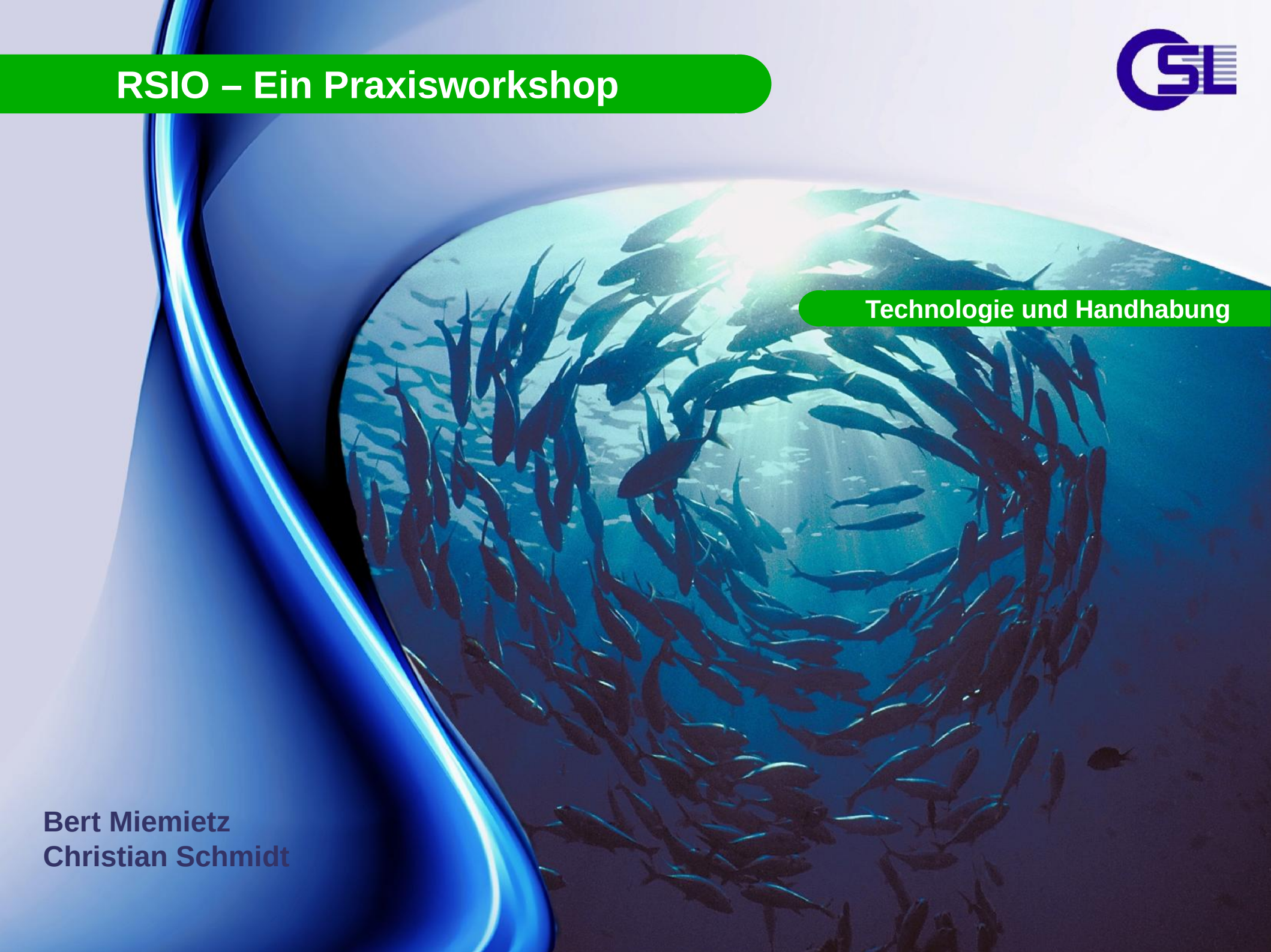
RSIO – Ein Praxisworkshop



Technologie und Handhabung

Bert Miemietz
Christian Schmidt

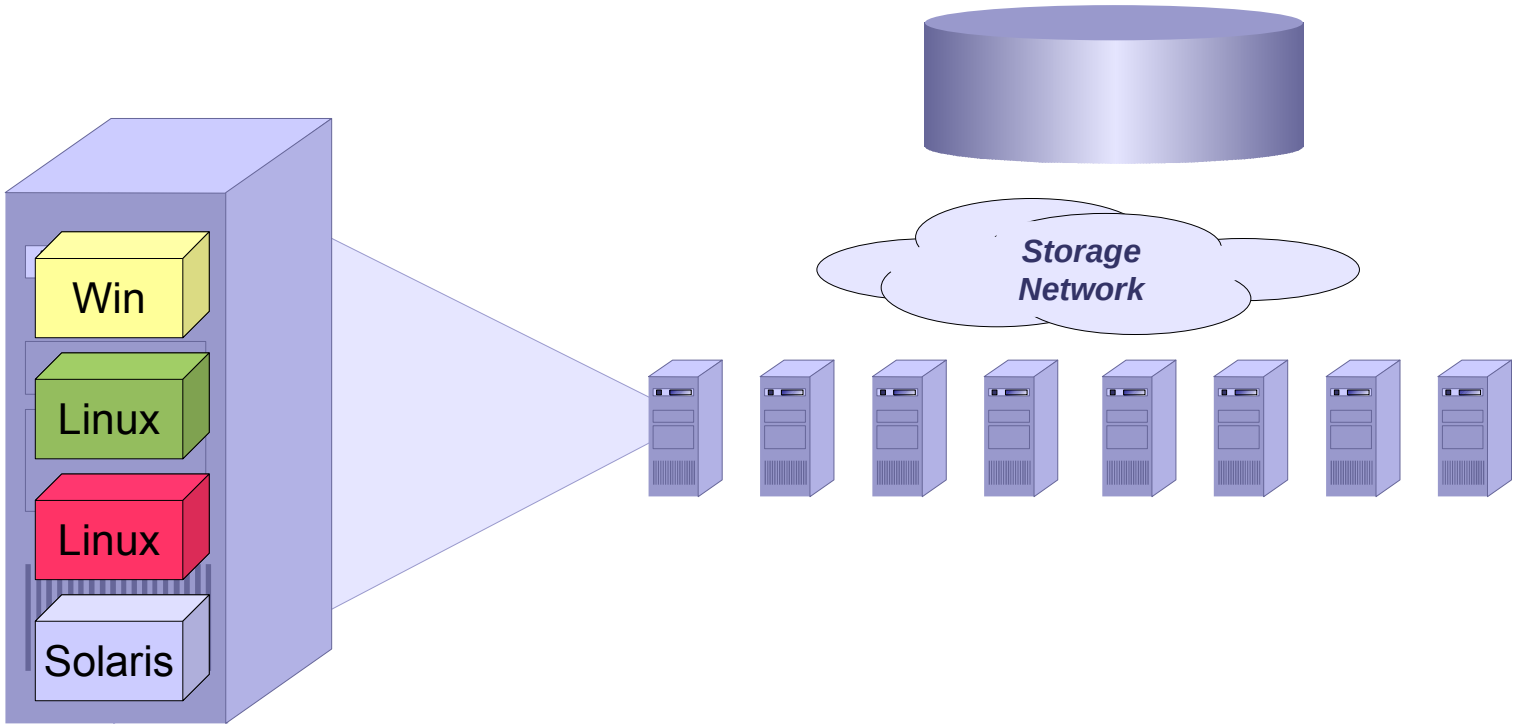
Warum Storage über Ethernet?
Anforderungen und Möglichkeiten

- **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 so schnell wie eine Festplatte*
  - *Gigabit-LAN heute mit applikationsadäquaten Durchsätzen*
  - *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*

Warum Storage über Ethernet?

Noch ein ganz wichtiger Punkt ...

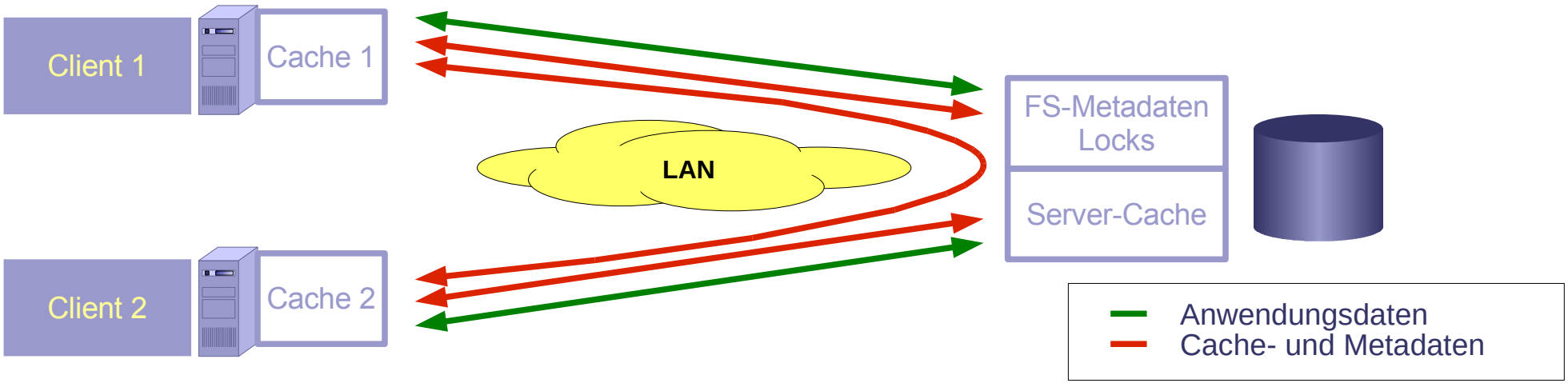


Vielfalt an Virtualisierungstechnologien, Plattformen ...
erhöht Uniformierungsdruck bei Connectivity

Storage über Ethernet heute: NFS, SMB, CIFS

- ***Was bietet NAFS***
  - *Spezialisierung auf Fileservices, dafür relativ einfache Handhabung*
  - *Filesharing*
  - *Keine komplexen RAID Funktionen*
  - *dateisystemtypische Funktionen wie Snapshots*
  - *Weite Verbreitung und Unterstützung der Protokolle*

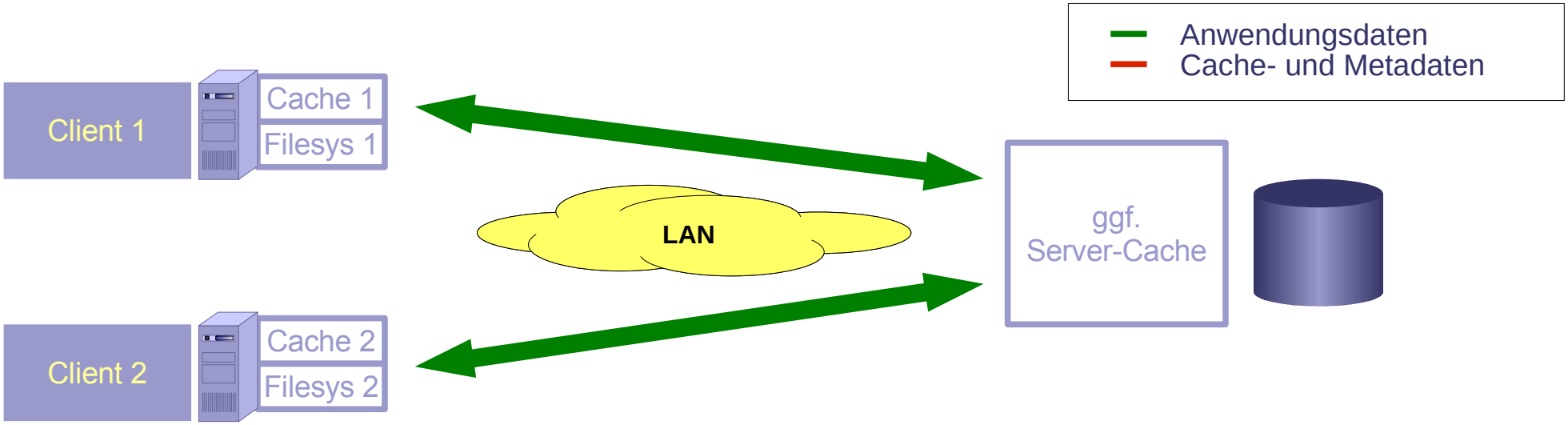
- ***Die Kehrseite***
  - *Aufwendige Integration mit Server OS (User- und Zugriffsmanagment)*
  - *Cache und Cohärenzproblematik*
  - *feste Bindung an File-Access-Semantik*
  - *nicht trivial: Skalierbarkeit, Parallelisierung, Hochverfügbarkeit, Multipathing*



Storage über Ethernet: RZ-Anwender brauchen Block-I/O

Jenseits von Filesharing überwiegen die Vorteile

- *volle Kontrolle des Client-OS über das Storage-Device*
- *nutzbar für beliebige Filesysteme und Applikationen, IO-Verhalten gut steuerbar*
- *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*



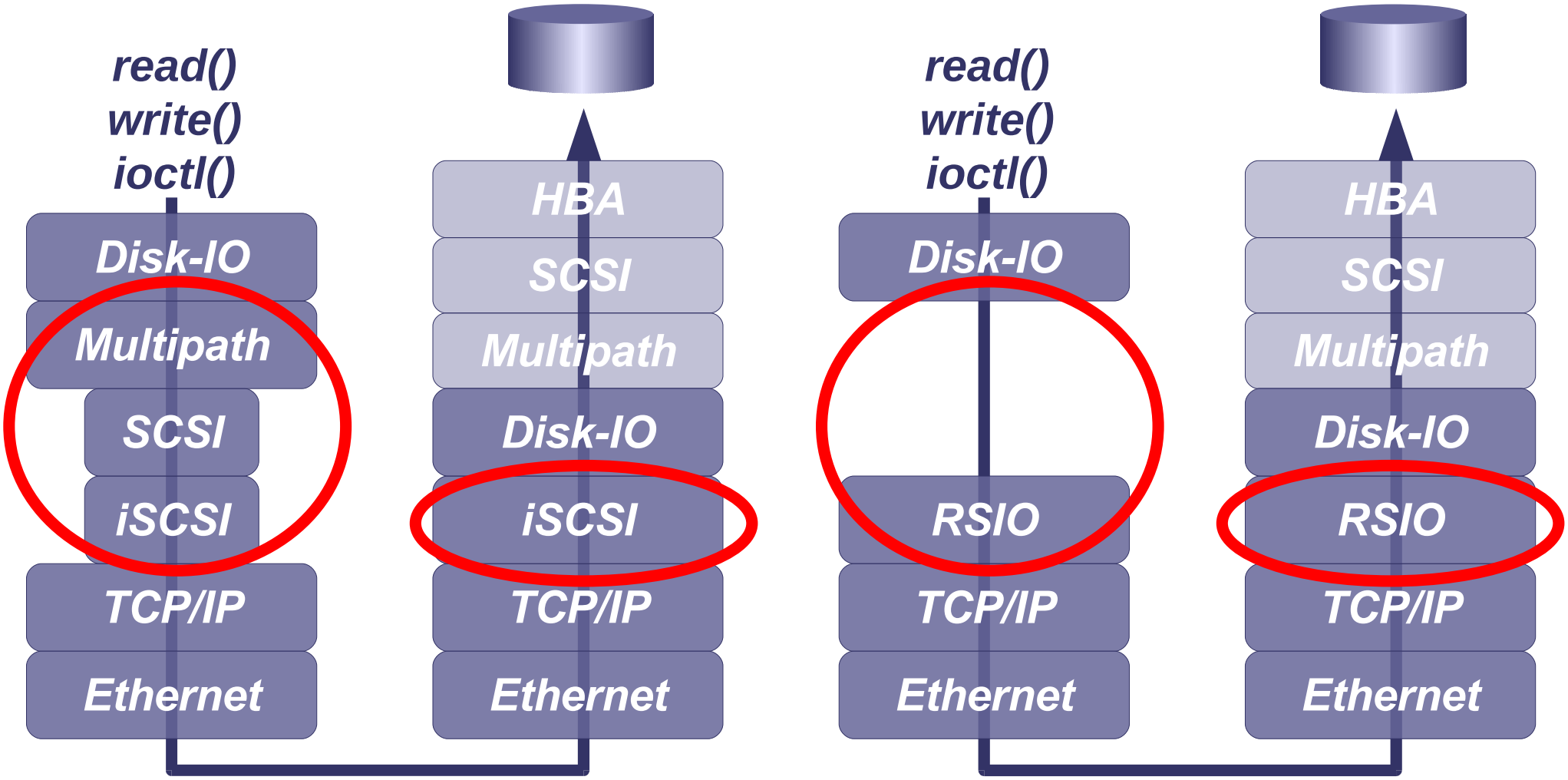
RSIO - Remote Storage I/O

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*

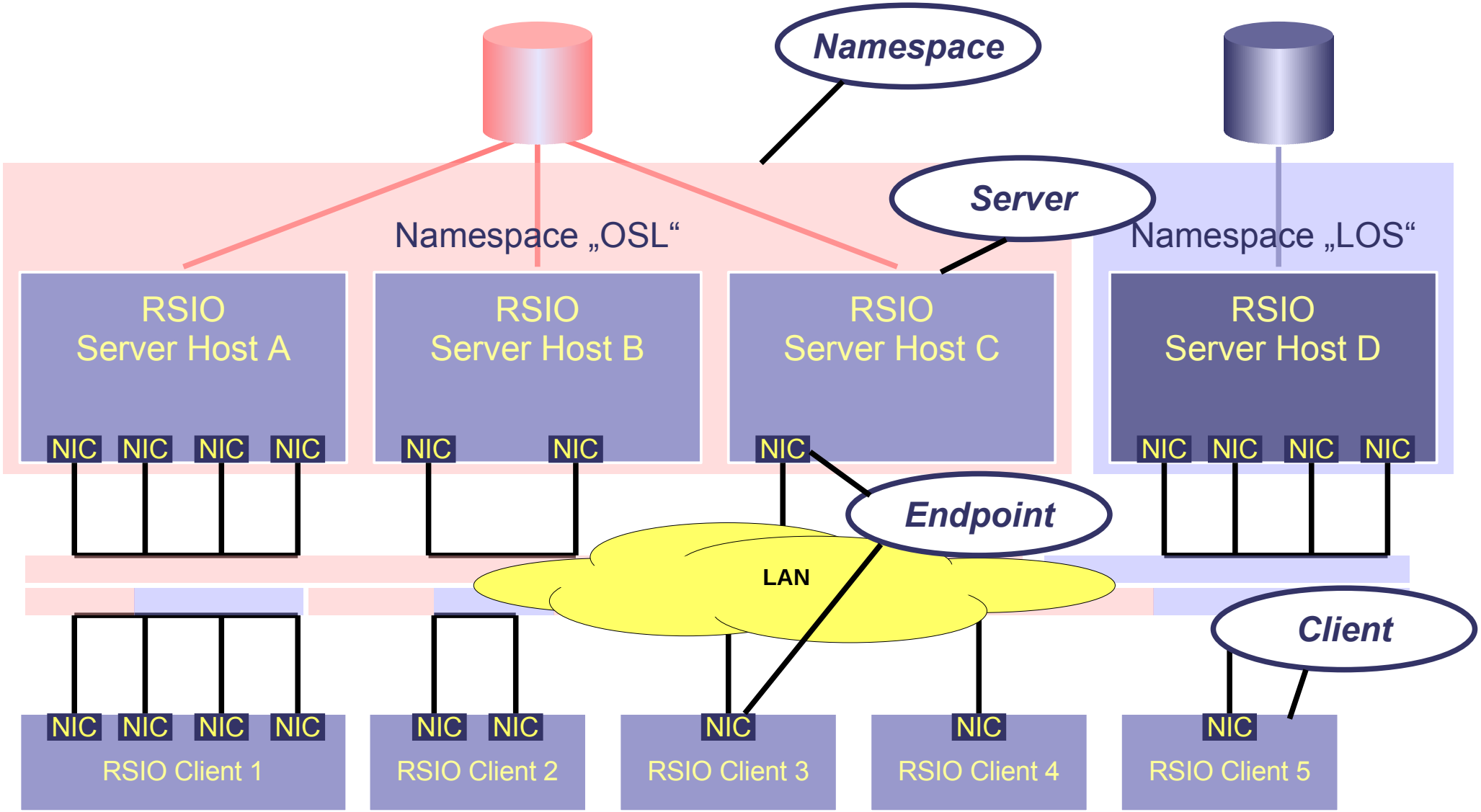
RSIO - Remote Storage I/O

Vergleich der Protokollstacks



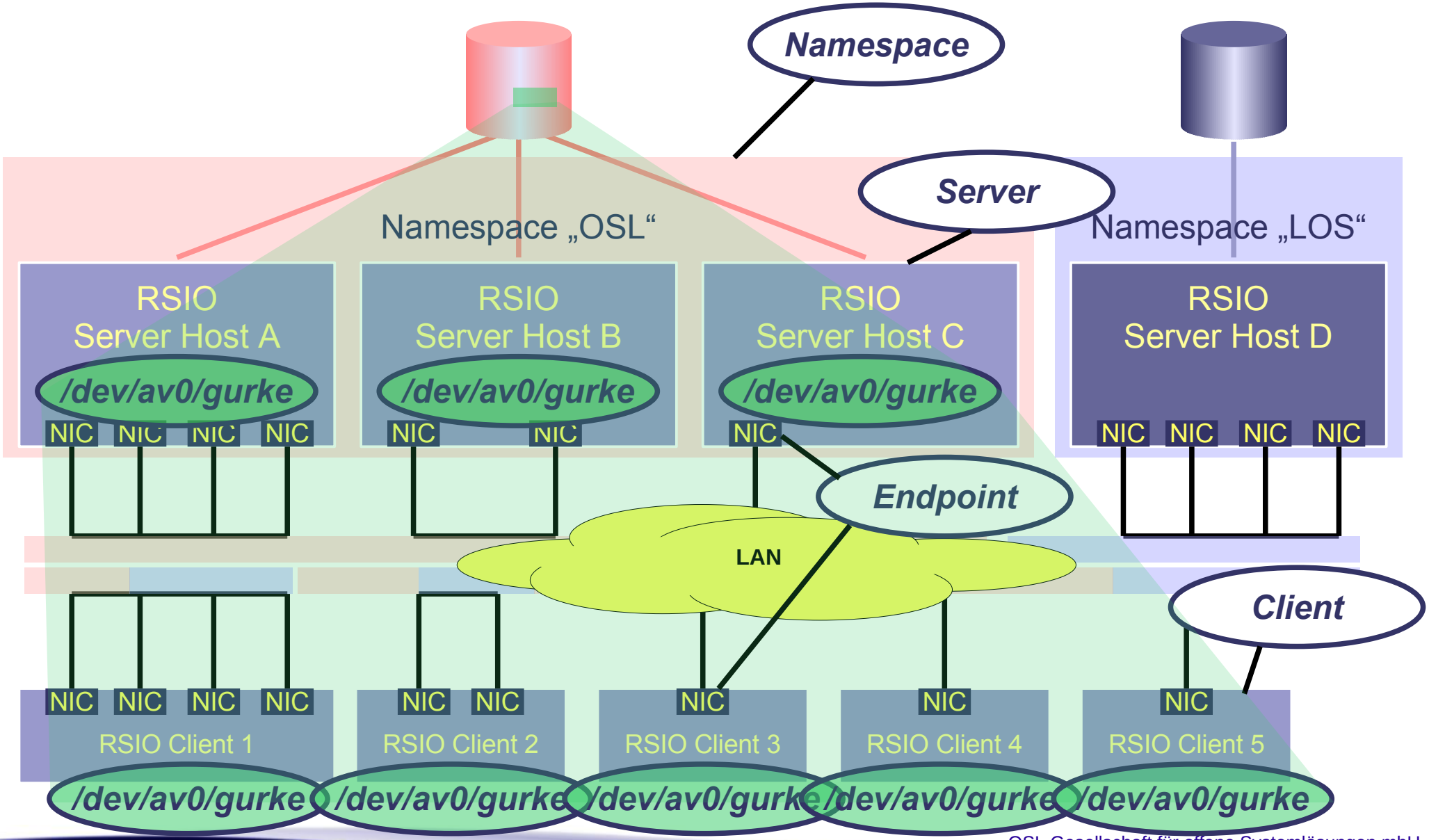
RSIO – Architektur im Überblick

Klar gegliedertes und flexibles administratives Konzept



RSIO – Architektur im Überblick

Klar gegliedertes und flexibles administratives Konzept



RSIO – Architektur im Überblick

Klar gegliedertes und flexibles administratives Konzept

- ***Ein Namespace definiert Server (und Clients) mit Zugriff auf dieselben Storage-Ressourcen -> Namensdienst***
- ***Jeder Server 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 Namespaces*
  - *Ermitteln verfügbarer Server*
  - *Ermitteln verfügbarer Verbindungen*
  - *Ermitteln der Schnittstelleneigenschaften*
  - *Test der Parameter auf der Übertragungsstrecke*

Wie OSL RSIO umgesetzt hat

Kombination mit dem OSL Storage Cluster

So meldet sich eine iSCSI-Lun (“format” - Solaris)

```
29. c3t227d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
    /iscsi/disk@0000iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59f2d03474f708.target-00001,0
30. c3t229d0 <DEFAULT cyl 1021 alt 2 hd 64 sec 32>
    /iscsi/disk@0001iqn.1986-03.com.sun%3A02%3A06df3360-bb85-ee33-bf59f2d03474f708.target-00001,0
```

Und so sieht der RSIO-Client Plattenressourcen

```
# rsconfig -q
000 osl
    clt: big-6
    srv: 000 big-5
         0  tvol1          disk        2097152 blocks of  512 bytes
         0  shadow         disk        2097152 blocks of  512 bytes
         0  ora_db         disk       10485760 blocks of  512 bytes
         0  postgres_db    disk       10485760 blocks of  512 bytes
         0  whole_zone     disk       41943040 blocks of  512 bytes
```

Und was ist mit der Performance?

Protokoll erlaubt hohe Performance und beeindruckende Skalierbarkeit

Server-Performance bei Cache Read / 8k

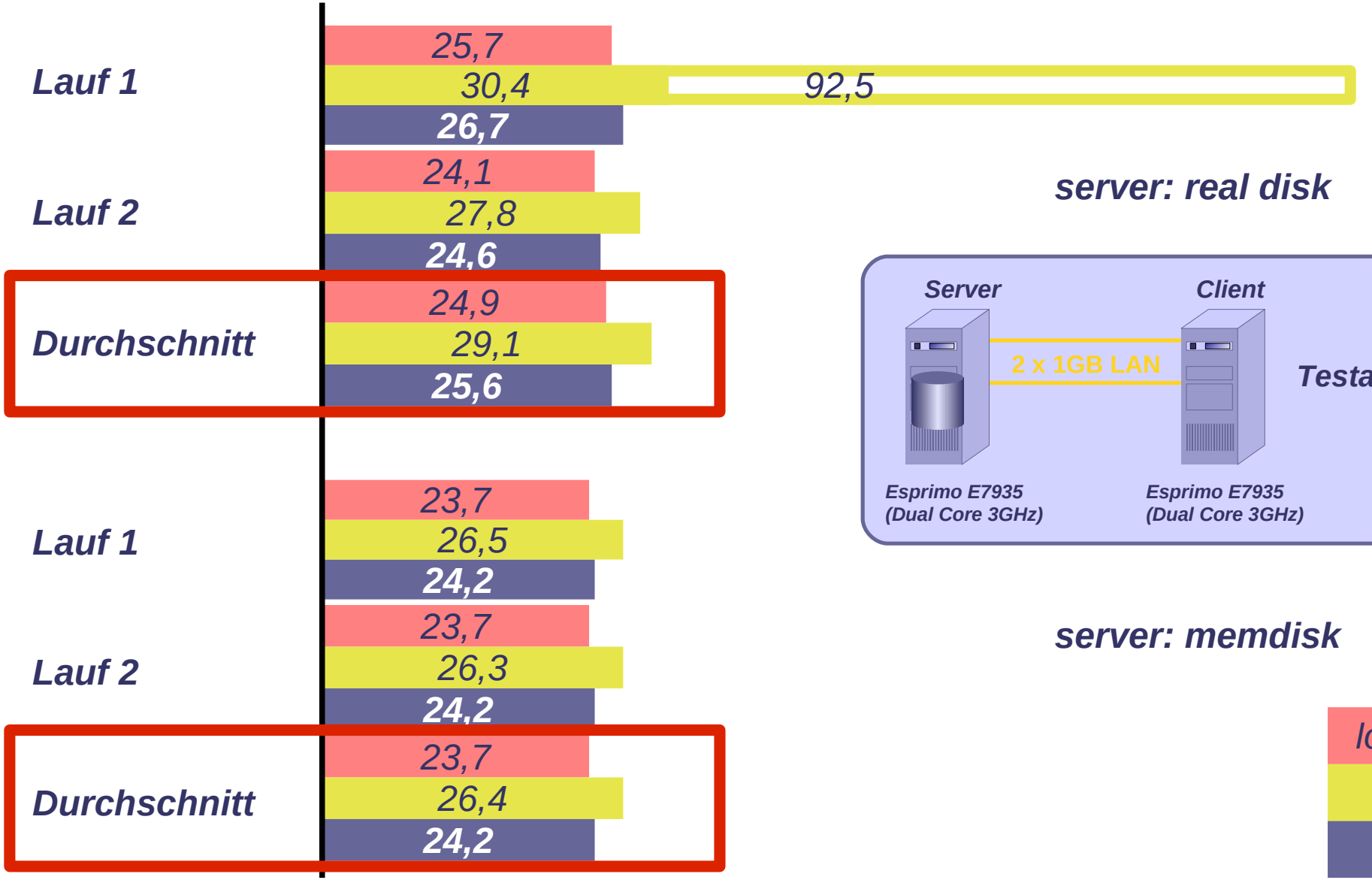
iSCSI	*10 Clients*	*100 Threads*	*7,6 Cores*	***31.000 IOPS***
iSCSI / comstar	*10 Clients*	*100 Threads*	*10,0 Cores*	***85.000 IOPS***
RSIO	*4 Clients*	*64 Threads*	*5,6 Cores*	***98.000 IOPS***
RSIO	*4 Clients*	*128 Threads*	*6,3 Cores*	***102.000 IOPS***

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***

Performance in der Praxis (single thread)

Vergleich zu NFS und DAS: Compilierung Storage Cluster Packages (Zeit in Sekunden)



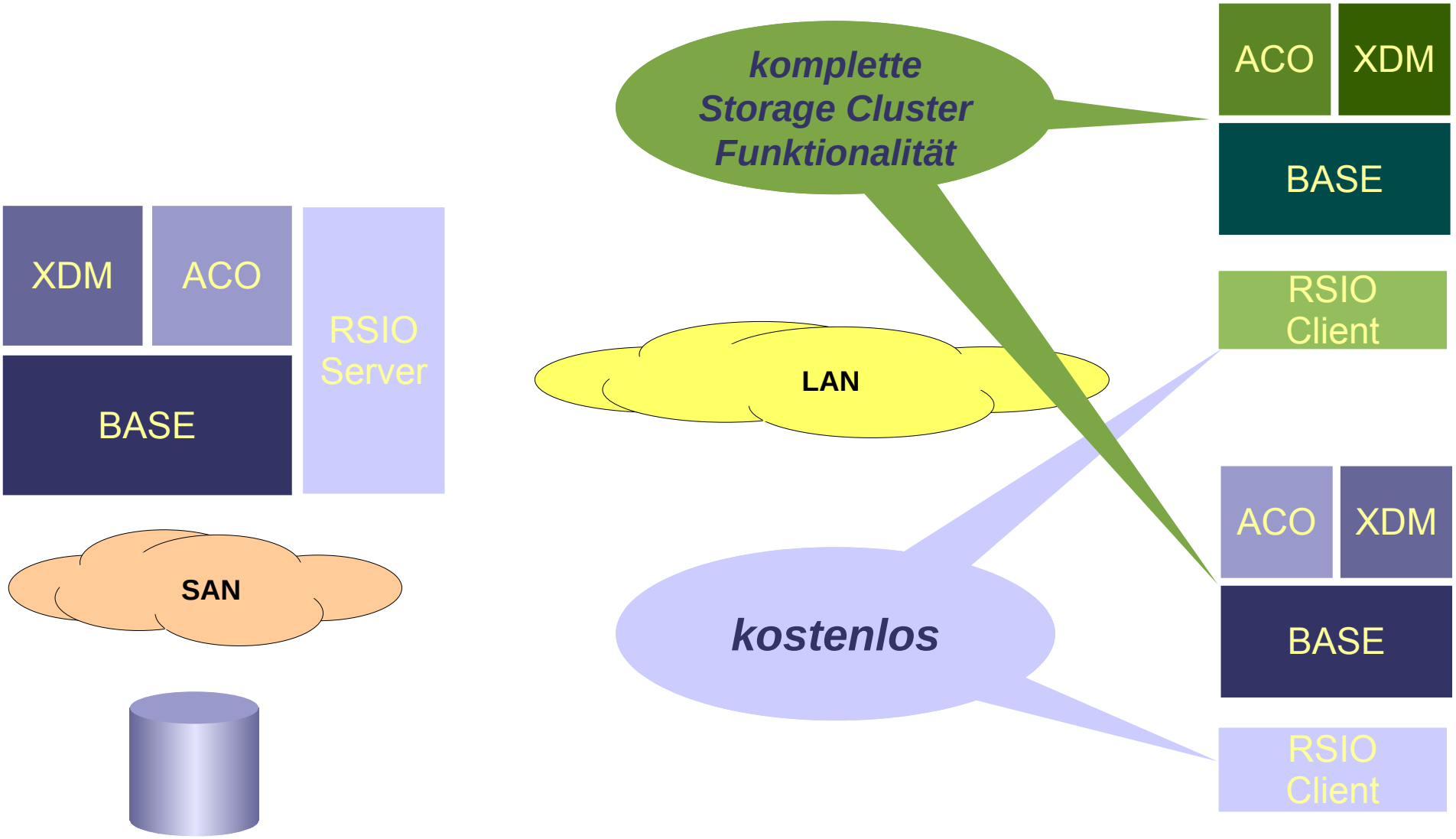
server: real disk

server: memdisk

local disk
NFS
RSIO

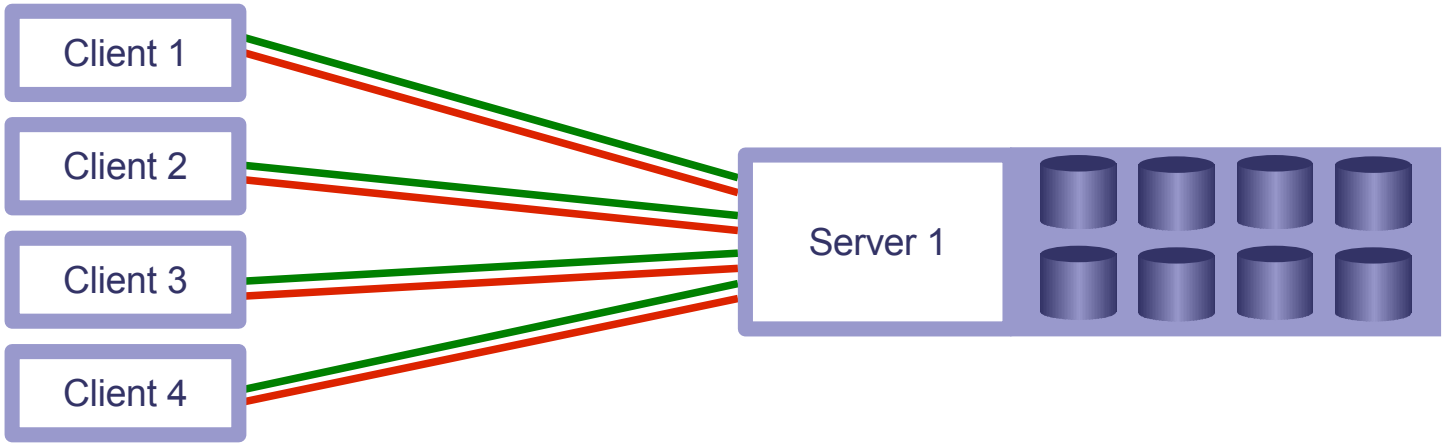
Was kann ich mit RSIO aufbauen?

Das Prinzip



RSIO – Block I/O over Ethernet

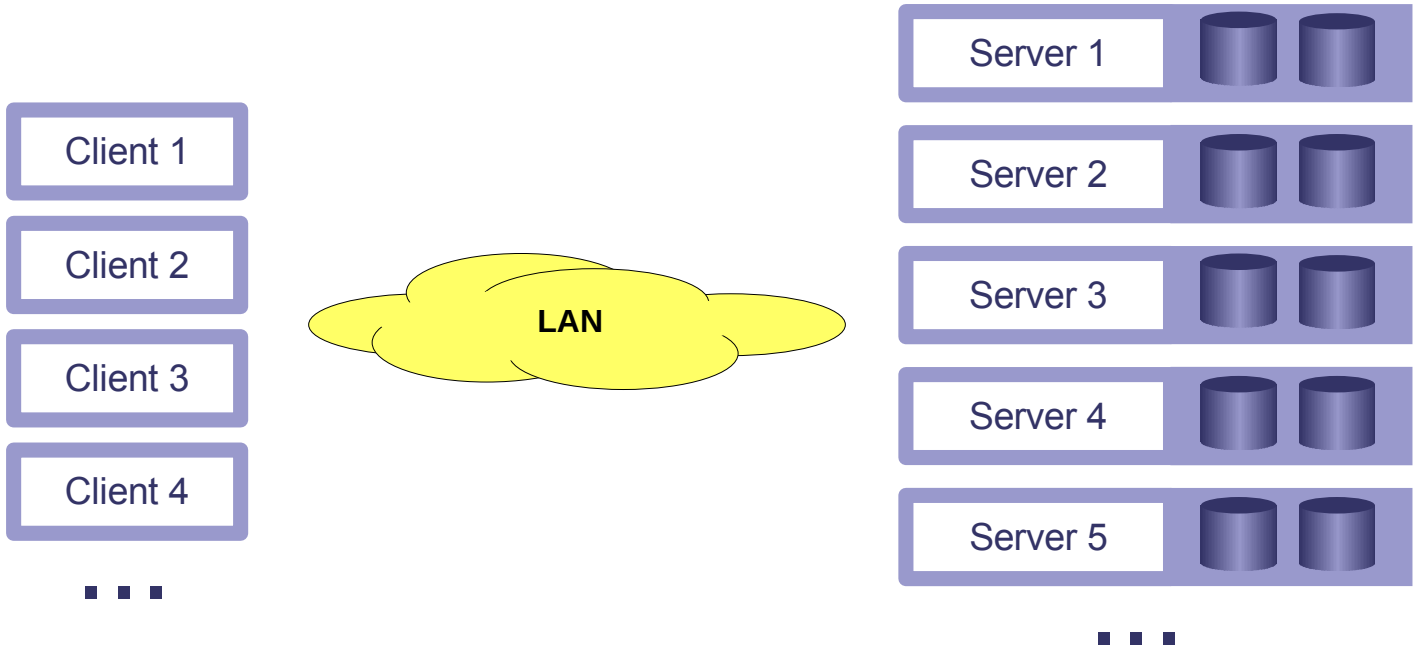
Beispiel 1: Einfacher Storage Server



- *Zugriff auf zentrales Speichersystem -> Global Pool, Global Namespace*
- *Virtualisierung und Cluster (HV) auf Clients einfach realisierbar*
- *Möglichkeit der Zentralisierung von Backup, Snapshots ...*
- *sehr preiswerte Speicheranbindung bei guter Performance*
- *redundante Datenpfade, Durchsatz je nach Bedarf skalierbar*

RSIO – Block I/O over Ethernet

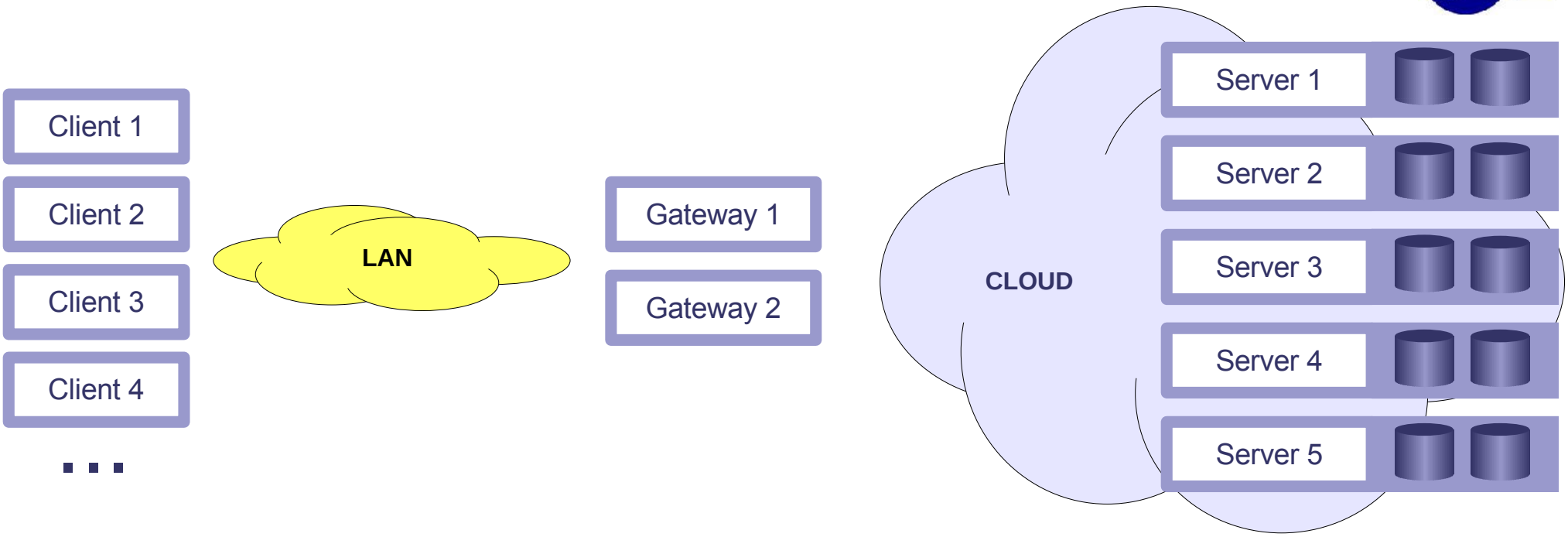
Beispiel 2: Storage Server Farm



- *Skalierung in Speichervolumen und Bandbreite*
- *jeder Server mit eigenem Namespace*
- *Storage-Kapazitäten “einsammeln” und so mit einfachen Mitteln große Kapazitäten und Bandbreiten darstellen*
- *nicht vergessen: Verfügbarkeit in der Server-Farm*

RSIO – Block I/O over Ethernet

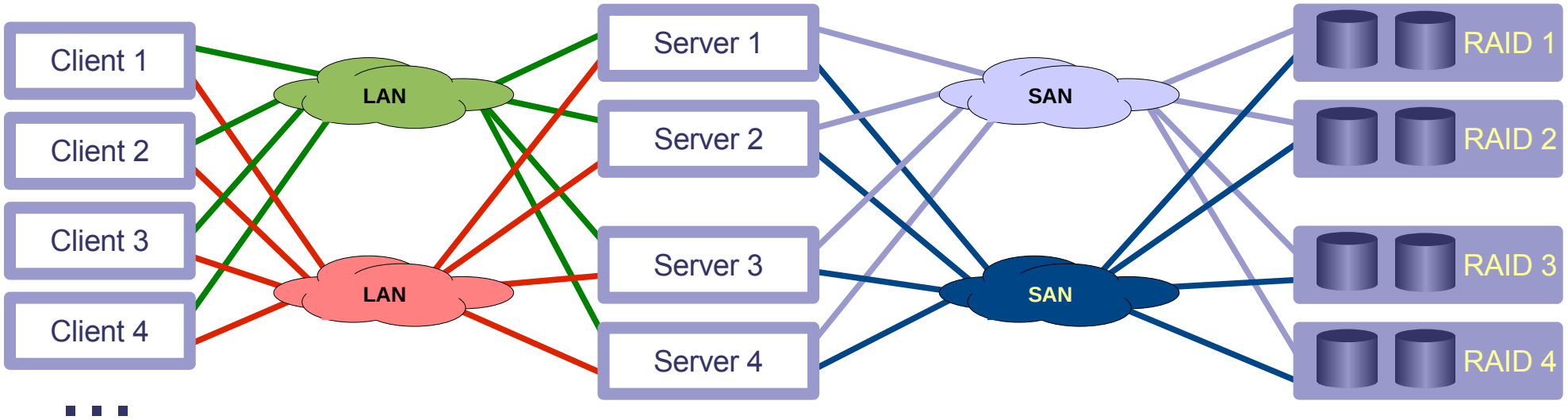
Beispiel 3: Szenario für Cloud Storage



- *Zugriff auf Speicherressourcen jenseits des LAN*
- *Mehrpfadigkeit, Bandbreite, Performance treten in den Hintergrund*
- *Gleichartige Administration wie bei RSIO im LAN*
- *Nutzt prinzipielle Routingfähigkeit von RSIO über IP*

RSIO – Block I/O over Ethernet

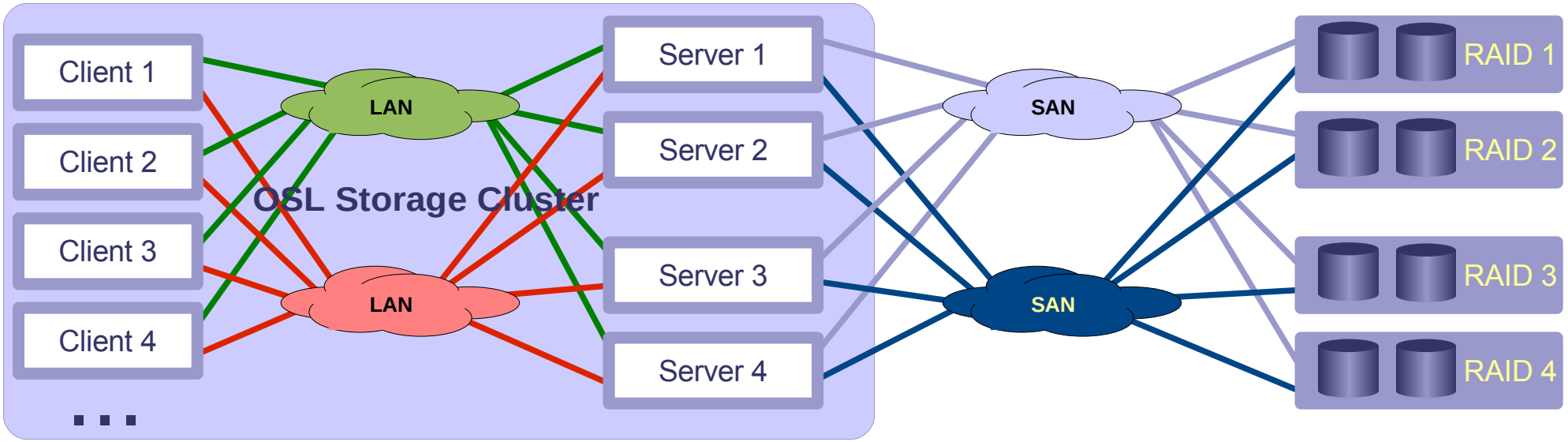
Beispiel 4: SAN-LAN-Konvergenz und geclusterte Storage-Server



- ***SAN ins LAN hinein verlängern***
- ***SAN-attached Server reichen "im Hintergrund" Storage-Ressourcen durch***
- ***verbesserte Ausnutzung des SANs, Performance-Rightsizing***
- ***hohe Performance, hohe Verfügbarkeit bei extrem niedrigen Kosten für RSIO-Clients***
- ***weitere Verbesserung von Performance und Systemauslastung möglich z. B. durch Nutzung freien Speichers als Cache***

RSIO – Block I/O over Ethernet

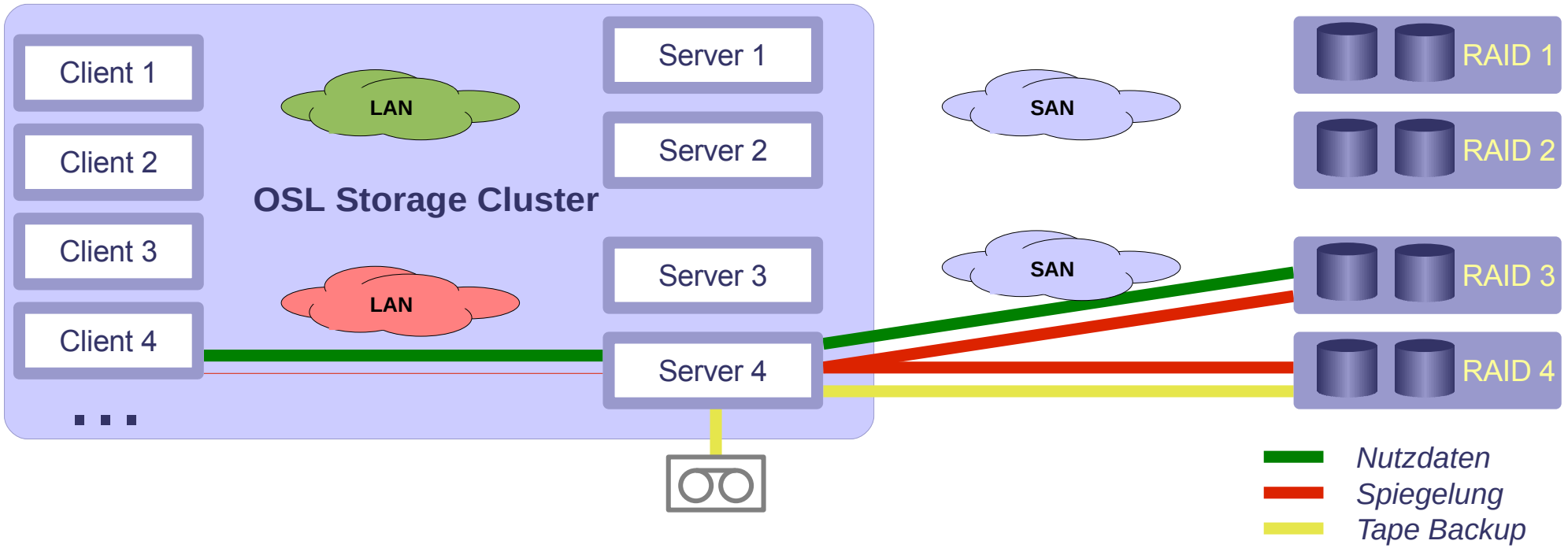
Beispiel 5: Server und Clients in einem Cluster verbinden



- ***alle Funktionen wie im vorherigen Beispiel***
- ***zusätzlich weitere Storage-Management-Funktionen:***
  - *Storage-Allokation, -Management vom Client aus*
  - *applikationsbezogene Speichervirtualisierung vollumfänglich auf Client nutzbar*
  - *Möglichkeit der transparenten Nutzung von Datenspiegelung, Backup to Disk etc.*
- ***Verschmelzung von Client und Server zu einer Einheit***
- ***run applications everywhere***

RSIO – Block I/O over Ethernet

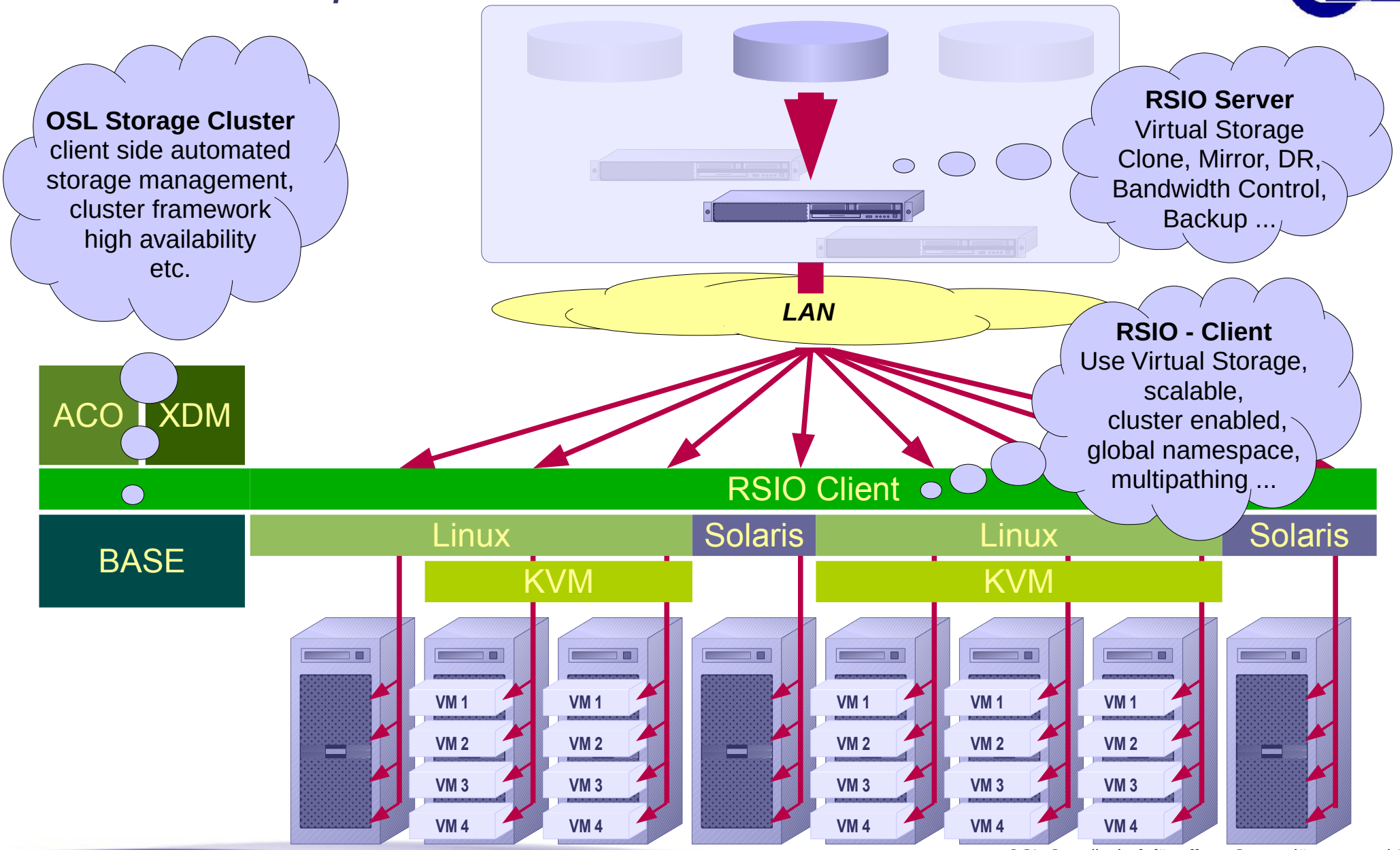
Beispiel 6: Hochgeschwindigkeitsbackup für LAN-attached Blockdevices



- ***über das LAN laufen nur Nutzdaten und die Steueranweisungen***
- ***LAN-less Backup:***
  - *hohe Geschwindigkeit*
  - *vollständige Steuerung vom Client aus*
  - *applikationsbezogene Aktionen*

RSIO – Block I/O over Ethernet

Nochmal zum Prinzip



OSL Gesellschaft für offene Systemlösungen mbH

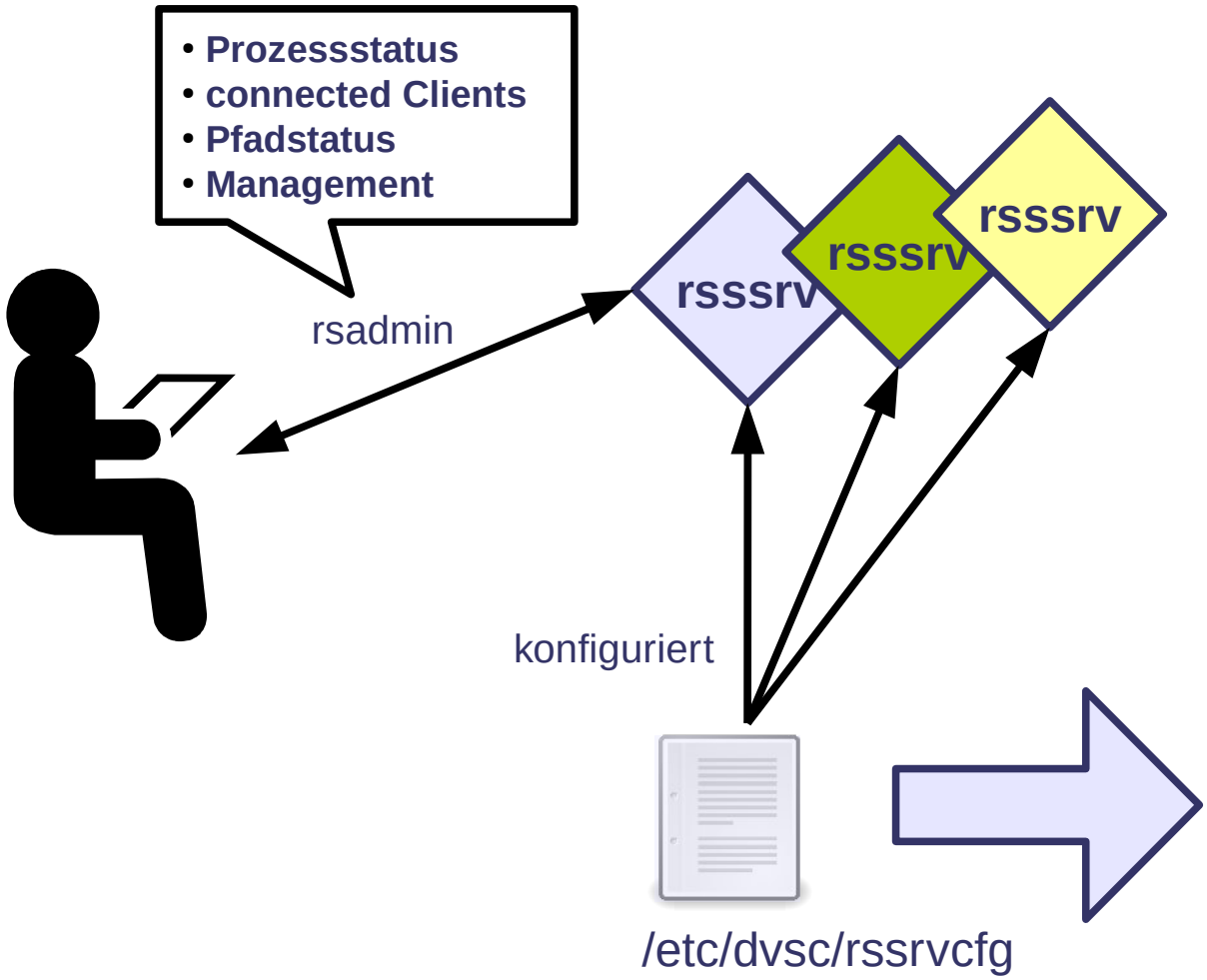
www.osl.eu

RSIO in der Praxis
Installation, Konfiguration und I/O-Tests

- **Installation und Konfiguration eines RSIO Clients**
  - Pakete installieren
  - Einfache Konfiguration mit Auto-Explorer und Konfigurationswerkzeug
  - Sofortiger Zugriff auf alle Plattenressourcen des RSIO Servers
    - Accessmanagment kann aktiviert werden
    - modularer Aufbau – Möglichkeiten für Verschlüsselung oder spezielle Authentifizierungsmaßnahmen
- **I/O Tests**
  - Anlegen eines Volumes (Stripe über 4 Disks)
  - sequentielles Lesen mit 8k Blocksize und 10 Threads
  - sequentielles Lesen mit 128k Blocksize und 10 Threads

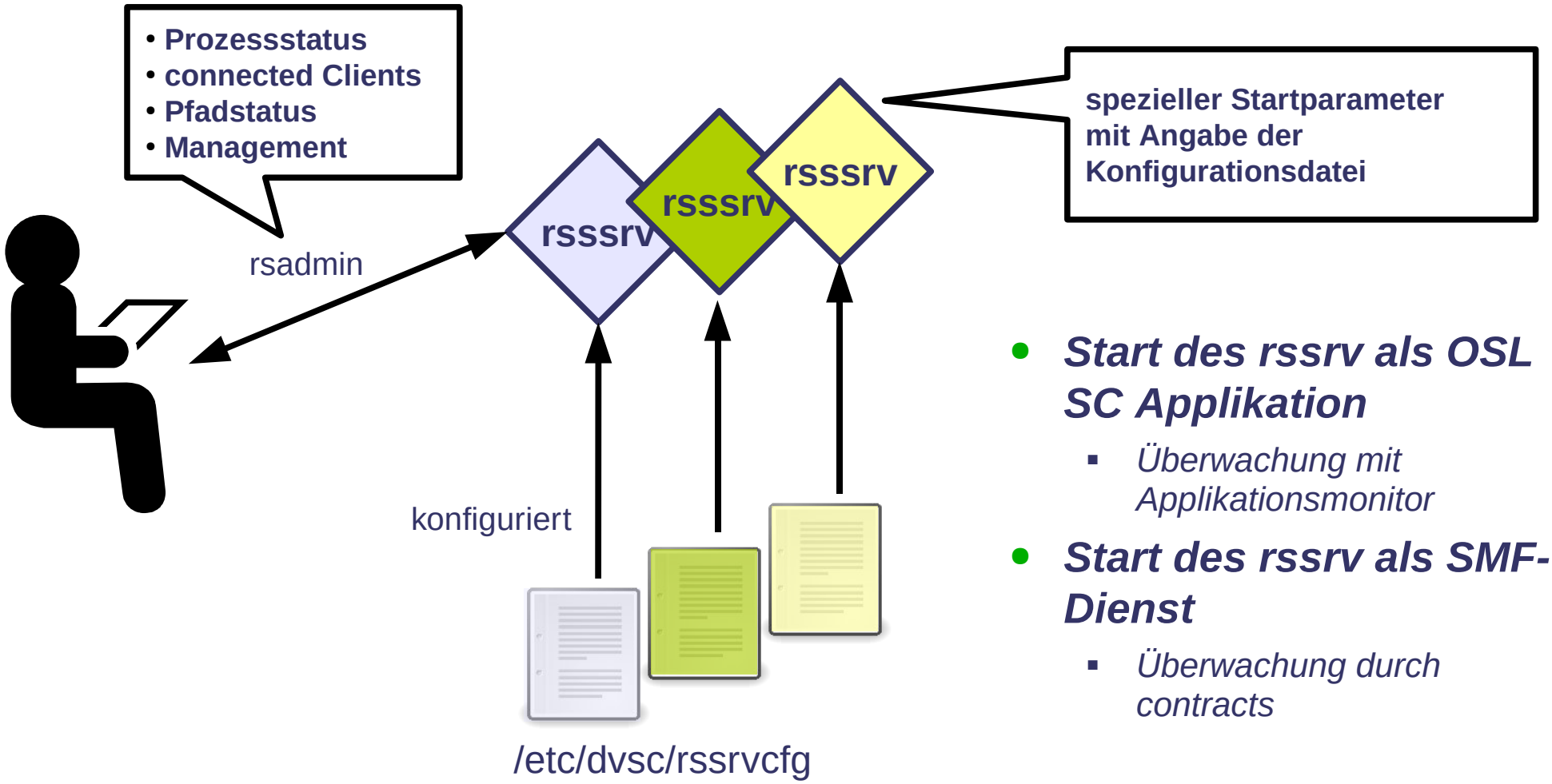
RSIO in der Praxis

Konfiguration und Betrieb des RSIO Servers




```
# cat /etc/dvsc/rssrvcfg

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

[server big-9]
serverid     = 1

[interface if1]
address      = 192.168.45.10

[interface if2]
address      = 192.168.46.10

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

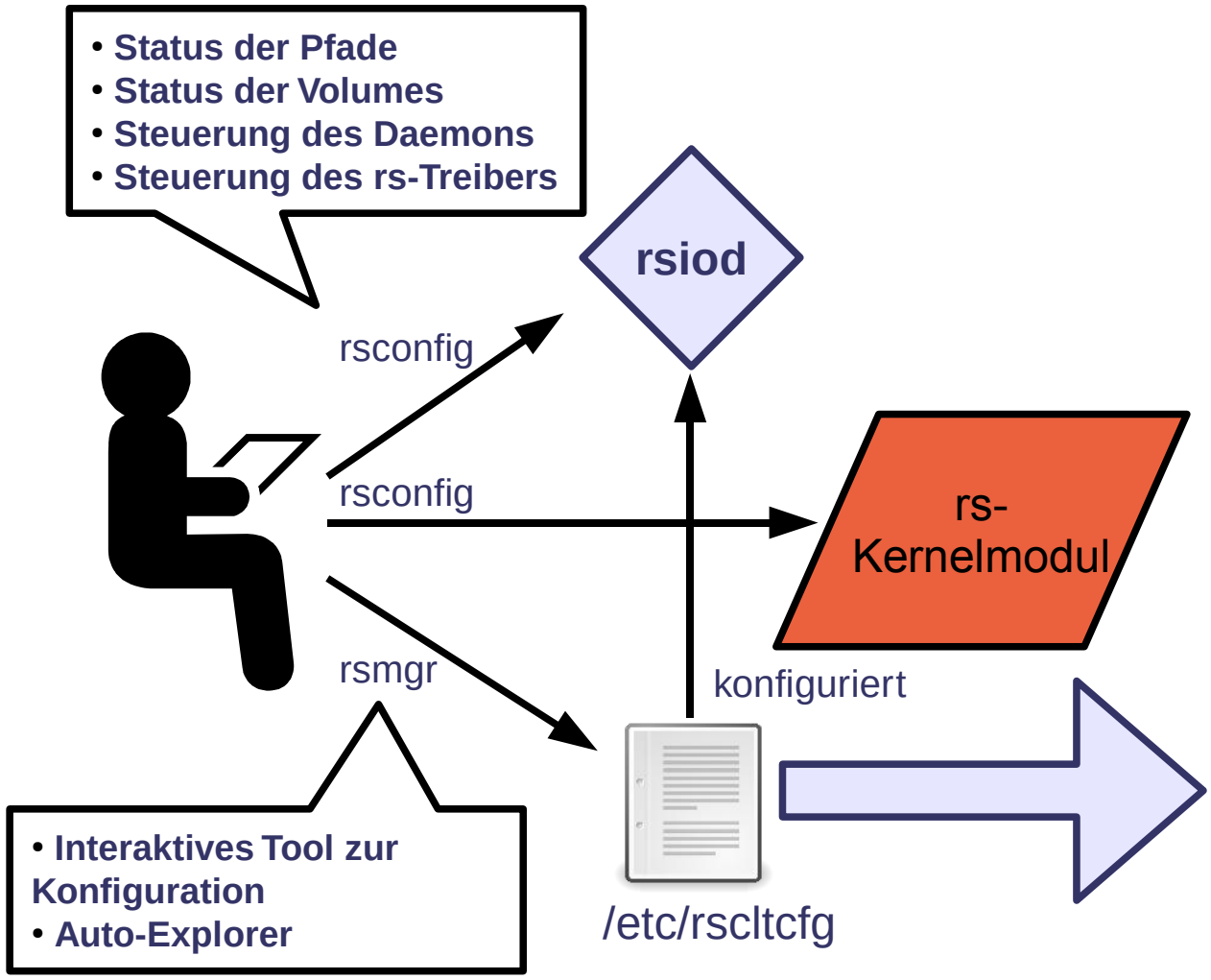
RSIO in der Praxis
Konfiguration und Betrieb des RSIO Servers



- ***Start des rssrv als OSL SC Applikation***
  - *Überwachung mit Applikationsmonitor*
- ***Start des rssrv als SMF-Dienst***
  - *Überwachung durch contracts*

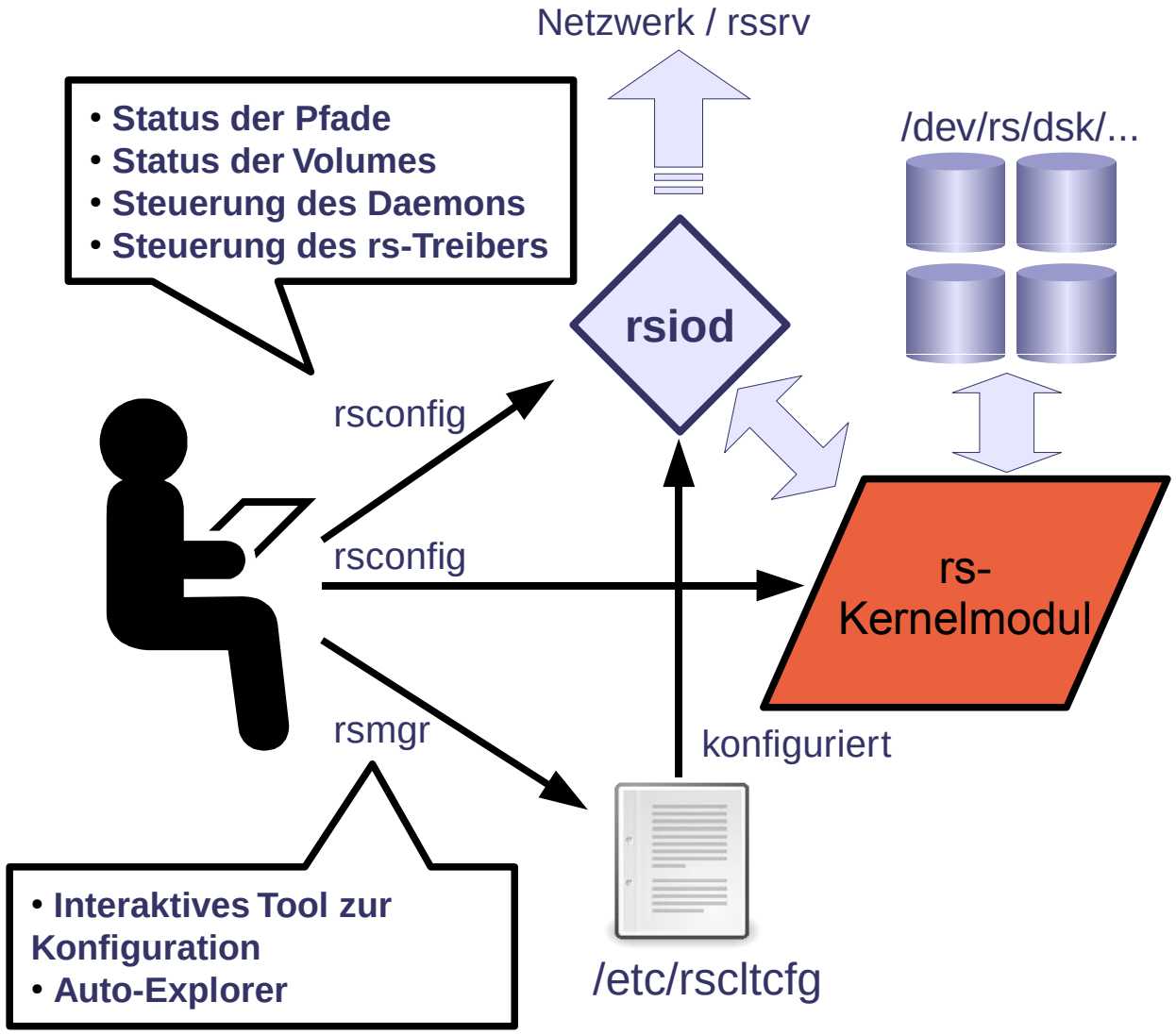
RSIO in der Praxis

Konfiguration und Betrieb des RSIO Clients




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

[defaults]
protocol     = tcp
port         = 5000

[interface if0]
address      = 192.168.45.20

[interface if1]
address      = 192.168.46.20

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

[server big-9-1]
address      = 192.168.45.10

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

RSIO in der Praxis
Konfiguration und Betrieb des RSIO Clients



- ***Start des rsiod als SMF-Dienst (Solaris)***
  - *Überwachung durch contracts*
- ***Starten des rsiod im Twin-Modus (Linux)***
  - *eigene Überwachung*

RSIO in der Praxis
Multipathing

- **RSIO Multipath benötigt keine OS- oder Switch-Features**
  - *RSIO ist Multilinkfähig und nutzt alle konfigurierten Verbindungen zu einem RSIO-Server*
  - *Bei verifiziertem Ausfall einer Verbindung wird diese geschlossen und der I/O über die verbleibenden Verbindungen fortgesetzt*
- **Restart des rsiod ist jederzeit möglich – auch bei laufendem I/O**
  - *Sicherheit gegen Hänger ...*

RSIO und OSL Storage Cluster

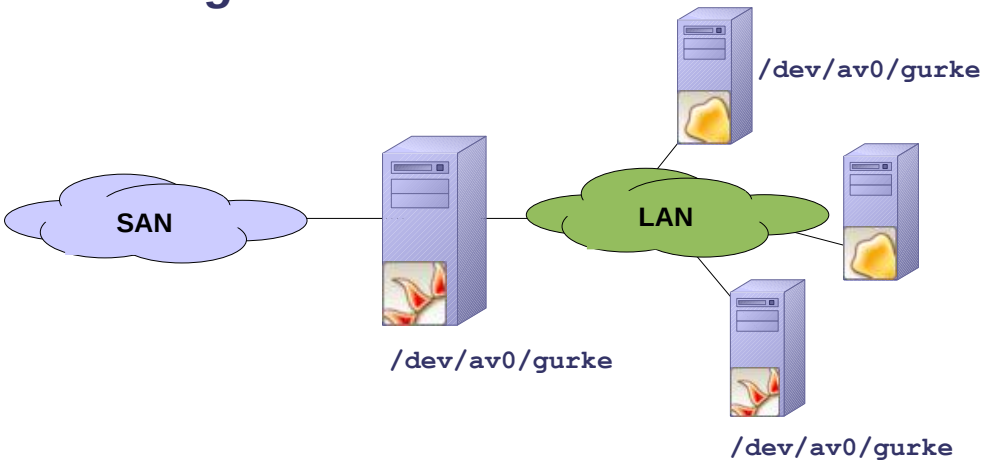
Integration in die Speichervirtualisierung

Technologietage 2011

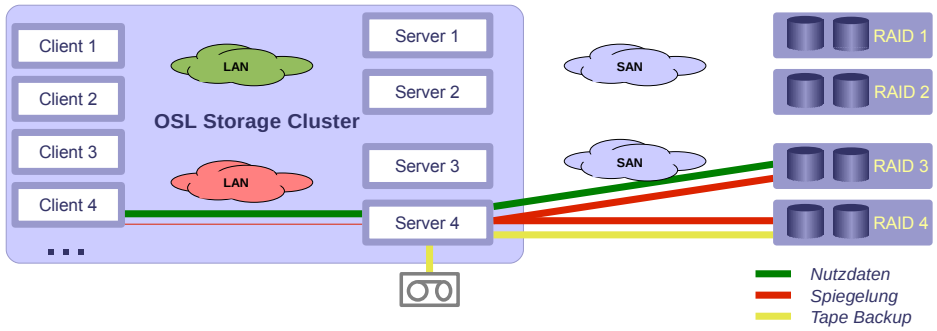
OSL Storage Cluster bietet hostbasierte Speichervirtualisierung, Clustering und diverse, speicherbezogene Zusatzfunktionen.

Diese Funktionen sind mit RSIO auch unter Linux verfügbar!

- *einheitlicher, plattformübergreifender Namensraum*
- *globaler Speicherpool auf allen RSIO Clients verfügbar*



- *Storageallokation ist sowohl vom RSIO Server als auch vom Client aus möglich*
- *XDM Operationen werden nur vom Client aus gesteuert, der Datentransfer geschieht auf dem Server*



OSL Gesellschaft für offene Systemlösungen mbH

www.osl.eu

RSIO und OSL Storage Cluster

Integration in die Speichervirtualisierung

- ***Speichermanagement ist sowohl vom RSIO Server als auch vom RSIO Client aus möglich***

  Beispiele:

  - *Anlegen von neuen Volumes*
  - *Clonen von Volumes*

- ***Auch mit RSIO ist die Storageadministration von jedem Node aus möglich.***

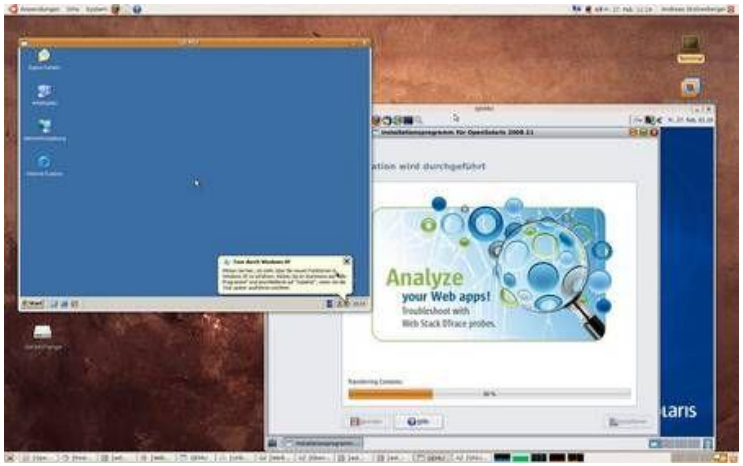
RSIO und OSL Storage Cluster
Integration in die Speichervirtualisierung

- **Mit RSIO und OSL Storage Cluster wird ein Cross-Plattform-Cluster möglich**
  - *Solaris und Linux in einem Cluster*
  - *Shared Storage Virtualization*
  - *gleiche Sichtweise auf Geräte erlaubt sogar Backup-Cluster*
  - *Einbinden von virtuellen Maschinen unter Linux*
    - *komplettes Management über die Clusterkommandos*
    - *Start, Stopp und Migration von VMs mit dem OSL-Applikationsbefehlssatz*
  - *Applikationen können sogar zwischen Solaris und Linux geschwenkt werden*
    - *Voraussetzung: gemeinsames Filesystem*

RSIO und OSL Storage Cluster

Virtuelle Maschinen im Cluster

- ***Eigentlich ist die Storageanbindung uninteressant – was zählt ist die Applikation***
  - *OSL Storage Cluster ACO ist auf RSIO Clients verfügbar (auch unter **LINUX**)*
- ***Und was ist mit Windows?***



Nachrichtenüberblick:

- ***Desktop-Virtualisierung von den KVM-Entwicklern***
- ***Ubuntu virtualisiert mit KVM***
- ***Red Hat kauft KVM-Entwickler***
- ***Kernel-Entwickler diskutieren über Xen***
- ***Red Hat Enterprise Linux 5.4 mit KVM als Beta erschienen***
- ***Red Hat Enterprise Virtualization 2.2 migriert VMs***
- ***Neues Frontend für KVM***
- ***IT-Schwergewichte unterstützen Virtualisierung mit KVM***

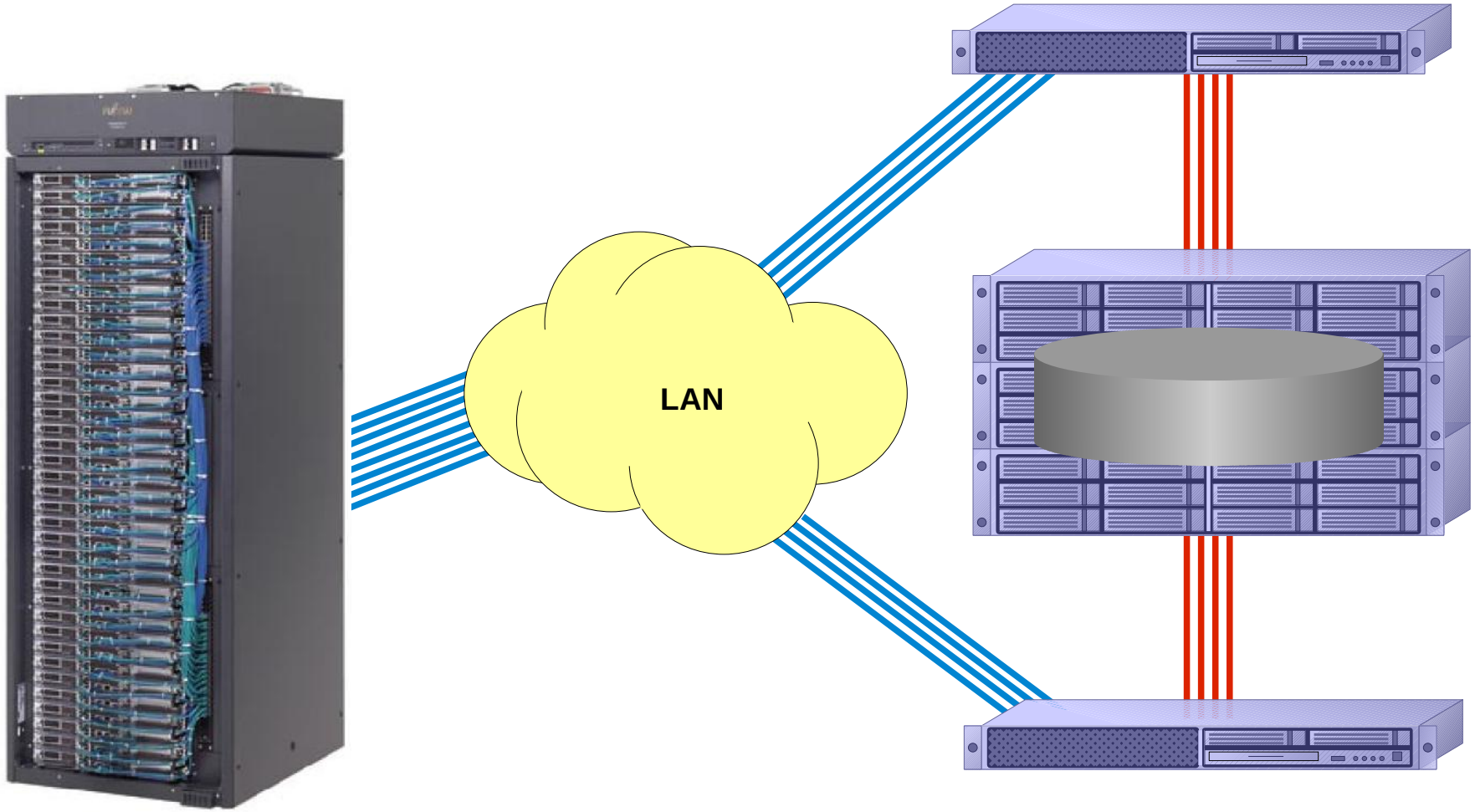
RSIO und OSL Storage Cluster

Virtuelle Maschinen im Cluster

- **KVM hat Zukunft**
  - *bevorzugte Virtualisierungslösung von RedHat und Novell*
- **Einfaches Management**
  - *libvirt stellt Schnittstellen bereit – auch für andere Virtualisierungslösungen*
  - *Verschiedene Managementoberflächen*
- **Intergration im OSL Storage Cluster**
  - *VM ist eine Applikation*
  - *Live-Migration zwischen den Clusternodes ist möglich*
  - *profitiert vom globalen Namespace*

RSIO und OSL Storage Cluster

Scale-Out-Infrastrukturen am Beispiel Fujitsu CX1000



RSIO und OSL Storage Cluster

Scale-Out-Infrastrukturen am Beispiel Fujitsu CX1000

- **Beispielsystem: Fujitsu Primergy CX1000**
  - *38-Node-Cluster mit bis zu 456 Prozessorkernen*
  - *ideal für VDI, Server VMs, Infrastruktur-as-a-Service , Software-as-a-Service, Cloud-Architekturen*
  - *mit OSL RSIO und OSL Storage Cluster ist eine ideale Lösung für die Speicheranbindung, Nodeüberwachung und Anwendungssteuerung vorhanden*
- **Ausblick auf die Möglichkeiten**
  - *Ab Storage Cluster 4.0: Virtual Nodes und Virtual Maschine Applications*
    - *Applikationen können auch auf virtuellen Nodes gestartet werden*
    - *Migration von Applikationen in Zonen oder VMs*
    - *Virtual Node ist selbst eine Applikation, die auf physikalischen Nodes gestartet werden kann*
  - *KVM Cluster mit Toolset für eine einfache Bereitstellung, Migration, Backup und Recovery von VMs*