RSIO - Storage Networking der nächsten Generation

SNW Europe 2010 – Hands-On-Lab
RSIO - Data Center Block I/O over Ethernet

RSIO, das neue Protokoll von OSL für Block-I/O über Standard-LAN-Infrastrukturen, zielt auf die Anforderungen von Rechenzentren: Virtualisierter Speicher über LAN, skalierbarer, hoher Durchsatz und hohe Verfügbarkeit, Unterstützung für parallele und geclusterte Infrastrukturen und mulithtreaded Connections, die von modernen CPU-Architekturen profitieren. Erfahren Sie Hintergründe zur Technologie und verschaffen Sie sich einen Überblick zu den vielfältigen Funktionen und Möglichkeiten, zur einfachen Administration, zu Einsatzszenarien und handfesten Vorteilen ...

Bert Miemietz &
Christian Schmidt

OSL Gesellschaft für
offene Systemlösungen mbH

Programm für das Hands-on-Lab

1. **Einführung in RSIO**
   - *Warum ein neues Storage-Protokoll?*
   - *Funktionsweise von RSIO*
   - *Einsatzszenarien RSIO*
   - *Performance-Eckdaten*
2. **Konfiguration und Betrieb**
   - *Server konfigurieren und starten*
   - *Einrichten der Clients*
3. **Umgang mit den Devices**
   - *Globaler Namensraum*
   - *Shared Storage*
4. **Erweiterte Funktionalitäten**
   - *Globales Volumemanagement*
   - *Vollintegrierte Clusterumgebungen*

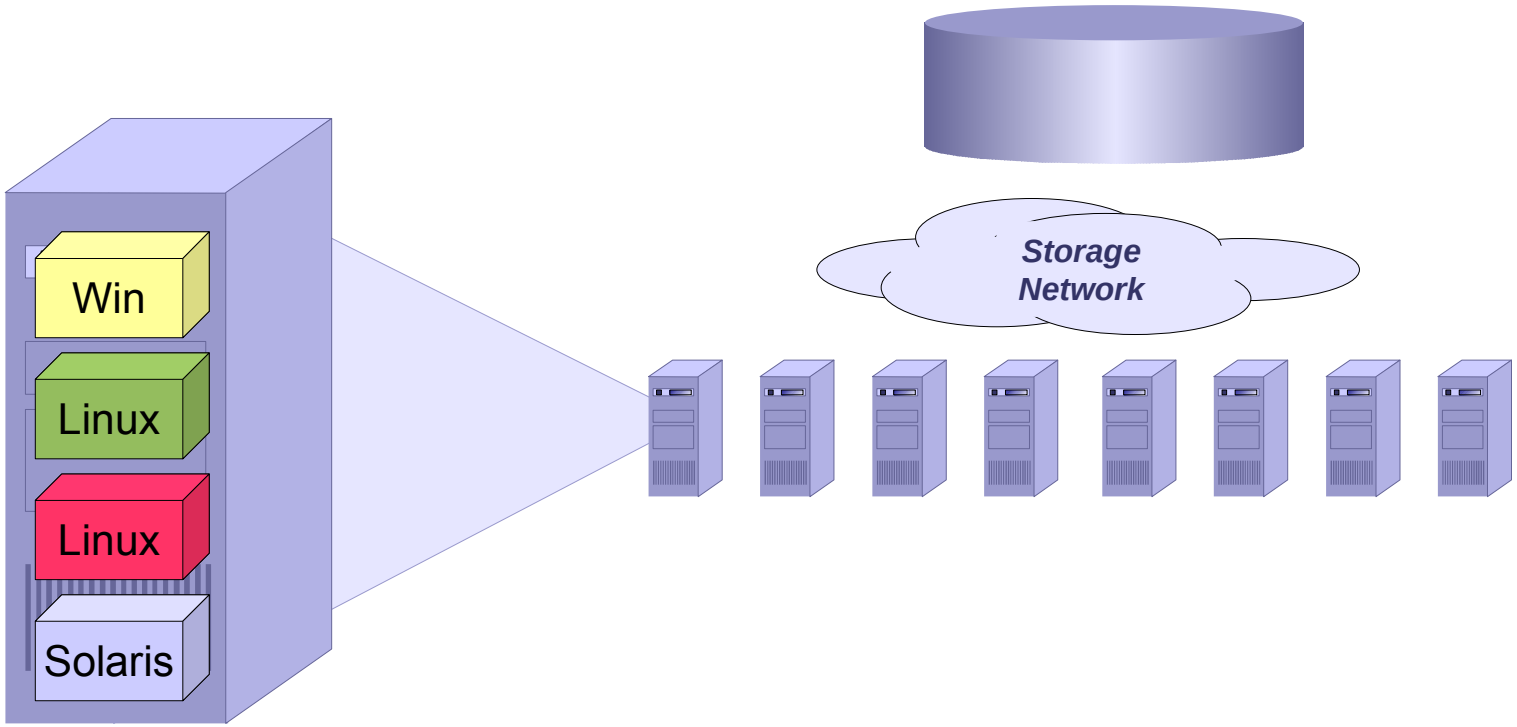
Warum Storage über Ethernet?
Anforderungen und Möglichkeiten

- **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?

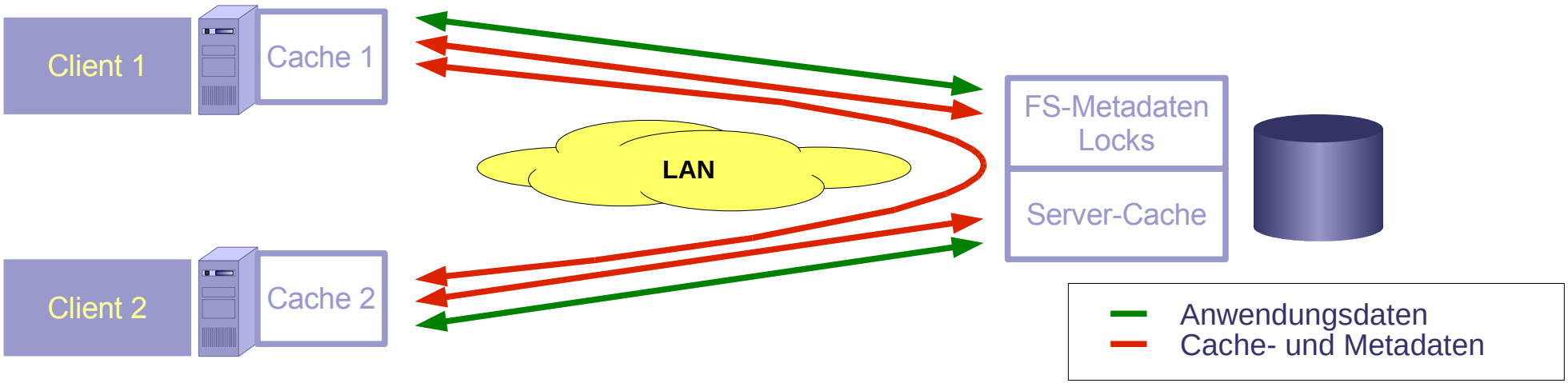
Noch ein ganz wichtiger Punkt ...



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

Storage über Ethernet heute: NFS, SMB, CIFS

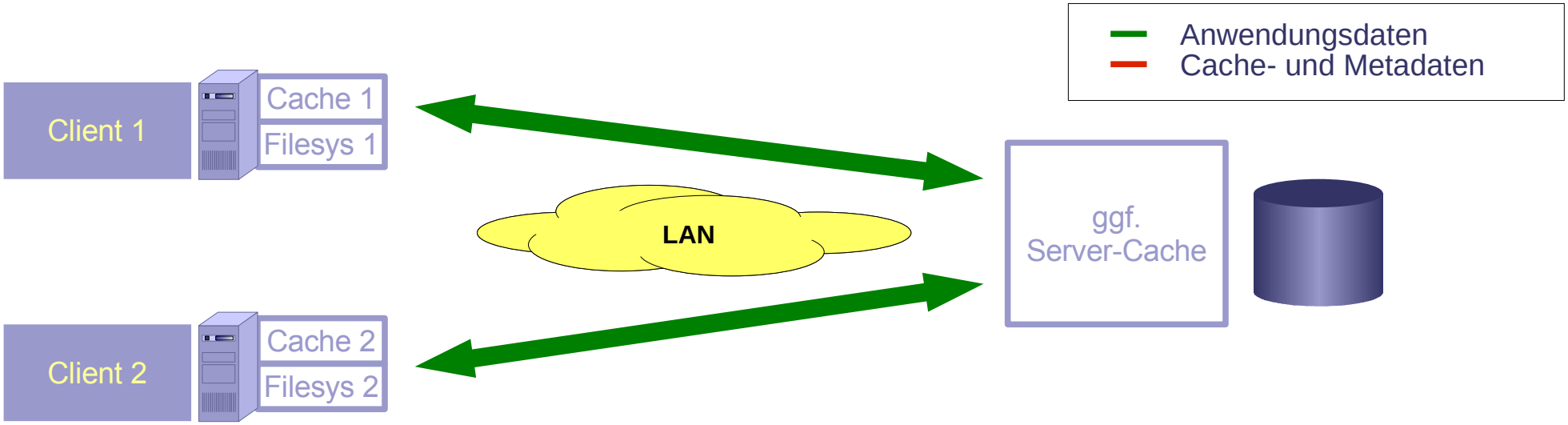
- ***Was bietet NAFS***
  - *Spezialisierung auf Fileservices, dafür relativ einfache Handhabung*
  - *Filesharing*
  - *Keine komplexen RAID Funktionen*
  - *dateisystemtypische Funktionen wie Snapshots*
  - *Weite Verbreitung und Unterstützung der Protokolle*
- ***Die Kehrseite***
  - *Aufwendige Integration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

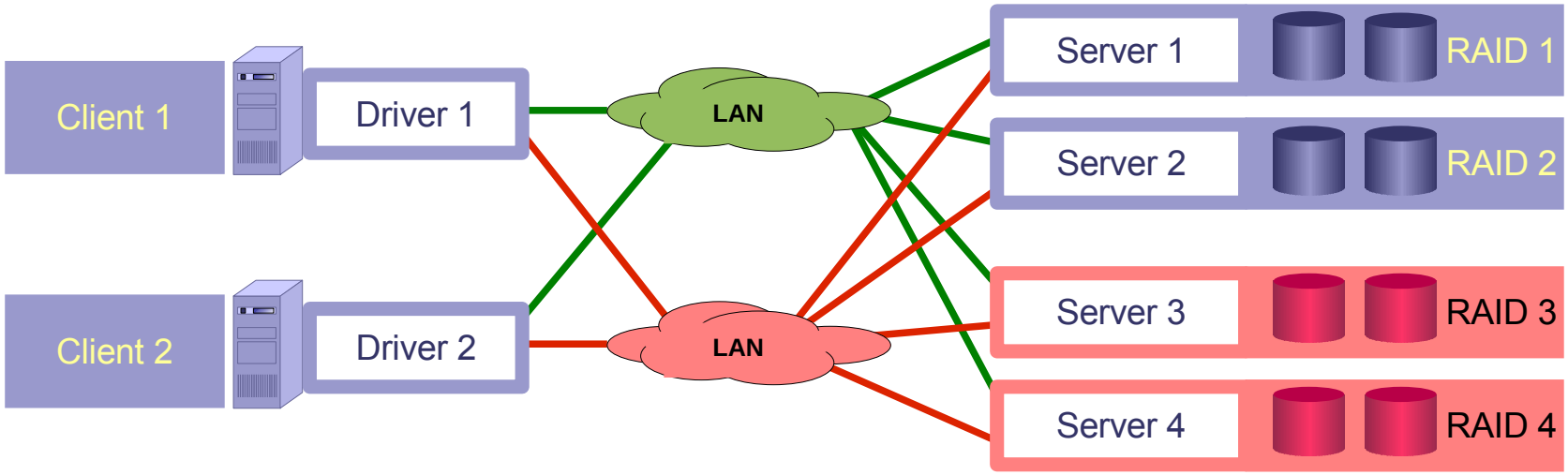
Jenseits von Filesharing überwiegen die Vorteile

- *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*



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*

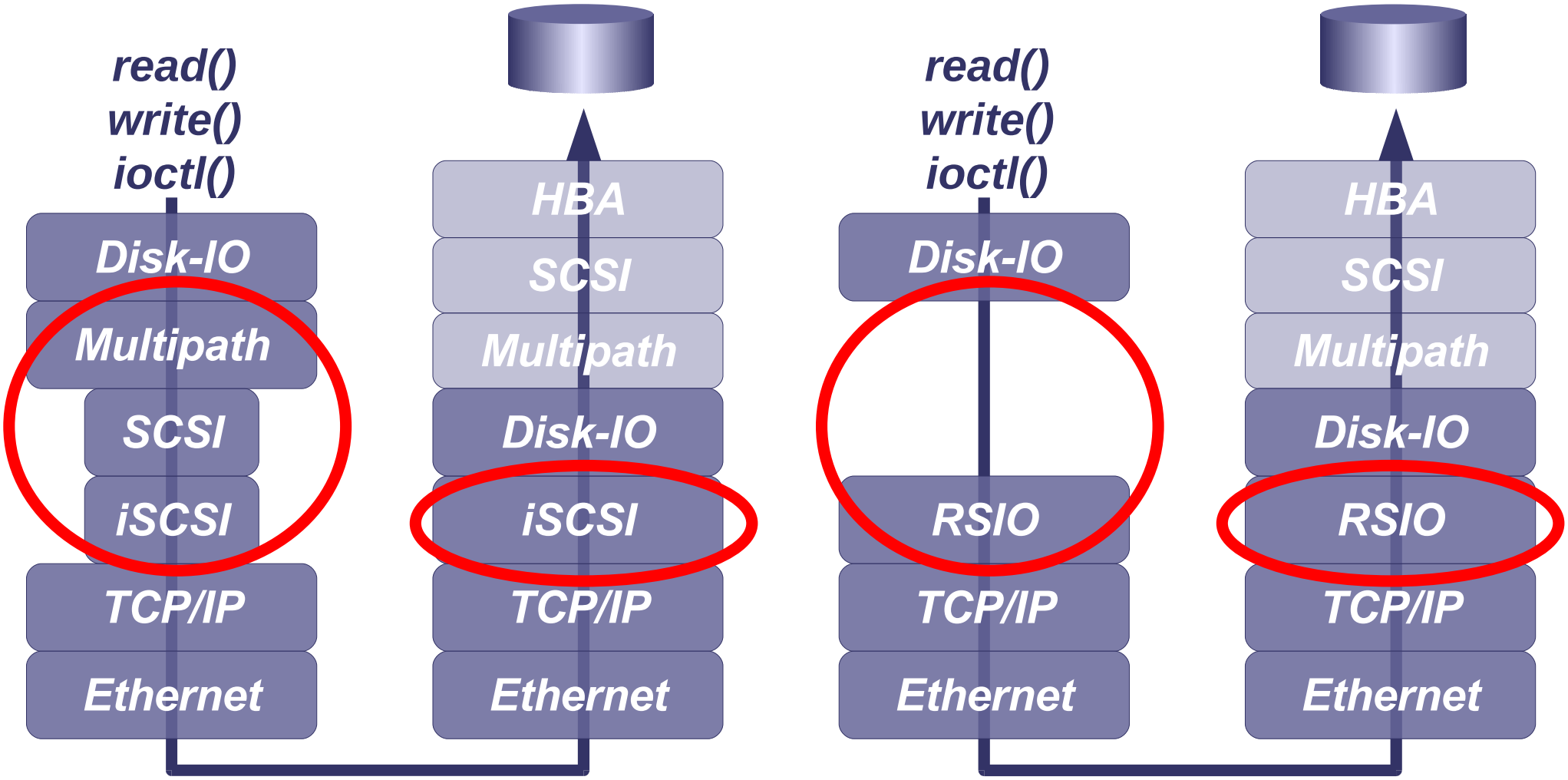
RSIO - Remote Storage I/O

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*

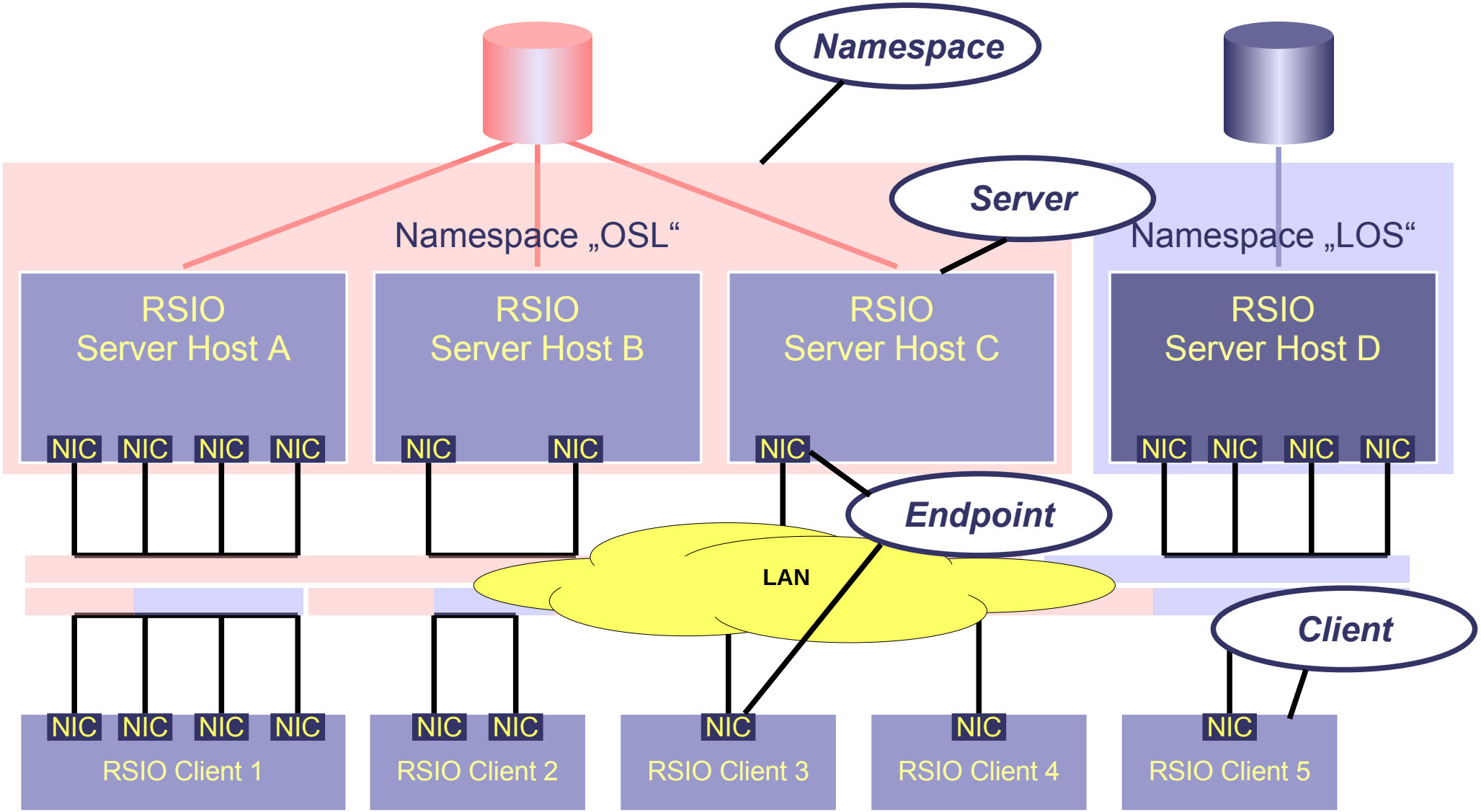
RSIO - Remote Storage I/O

Vergleich der Protokollstacks



RSIO – Architektur im Überblick

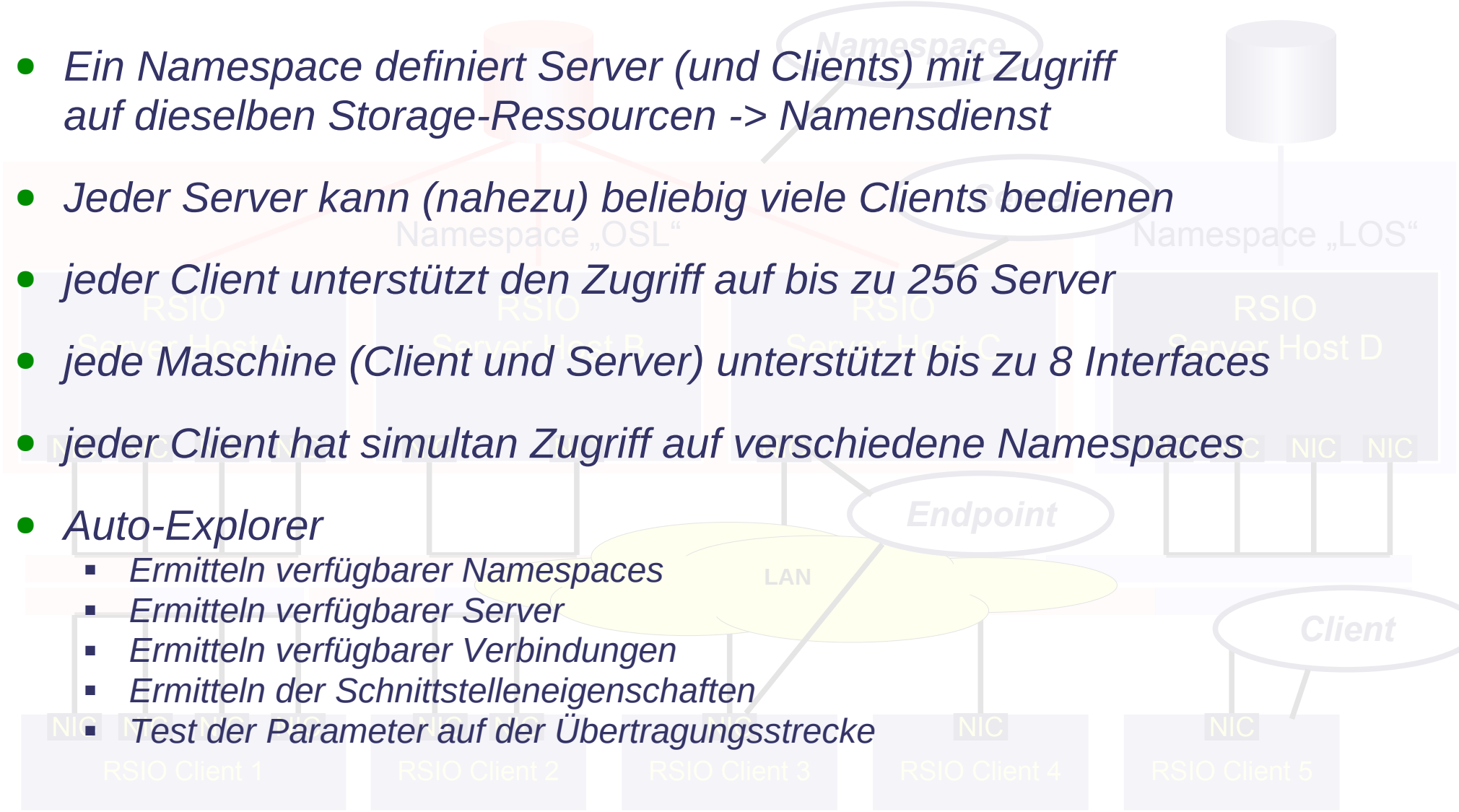
Klar gegliedertes und flexibles administratives Konzept



Parameter der RSIO-Architektur

Flexible Client-Server-Implementierung

- *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

Vergleich der Darstellung von Ressourcen auf dem Client

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***

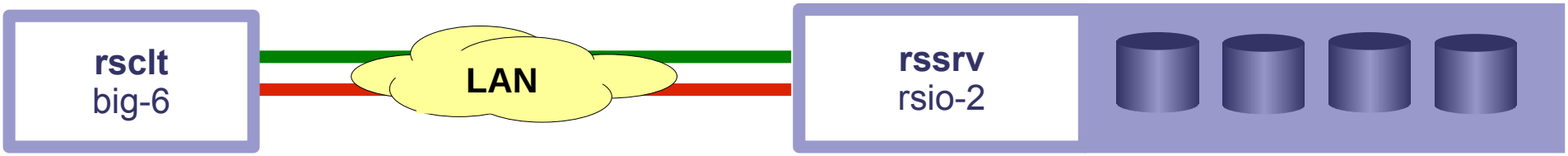
Client-Performance Throughput

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***

Erste Demonstration
Unser Testaufbau

Fujitsu-Siemens
Primergy RX-100

Fujitsu
Esprimo E7935



- *1 x Dual-Core CPU*
- *nur Root-Disk*
- *2 x 1GBit LAN*
- *Solaris*
- *RSIO Client*

- *1 x Quad-Core CPU*
- *4 x 2TByte ECO-PC-Disk 5400rpm*
- *2 x 1 Gbit LAN*
- *Solaris*
- *OSL Storage Cluster + RSIO Server*

Übung zum Mitmachen

RSIO-Client unter:

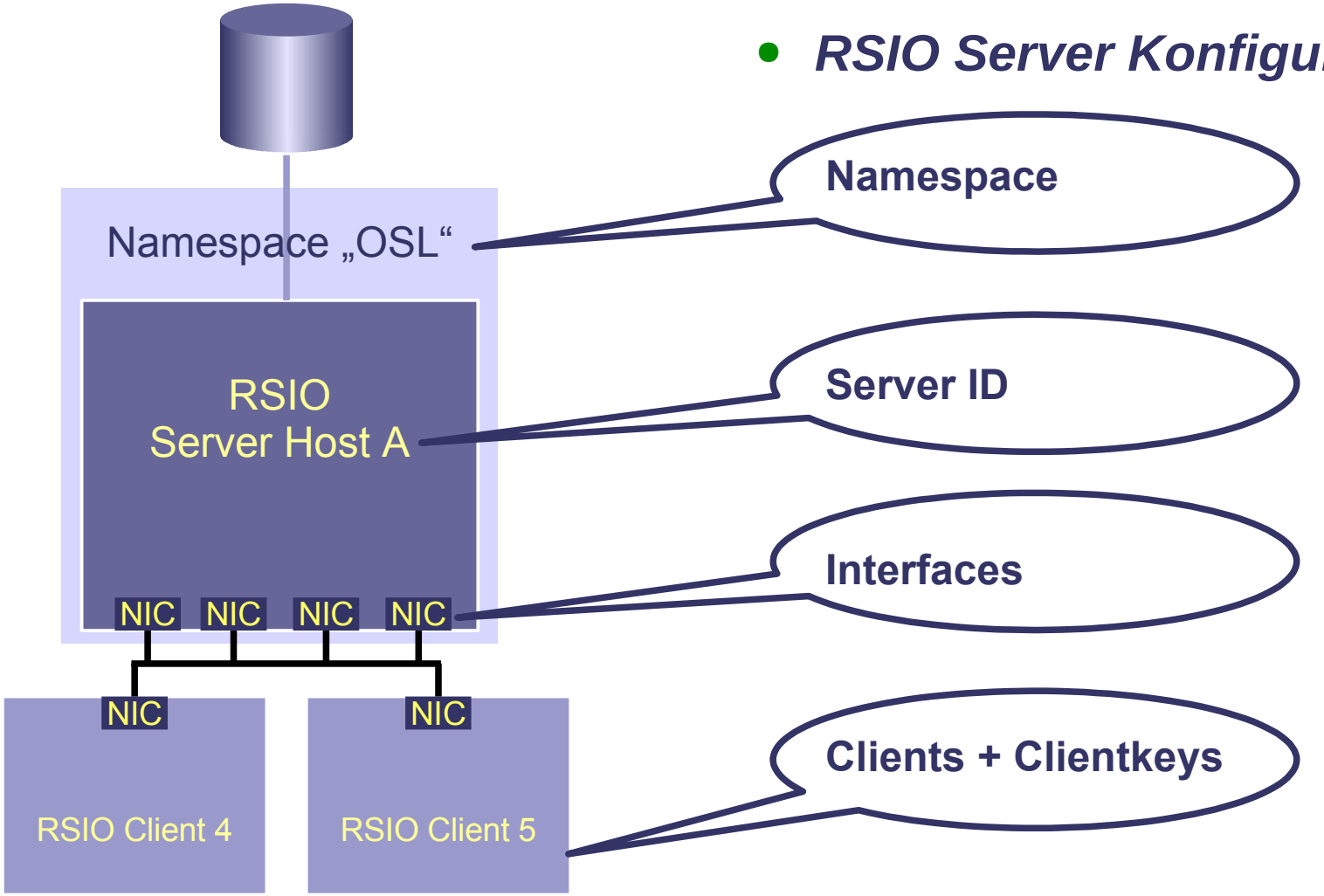
SLES11

OpenSuSE 11.3

RHEL 5.5

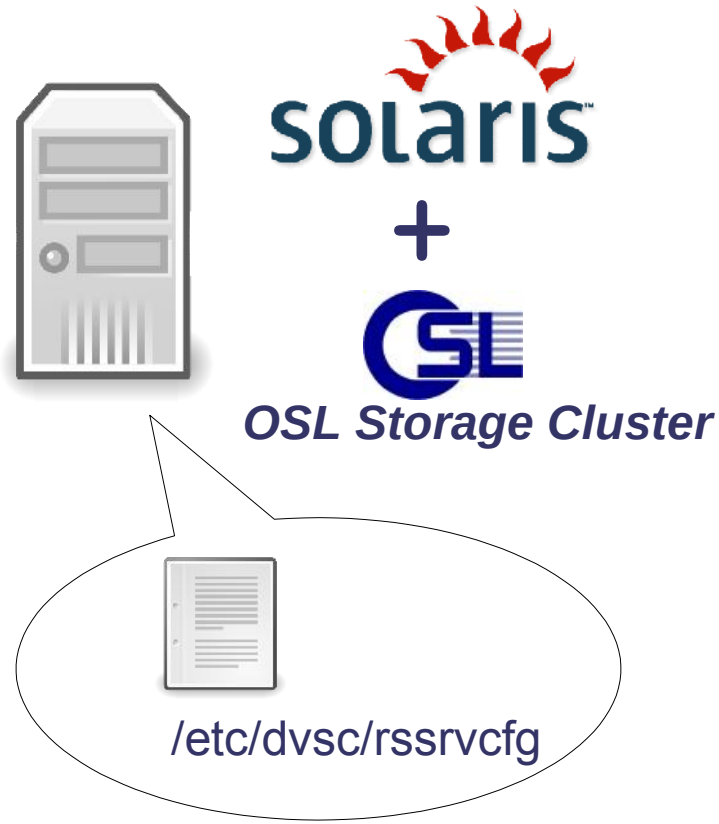
Solaris 10

RSIO – Konfiguration und Betrieb
Server Konfiguration



- *RSIO Server Konfiguration*

RSIO – Konfiguration und Betrieb
Server Konfiguration



- ***RSIO Server***
  - *Solaris Server*
  - *benötigt derzeit OSL Storage Cluster*
  - *stellt vollständig virtualisierten Storage zur Verfügung*
- ***Server Prozess***
  - ***`rssrv`***
  - *Konfiguration über eine einfache Textdatei*
  - *Ein Listener-Prozess und ein Prozess für jeden Client*
- ***Logmeldungen des Servers***
  - ***`rsslogcat`***

RSIO – Konfiguration und Betrieb

Server Konfiguration

```
# 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
```

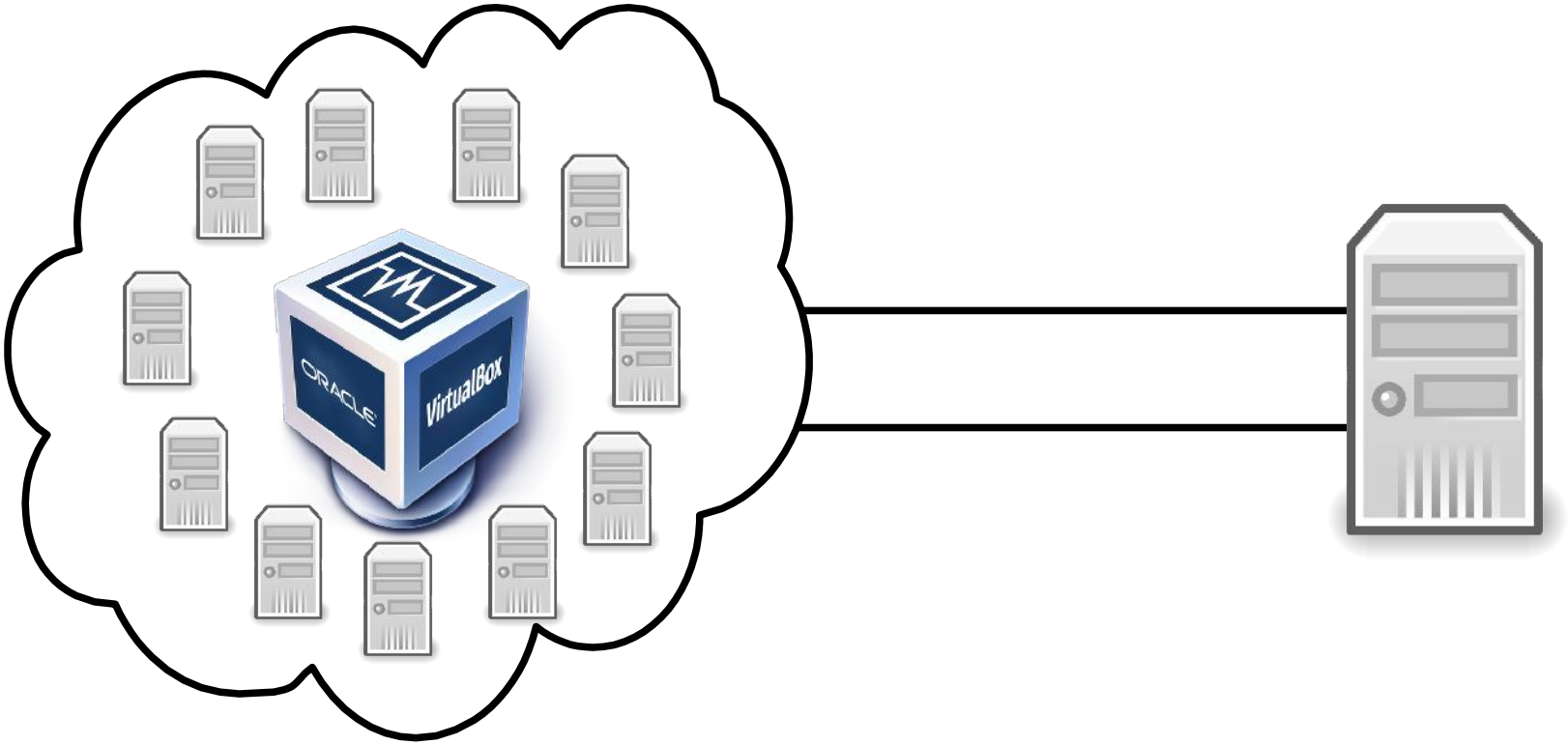
Namespace Konfiguration
Name, ID und Standardeinstellungen

Server Name und ID

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

Client Sektion
Ein Eintrag für jeden Client (Name und Schlüssel)

RSIO – Konfiguration und Betrieb
Hands-On Lab Aufbau



- *VirtualBox Server mit 15 VMs*
- *5x SLES11, 4x OpenSuse, 4x Red Hat, 2x Solaris*
- *3 GigaBit-Interfaces pro VM, 2 für RSIO*

- *RSIO Server "rsio-1"*
- *Solaris 10 mit OSL Storage Cluster*
- *2 Application Volumes pro VM*

RSIO – Konfiguration und Betrieb
Client Konfiguration

- **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*
- **Konfigurationsdatei: `/etc/rscltcfg`**
  - *Lesbare ASCII Konfigurationsdatei*
  - *Konfigtool: `crsio`*

RSIO – Konfiguration und Betrieb
Client Konfiguration

```
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
```

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

Interface Sektion
Alle Clientinterfaces die von rsiod genutzt werden sollen.

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

Server Interfaces
Pro Serverinterface ein Abschnitt.

RSIO – Konfiguration und Betrieb
Client Konfiguration

- **Hands-On!!!**
  - Jeder Teilnehmer erhält eine eigene VM
  - Konfiguration und Inbetriebnahme von RSIO
    - Einrichten der Konfiguration mit crsio
    - Starten des rsio-Daemons
    - Handhabung von rsconfig
    - Anzeigen der Log-Meldungen
    - Attach und Detach der Devices
  - Nutzen von RSIO Devices
    - Erstellen und Mounten eines Filesystems
    - Raw-Device IO

RSIO – Konfiguration und Betrieb
Client Konfiguration

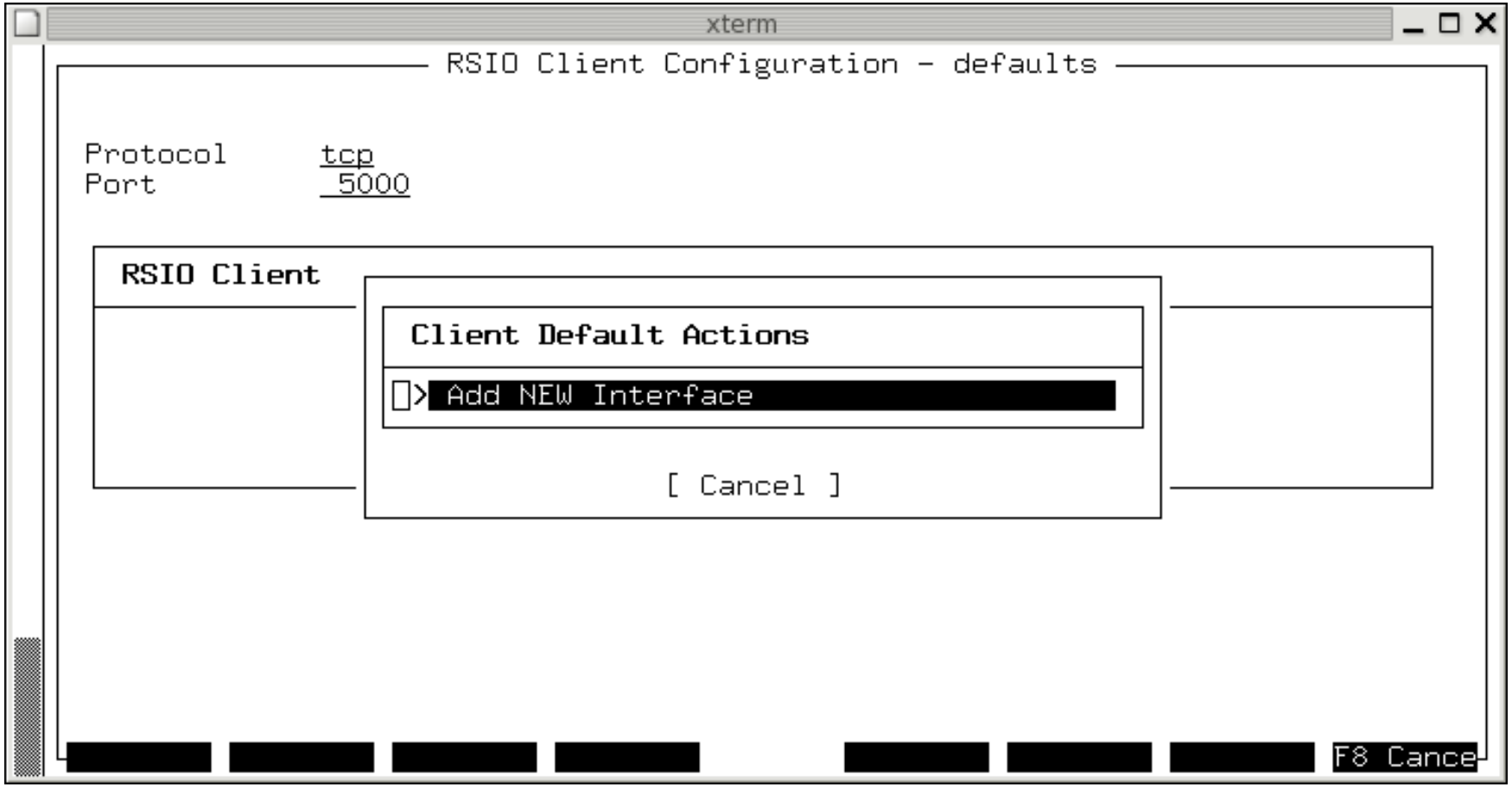
- ***Virtuelle Maschinen***
  - *Suse Enterprise Linux Server 11*
    - *sles11-1 – sles11-5*
  - *OpenSuse 11.3*
    - *osuse-1 – osuse-4*
  - *RedHat Enterprise Linux*
    - *redhat1 – redhat4*
  - *Solaris 10 Update 8*
    - *sol10-2 und sol10-3*
- ***Anmelden über putty***
  - *Username: root Passwort: root*

RSIO – Konfiguration und Betrieb
Client Konfiguration

- ***1. Schritt: Anlegen der Clientkonfiguration***
  - *Welche Interfaces sollen genutzt werden?*
  - *Welches Protokoll und welcher Port sollen die Clients nutzen?*
  - *Gibt es interfacespezifische Abweichungen beim Port oder Protokoll*
- ***Konfiguration erfolgt mit dem Curses-Tool `crsio`***
  - *Interfaces:* ***192.168.10.x*** *und* ***192.168.20.x***
  - *Protokoll:* ***tcp***
  - *Port:* ***5000***
  - *Alle Interfaces nutzen die* ***default-Einstellungen***

RSIO – Konfiguration und Betrieb

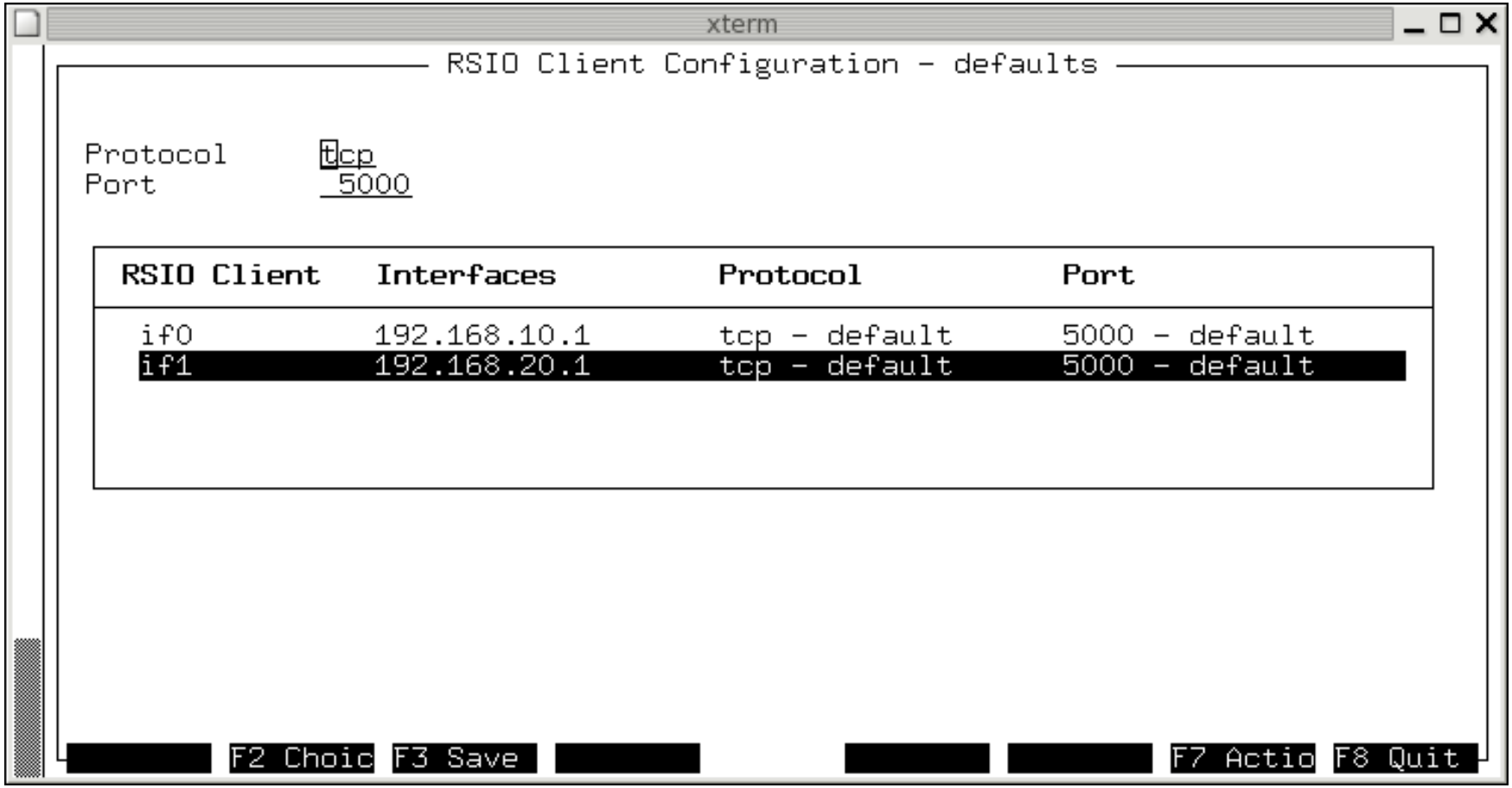
Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 1: Hinzufügen der Clientinterfaces aus den Netzen 192.168.10.x und 192.168.20.x

RSIO – Konfiguration und Betrieb

Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 2: Abspeichern der Clientkonfiguration mit [F3 Save]

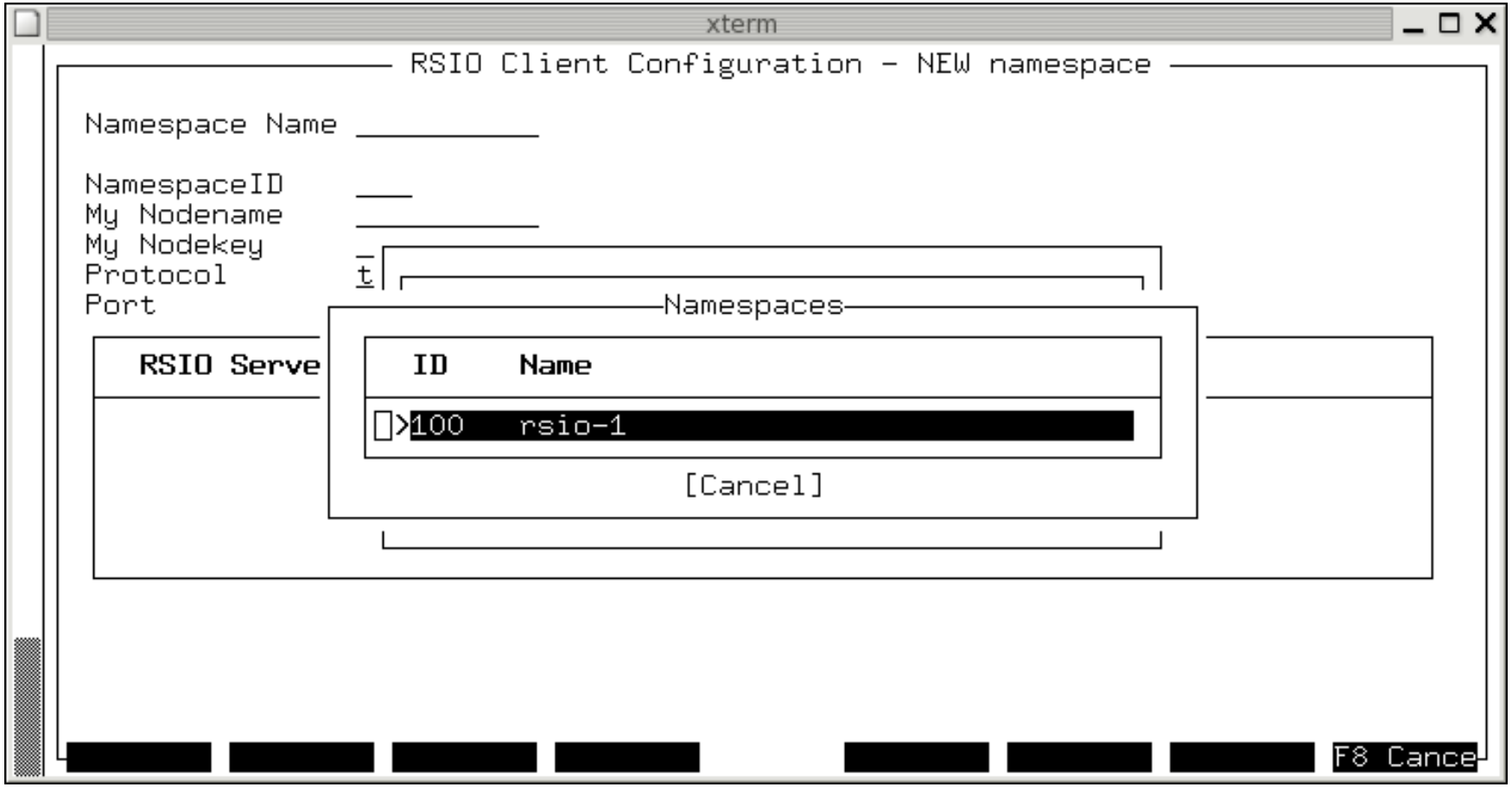
RSIO – Konfiguration und Betrieb
Client Konfiguration – Erstellen der Konfiguration mit crsio

- ***Speicher wird in Namespaces gruppiert***
- ***Alle Server im selben Namespace sollen auf den gleichen (shared) Storage Zugriff haben***
- ***Clients können sich in mehreren Namespaces registrieren***

- ***Einrichten des Namespace für das Hands-On Lab***
  - *Über den Menüpunkt "Add NEW Namespace" eine neue Namespacekonfiguration starten*
  - *Nach verfügbaren Namspaces scannen ([F7 Action] Find NEW Namespace)*
  - *Clientidentifikation ausfüllen*
  - *Nach neuen Namespaceservern scannen ([F7 Action] Find New Namespace Servers)*

RSIO – Konfiguration und Betrieb

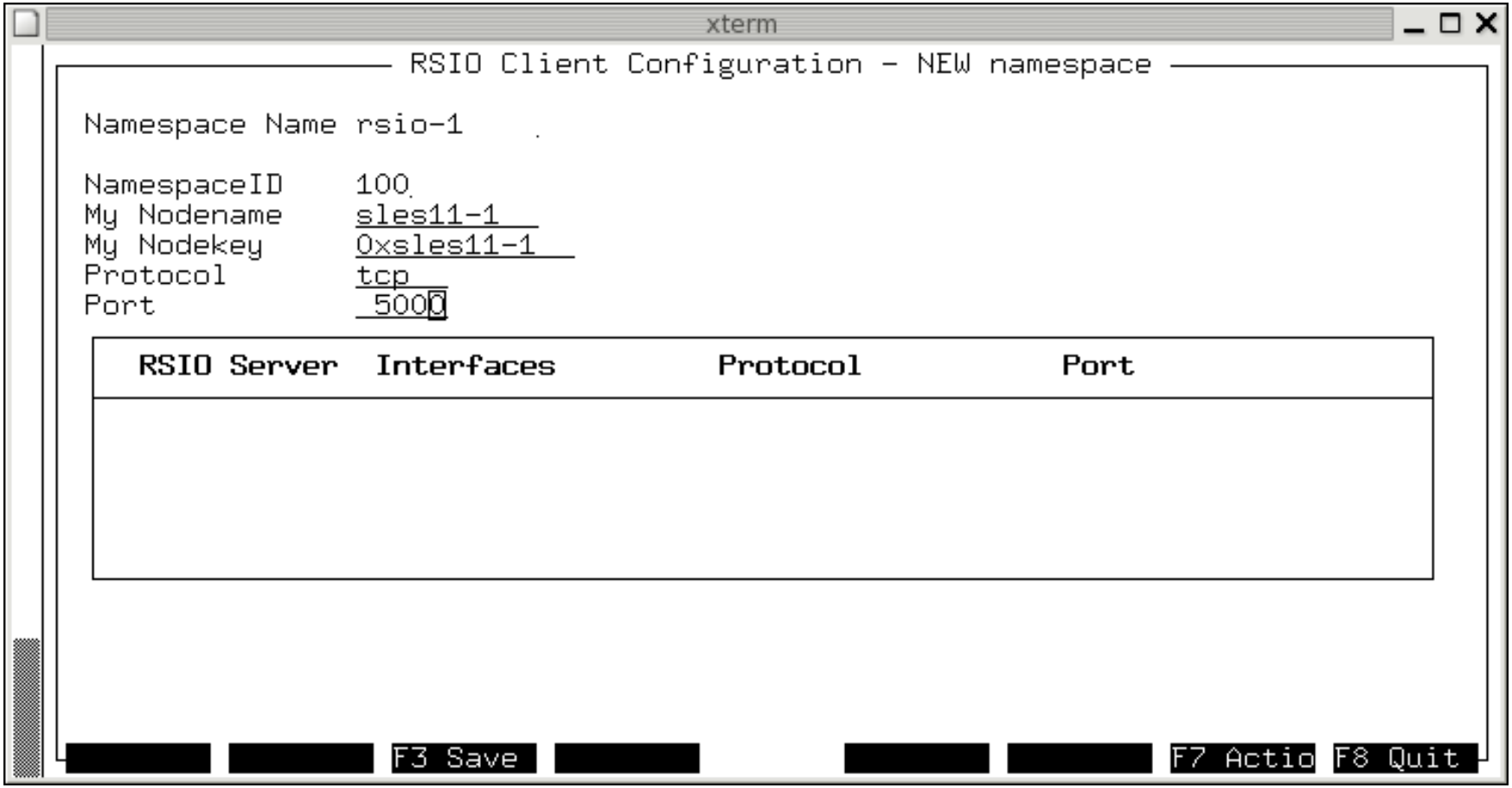
Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 3: Anlegen eine neuen Namspaces ([F7 Action] -> Find NEW Namespace)

RSIO – Konfiguration und Betrieb

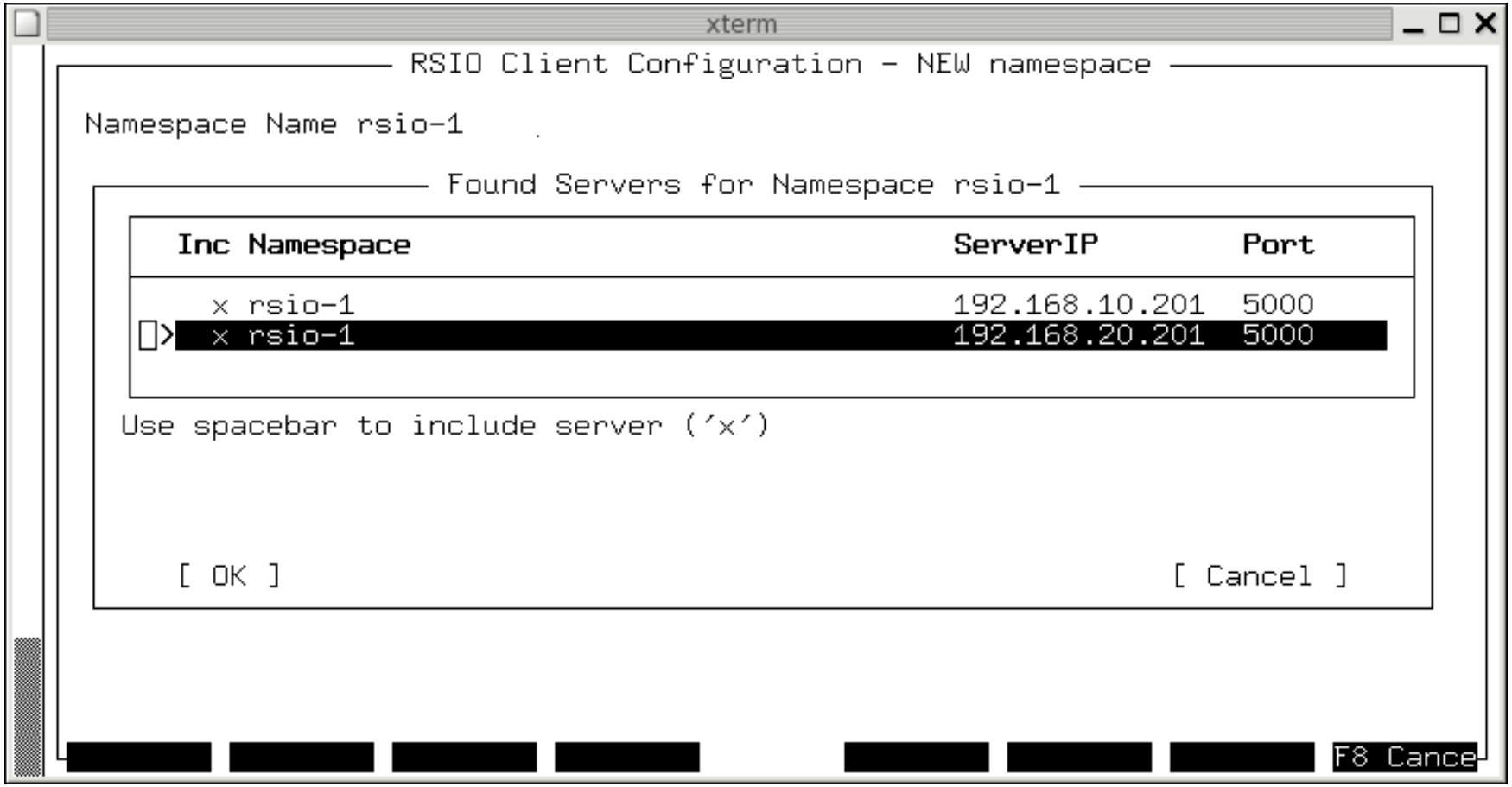
Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 4: Namespace Anmeldedaten eingeben

RSIO – Konfiguration und Betrieb

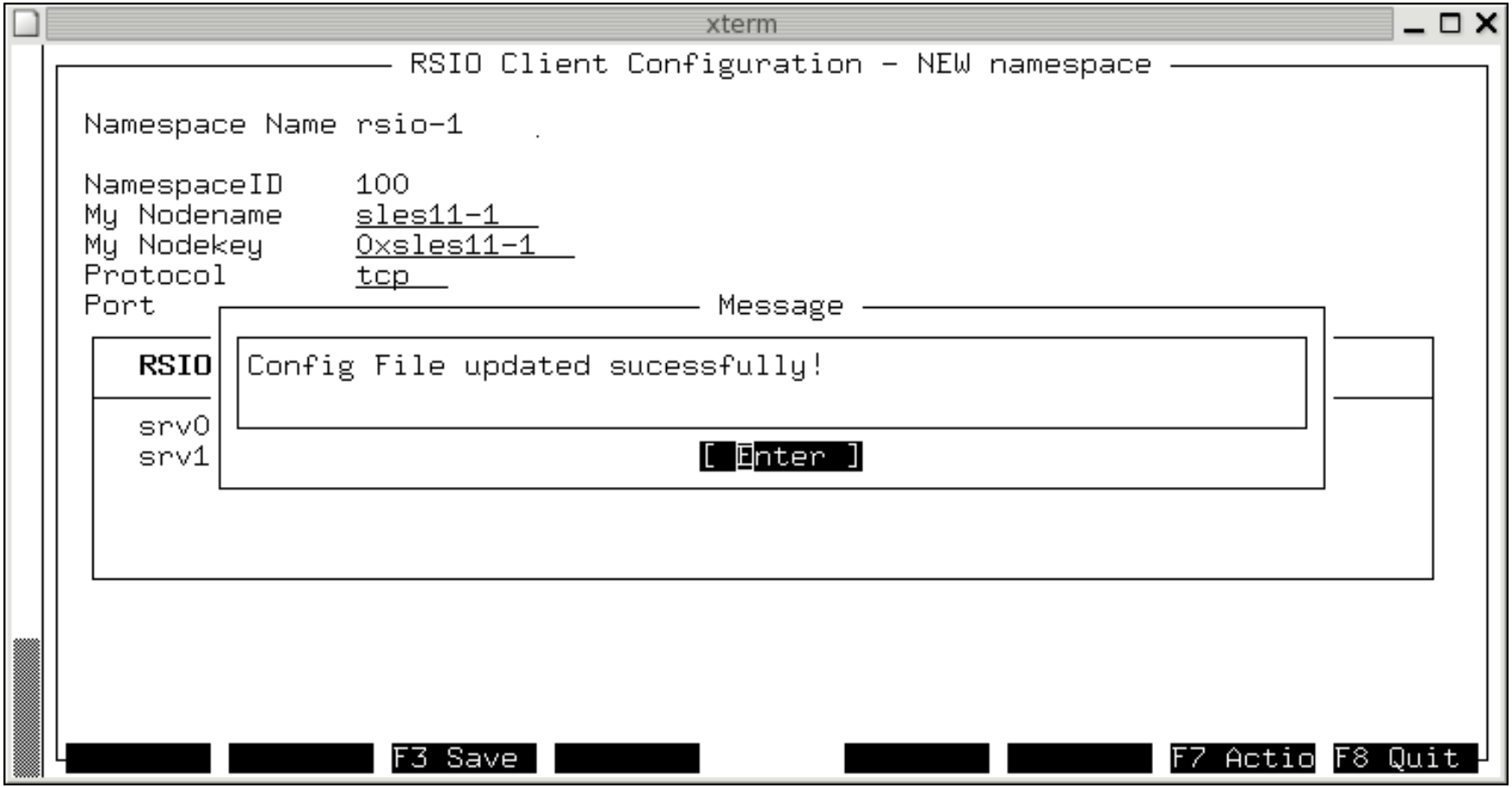
Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 5: Namespace Server finden ([F7 Action] Find NEW Namespace Server)

RSIO – Konfiguration und Betrieb

Client Konfiguration – Erstellen der Konfiguration mit crsio



Schritt 6: Abspeichern der Konfiguration mit [F3 Save]

RSIO – Konfiguration und Betrieb
RSIO In Betrieb nehmen

- **Die Konfiguration ist nun abgeschlossen**
  - Die Konfigurationsdatei liegt unter **`/etc/rscltcfg`**
- **Zum Nutzen von RSIO muss der RSIO-Daemon rsiod gestartet werden**
  - Aufruf #**`rsiod`**
- **Die Arbeit des Daemon wird geloggt**
  - Ausgabe mit **`rsclogcat [-f]`**
  - 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 (rs-Treiber) schreibt Logmessages in syslog

RSIO – Konfiguration und Betrieb
RSIO In Betrieb nehmen

- ***Zentrales Administrationskommando: `rsconfig`***
- ***Anzeigen der Netzwerk Multipath Informationen***

```
sles11-1# rsconfig -m status
0 if0 IP(TCP) 192.168.10.1/5000
    rep: 000 IP(TCP) 192.168.10.9/5000 connected  tx: ok rx: ok
1 if1 IP(TCP) 192.168.20.1/5000
    rep: 001 IP(TCP) 192.168.20.9/5000 connected  tx: ok rx: ok
```

  - *Multipathfunktionalität ist zentraler Bestandteil des RSIO Protokolls*
  - *Automatische Lastverteilung über die verfügbaren Kanäle*
  - *Automatisches Failover bei Pfadausfällen*
  - *Zusätzlicher IP Multipath oder Storage Multipath ist nicht notwendig*

RSIO – Konfiguration und Betrieb
RSIO In Betrieb nehmen

- ***Anzeigen der verfügbaren Volumes***

```
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
```

  - *Diese Volumes sind über die konfigurierten Namspaces und Server verfügbar*
  - *Detaillierte Ausgabe mit den Schaltern -q[[[v]v]v]*

- ***Lokales Anlegen und Entfernen der Deviceknoten***

```
sles11-1 # rsconfig -a    # Attach der Volumes
sles11-1 # rsconfig -d    # Detach der Volumes
```

- ***Attach der Volumes***
  - *Alle Volumes werden per default unter /dev/av[0-3] angelegt*
    - *Pfad der Volumes wird vom Server vorgeschlagen*
    - *Anzeigen der lokalen RSIO Volumes mit* ***rsconfig***

```
sles11-1# rsconfig -lvv
osl:oracle@0                                    31457280 blocks,   1 server(s)
    c: /dev/av0/roracle
    b: /dev/av0/oracle

osl:RSIO_oracle@0                               20971520 blocks,   1 server(s)
    c: /dev/av0/rRSIO_oracle
    b: /dev/av0/RSIO_oracle
```

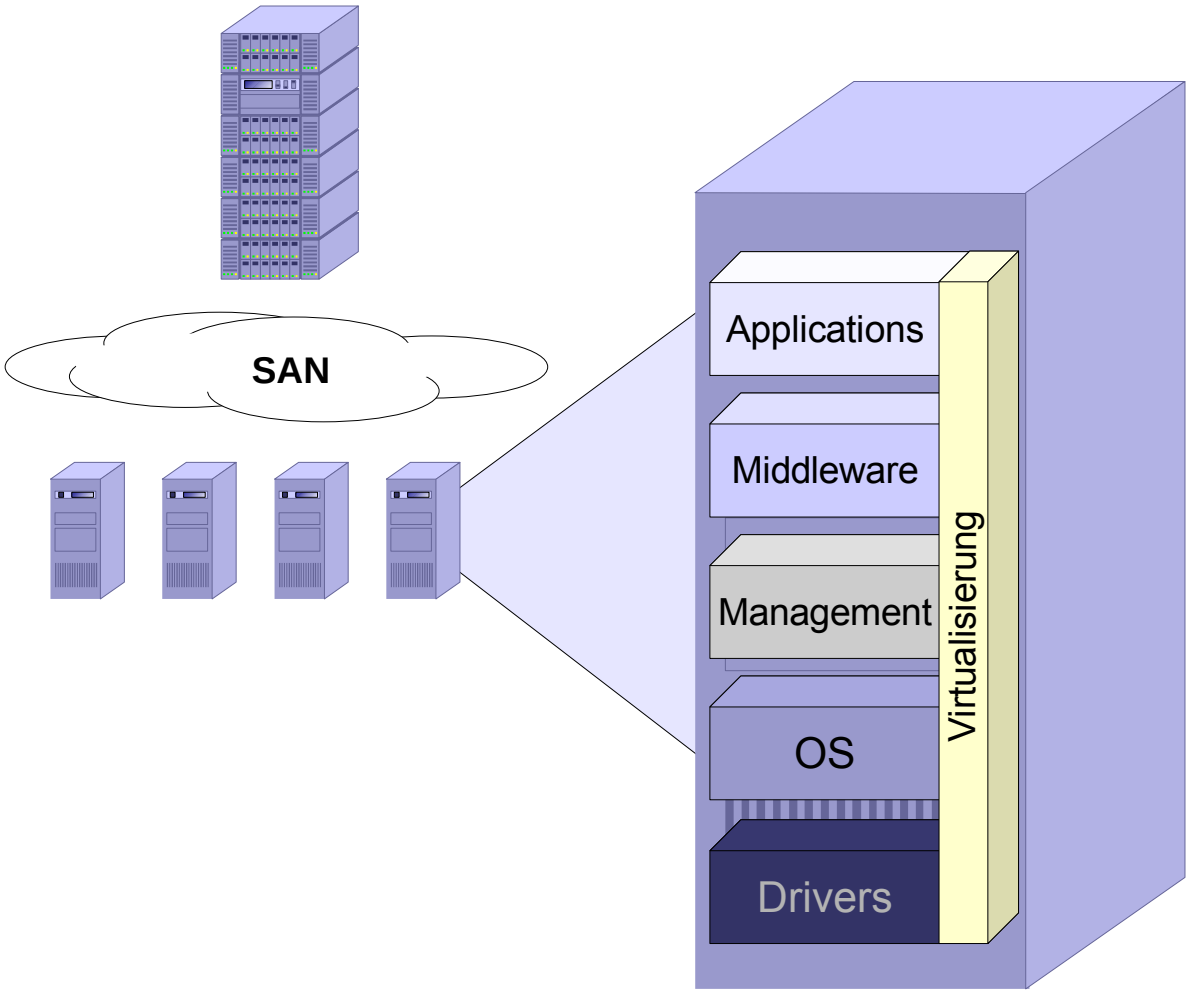
RSIO

und

OSL Storage Cluster

Was ist OSL Storage Cluster?

Hostbasierte, clusterfähige Speichervirtualisierung

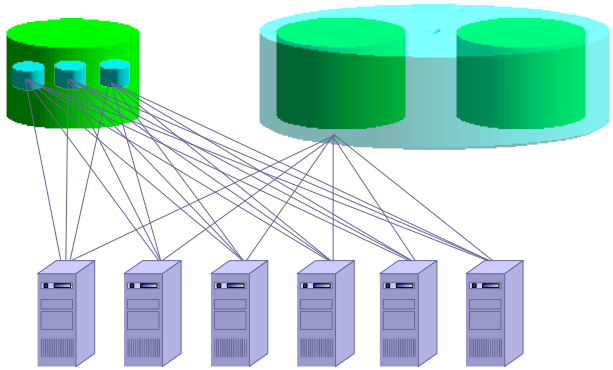


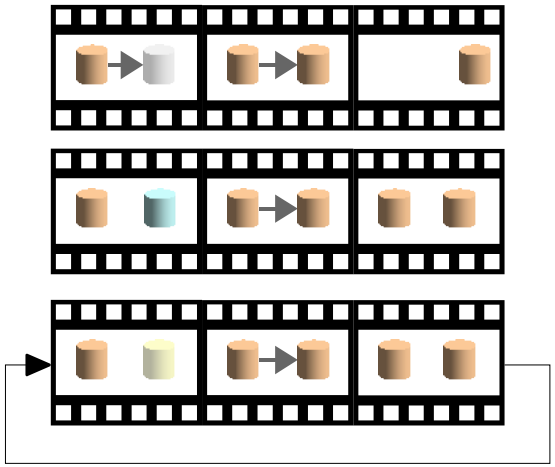
- ***Nähe zur Software, die auf dem Host läuft***
- ***einfache Interaktion und Steuerung***
- ***einfache Administration***
- ***es ist nur Software***
  - flexibel und kostensparend
  - nicht gebunden an Hardware Lifecycle
  - keine Performance- oder Verfügbarkeits-engpässe
- ***keine Festlegung auf einen bestimmten Hardwareanbieter***
- ***Einfachste Migrationen möglich:***
  - zwischen Speichersystemen
  - zwischen Anschlußtechnologien
- ***einfache Verbindung mit HV und DR***

Hostbasierte Speichervirtualisierung

Spitzentechnologie von OSL – Funktionsübersicht

- Basis-Virtualisierung
- clusterweit
- Globale Pools
- Daten verschieben
- Daten klonen
- Daten spiegeln
- Sonderfunktionen





keine spezielle Hardware erforderlich !

Physical Volumes + Application Volumes
linear oder integriert (simple, concat, stripe)
Hardwareabstraktion und IO-Multipathing
systemgestützte Speicherallokation
Online-Konfig./Dekonfig./Vergrößerung

globale Geräte / globaler Namesraum
vollautomatisiertes Zugriffsmanagement

globale Pools (hostübergreifend)
globales Inventory (Verzeichnis)
kein Verschnitt von Kapazitäten

Daten online verschieben / reorganisieren
minimaler Einfluß auf laufendenApplikations-I/O

Online-Datenkopien auf wahlfreie Ziele
atomare Operationen für mehrere Volumes

permanente Master-Image-Beziehungen
mehrere Images + OSL-Universen
inkrementelle Resynchronisation
Überbrückung von Fehlern auf dem Master

XVC (Extended Volume Controls)
z.B. Pause, Stop, Trigger, Aktionen
Bandbreitensteuerung
detaillierte Statistik

Und es geht noch mehr ...

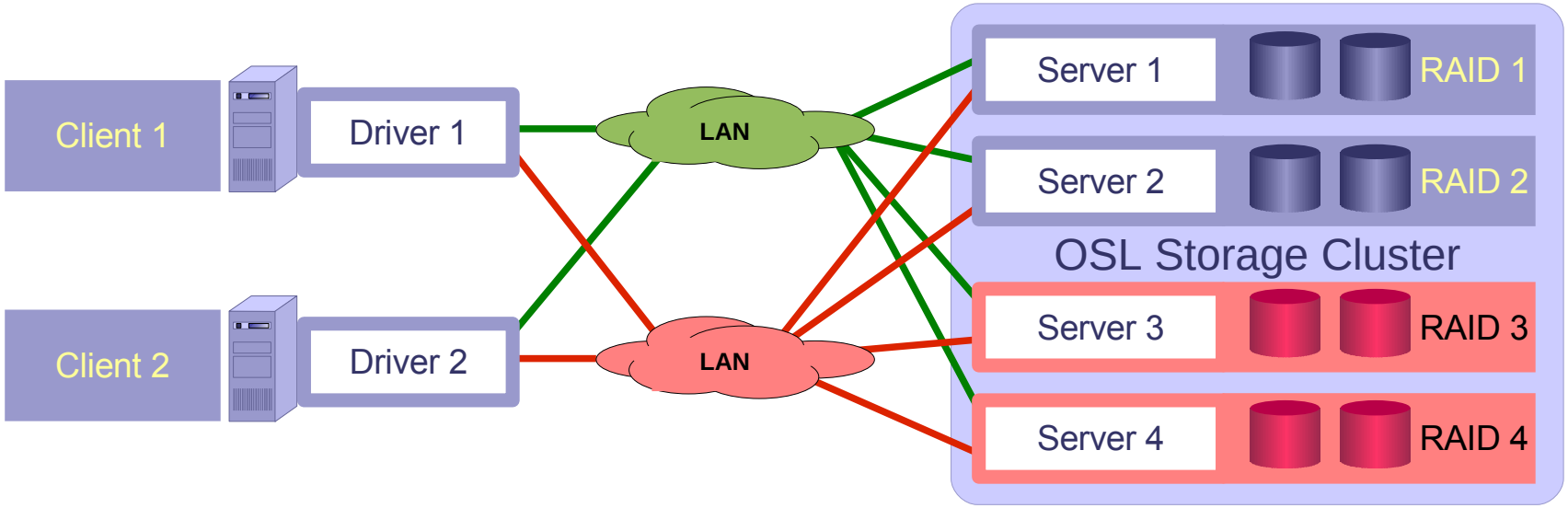
über HV, Backup und DR bis hin zum applikationsbezogenen Speichermanagement

Application Awareness	Application Control Option	Applikationen spiegeln
Bandbreiten-steuerung	clusterweite Steuerung von Applikationen	Applikationen klonen
User-Management	virtualisierte (hardwareabstrakte) Ablaufumgebungen	B2D / DASI / DR -Werkzeuge
	Hochverfügbarkeit	
	resssourcenbasiertes Selbstmanagement	
	Application Resource Description	

clusterfähige Speichervirtualisierung	Extended Data Management
globale Storage-Pools (rechnerübergreifend)	Integration von RAID-basierten Datenkopien / Snapshots
globales Disk-Inventory	hostbasierte Spiegelung
globale Geräte / globaler Namensraum	online Datenmigration
Cluster-Volumemanager mit automatisierter Allokation	Daten klonen
Disk Access Management	
I/O-Multipathing	

RSIO und OSL Storage Cluster

Das perfekte Team für SAN und LAN



RSIO Client

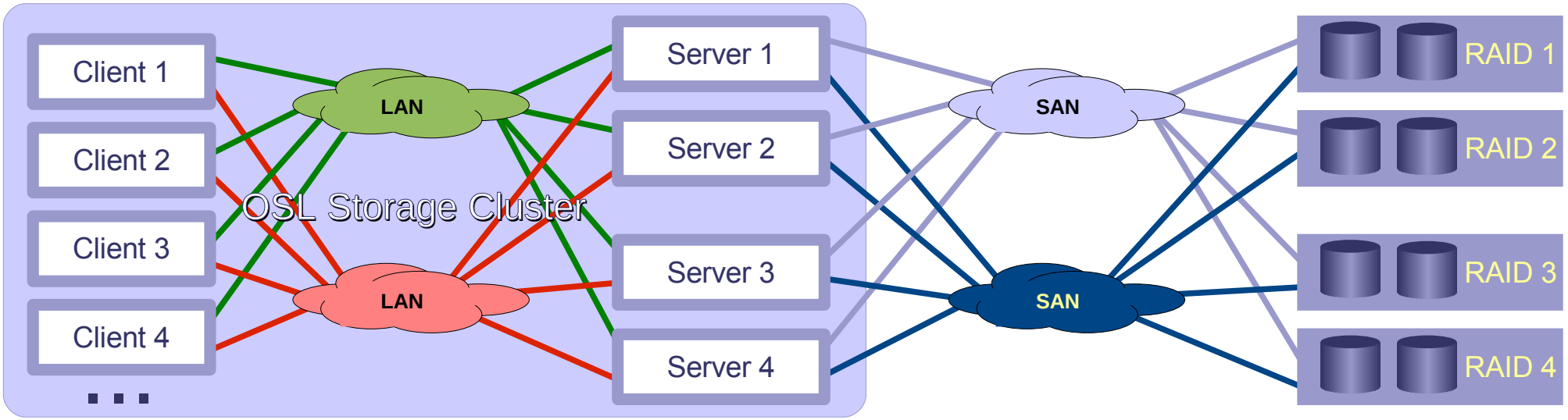
- *Zugriff auf virtualisierten Speicher*
- *Zugriff auf Global Storage*
- *Nutzung des Global Namespace*
- *Multithreaded RSIO-Client*

OSL Storage Cluster + RSIO Server

- *Speichervirtualisierung*
- *Global Storage Pool*
- *Global Namespace*
- *Access Management*
- *Multithreaded RSIO Server*

Und dann im richtigen RZ

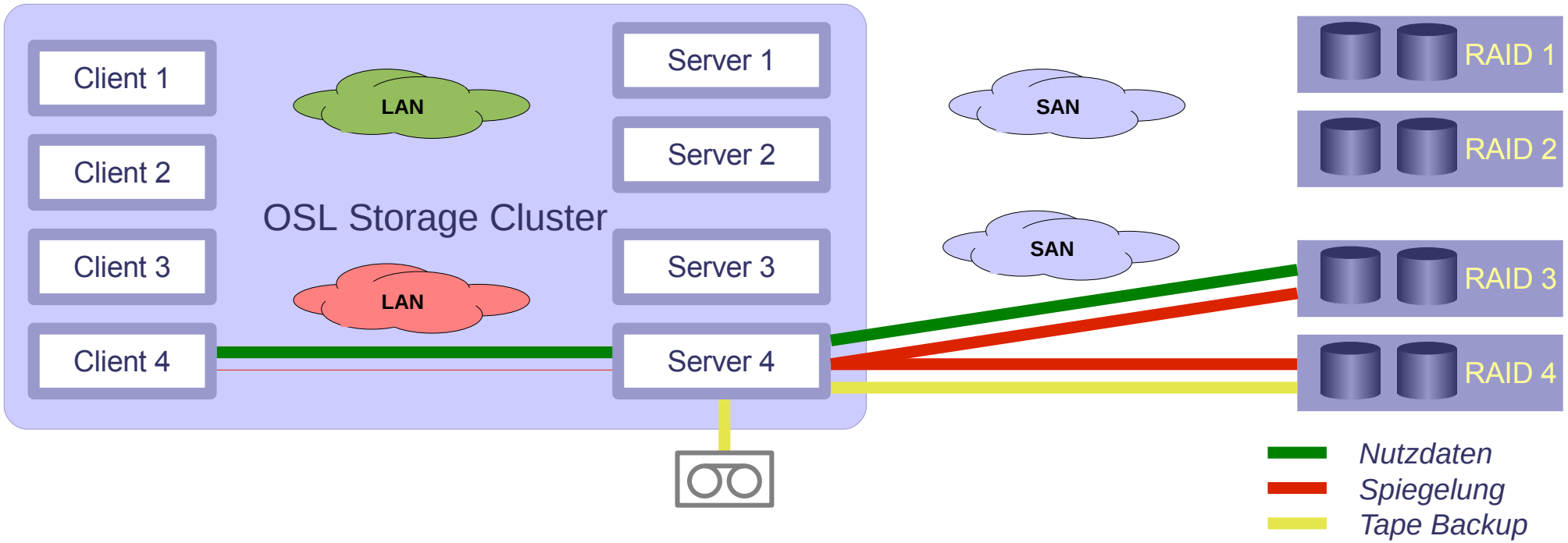
Server und Clients in einem Cluster integrieren



- *alle Funktionen wie im vorherigen Beispiel, dazu SAN-LAN-Integration*
- *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*
- *Alles aus einem Kabel: Storage, HV, Backup*
- *run applications everywhere*

Ein weiterer Vorteil von RSIO und OSL SC

Hochgeschwindigkeits-Backup 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*

Zusammenfassung

RSIO - Data Center Block I/O over Ethernet

- *direkter Transport aller relevanten IO-Aufrufe (read, write, ioctl)*
- *eigene Frames -> für verschiedene Träger geeignet*
  Implementierung über TCP(IP) natürlich routingfähig
- *integriert Verbindungsaufbau, Überwachung, Path-Multiplexing, Trunking, Selbstkonfiguration, Error Recovery*
- *Im LAN bereits mit heutiger Technik beeindruckende Skalierbarkeit und Durchsätze*
- *administorfreundliche Gliederung: Namespace -> Server -> Client*
- *damit sind vielfältige Storage-Szenarien abbildbar:*
  - *einfache Server und Clients*
  - *Cluster von Storage-Servern (Targets) und Cluster von Storage Clients (Initiators)*
  - *Storage Server Farms und Cloud-Konzepte*
- *besondere Eignung für Kombination mit Speichervirtualisierung*
- *”Huckepack” -Transport für andere Dienste (z. B. HV, Backup ...- alles über ein Kabel)*
- *Clients können mit Servern zu einem intelligenten Cluster verschmelzen, z. B. für:*
  - *automatisierte, applikationsbezogene Speicherverwaltung vom Client aus*
  - *Failover-Cluster*

RSIO - Storage Networking der nächsten Generation

F r a g e n ?

Bert Miemietz &
Christian Schmidt

OSL Gesellschaft für
offene Systemlösungen mbH

RSIO - Storage Networking
der nächsten Generation

Danke für Ihre Aufmerksamkeit!

Wir freuen uns auf weitere Gespräche

Stand 26

Bert Miemietz &
Christian Schmidt

OSL Gesellschaft für
offene Systemlösungen mbH