

Tutorial: OSL RSIO

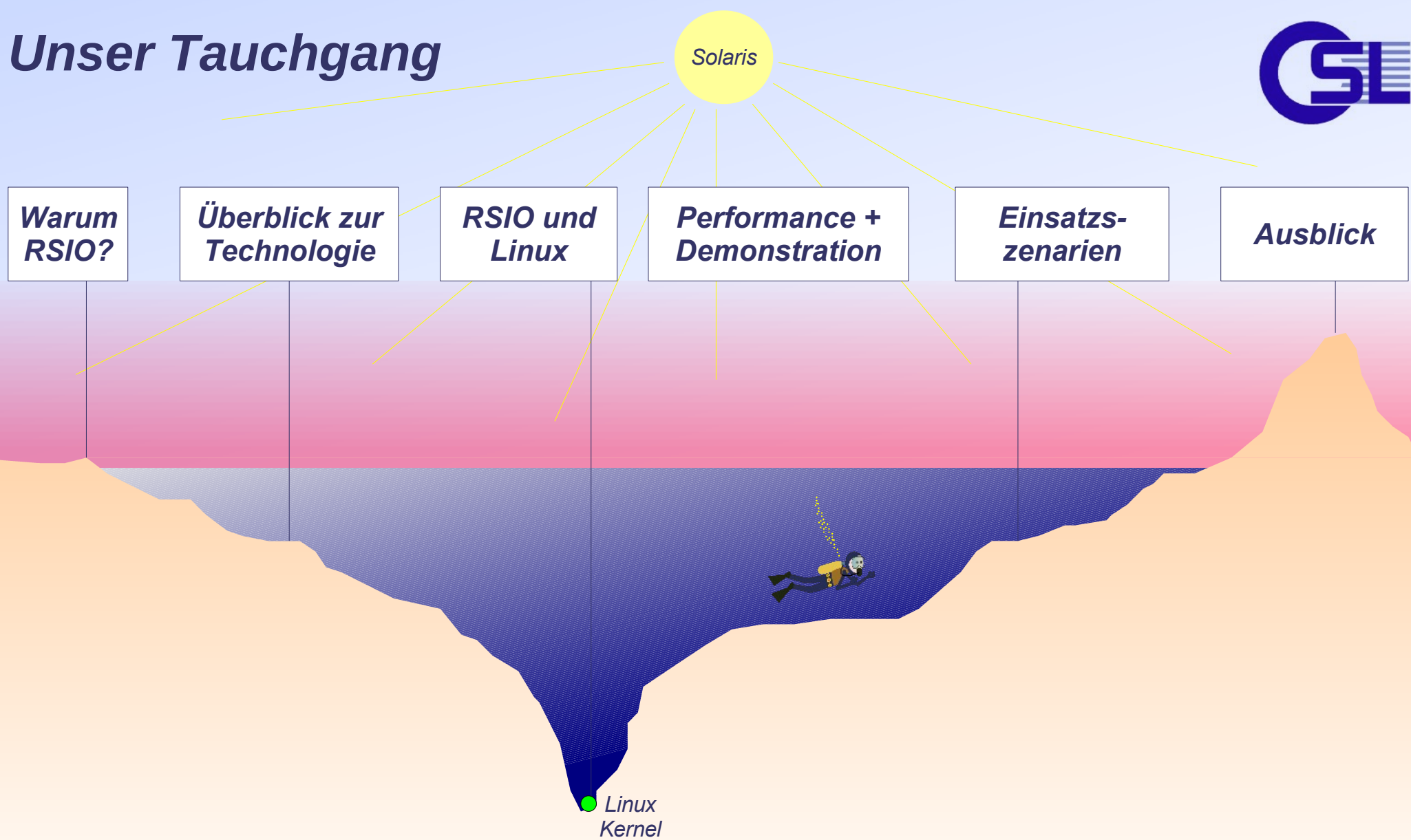
Storage Networking der nächsten Generation

iX Day rund um Solaris 2010
Stuttgart – September 2010

Bert Miemietz
Christian Schmidt

OSL Gesellschaft für
offene Systemlösungen mbH

Unser Tauchgang



OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Warum RSIO?

Klassisch: Spezialisierte Netzwerke

Verschiedene Netzwerke für verschiedene Anwendungen



Fibre Channel

- *Spezialprotokoll*
- *Block I/O*
- *Kanaleigenschaften*
- *Niedrige Latenz*
- *Hoher Durchsatz*
- *Niedrige CPU-
Belastung*

Ethernet / IP

- *Universalprotokoll*
 - *Primär von Applikationen getrieben*
 - *Zweck: Kommunikation*
 - *Client/Server-Applikationen*
 - *Implementierung wesentlich im OS
-> höhere CPU-Belastung*
 - *Seit langem auch für Storage
genutzt (NFS, Backup ...)*
- *NFS, SMB ...*
 - *Backup*
 - *IP over FC*
 - *FC over IP*

Klassisch: Spezialisierte Netzwerke

Verschiedene Netzwerke für verschiedene Anwendungen



Fibre Channel

- Spezialprotokoll

Aus Sicht des Anwenders sind sich beide

- Kanaleigenschaften

- Niedrige Latenz

- Hoher Durchsatz

- Niedrige CPU-
Belastung

Ethernet / IP

- Universalprotokoll

- Zweck: Kommunikation

- Client/Server-Applikationen

- Implementierung wesentlich im OS
> höhere CPU-Lastung

- Seit langem auch für Storage
genutzt (NFS, Backup ...)

**Netzwerktechnologien heute im Verhalten
oftmals ähnlicher als erwartet**

Warum Storage über Ethernet?

Anforderungen und Möglichkeiten



- **Anforderungen und Erwartungen**

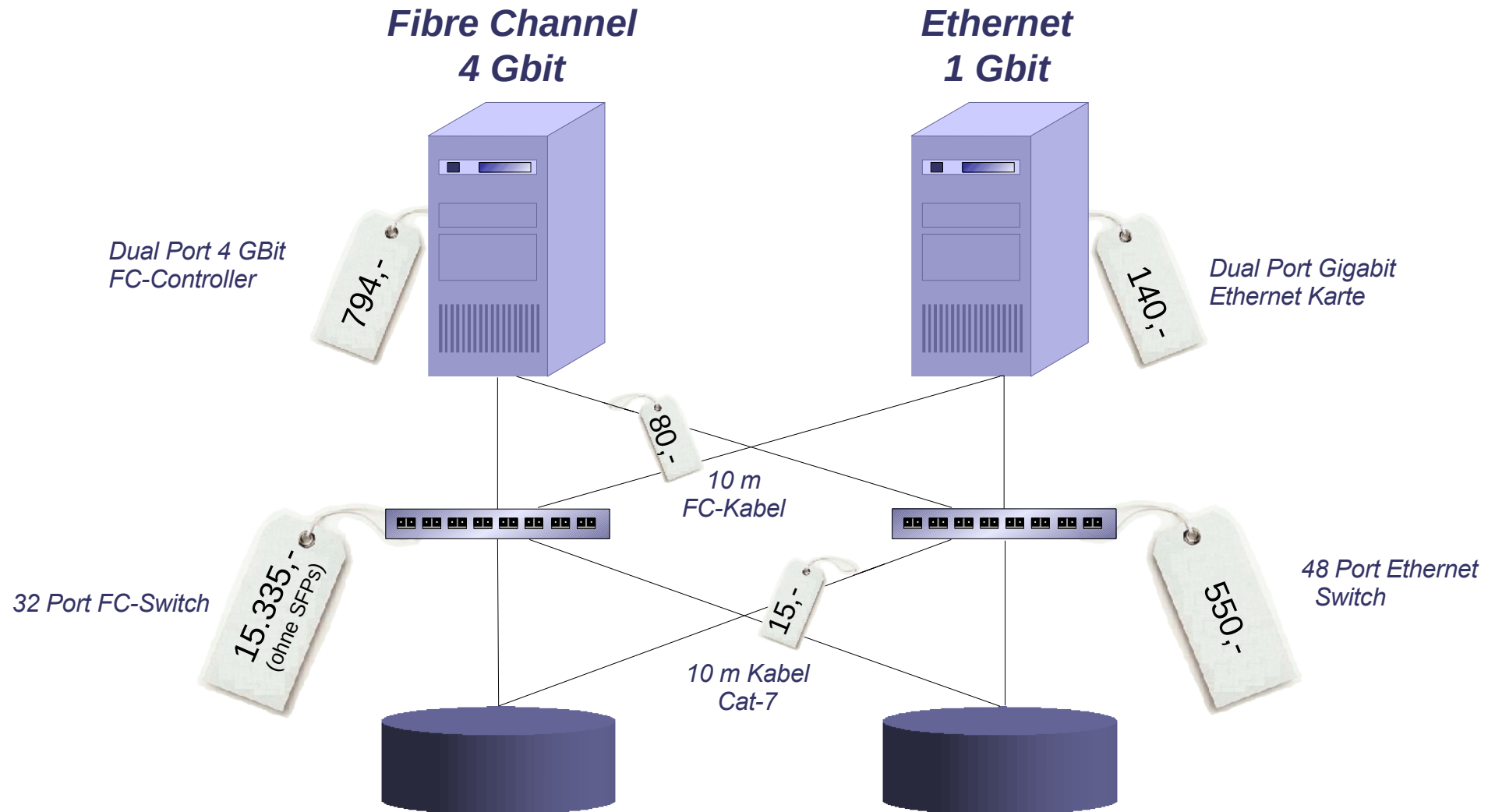
- *Erfordernisse der Anwendungen und Protokolle (Kommunikation, Filesharing etc.)*
- *Preisliche Motivationen*
- *Einheitliche Infrastruktur, weniger Ports ?*
- *Einfachheit, Flexibilität ?*
- *Virtualisierungstechnologien, Verfügbarkeit von Treibern*
- *Zusatzfunktionen (Konvertierungen, Filesystemsnapshots ...)*

- **Möglichkeiten**

- *Gigabit-LAN heute vergleichsweise preiswert*
- *Gigabit-LAN heute in vernünftiger Relation zur Geschwindigkeit einer Festplatte bzw. eines RAID-Systems*
- *Gigabit-LAN heute in günstiger Relation zu Durchsatz-Anforderungen der Applikationen*
- *Mehrere Gigabit-Ports je Server*
- *Ethernet ist eigentlich (fast) kein Ethernet mehr -> Switching-Technologie*
- *RAID-Systeme / Filer sprechen direkt die erforderlichen Protokolle*
- *Neue Performance-Erwartungen an 10GBit-Ethernet*

Was ist mit den Kostenvorteilen?

Ein Vergleich mit Fibre Channel (Stand 10/2009)



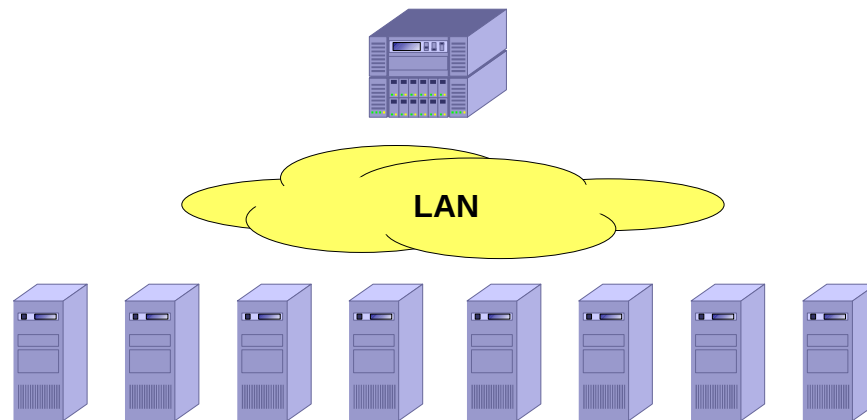
OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Storage über Ethernet heute: NFS, SMB, CIFS

NA(F)S präsentiert sich mit handfesten Vorteilen



- *Spezialisierung auf Fileservices, dafür relativ einfache Handhabung*
- *kann komplexe RAID-Funktionen verbergen*
- *dateisystemtypische Funktionalitäten wie Snapshots und weitere Sonderfunktionen*
- *ermöglichen Filesharing*
- *weite Verbreitung und Unterstützung der wichtigsten Protokolle*
- *im Rahmen des heute Vorstellbaren Möglichkeiten nahezu ausgereizt*



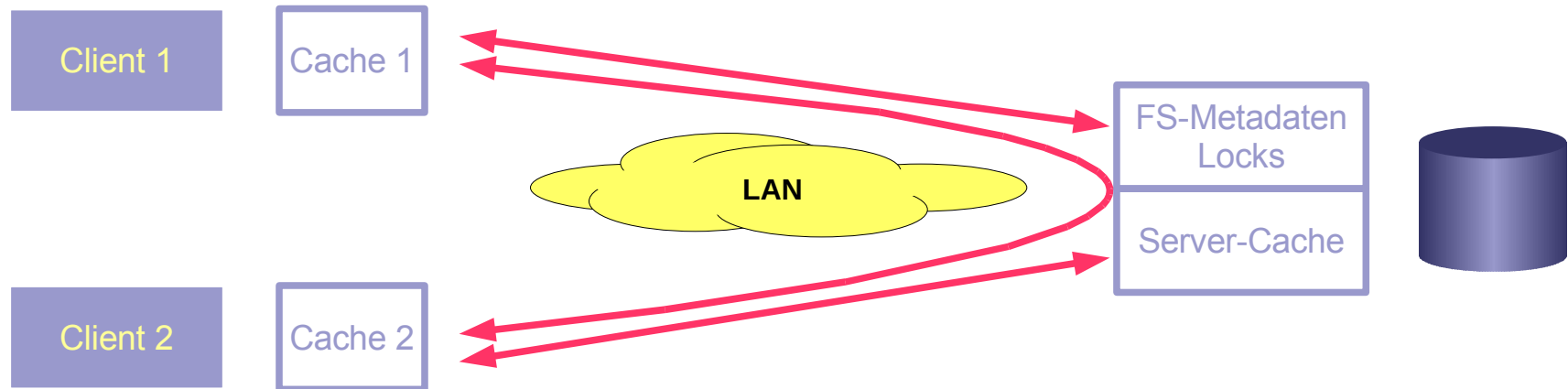
OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Die Kehrseite des NAFS-Ansatzes

Es gibt auch prinzipbedingte Nachteile



- *aufwendige Integration mit Server-OS (Zugriffskontrolle, User-Management)*
- *Cache- und Cohärenzproblematik, schwierige Nutzung der Client-Ressourcen*
- *nicht trivial: Skalierbarkeit, Hochverfügbarkeit, Multipathing*
- *feste Bindung an File-Access-Semantik*
- *mit zunehmender Funktionalität auch Zunahme von Komplexität und ggf. Inkompatibilitäten*

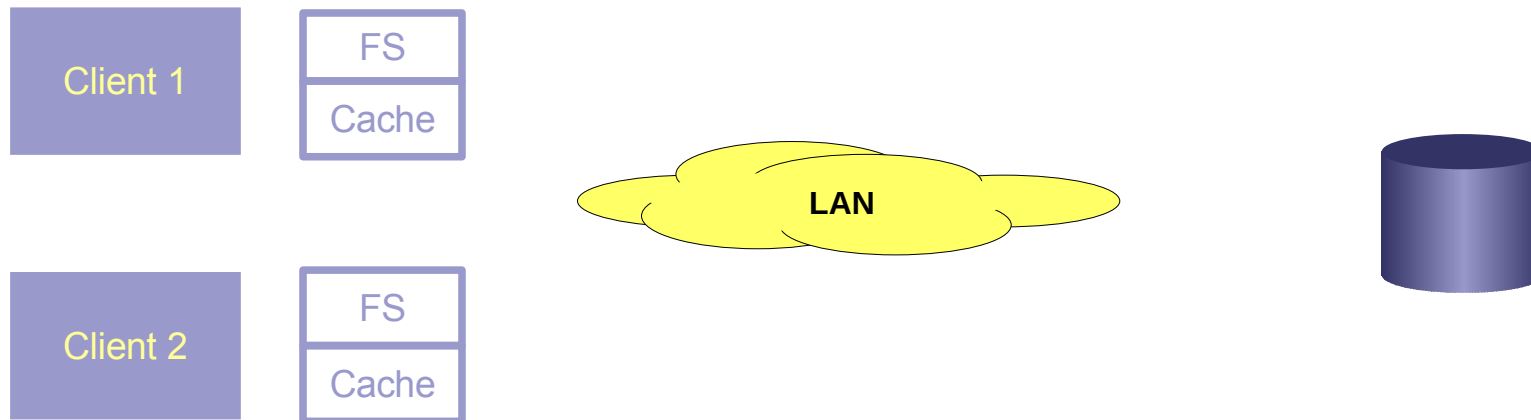


Starke Argumente für Block-I/O im RZ

Jenseits von Filesharing überwiegen die Vorteile



- *Volle Kontrolle des Client-OS über das Storage-Device*
- *Nutzbar für beliebige Filesysteme*
- *Keine Kopplung an Server-OS (Isolation, privates Identity Management)*
- *Nur Übertragung von I/O, nicht von Cache-Inhalten*
- *Cache liegt beim Client -> schnellster Zugriff, Client-Caches summieren sich auf*
- *Einfache Administration, schlankes Protokoll, hohe Geschwindigkeit*



Block-I/O über Ethernet/IP - Performance

Performance hat viele Aspekte



- **Latenz / Service Time**

- *Relevant für einzelnen I/O und single threaded I/O*

- **Maximaler Durchsatz**

- *Relevant für multithreaded I/O*

- **Skalierbarkeit / Parallelisierbarkeit**

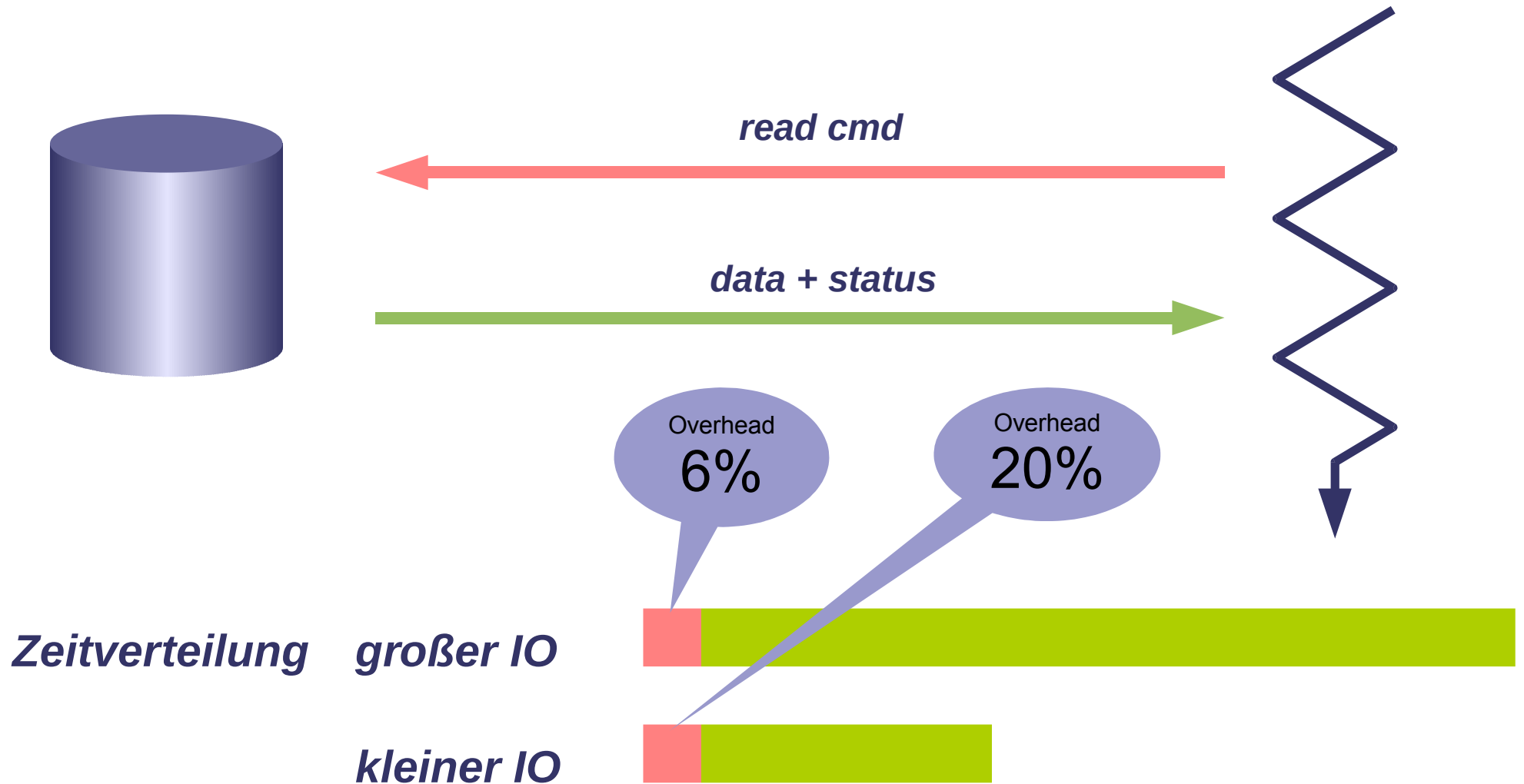
- *Thema für Multipathing*

- **IO-Größe**

- *Relevant für Nutzdatenanteil am Gesamtdurchsatz*

Bedeutung von Latenzen und IO-Größe

Heutige Paradigmen setzen Grenzen



Storage over Ethernet: Akzeptable Latenzen?

Betrachtungen zur Performance



Zeit zur Übertragung von 32 Byte an 1 GBit-Lanboard

0,3

Zeit zur Übertragung von 512 Byte an 1 GBit-Lanboard

5

Zeit zur Übertragung von 8kByte an 1 GBit-Lanboard

80

memcpy 8k auf 600MHz-System

2

Threadwechsel per CV auf 600MHz-System

5

32Bit-Typkonvertierung auf 600MHz-System

0,02

100µ s

Storage over Ethernet: Akzeptable Latenzen?

Betrachtungen zur Performance



Zeit zur Übertragung von 32 Byte an 1 GBit-Lanboard

Zeit zur Übertragung von 512 Byte an 1 GBit-Lanboard

Es geht also:

Zeit zur Übertragung von 8kByte an 1 GBit-Lanboard

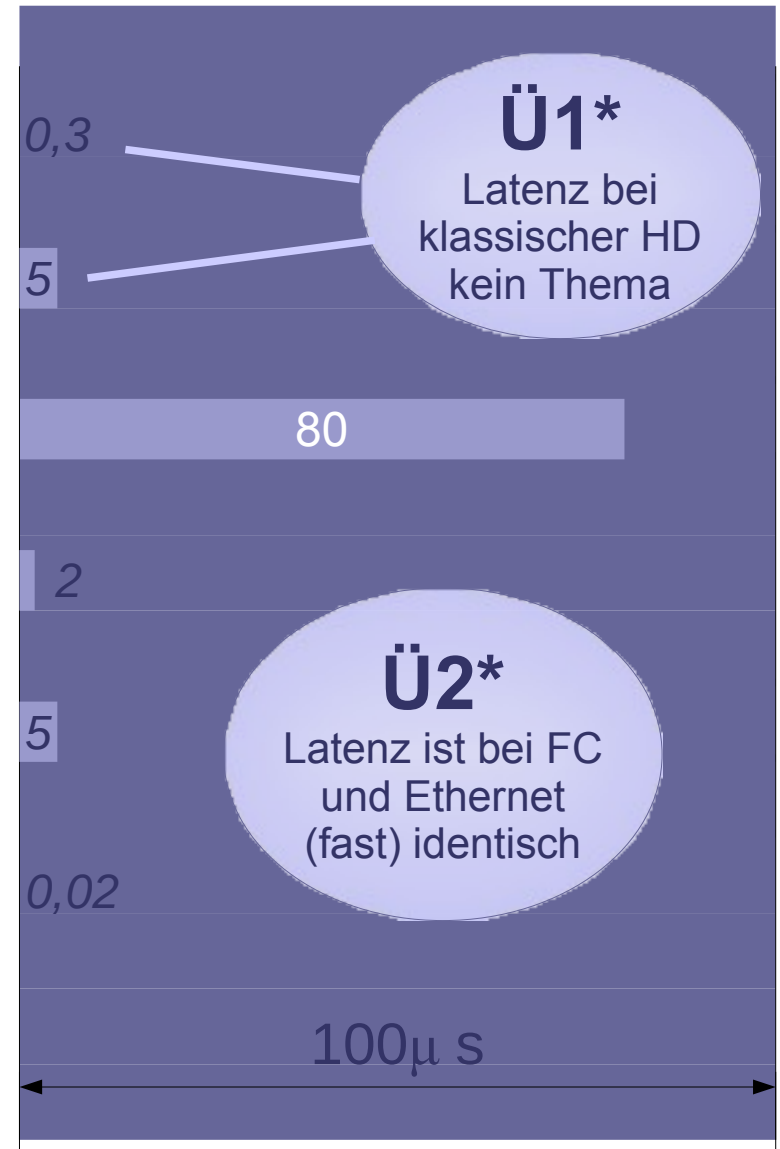
1. Weniger um die Hardware.

2. Entscheidend um das Protokoll.

3. Entscheidend um die Systemsoftware.

4. Nicht unerheblich um die Applikation.

32Bit-Typkonvertierung auf 600MHz-System



Block-I/O mit iSCSI

Bekanntes Protokoll über neues Medium



- *Low-Level-Protokoll auf IP umgesetzt*
- *Plattformunabhängig auf Server-Seite (Target)*
- *Starke Bindung an TCP, Offload-Engines auf Initiatorseite dennoch selten*
- *Tiefer Stack – nicht unerheblicher CPU-Bedarf*
- *Zahlreiche SCSI-Funktionen, aus Sicht der Anwendungen aber dennoch Verlust an Funktionalität*
-> Storage-Management meist über andere Protokolle
- *Vernetzte, geclusterte Speichersysteme ohne speziellen Support*
- *Weitere Schwierigkeiten*
 - *Multipathing*
 - *Clustering / Parallelisierung*
 - *Nomenklatur*
 - *Target Portal Groups*

Block-IO: iSCSI (Solaris) auf der Kommandozeile

Die Target-Seite (Server)



```
[root@big-5] iscsitadm list target
Target: target-0
  iSCSI Name: iqn.1986-03.com.sun:02:06df3360-bb85-ee33-bf59-f2d03474f708.target-0
  Connections: 2
[root@big-5] iscsitadm list target -v
Target: target-0
  iSCSI Name: iqn.1986-03.com.sun:02:06df3360-bb85-ee33-bf59-f2d03474f708.target-0
  Connections: 2
    Initiator:
      iSCSI Name: iqn.1986-03.com.sun:01:baladef3ffff.4a431482
      Alias: unknown
    Initiator:
      iSCSI Name: iqn.1986-03.com.sun:01:baladef3ffff.4a431482
      Alias: unknown
  ACL list:
    Initiator: big-6
  TPGT list:
    TPGT: 1
  LUN information:
    LUN: 0
      GUID: 600144f04aae26ed000000ae488148e00
      VID: SUN
      PID: SOLARIS
      Type: disk
      Size: 1.0G
      Backing store: /dev/av0/target
      Status: online
```

Block-IO: iSCSI (Solaris) auf der Kommandozeile

Die Initiator-Seite (Client)



So meldet sich die Platte ohne MPXIO in “format”

```
29. c3t227d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>  
    /iscsi/disk@0000iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59-  
f2d03474f708.target-00001,0  
31. c3t229d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>  
    /iscsi/disk@0001iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59-  
f2d03474f708.target-00001,0
```

und so mit MPXIO

```
28. c4t600144F04AAE26ED000000AE488148E00d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>  
    /scsi_vhci/disk@g600144f04aae26ed000000ae488148e00  
29. c4t600144F04AAE26EE000000AE488148E00d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>  
    /scsi_vhci/disk@g600144f04aae26ee000000ae488148e00
```

weiterführende Komplikationen:

- *MPXIO, Routing, Load-Balancing*
- *Target Portal Groups*
- *Raw-Device Access und IO-Caching*
- *Failover von Targets zwischen verschiedenen Servern*

```
[root@big-6] rsconfig -q
```

```
000 osl
```

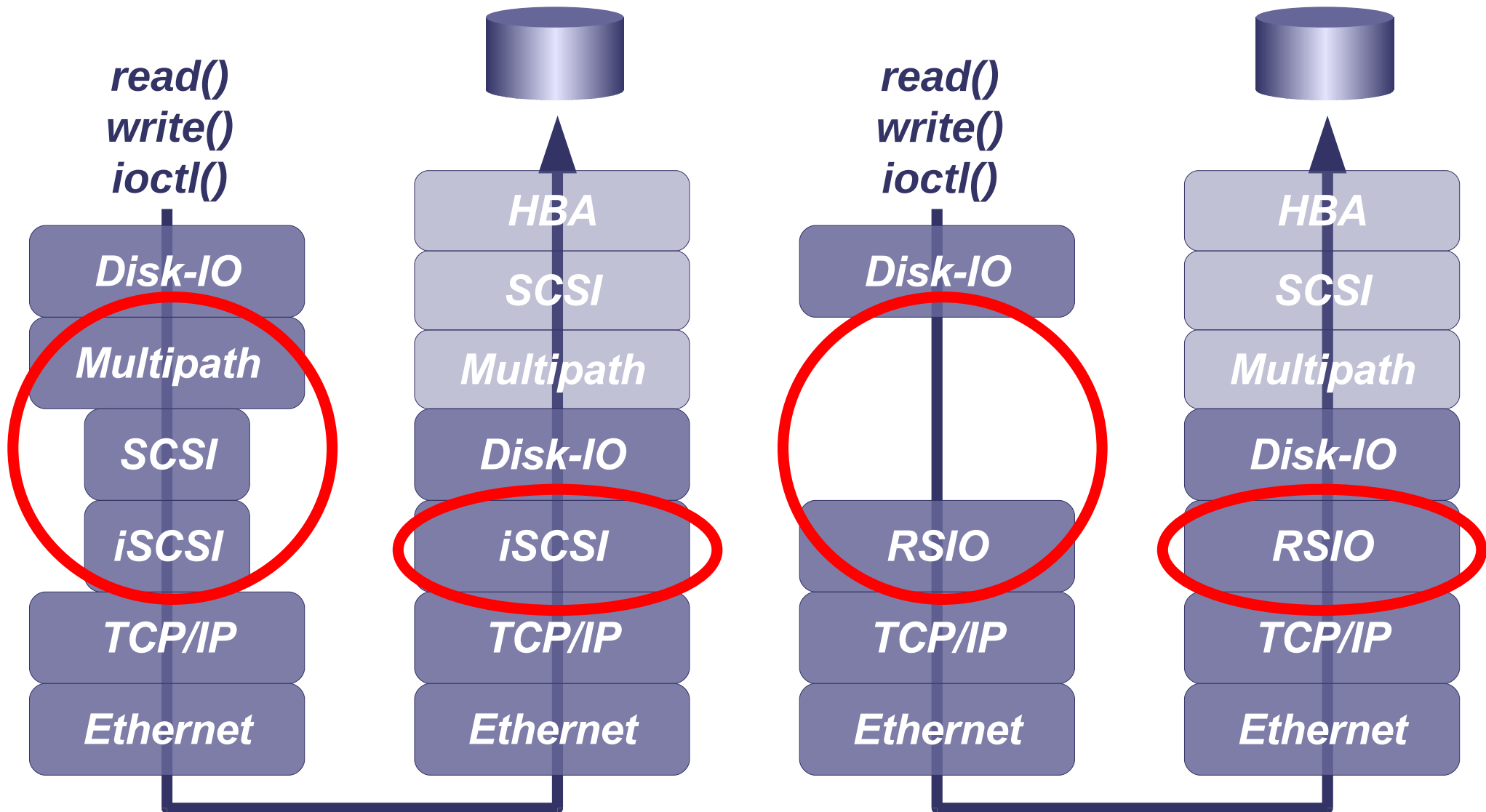
```
    clt: big-6
```

```
    srv: 000 big-5
```

0	tvoll	disk	2097152	blocks	of	512	bytes
0	shadow	disk	2097152	blocks	of	512	bytes
0	z1_root	disk	10485760	blocks	of	512	bytes
0	sparse	disk	10485760	blocks	of	512	bytes
0	whole	disk	41943040	blocks	of	512	bytes
0	iscsit_cfg	disk	20480	blocks	of	512	bytes
0	target	disk	2097152	blocks	of	512	bytes
0	tconf	disk	20480	blocks	of	512	bytes
0	p07	disk	585920023	blocks	of	512	bytes
0	p08	disk	585920023	blocks	of	512	bytes
1	b07	disk	976545023	blocks	of	512	bytes
1	b08	disk	976545023	blocks	of	512	bytes
2	b09	disk	976545023	blocks	of	512	bytes
2	b10	disk	976545023	blocks	of	512	bytes
3	b11	disk	976545023	blocks	of	512	bytes
3	b12	disk	976545023	blocks	of	512	bytes

Block-I/O über Ethernet/IP – ohne SCSI

Ein anderer Ansatz



RSIO - Remote Storage IO

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



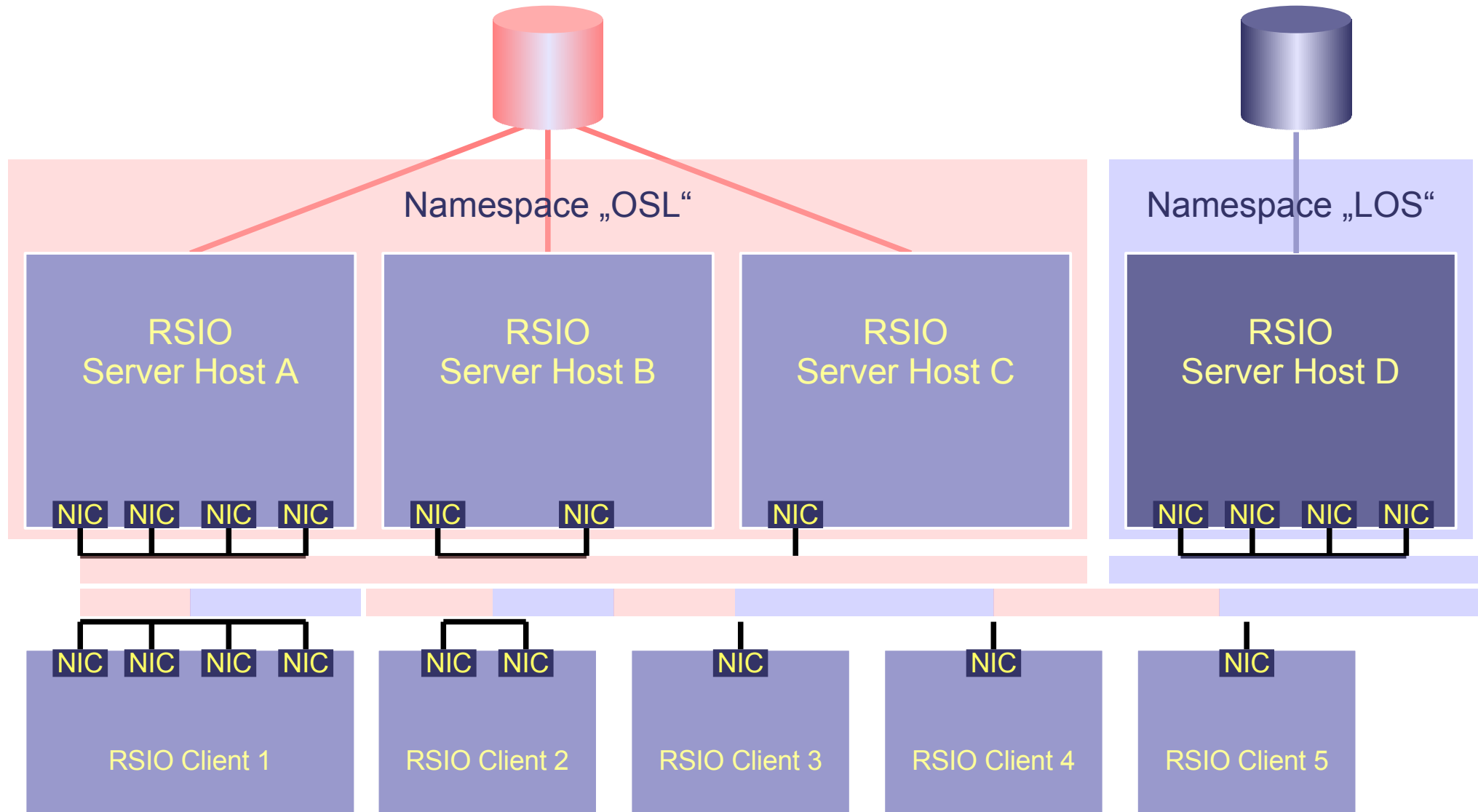
- *eigenes, von OSL entwickeltes Protokoll*
- *natürliche Erweiterung des Speichervirtualisierung auf LAN (Ethernet)*
- *voller Clustersupport möglich (Client und Server), vorrangig natürlich:*
- *Einbindung in OSL-Clustertechnologie*
- *alle relevanten IO-Aufrufe (read, write, ioctl ...) abbildbar*
- *hochportable Implementierung*
- *internes Layout berücksichtigt moderne CPU- und Serverkonzeptionen*
- *guter Durchsatz / gute Verfügbarkeit mit heutiger preiswerter Standardtechnik*
- *integrierte und bequeme Administration vom Host aus*
- *Erweiterbarkeit, Raum für intelligente IO-Lösungen*

OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Überblick zur Implementierung

RSIO - Architektur im RZ

Klar gegliedertes und flexibles administratives Konzept



OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Parameter der RSIO-Architektur

Flexible Client-Server-Implementierung



- *Ein Namespace definiert Server (und Clients) mit Zugriff auf dieselben Storage-Ressourcen*
- *Auf einem Serverhost können (nahezu) beliebig viele Server(prozesse) laufen*
- *Jeder Serverhost kann (nahezu) beliebig viele Clients bedienen*
- *jeder Client unterstützt den Zugriff auf bis zu 256 Server*
- *jede Maschine (Client und Server) unterstützt bis zu 8 Interfaces*
- *jeder Client hat simultan Zugriff auf verschiedene Namespaces*
- *Auto-Explorer*
 - *Ermitteln verfügbarer Verbindungen*
 - *Ermitteln der Schnittstelleneigenschaften*
 - *Test der Parameter auf der Übertragungsstrecke*

Und was ist mit Linux?

Was der technisch Interessierte zu diesem Thema wissen muß



- **GNU:**

- **gcc** sehr leistungsfähiges Werkzeug
- auf vielen Plattformen verfügbar
- Unterstützung aller relevanten Programmier Techniken und **Standards**

- **LINUX:**

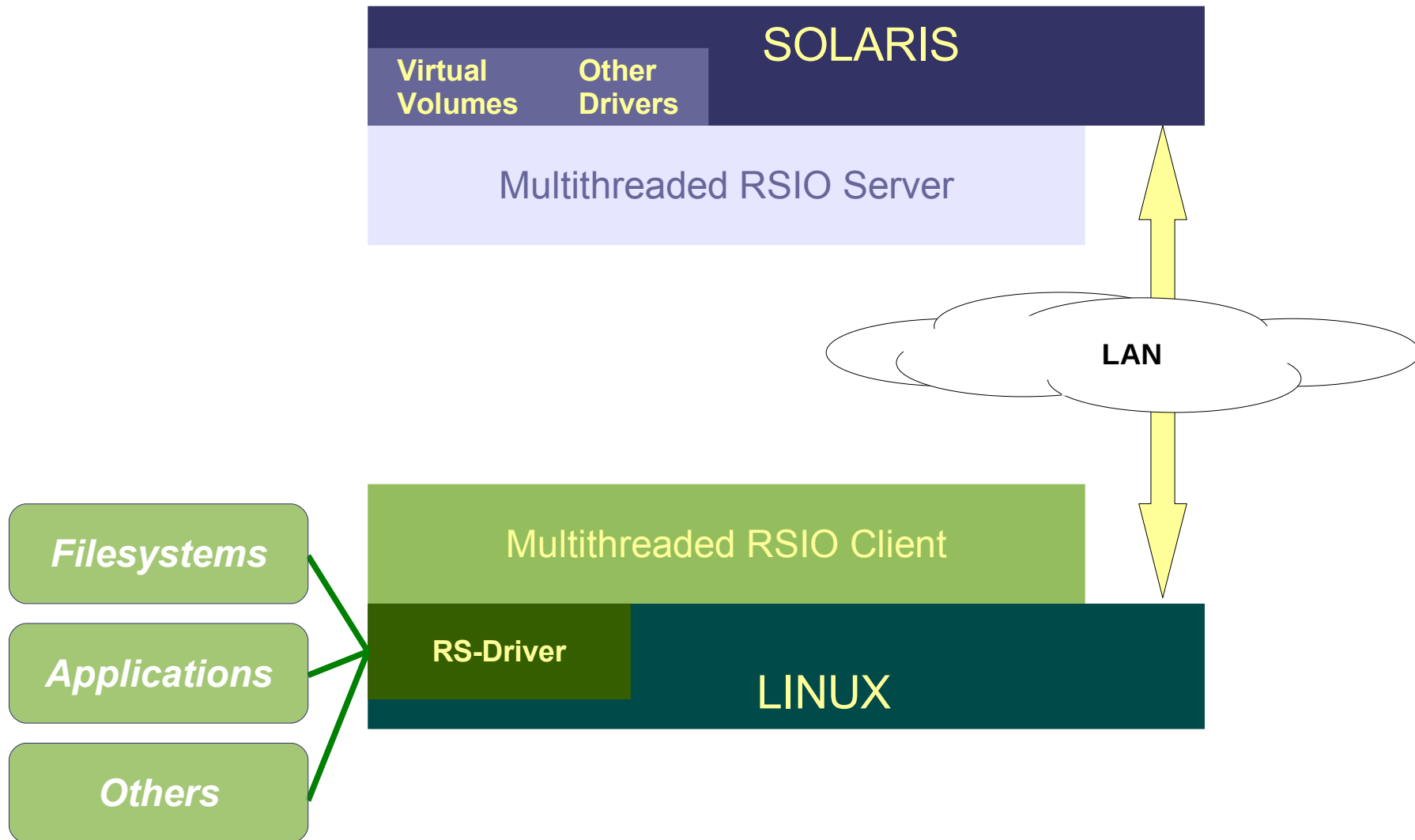
- leistungsfähiges OS
- verfügbar auf vielen Hardware-Plattformen
- aber: große Vielfalt (ubuntu, fedora, openSUSE, **debian**, Mandriva, LinuxMint, PCLinuxOS, **slackware**, gentoo linux, CentOS, **Red Hat**, SLES, SLED)
- Gemeinsamkeit: Kernel

- **Kernel**

- keine langfristig stabilen Schnittstellen -> Dokumentation problematisch
- Portabilität? (POSIX-Namespace-Pollution)
- Kernelprogrammierung naturgemäß proprietär (Kernel Build System, Macros etc.)
- Konzepte deutlich von anderen Systemen abgesetzt (z. B. Multithreading)
- Wartbarkeit?

Wie lösen wir das Problem?

Ein einheitliches Modell, nicht nur anwendbar für Linux und Solaris



Wie lösen wir das Problem?

Ein einheitliches Modell, nicht nur anwendbar für Linux und Solaris



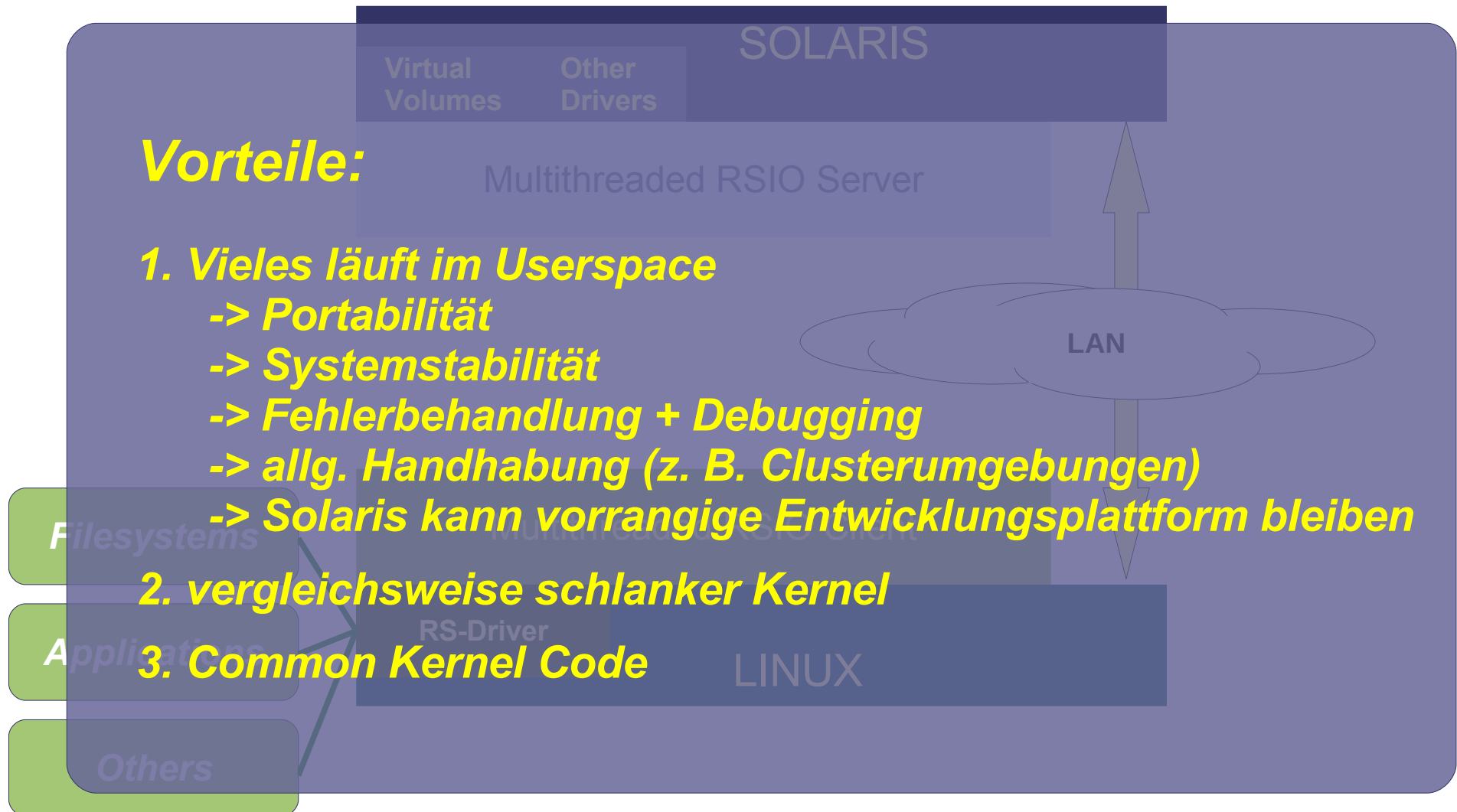
Vorteile:

1. Vieles läuft im Userspace

- > Portabilität
- > Systemstabilität
- > Fehlerbehandlung + Debugging
- > allg. Handhabung (z. B. Clusterumgebungen)
- > Solaris kann vorrangige Entwicklungsplattform bleiben

2. vergleichsweise schlanker Kernel

3. Common Kernel Code

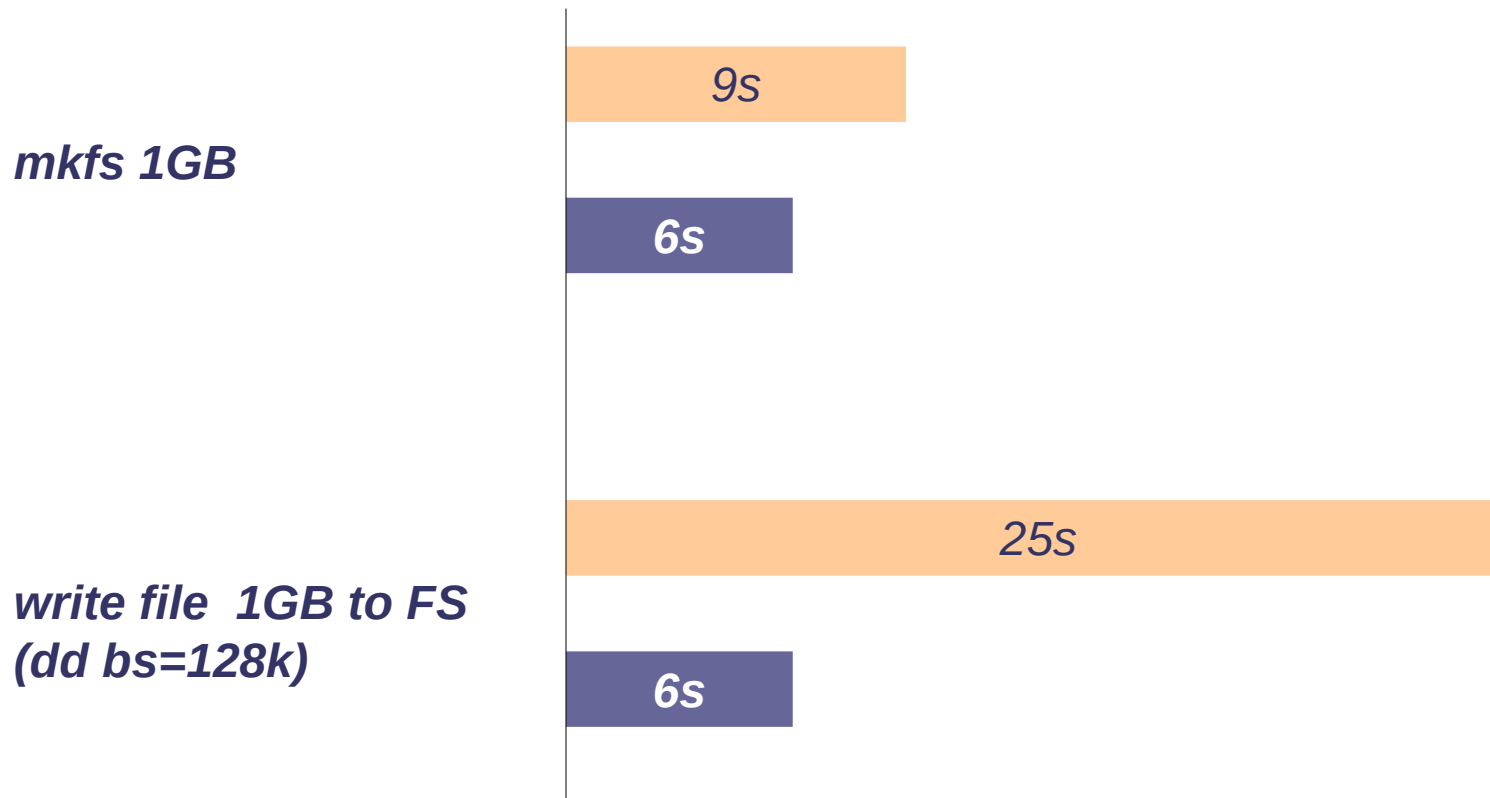


- **Entwicklungsplattform: SUSE Linux Enterprise Server 11 (x86_64)**
 - Kernel: 2.6.27.19-5-default
- **Ladbares Kernel-Modul (analog Solaris: “rs”)**
 - Modul heißt analog zu Solaris “rs.ko”
 - Vorabinfo: `modinfo`
 - Laden: `insmod` (+ ggf. Optionen)
 - Anzeigen: `lsmod`
 - Entladen: `rmmod`
- **Besonderheiten**
 - Logging und Debug-System analog zu Solaris implementiert
 - sysfs-Integration, Management der Disk-Device-Nodes via `rsconfig`
 - Reservierung Major-Nummer für Block- und Char-Devices (Default: local 246)
 - Effektiv im Moment für Anwender nur Nutzung der Block-Devices (Disk)
 - modprobe-Konfiguration noch nicht enthalten

Performanceparameter

Performance in der Praxis

Wer viel mißt, mißt ... Vergleiche zu iSCSI / Filesysteme



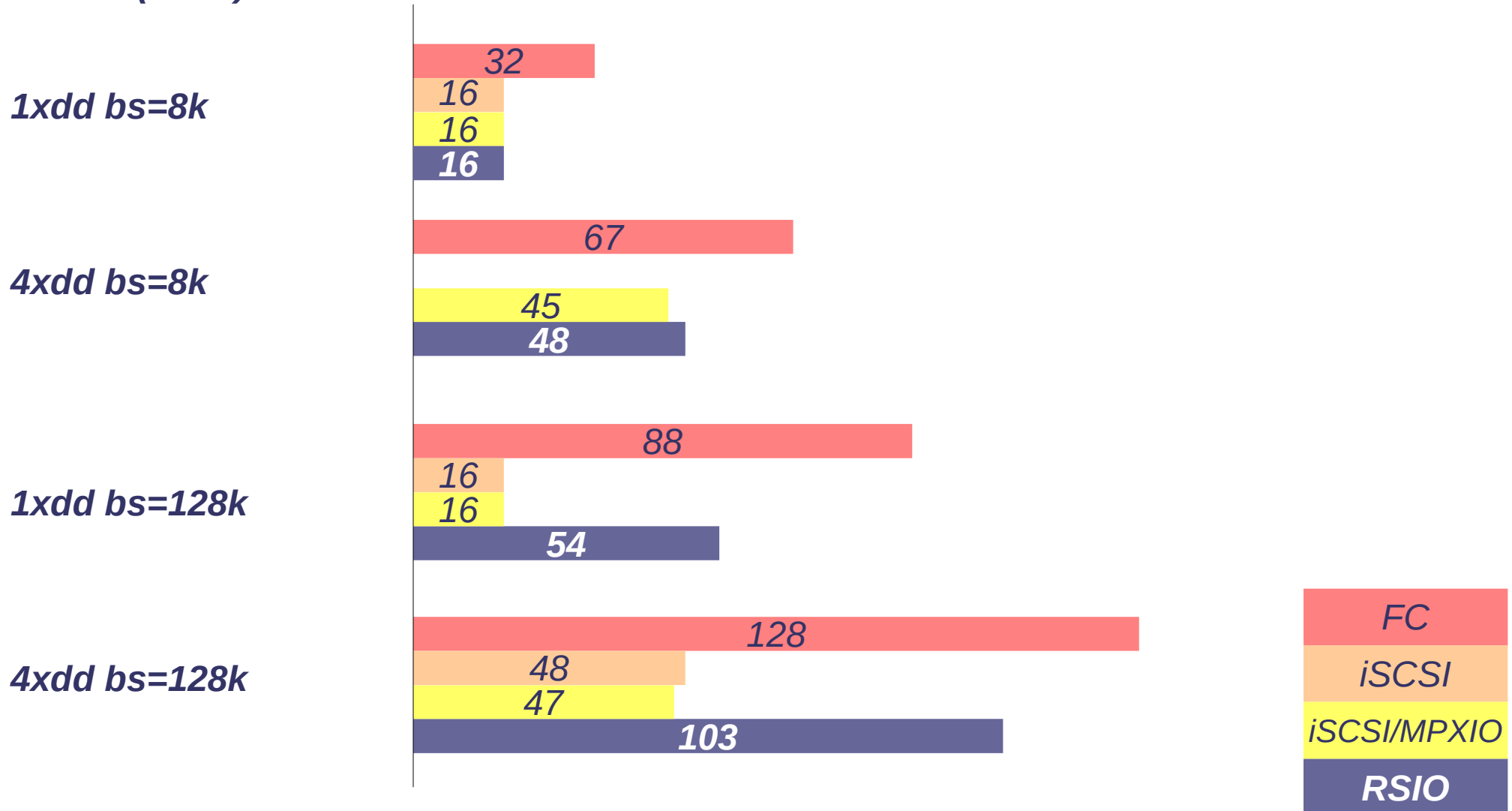
OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

Performance in der Praxis

Wer viel mißt, mißt ... Vergleiche zu FC und iSCSI / Character Device (raw IO)



WRITE (MB/s)



Aber ist das nicht zu langsam?

Performance bei Cache-Reads -> der Netzwerkprotokoll-Test



Server-Performance bei Cache Read / 8k

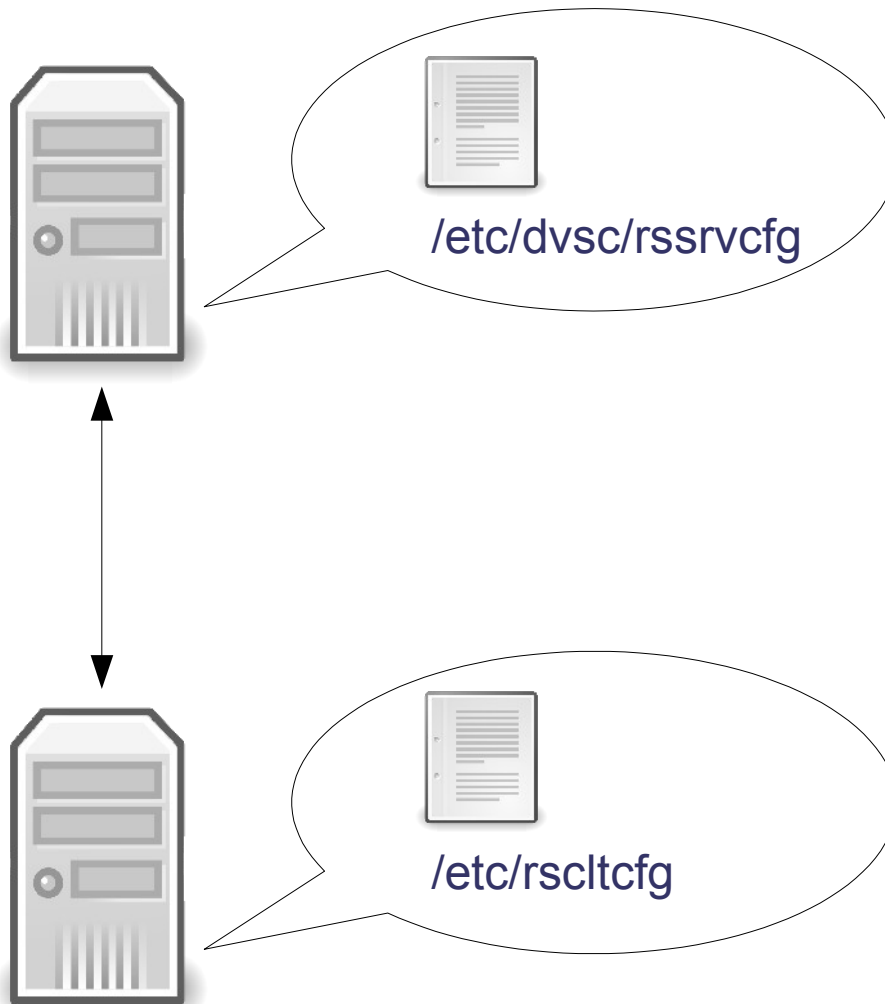
iSCSI	10 Clients	100 Threads	7,6 Cores	31.000 IOPS
iSCSI / comstar	10 Clients	100 Threads	10,0 Cores	85.000 IOPS
RSIO	4 Clients	64 Threads	5,6 Cores	98.000 IOPS
RSIO	4 Clients	128 Threads	6,3 Cores	102.000 IOPS

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

Praxis

- ***Konfiguration & Systemintegration***
- ***Abfragen***
- ***Zugriffe***



• **RSIO Server Konfiguration**

- *Namespace*
- *Server ID*
- *Interfaces*
- *Clients + Clientkeys*

• **RSIO Client Konfiguration**

- *Namespace + Clientidentifikation*
- *Interfaces*
- *Server*

- **RSIO Server: *rssrv***
 - Benötigt vv-Treiber (OSL SC Base)
 - Derzeit nur unter Solaris
 - Stellt alle Application Volumes des RSIO-Servers den Clients zur Verfügung
- **Ausgabe der Server Logfiles: *rsslogcat [-f]***
 - Loggt alle RSIO Ereignisse des Servers: Start & Stop, Aushandlung der Clientverbindung, Pfadfehler, Beendigung von Clientverbindungen
 - Selbes Verhalten wie bei der OSL-SC Logfacility

```
big-9@rsio 2010_09_14-06:27:16 (GMT) INFO (osl:sles1/1-ctl): new child server started
big-9@rsio 2010_09_14-06:27:16 (GMT) INFO (osl:sles1/1-ctl): client is: Linux
```

```
# cat /etc/dvsc/rssrvcfg
```

```
[namespace osl]
namespaceid  = 200
protocol     = tcp
port        = 5000
```

Namespace Konfiguration

Name, ID und Standardeinstellungen

```
[server big-9]
serverid     = 1
```

Server Name und ID

```
[interface if1]
address      = 192.168.45.10

[interface if2]
address      = 192.168.46.10
```

Interface Sektionen

Ein Eintrag für jedes zu nutzende Interface.
Port und Protokoll kann auf für jedes Interface
individuell gesetzt werden.

```
[clients]
big-2  0xbig2
venus  0xvenus
big-6  0xbig6
```

Client Sektion

Ein Eintrag für jeden Client (Name und Schlüssel)

- **Client Dämon: *rsiod***
 - *Läuft im Userspace*
 - *Kommuniziert mit allen konfigurierten und erreichbaren Servern*
 - *Schnittstelle zwischen dem rs-Gerätetreiber und dem Netzwerk*
- **Konfigurationsprogramm: *rsconfig***
 - *Attach und Detach von rs-Devices*
 - *Anzeigen der Geräte (lokale Sicht, Namespace Sicht)*
 - *Anzeigen von Multipfadinformationen*

```
sles1:~ # rsconfig -q
000 osl
    clt: sles1
    srv: 000 big-9
          0  RSIO_oracle      disk      20971520 blocks of  512 bytes
          0  S10_oracle      disk      20971520 blocks of  512 bytes

sles1:~ # rsconfig -lv
osl:RSIO_oracle@0      20971520 blocks,    1 server(s)
osl:S10_oracle@0      20971520 blocks,    1 server(s)

sles1:~ # rsconfig -m status
0 if0 IP(TCP) 192.168.45.20/5000
    rep: 000 IP(TCP) 192.168.45.10/5000 connected tx: ok rx: ok
1 if1 IP(TCP) 192.168.46.20/5000
    rep: 001 IP(TCP) 192.168.46.10/5000 connected tx: ok rx: ok
```

- **Attach der Volumes**

- *Alle Volumes werden per default unter /dev/av[0-3] angelegt*
 - *Selber Pfad wie auf dem RSIO Server*
 - *Applikationen können sowohl auf dem Server, als auch auf dem Client laufen -> keine Anpassungen in der ARD notwendig*
- *RSIO Clients können in mehreren Namespaces registriert sein*
 - *Volumes können nicht mehr unter dem selben Pfad eingehangen werden wie auf dem Server, da Namen doppelt vorkommen können*
 - *RSIO Volumes können beim Attach relociert werden (**rsconfig -ar**)*

```
root@big-2# rsconfig -ar
root@big-2# rsconfig -lvv
osl:oracle@0                                31457280 blocks,    1 server(s)
  c: /dev/rs/rdisk/osl:oracle@0
  b: /dev/rs/dsk/osl:oracle@0

osl:RSIO_oracle@0                            20971520 blocks,    1 server(s)
  c: /dev/rs/rdisk/osl:RSIO_oracle@0
  b: /dev/rs/dsk/osl:RSIO_oracle@0
```


Der RS-Treiber stellt die Geräte bereit und ist die Schnittstelle zu den Userspace Anwendungen

- ***Userspace Programme von Solaris und Linux sind fast identisch – die Unterschiede werden beim Treiber sichtbar***
 - *keine Charakterdevices unter Linux*
 - *begrenzter Vorrat an Minor-Nummern für die Geräte*
 - *aggressives Caching unter Linux*
 - *Modulhandling*
 - *Treiberkonfiguration*

- *keine Charakterdevices unter Linux*

Linux:

```
sles1:~ # rsconfig -lvv
osl:SLES11_oracle@0          20971520 blocks,    1 server(s)
  c:
  b: /dev/av0/SLES11_oracle
```

Solaris:

```
root@big-2# rsconfig -lvv
osl:SLES11_oracle@0          20971520 blocks,    1 server(s)
  c: /dev/av0/rSLES11_oracle
  b: /dev/av0/SLES11_oracle
```

- **Modulhandling und Treiber Konfiguration**

Linux:

- *Laden und Entladen: insmod, rmmod*
- *Module anzeigen: lsmod*
- *Konfigurieren: modprobe.conf*
- *Automatisches Laden: modprobe, /etc/sysconfig/kernel*

Solaris:

- *Laden und Entladen: modload, modunload*
- *Module anzeigen: modinfo*
- *Konfigurieren: /kernel/drv/rs.conf*
- *Automatisches Laden: beim Zugriff auf das Device (/etc/name_to_major)*

- **Der Client pflegt ein eigenes Logfile**

- Ausgabe mit `rsclogcat [-f]`
- Selbes Verhalten wie die Storage Cluster Logfacility
- Logging von Verbindungsauf- und abbau und Netzwerkereignisse

```
big-2@rsio 2010_09_14-15:02:32 (GMT) INFO (6-if0-tx0): started transmit  
for rep 192.168.45.10/500
```

- Das Kernelmodul loggt in syslog (IO-Events)

```
Sep 14 16:50:25 big-2 rs: [ID 567783 kern.notice] NOTICE: generic -  
rs_txwait: interrupted
```

```
root@big-2# cat /etc/rscltcfg
```

```
[defaults]
protocol    = tcp
port        = 5000
```

Default-Werte für Protokoll und Port.
Abweichende Werte können in der Namespace und Interface Sektion angegeben werden

```
[interface if0]
address     = 192.168.45.20

[interface if1]
address     = 192.168.46.20
```

Interface Sektion

Alle Clientinterfaces die von rsiod genutzt werden sollen.

```
[namespace osl]
namespaceid = 200
nodename    = big-2
nodekey     = 0xbig-2
```

Namespace Sektion

Für jeden Namespace einen eigenen Abschnitt mit der Clientidentifikation und den Namespace Angaben

```
[server big-9-1]
address     = 192.168.45.10

[server big-9-2]
address     = 192.168.46.10
```

Server Interfaces

Pro Serverinterface ein Abschnitt. Server gehören zum vorhergehenden Namespace.

Ende des Praxisteils

Hinweise zur Positionierung

Performanceaspekte



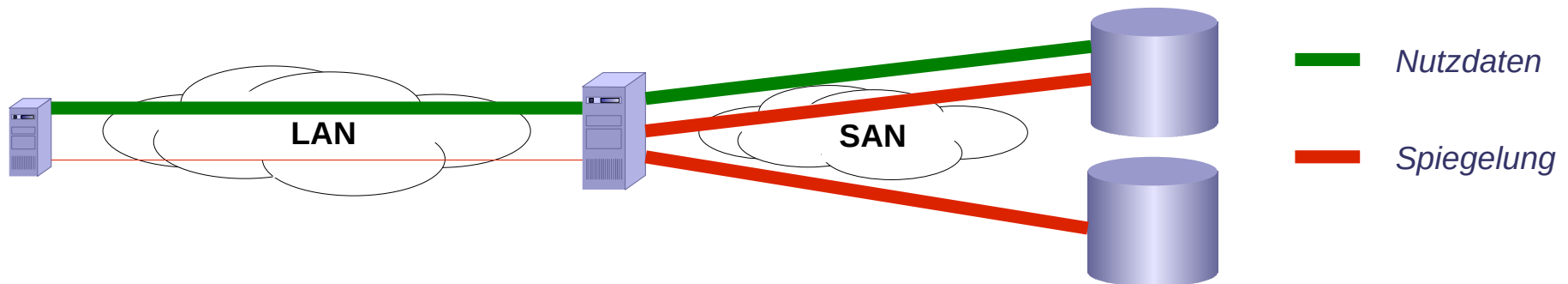
- *exzellente Performance*
 - *keine Spezialsettings für TCP/IP -> Performance "out of the Box"*
- *aber: LAN ist kein SAN*
 - *brauche ich extreme Performance bei niedrigster CPU-Last: FC*
- *RSIO wird oft schneller sein als lokale Platten*
 - *es wird Virtual Storage aus spezialisierten Systemen genutzt*
 - *Ausnutzung von Cache-Algorithmen der Speichersysteme*
- *Standard-Server hat mit nur 2x1Gigabit > 200MByte/s zur Verfügung*
- *Intelligente RSIO-Server -> nicht alle Daten müssen durch's LAN*

RSIO - was heißt intelligenter Server?

Virtual Storage und Transportvermeidung



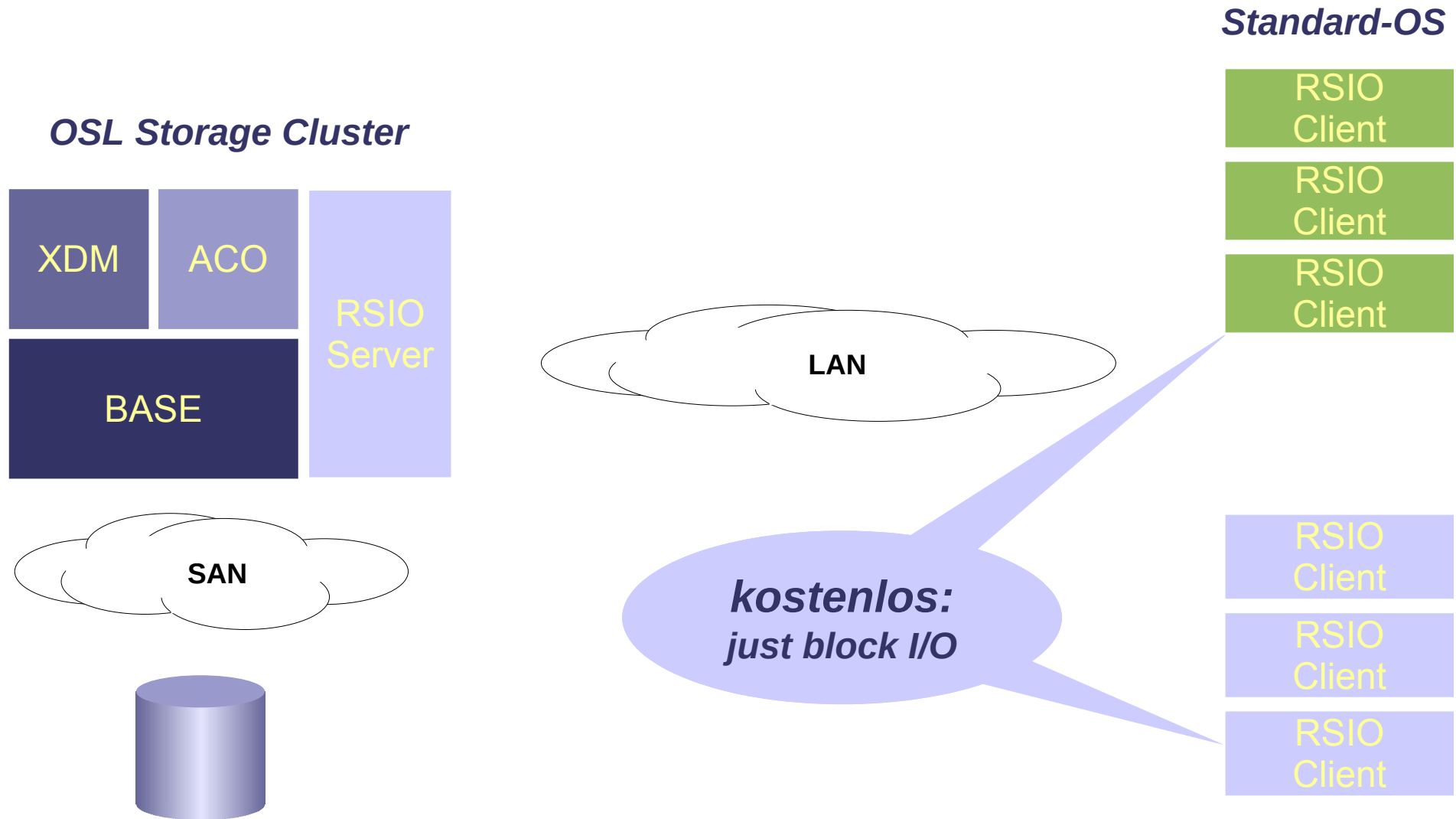
- automatische Nutzung des I/O-Caches des OS
 - Daten werden effizienter transportiert
- in Verbindung mit OSL Storage Cluster: Virtual Storage
 - kein “format”, “fdisk” auf dem Client
 - global Namespace
 - Disk-Striping etc.
- Server kann weitere Optimierungen anbieten
 - Optimierung der I/O-Aufträge
 - zusätzliche Cache-Algorithmen
- Datenspiegelung etc. ohne Transport der Daten via LAN
 - volle Steuerung vom Client aus
 - nur Control-Daten laufen über LAN, Nutzdaten bleiben im Backend



OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

RSIO - was kann ich damit anfangen?

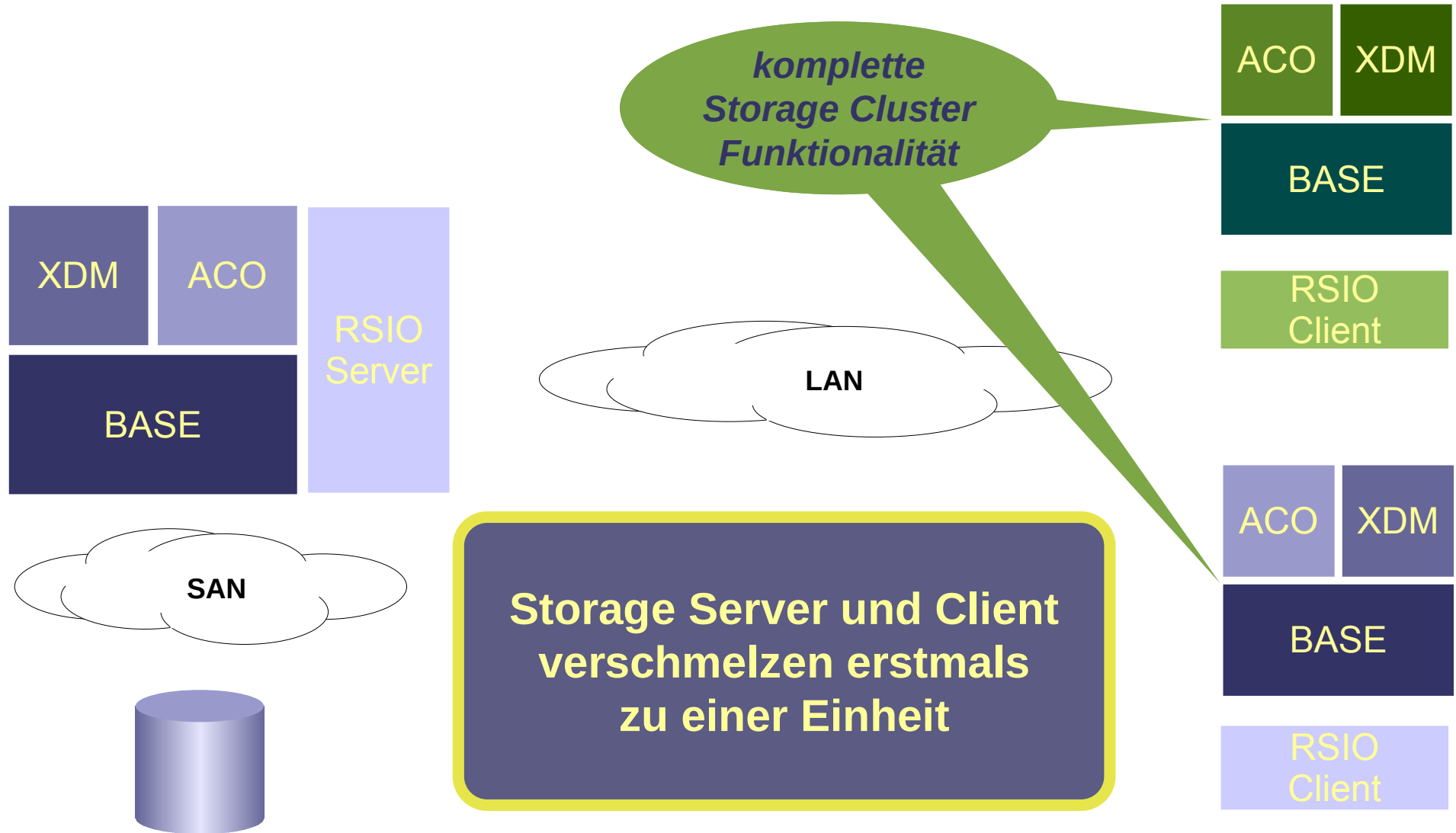
Ein einfaches Szenario in Verbindung mit OSL Storage Cluster



OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

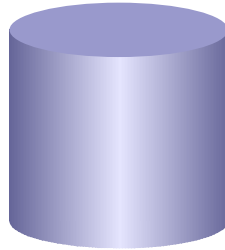
RSIO - was kann ich damit anfangen?

Volle Clusterfunktionalität für Server und Client



RSIO – Entdecke die Möglichkeiten

SAN-LAN-Konvergenz



RSIO-Server



SAN

LAN

LAN

Anwender

Komplexität reduzieren

SAN-Ports einsparen

Portauslastung verbessern

Hosts preiswert und einfach anbinden

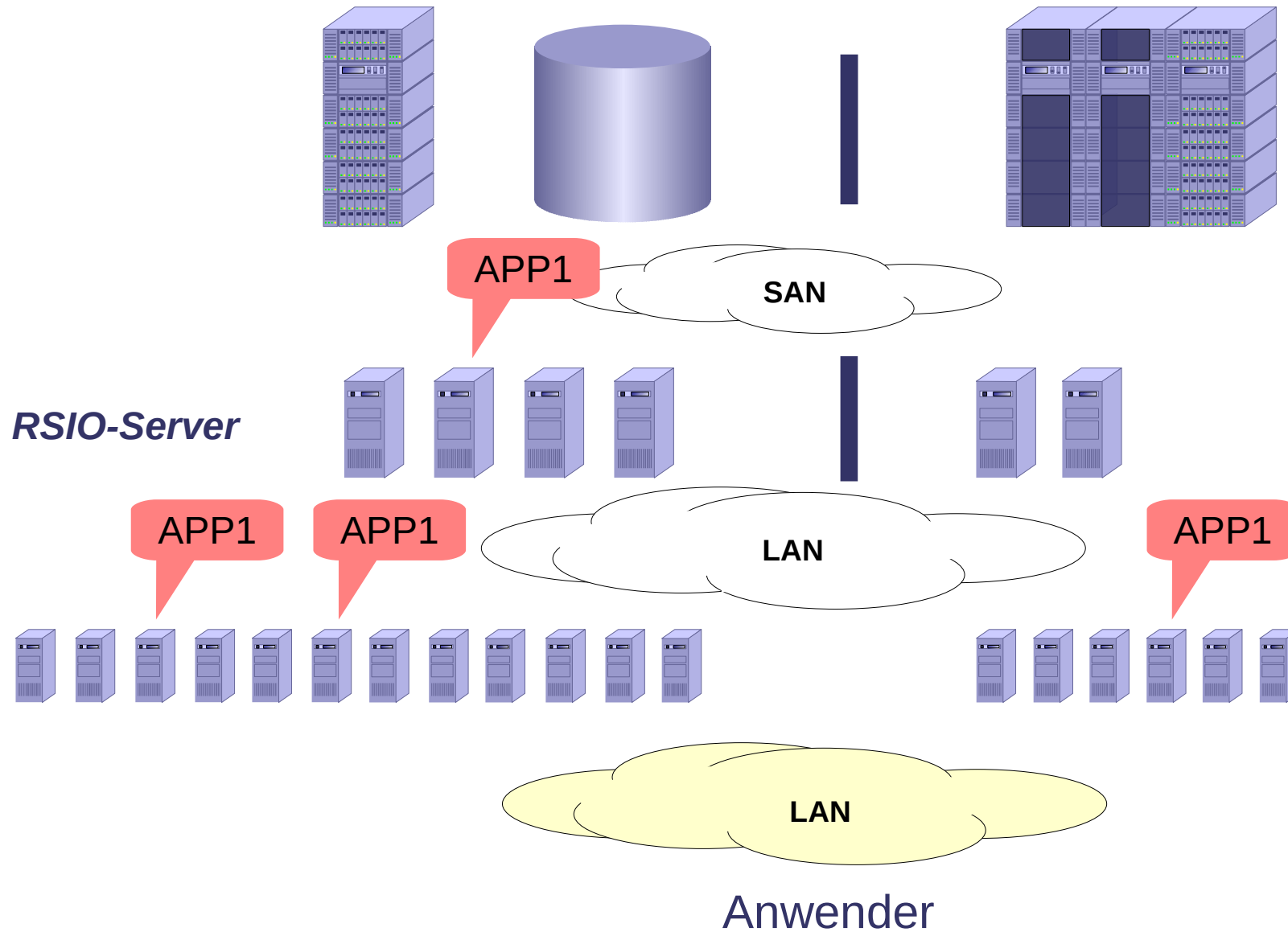
Performance-Rightsizing

erleichterte OS-Virtualisierung

Standardkomponenten

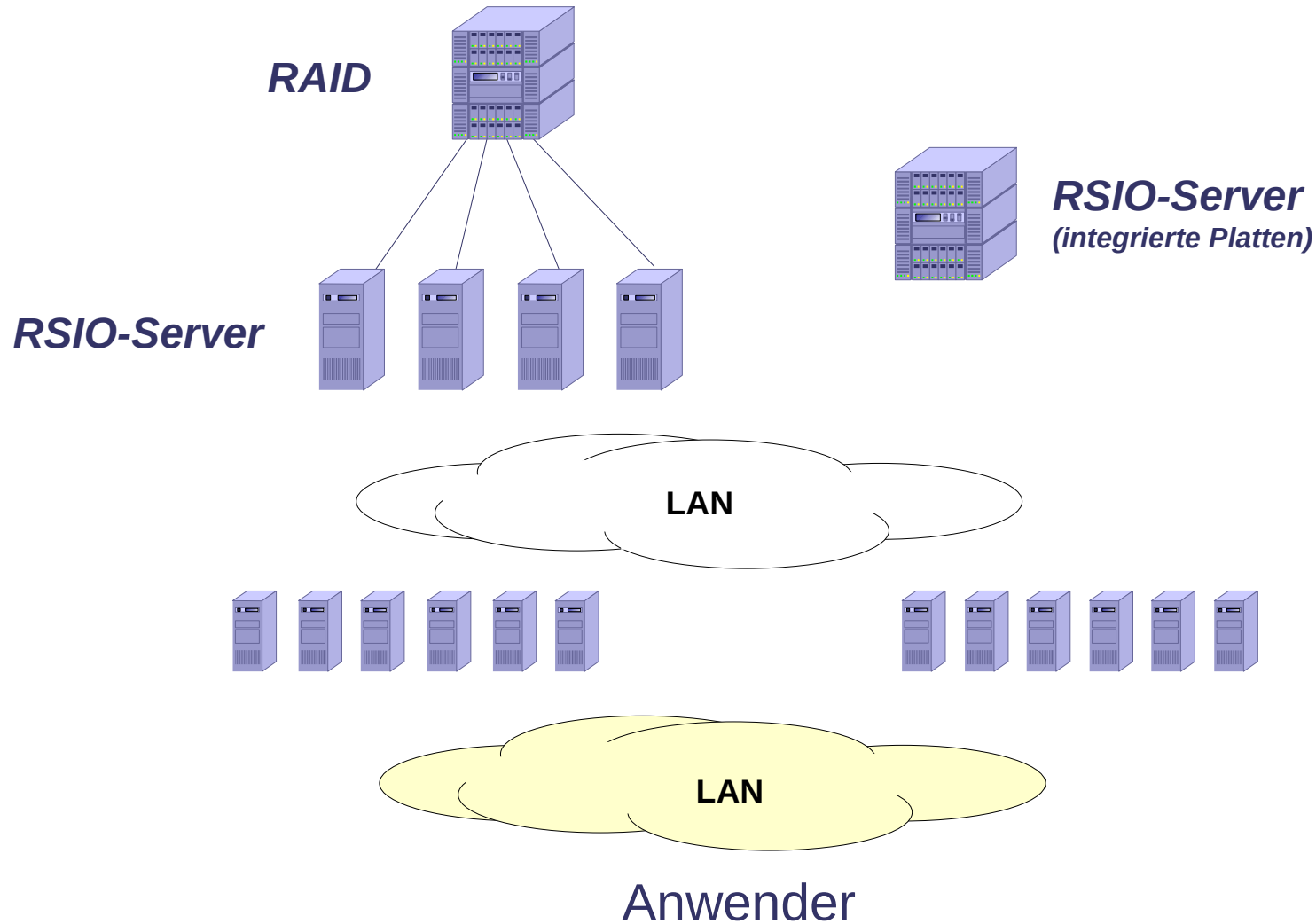
RSIO – Entdecke die Möglichkeiten

Es geht nicht nur um I/O: Run Applications Anywhere



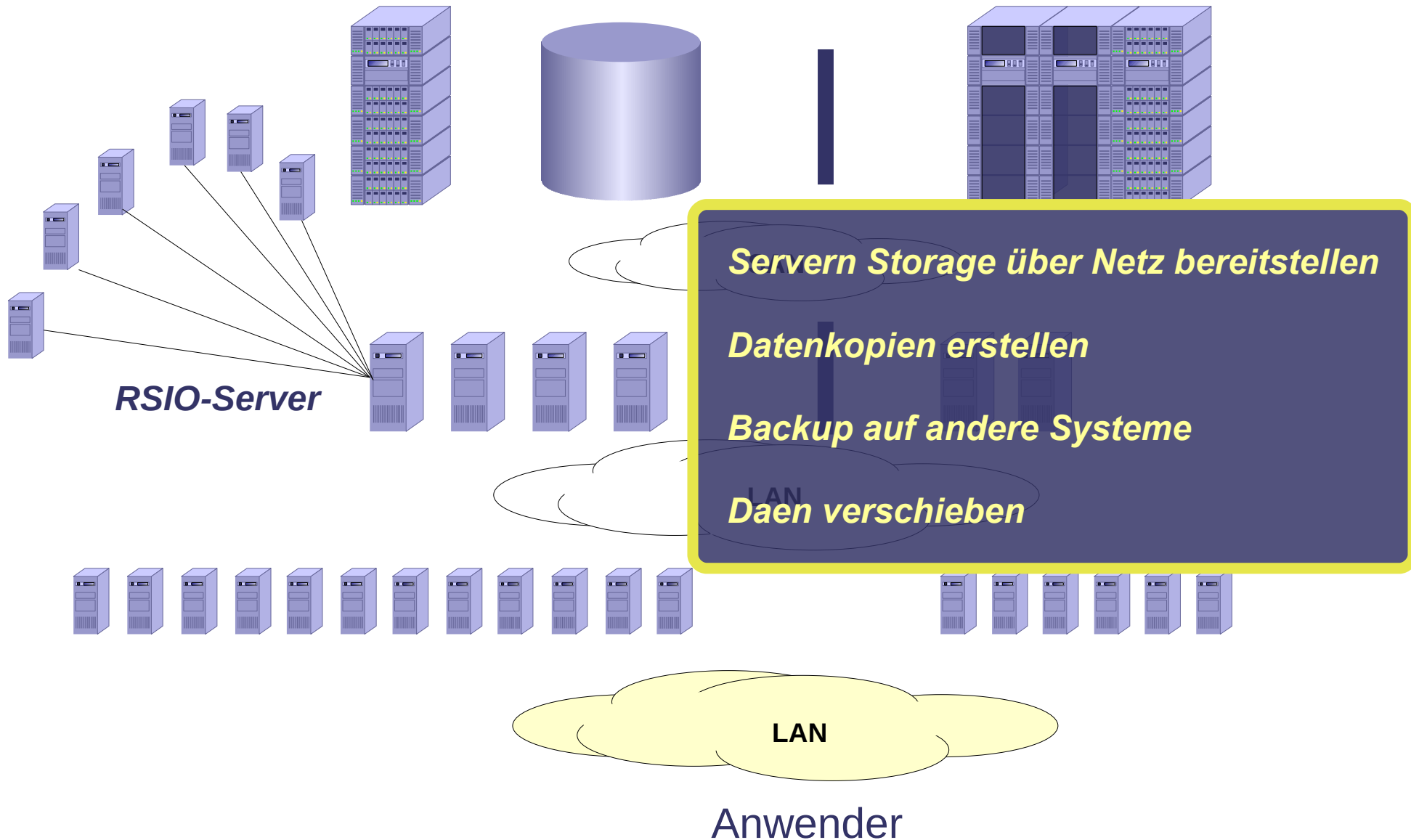
RSIO – Entdecke die Möglichkeiten

Für nicht ganz so große Anwender: Gar kein SAN mehr



RSIO – Entdecke die Möglichkeiten

Für Freidenker: Storage-Engpässe überbrücken / Storage einsammeln



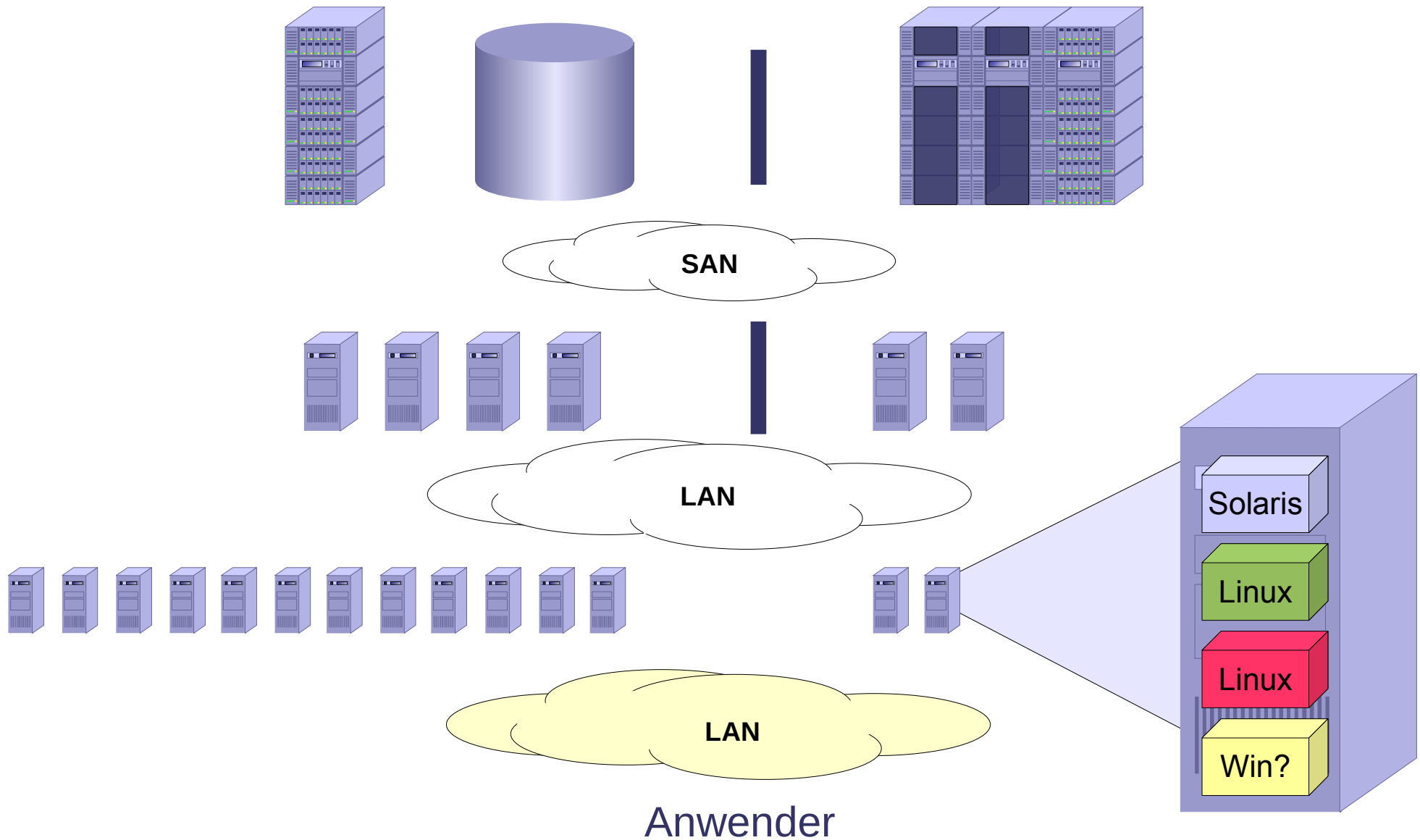
OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

*Welten verbinden:
Zeit für etwas Praxis ...*

OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

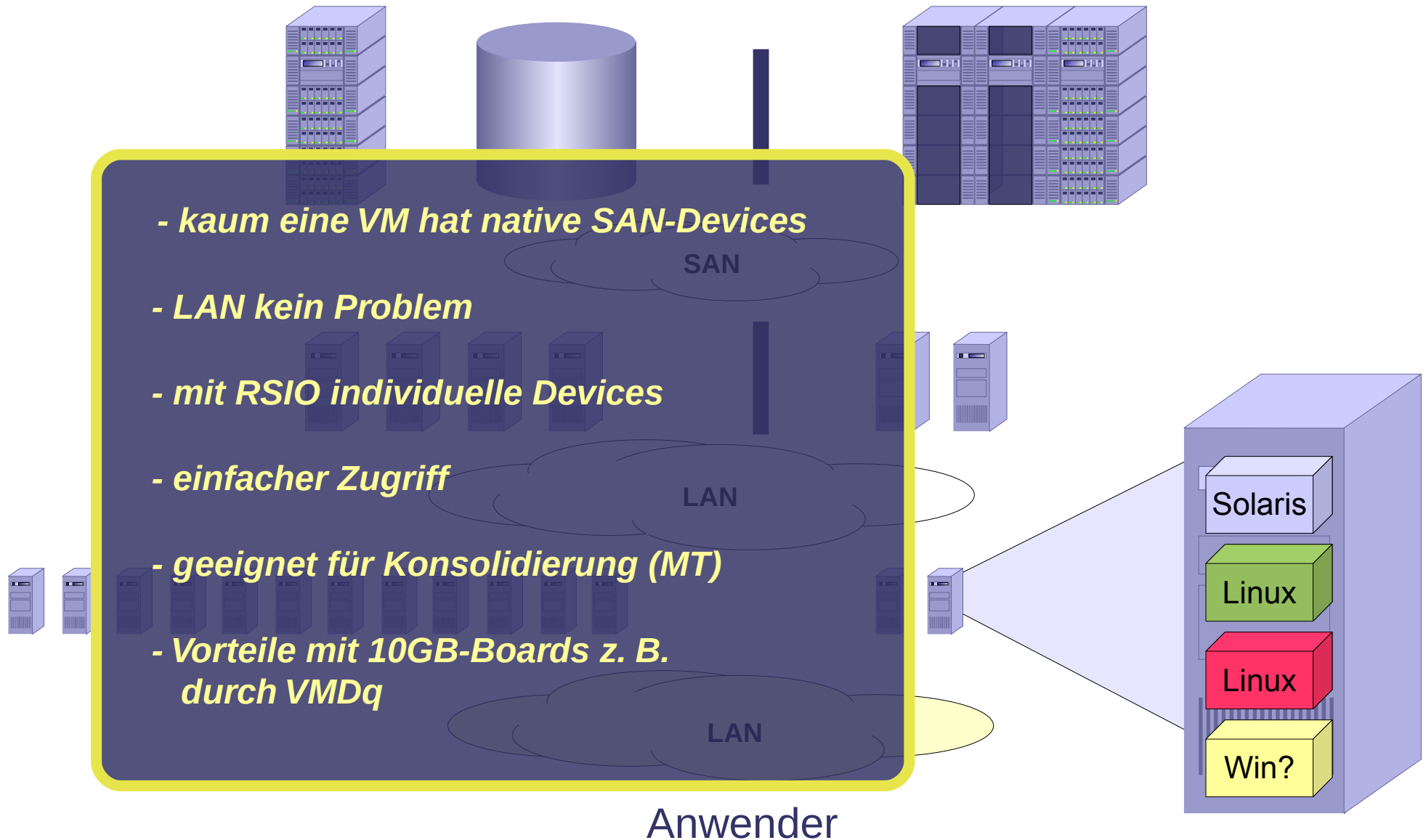
RSIO - Passend zu Virtuellen Maschinen

Drastische Vereinfachung der Storage-Anbindung für virtuelle Maschinen



RSIO - Passend zu Virtuellen Maschinen

Drastische Vereinfachung der Storage-Anbindung für virtuelle Maschinen



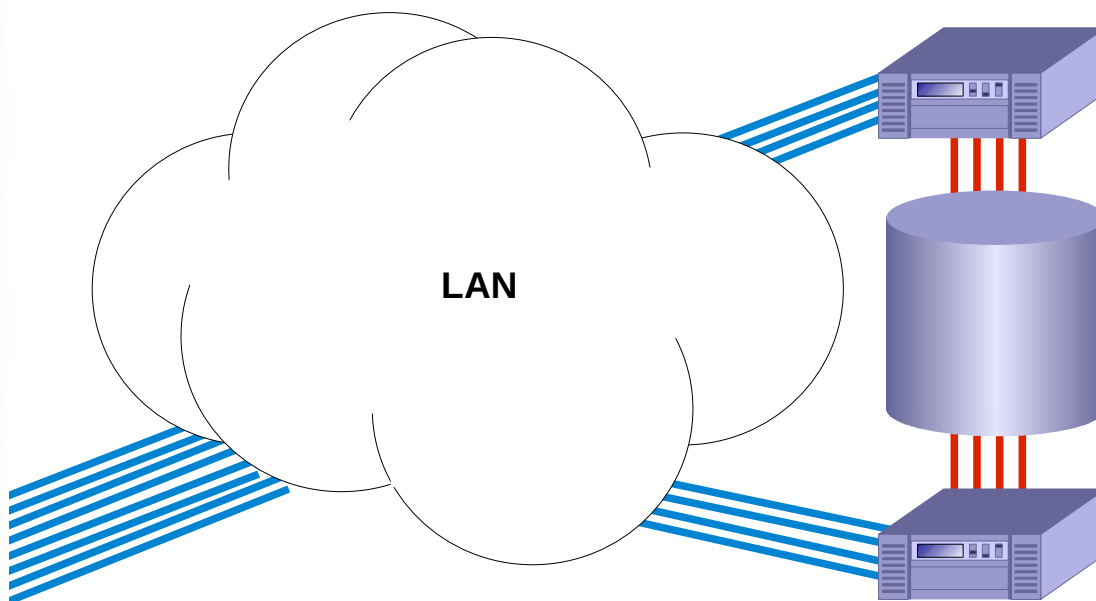
RSIO + SC - Passend zu Realen Maschinen

Storage, Management und HA für Cheap Server Farms



38 x ca. 300 RIP -> ca. 11.500 RIP

zum Vergleich: M9000 32x SPARC64 VI 2400MHz ca. 1200 RIP



**RSIO bringt Storage heran
OSL SC liefert geeignetes Framework
für Management, Backup/DR, HA**

OSL Gesellschaft für offene Systemlösungen mbH
www.osl.eu

- *noch nie war es so einfach, Standard-Server in leistungsfähige Storage-Server zu verwandeln*
- *noch nie war es so einfach und so preiswert, Server performant an Storage anzubinden*
- *noch nie bildeten Storage-Server und Storage-Clients einen so hochintegrierten, leistungsfähigen Cluster*
- *noch nie gab es alles über ein Kabel (Virtual Storage, HA-Cluster, Admin)*
- *noch nie gab es diesen Grad an Applikationsbewußtsein und diese Verflechtung mit HA/DR in der Storage-Administration*
- *noch nie gab es so eine weitreichende Integration von Speichervirtualisierung und Backup to Disk/Tape*
- *noch nie gab es diese Verflechtung von Solaris und Linux*
- *noch nie bot eine Clusterlösung diese Offenheit für verschiedene VMs*
- *noch nie gab es so vielfältige Kombinationsmöglichkeiten*
- *noch nie gab es diesen Komfort auf mehreren Plattformen*



OSL RSIO

Remote Storage I/O



Storage-Networking der nächsten Generation

Wir freuen uns auf Projekte und Gespräche:

SNW Europe
iX Solaris Day

Frankfurt/M.
Stuttgart

26./27.Oktober
30.9. - 1.10.

OSL Gesellschaft für
offene Systemlösungen mbH