfung mit einer clusterfähigen Speichervirtualisierung?

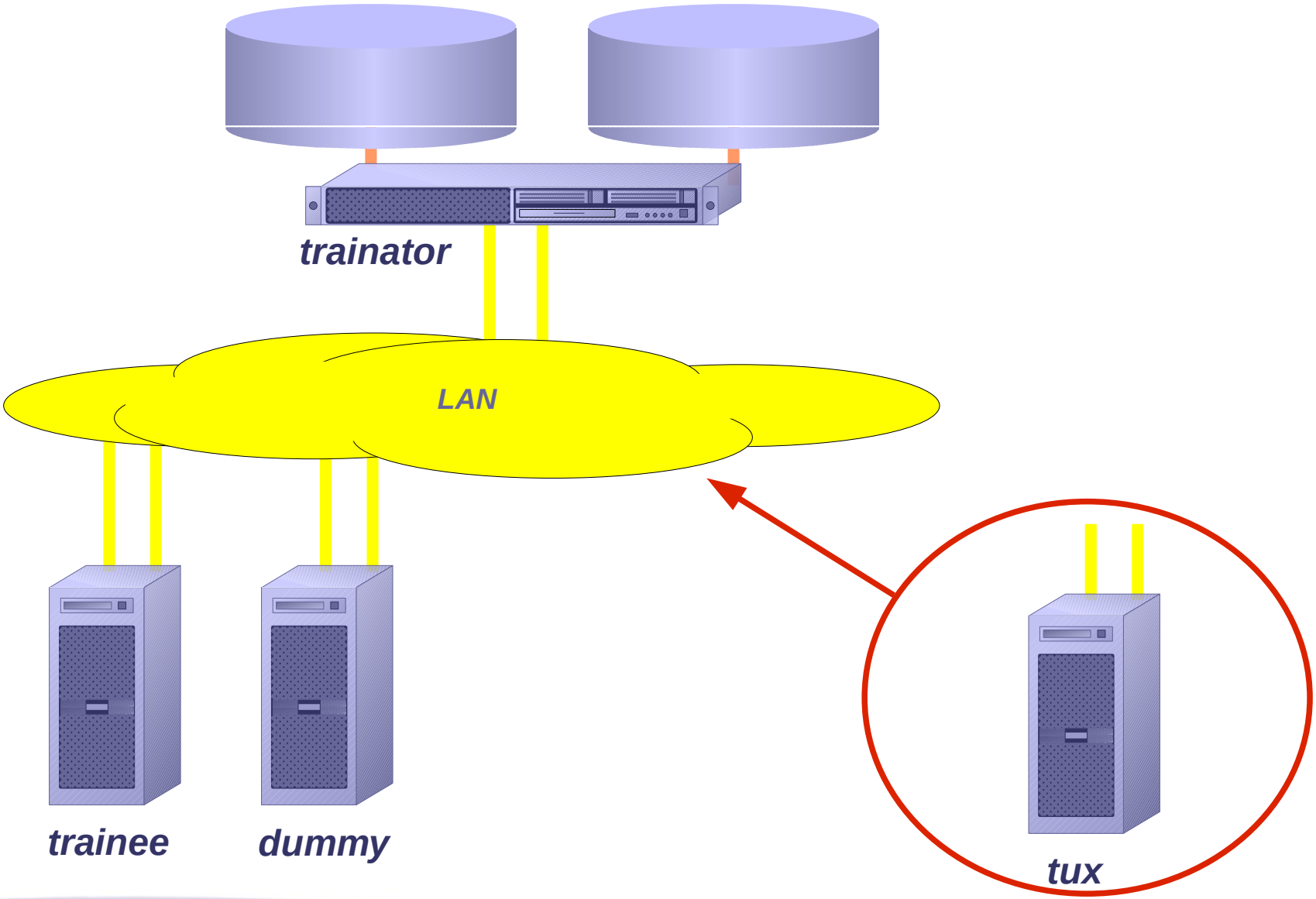


RSIO - Remote Storage I/O

So sieht es aus

RSIO Administration in der Praxis

Beispielkonfiguration



OSL Gesellschaft für offene Systemlösungen mbH

www.osl.eu

RSIO Serveradministration - Beispiele

Anzeigen von Clients / Hinzufügen eines Clients

```
srv# rsadmin -q
0001 trainee
0002 dummy

srv# rsadmin -qv
  id node name        node key                                  mode
-------------------------------------------------------------------------
0001 trainee          0139dfX982                                rsio client
0002 dummy            1234567890                                rsio client



srv# rsadmin -c tux -k Linuxtag
node tux added with rs nodeid 3

srv# rsadmin -qv
  id node name        node key                                  mode
-------------------------------------------------------------------------
0001 trainee          0139dfX982                                rsio client
0002 dummy            1234567890                                rsio client
0003 tux              Linuxtag                                  rsio client
```

RSIO Serveradministration - Beispiele

Anzeigen von Clients / Hinzufügen eines Clients

```
srv# rsadmin -q
0001 trainee
0002 dummy

srv# rsadmin -qv
  id node name        node key                                  mode
-------------------------------------------------------------------------
0001 trainee          0139dfX982                                rsio client
0002 dummy            1234567890                                rsio client



srv# rsadmin -c tux -k Linuxtag                                 create client
node tux added with rs nodeid 3

srv# rsadmin -qv                                                query all clients
  id node name        node key                                  mode
-------------------------------------------------------------------------
0001 trainee          0139dfX982                                rsio client
0002 dummy            1234567890                                rsio client
0003 tux              Linuxtag                                  rsio client
```

RSIO Serveradministration - Beispiele

Anlegen eines Volumes und Zugriff erlauben (grant access)

```
srv# smgr -c eisberg -S 1g

srv# rsadmin -g eisberg tux
access to eisberg@0 granted to tux

srv# rsadmin -qa
0 testvol          : trainee
0 probevol         : dummy,trainee
0 eisberg          : tux
```

Sofortige Sichtbarkeit auf dem Client

```
clt# rsconfig -q
000 rsio
    clt: tux
    srv: 000 trainator
         0  eisberg         disk         2097152 blocks, blocksize  512 bytes
clt# rsconfig -qvv
000 rsio, mode: simple rsio client
    clt: tux
    srv: 000 trainator
         0  eisberg         disk         2097152 blocks, blocksize  512 bytes
            c: /dev/av0/reisberg
            b: /dev/av0/eisberg
```

RSIO Serveradministration - Beispiele

Anlegen eines Volumes und Zugriff erlauben (grant access)

```
srv# smgr -c eisberg -S 1g                    create volume (1 gigabyte)

srv# rsadmin -g eisberg tux                   grant access
access to eisberg@0 granted to tux

srv# rsadmin -qa                              query all volume permissions
0 testvol          : trainee
0 probevol         : dummy,trainee
0 eisberg          : tux
```

Sofortige Sichtbarkeit auf dem Client

```
clt# rsconfig -q
000 rsio                                      namespace
    clt: tux                                  server
    srv: 000 trainator                        my client name
         0  eisberg          disk        2097152 blocks, blocksize  512 bytes
clt# rsconfig -qvv
000 rsio, mode: simple rsio client
    clt: tux
    srv: 000 trainator                        device path proposal
         0  eisberg          disk        2097152 blocks, blocksize  512 bytes
            c: /dev/av0/reisberg
            b: /dev/av0/eisberg
```

RSIO Clientadministration - Beispiele

Volume Attach und Filesystem erzeugen

```
clt# ls -lai /dev/av0/eisberg
ls: cannot access /dev/av0/eisberg: No such file or directory
clt# rsconfig -a
clt# ls -lai /dev/av0/eisberg
66274 brw------- 1 root root 246, 8 May 24 20:33 /dev/av0/eisberg
clt# rsconfig -lvv
rsio::eisberg@0                                  2097152 blocks,   1 server(s)
    c: -
    b: /dev/av0/eisberg
clt# mkfs -t ext2 /dev/av0/eisberg
mke2fs 1.41.1 (01-Sep-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 35 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
clt# mount /dev/av0/eisberg /mnt
clt# df -h /mnt
Filesystem            Size  Used Avail Use% Mounted on
/dev/av0/eisberg     1008M  1.3M  956M   1% /mnt
```

RSIO – Und wieder auf dem Server

Welche Clients sind “connected”?

```
srv# rsadmin -lvv
000 rsio
    000 (id   3) tux              Linux 2.6  LP64   little endian
srv# rsadmin -lvvv
000 rsio
    000 (id   3) tux              Linux 2.6  LP64   little endian
    IP(TCP) 192.168.1.100/5000<->192.168.1.110/5000 connected  tx: ok rx: ok
    IP(TCP) 192.168.2.100/5000<->192.168.2.110/5000 connected  tx: ok rx: ok
```

Welche Volumes sind im Zugriff welcher Clients?

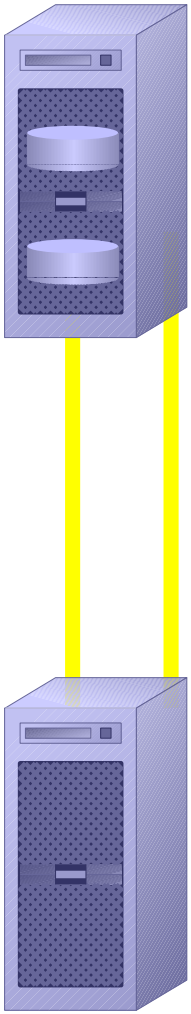
```
srv# rsadmin -qA
0 testvol                0 ---                98  0x00000024
0 probevol               0 ---                98  0x00000002
0 eisberg                3 tux                98  0x00000030  RW
```

RSIO - Remote Storage I/O

Szenarien

Das einfachste Szenario

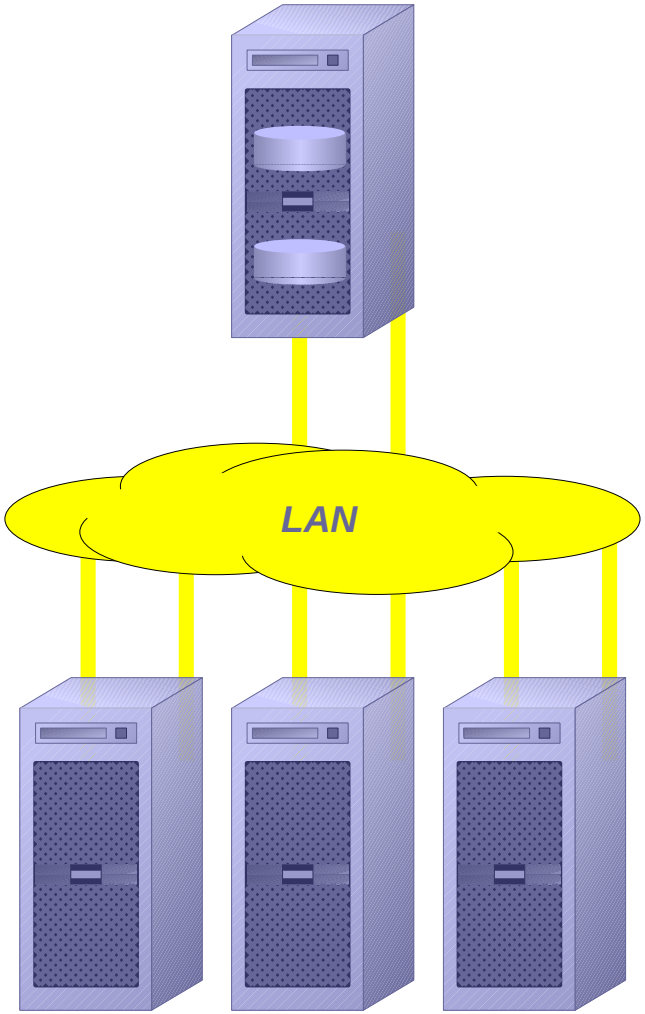
Eine Maschine als RSIO-Device-Server



- *Client kann Storage des Servers nutzen*
- *“FC-Funktion ohne FC”*
- *Multipathing*
- *Performance gleich oder besser als mit lokalen Platten (Profitieren von serverseitiger Virtualisierung)*

Das einfachste Szenario

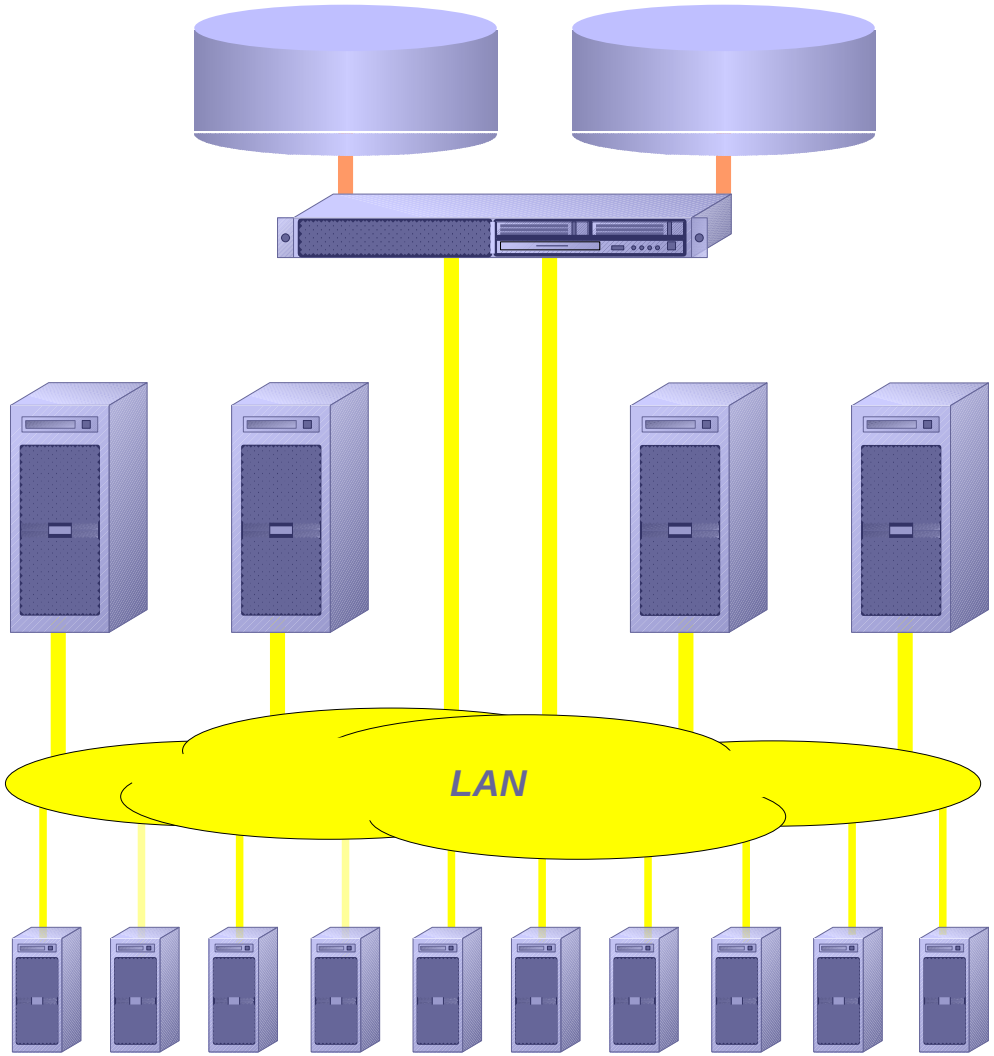
Eine Maschine als RSIO-Device-Server – natürlich auch für mehrere Clients



- *Clients können Storage des Servers nutzen*
- *SAN ohne FC (SAN per LAN)*
- *Multipathing*
- *Performance gleich oder besser als mit lokalen Platten (Profitieren von serverseitiger Virtualisierung)*
- *Global Namespace*
- *Device-Sharing möglich*
- *hostübergreifender Storage-Pool*

Immer noch ein einfaches Szenario

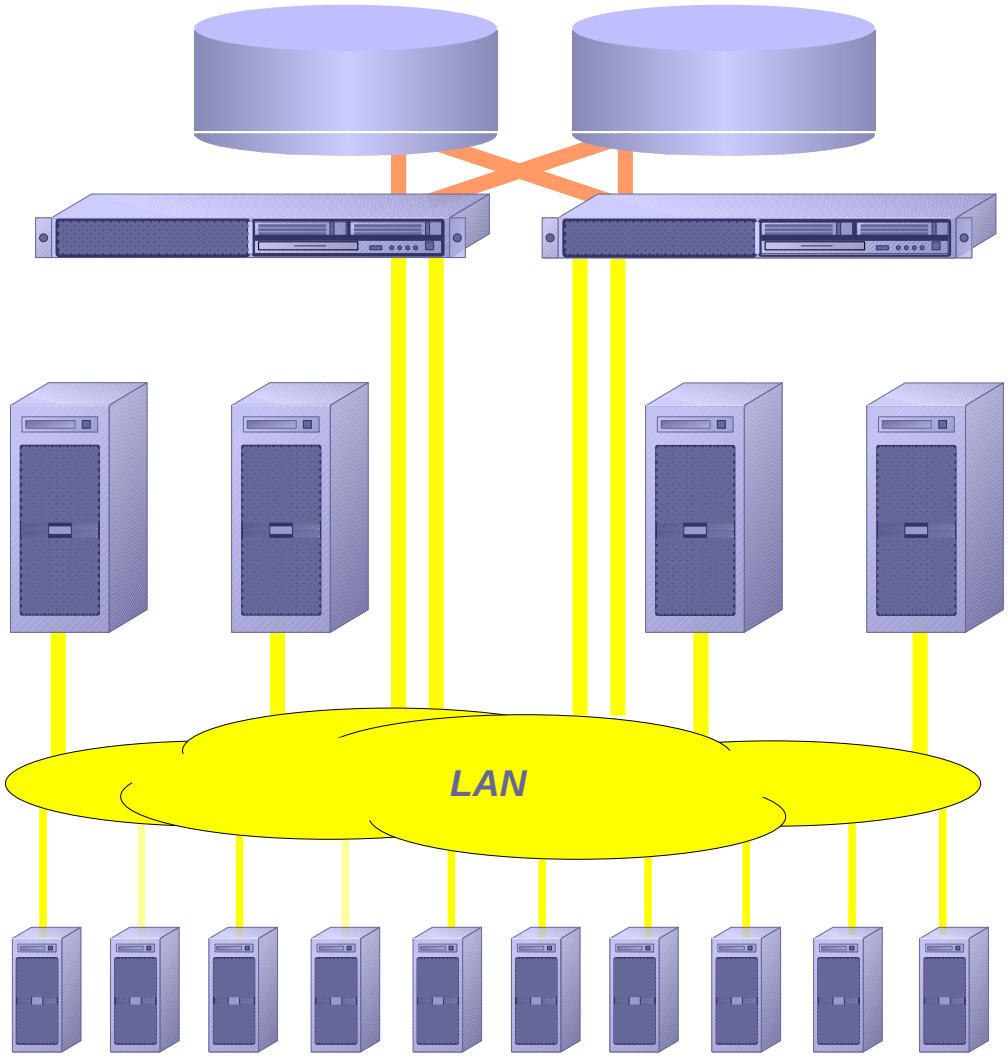
Einbeziehung von FC-Devices - RSIO-Server als LAN-to-FC-Gateway



- *bestmögliches Fanout-Verhältnis für die FC-Ports der RAID-Systeme*
- *preiswerteste und trotzdem performante und zuverlässige Storage-Anbindung*

Für höhere Ansprüche

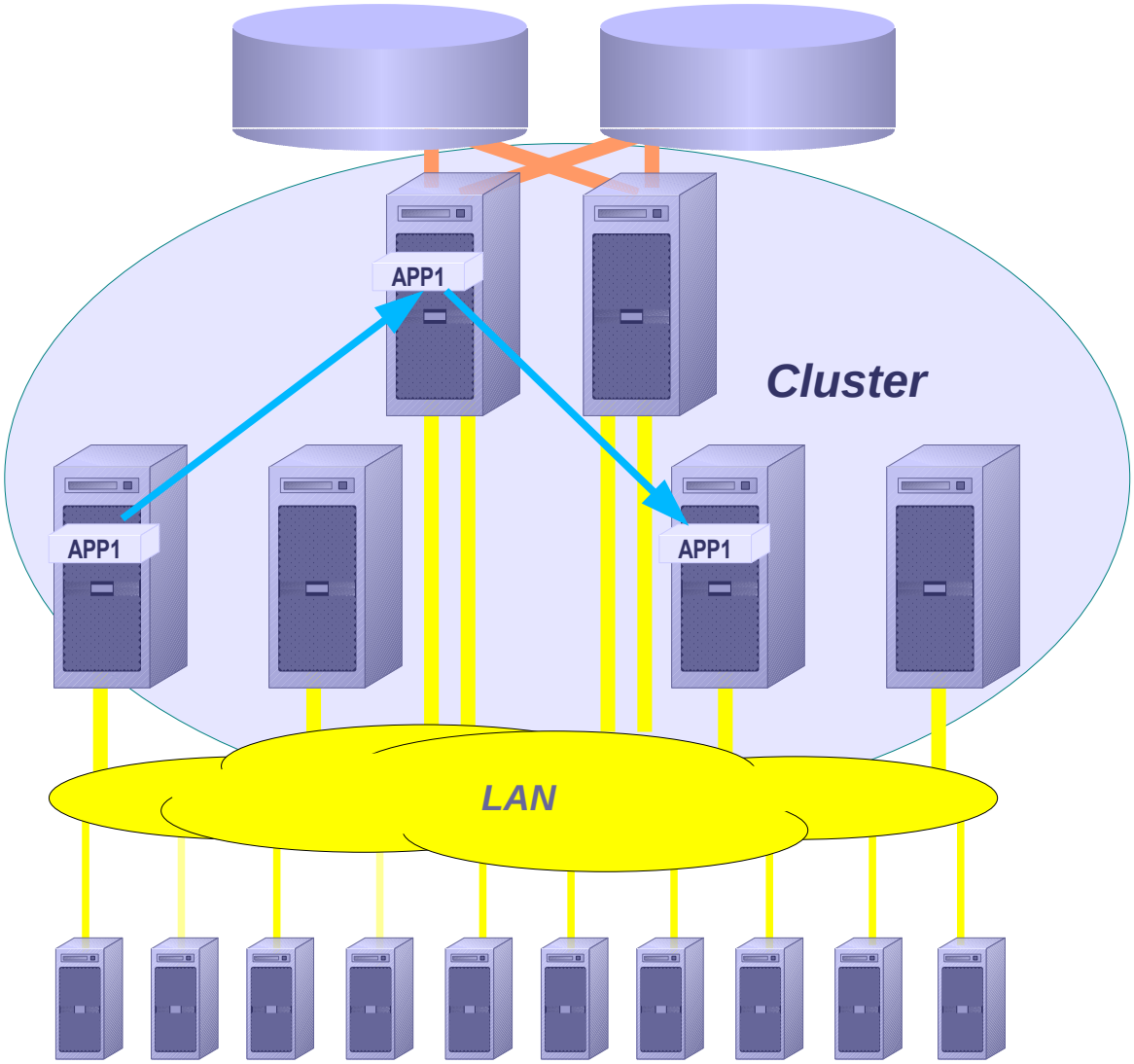
Geclusterte Storage-Server



- *Multipathing für die Clients über mehrere Storage-Server*
- *Bei nicht zu vielen Storage-Servern weiterhin Verzicht auf FC-Switches möglich*

Ausbau zum multifunktionalen Storage-Cluster

Storage-Server und Storage-Clients in einem Cluster

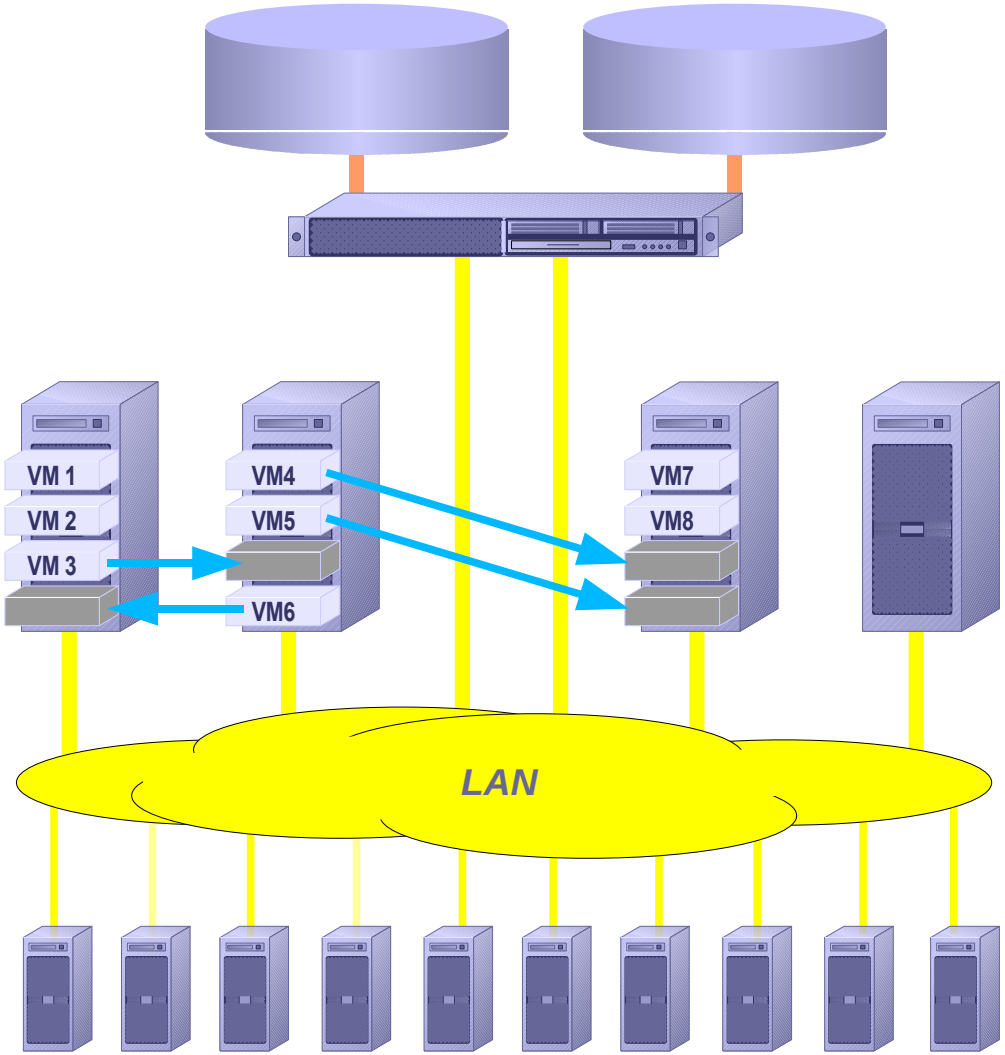


Mit Installation der Storage-Cluster-Pakete läßt sich ein RSIO-Client zum vollwertigen Cluster-Mitglied machen, d.h.:

- *Storage-/Device-Allokation vom Client aus*
- *Spiegeloperationen vom Client aus*
- *automatisches Access-Management*
- *identische Gerätesicht auf Storage-Servern und -Clients*
- *optionale HV für Applikationen (Applikationen auf jedem Clusternode lauffähig, gleich ob Storage-Server oder -Client)*

Storage Cluster, RSIO und KVM

Einheitliche Netzwerk-Infrastruktur, einfaches Handling



In Verbindung z.B. mit KVM erhalte ich einen VM-Cluster:

- *Storage, LAN, Live-Migration etc. über die gleiche Infrastruktur: Ethernet*
- *Hochverfügbarkeit*
- *kein Trennung von Storage- und Control-Verbindungen*
- *kein Split Brain*
- *zentrale Administration möglich*
- *Alle Funktionen der Speichervirtualisierung für VM-Abbilder nutzbar*
- *via IP Zugriff auf die gesamte Storage-Welt einschl. Storage-Virtualisierung aus der VM heraus installierbar / konfigurierbar*

RSIO - Zusammenfassung

Flexibel, modern und hochperformant

- *direkter (schlanker) Transport aller relevanten IO-Aufrufe (read, write, ioctl)*
- *integriert Verbindungsaufbau, Überwachung, Path-Multiplexing, Trunking*
- *fähig zu Selbstkonfiguration und Error Recovery*
- *vielfältige Einsatzszenarien:*
  - *einfache Server und Clients, ggf. mit Multipathing*
  - *Cluster von Storage-Servern (Targets)*
  - *Cluster von Storage Clients (Initiators)*
  - *integrierte Cluster von Servern und Clients*
  - *Storage Server Farms*
  - *Cloud-Konzepte*
  - *(K)VM-Cluster*
- *besondere Eignung für Kombination mit Speichervirtualisierung*
  - *eingängige Geräte-Namen*
  - *Partitionierung auf Clientseite entfällt*
  - *On-Demand-Allokation und Online-Rekonfiguration*
  - *Cross-Platform Shared Device Access möglich*
  - *ermöglicht Administration vom Client aus*

Mehr Informationen

www.osl.eu