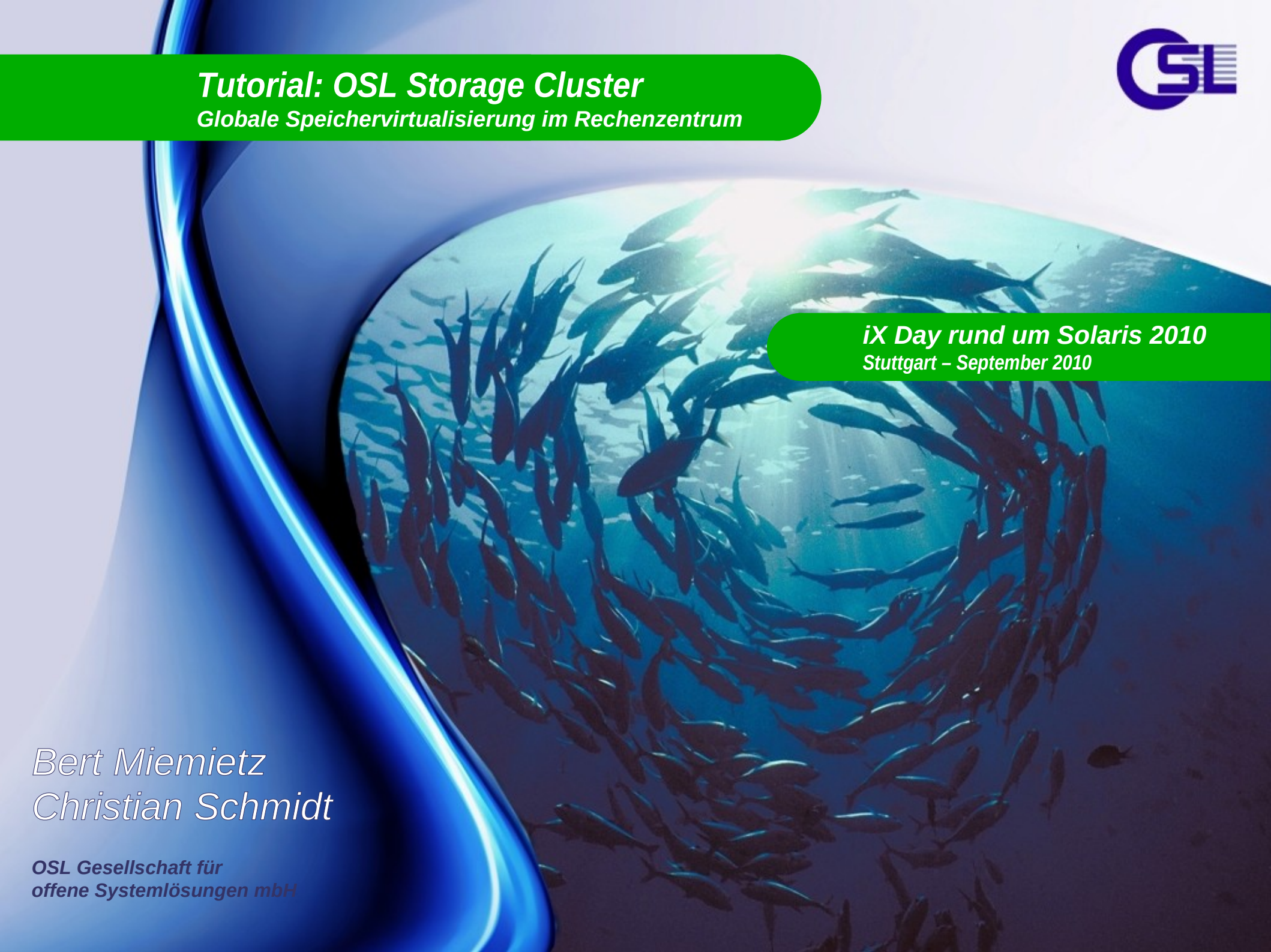
Tutorial: OSL Storage Cluster

Globale Speichervirtualisierung im Rechenzentrum



iX Day rund um Solaris 2010

Stuttgart – September 2010

Bert Miemietz
Christian Schmidt

OSL Gesellschaft für
offene Systemlösungen mbH

Überblick

Was heißt Shared Storage Clustering

Nutzung moderner RZ-Infrastrukturen für neuartige Management-Konzepte

OSL Storage Cluster:

- *Lösung zur Integration von Unix-Servern mit modernen, RAID-basierten Speicherinfrastrukturen*
- *erweitert OS um aufeinander abgestimmte Virtualisierungs-, Management- und Cluster-Funktionalitäten*
- *Speicher- bzw. Volume-Management, Virtualisierung, System- und Applikationsmanagement sowie Clustering werden als Einheit begriffen*
- *das administrative Konzept und die Software selbst zielen auf flexible, virtualisierte Administrations- und Ablaufumgebungen*
- *deutliche Vereinfachung der Abläufe und administrativen Aufgaben im RZ*

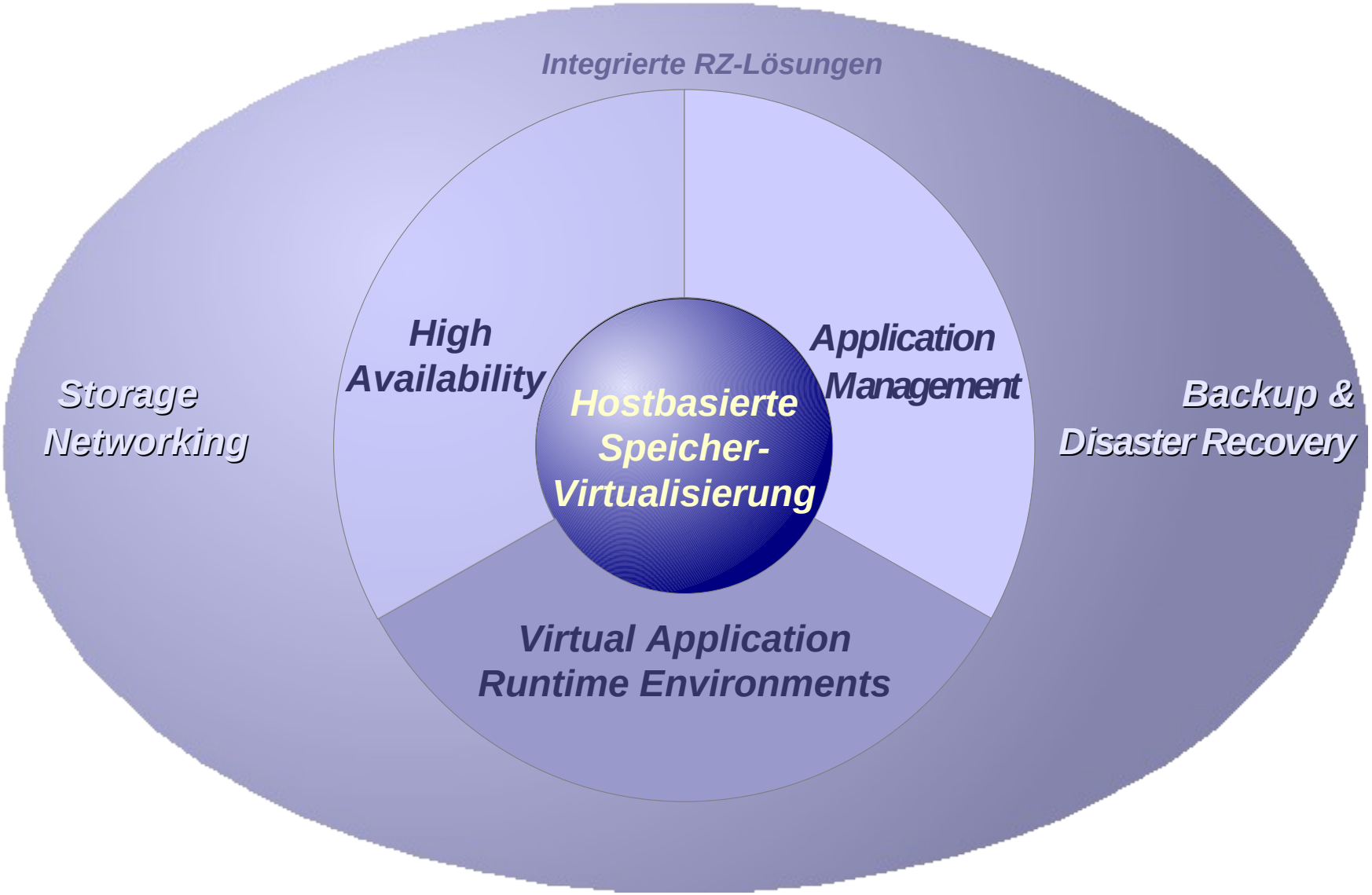
"Die Verbindung aus Softwaretechnologie und durchdachter, langfristig angelegter RZ-Organisation beim Anwender hilft, Ressourcen effektiv auszunutzen, Kosten zu senken und zusätzliche Freiheit bei der Auswahl der Systemplattformen zu gewinnen."

Kern des OSL Storage Clusters:

globale, hostbasierte Speichervirtualisierung

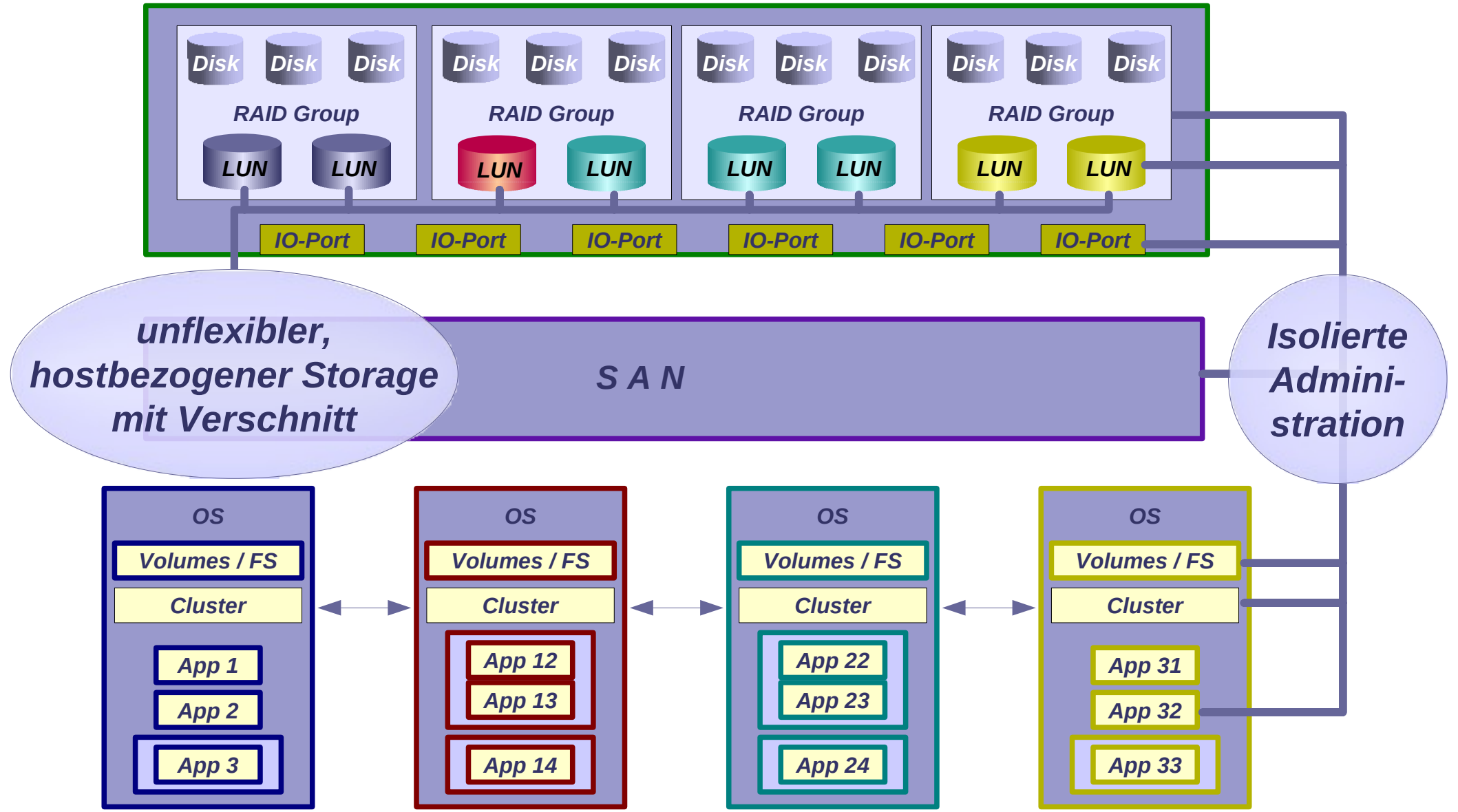
Was steckt also hinter dem Konzept von OSL?

Virtualisierung - Clustering



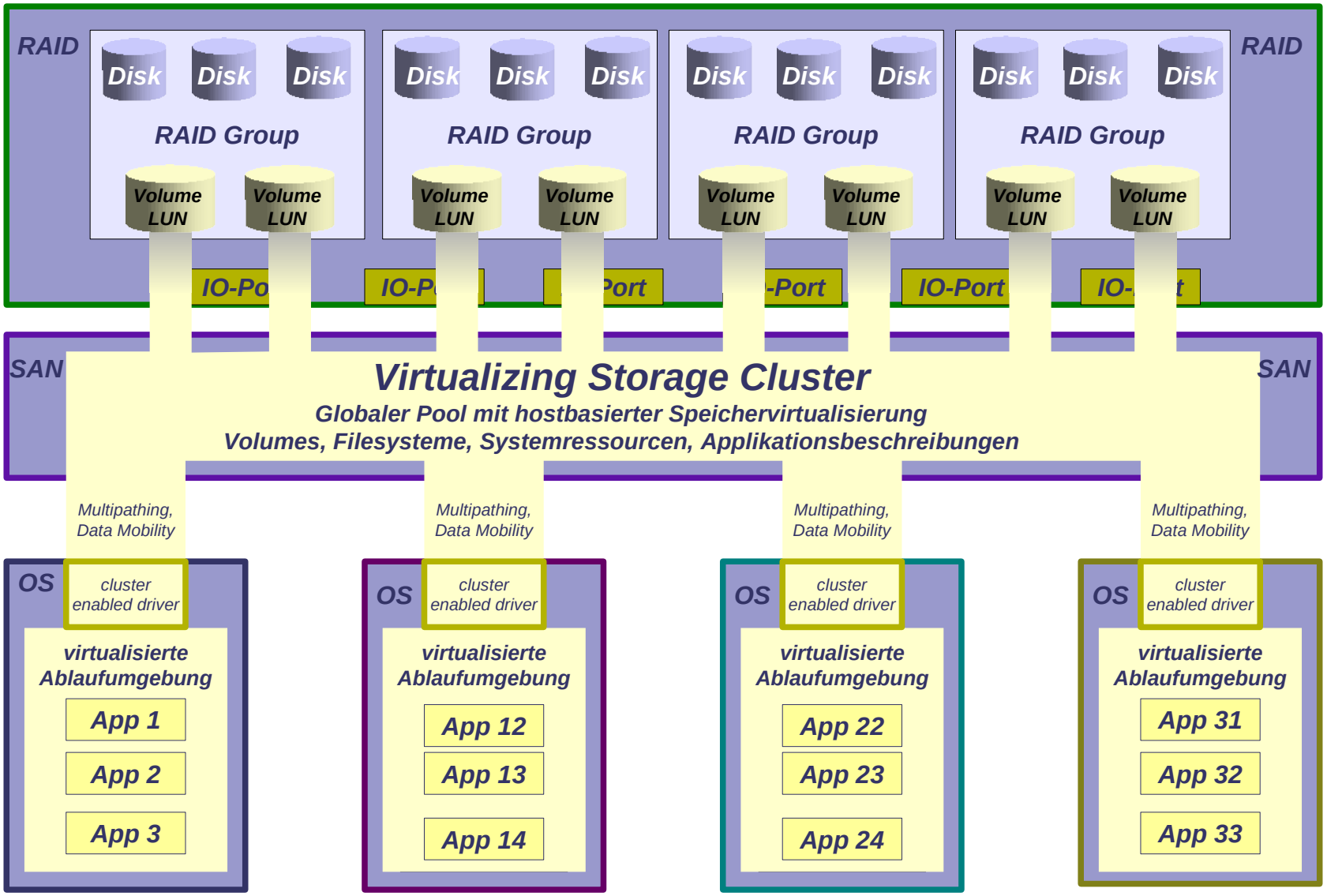
Blick in das durchschnittliche Rechenzentrum

Moderne Infrastruktur, vielschichtige Administration



Zentralisierung + Netztopologie = Cluster

OSL Storage Cluster: Vereinfachung durch Integration



Für Eilige: Differenzierungsmerkmale im Überblick

Was unseren Shared Storage Cluster von anderen Konzepten unterscheidet

- *Administration ausschließlich vom Host aus (Allokieren, Volume erzeugen, Filesystem erzeugen ...)*
- *Speichersystem einmal in Betrieb -> quasi nichts mehr daran zu tun*
- *symmetrisches Konzept*
  - Administration von jeder Maschine aus
  - no single point of failure
- *keine zusätzliche Hardware für Heartbeat o.ä.*
- ***globaler** Storage-Pool*
  - enorme Flexibilität
  - kein Verschnitt
  - optimale Auslastung, auch unter Performance-Aspekten
  - erweitert damit auch Einsatzmöglichkeiten von ZFS
- *Clusterfähigkeit / global devices / namespace von Anfang an*
- *Integration mit Anwendungssteuerung - > Application Awareness*
- *beeindruckende Skalierbarkeit, keine Performance-Engpässe*

- ***enorme Vereinfachungen, größere Stabilität***

Leistungsumfang im Detail

Speicher-Virtualisierung, Anwendungssteuerung, HV, Backup und DR

Application Awareness

Bandbreiten-steuerung

User-Management

Application Control Option

- clusterweite Steuerung von Anwendungen
- virtualisierte (hardwareabstrakte) Ablaufumgebungen
- Hochverfügbarkeit
- ressourcenbasiertes Selbstmanagement

Application Resource Description

Application Mirrors

Application Clones

B2D / DASI / DR -Tools

Clusterfähige Speichervirtualisierung

- globale (hostübergreifende) Storage Pools
- Global Disk Inventory
- Global Devices / Global Namespace
- Cluster-Volumemanager mit automatischer Allokation
- Disk Access Management
- IO-Multipathing

Extended Data Management

- Integration RAID-basierter Datenkopien / Snapshots
- Hostbasierte Datenspiegelung
- Live Data Migration
- Data Clones

clusterweite Speichervirtualisierung

Speicher an die Solaris-Systeme anbinden

"Foreign" und "native" Disks

- *neue LUNs sind "foreign" und "unused"*
- *Inventarisierung nimmt Disks in den globalen Pool auf*
- *einheitliche, hardwareabstrakte Sicht durch alle Cluster-Nodes*

Vor Erst-Inventarisierung:

Nach Erst-Inventarisierung:

Nach RAID-Erweiterung:

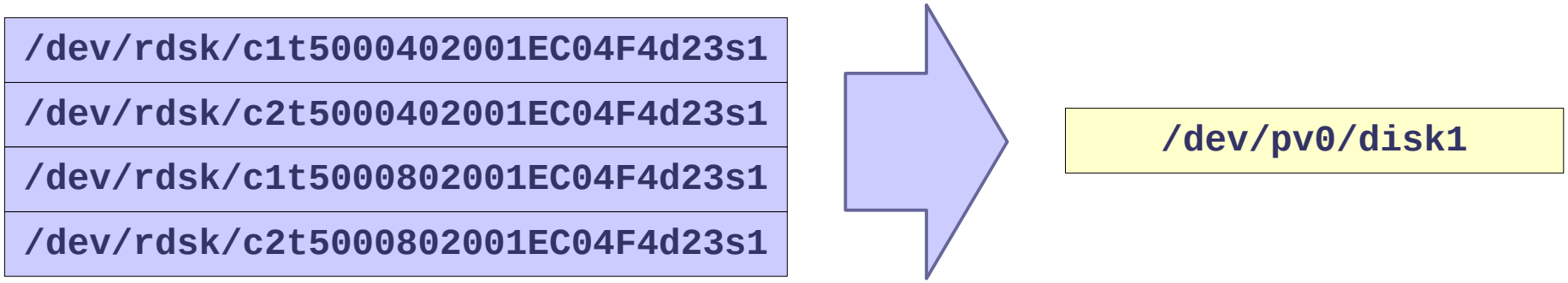
Nach 2. Inventarisierung:

Vorteile

- *leichte Identifizierung neuer Disks*
- *Solaris-Geräteadressen uninteressant*
- *sprechende und zugleich einfache Gerätenamen*
- *Bildung eines globalen Pools*

Physical Volumes

Global wirksame Hardwareabstraktion



- *Gerätenamen werden vom Administrator gewählt -> es entfällt die Notwendigkeit, mit schwierigen Controller-Nummern oder SCSI-Adressen zu arbeiten*
- *SCSI-Adressen und Mappings des RAID-Systems spielen bei »native« Disks keine Rolle mehr und können (offline) geändert werden, ohne daß irgendwelche Konfigurationsänderungen in OSL SC erforderlich sind (wohl aber u. U. sd.conf etc).*
- *Slices (format) werden nicht mehr für die Aufteilung der Platten benutzt.*
- *Damit kann die Platte bei Neuaufteilungen online bleiben.*
- *Mehrere Datenpfade werden zu einem einzigen Geräteknoten zusammengefasst. Bei Ausfall eines Datenpfades ist weiter ein Zugriff auf die Platte möglich, sofern ein alternativer Pfad vorhanden ist (IO-Multipathing).*
- *Alle Platten werden optimal über die Kanäle verteilt (Load Balancing).*

Praxisteil 1

- *Inventarisierung*
- *Physical Volumes und Pfade*
- *Storage Pool*

OSL Storage Cluster
Praxisteil 1: Foreign und Native Disks

```
[root@erde] dkadmin -ni
Running with clustername:                iscsi
Building device table:                   ok

Found foreign disk with following properties:
device path:             /dev/rdsk/c2t5d23s1
vendor / product:        SUN / DISK
serial number:           632D7033:"S;
capacity (MByte):        47683
disk format:             sunSPARC
dvsc product specific.: ????|               |                |              |
former DVSC volume:      seems never used by DVSC with this device type
alternate disk info:     o.k.
current VTOC will be:    *** DESTROYED ***

new volume name or [RETURN] to skip device: disk1
new volume group or [RETURN] for "default":
```

OSL Storage Cluster
Praxisteil 1: Foreign und Native Disks

- **Es wird ein globaler Storage Pool gebildet**
  - **Übersicht über den gesamten Pool**

    ```
    smgr -q summary
    ```

  - **Übersicht über die Inventarisierten physical Disks**

    Lokal: `pvadmin -l`

    Global: `pvadmin -q`

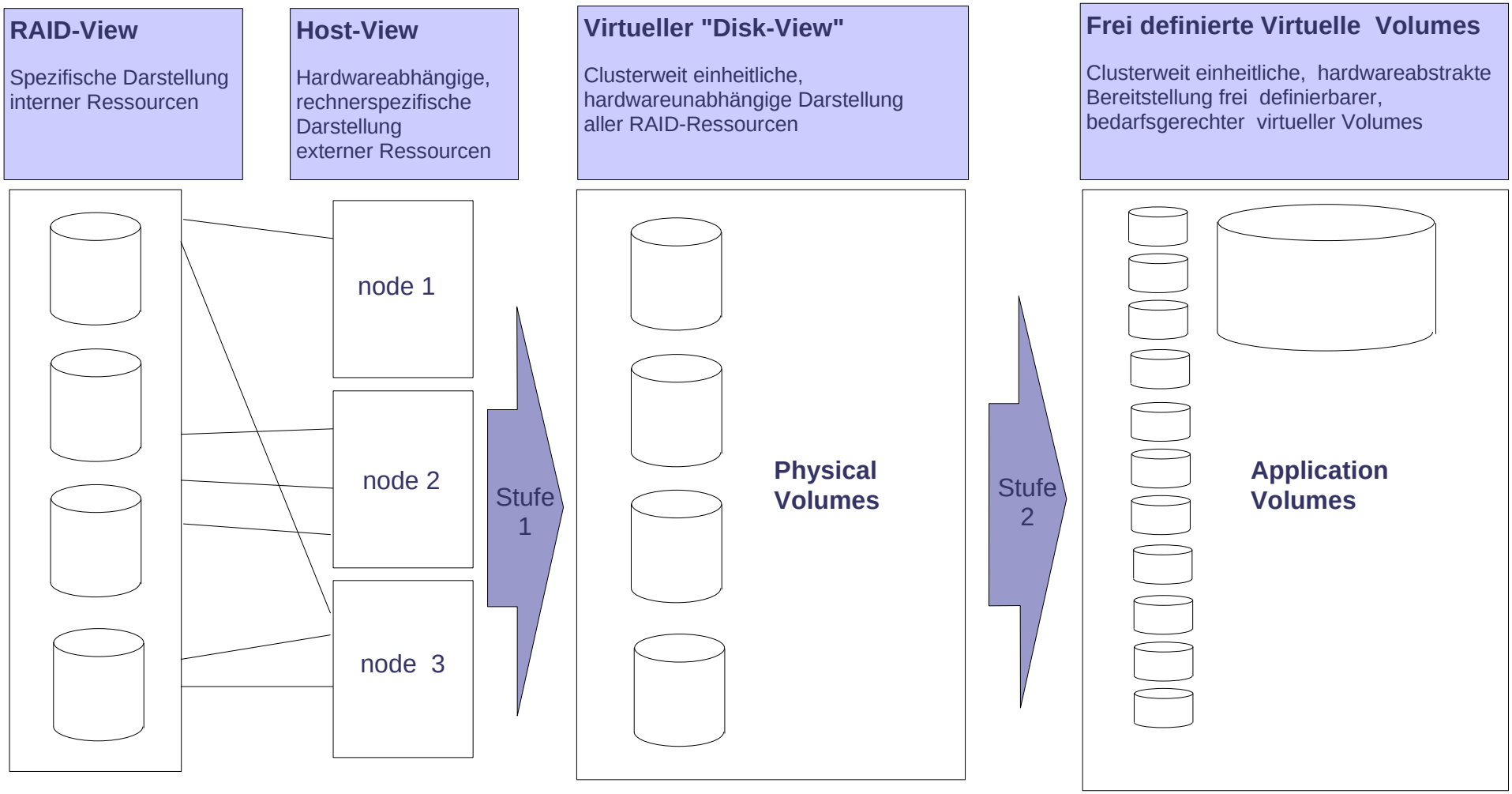
  - **Übersicht über die genutzten Pfade der physical Disks**

    ```
    pvadmin -lvv
    ```

Ende Praxisteil 1

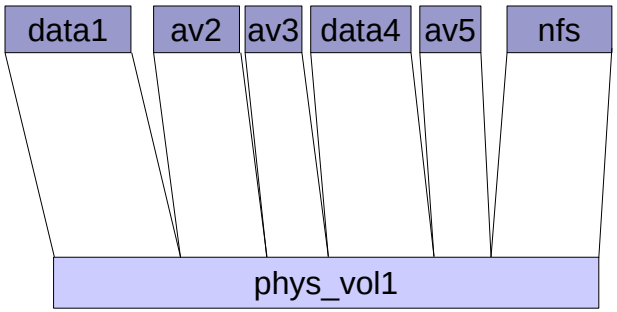
Wie weiter?

Wie wir den globalen Pool nutzen

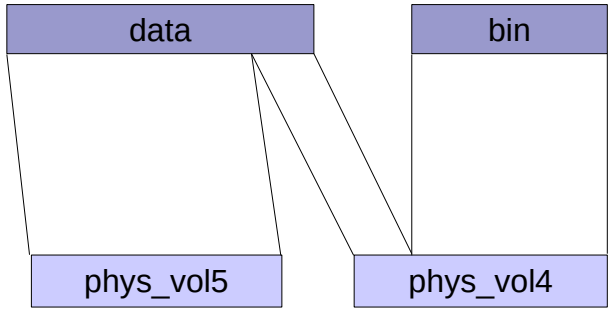


2. Stufe: Application Volumes

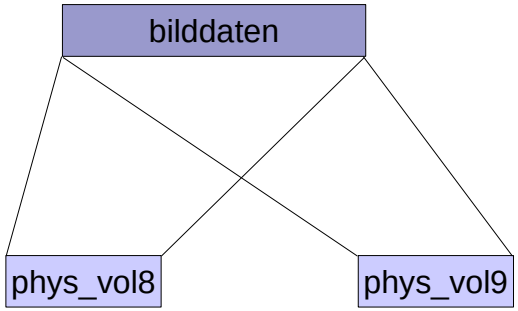
Das Wichtigste im Schnelldurchlauf



simple

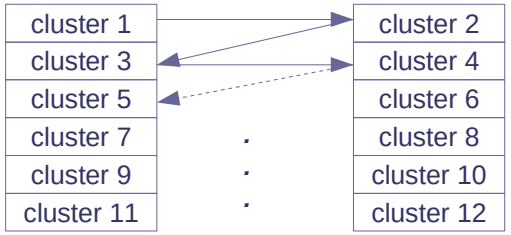


concat



stripe

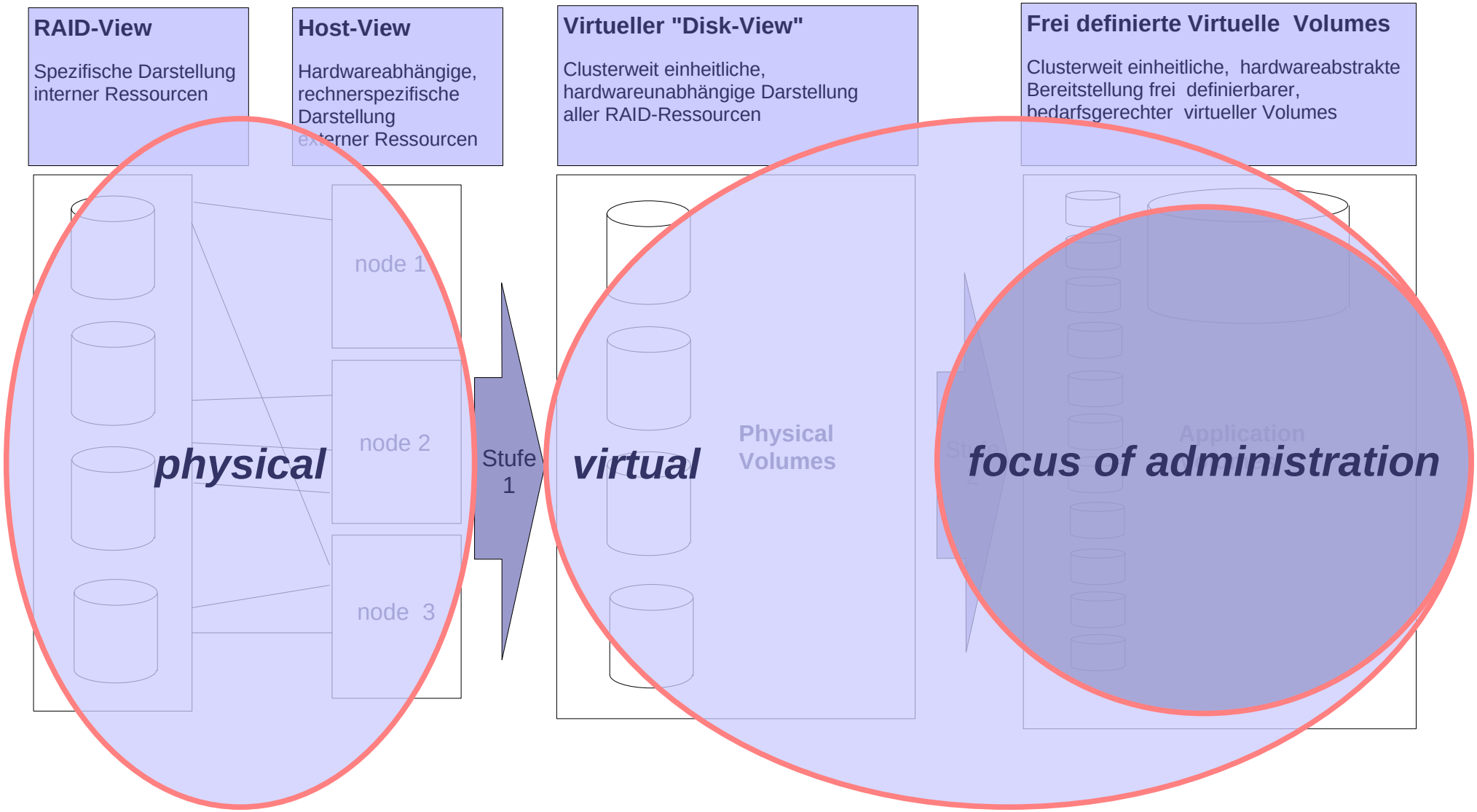
- *es stehen unterschiedliche Typen zur Verfügung*
  - Bereitstellen benötigter Größen
  - mögliche Modifikation Performance-Attribute
- *Erzeugung aus dem globalen Pool*
- *Application Volumes liegen direkt auf Physical Volumes (flache Hierarchie)*
- *frei wählbare Namen*



(Applikations-) Daten gehören per Definition auf Application Volumes!
Datenspeicherung direkt auf Physical Volumes oder Disks ist tabu.

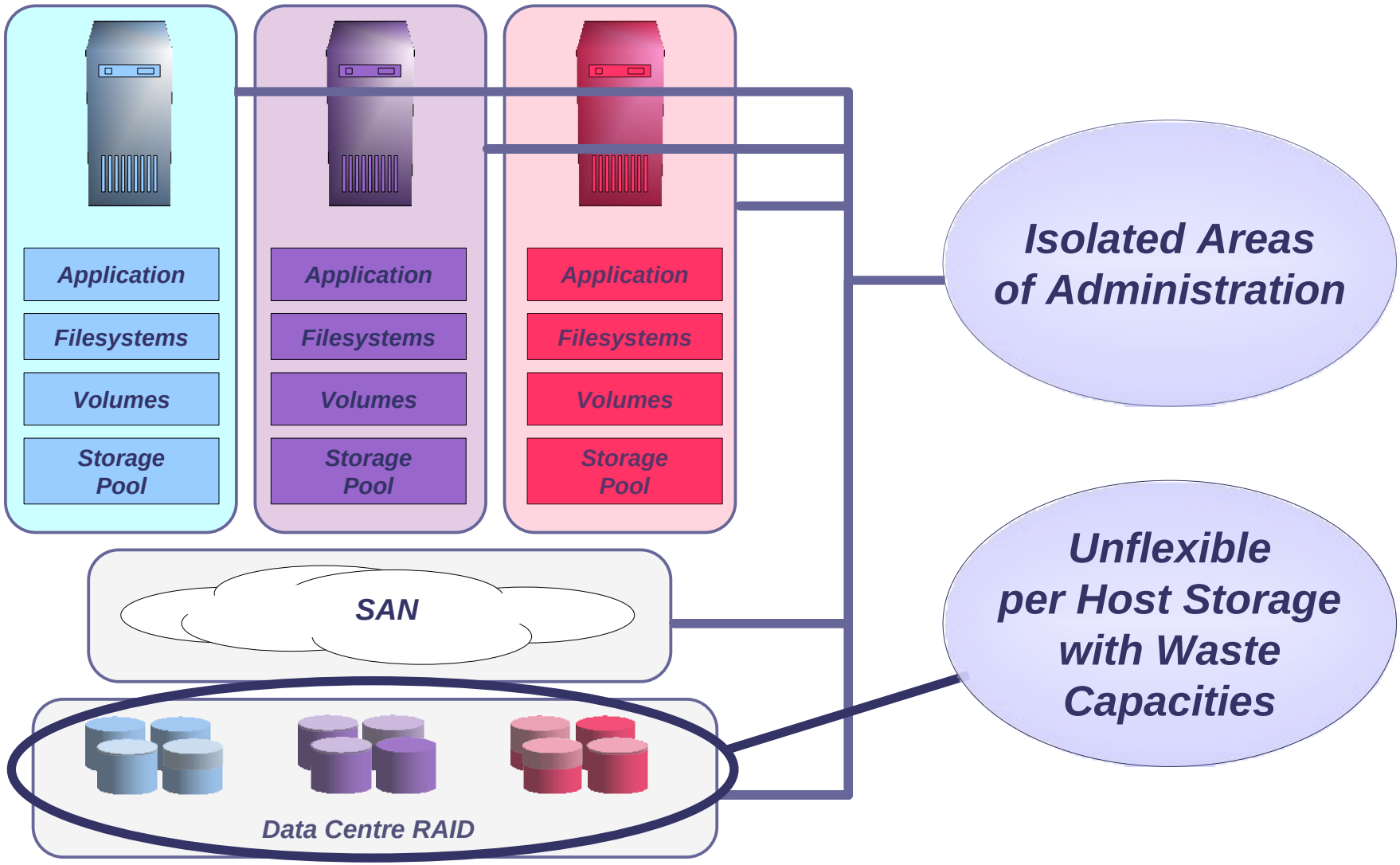
Was haben wir erreicht?

Vereinfachung in der Administration und mehr Flexibilität



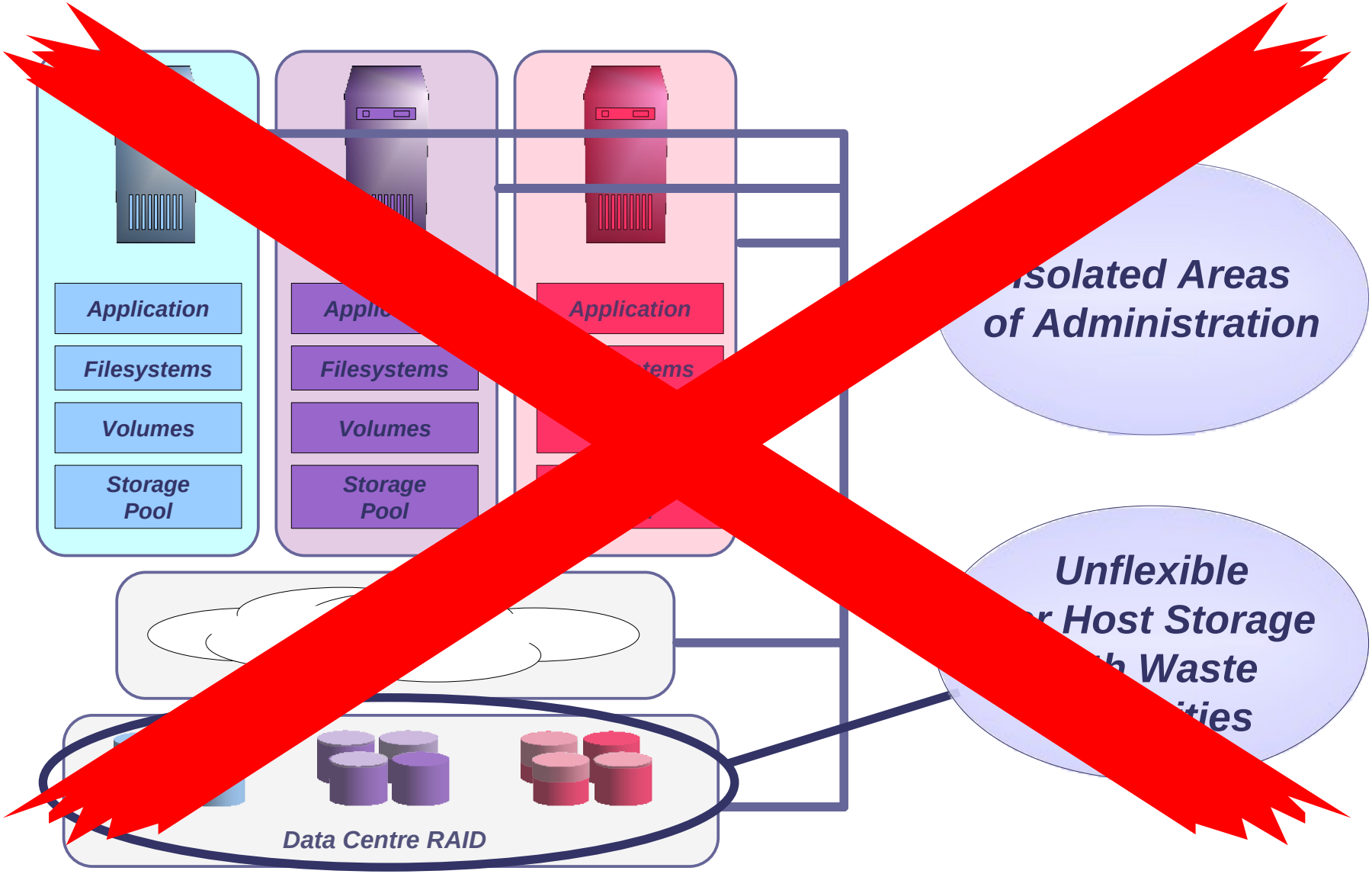
Und was noch?

Verschnittfreie Nutzung der Ressourcen



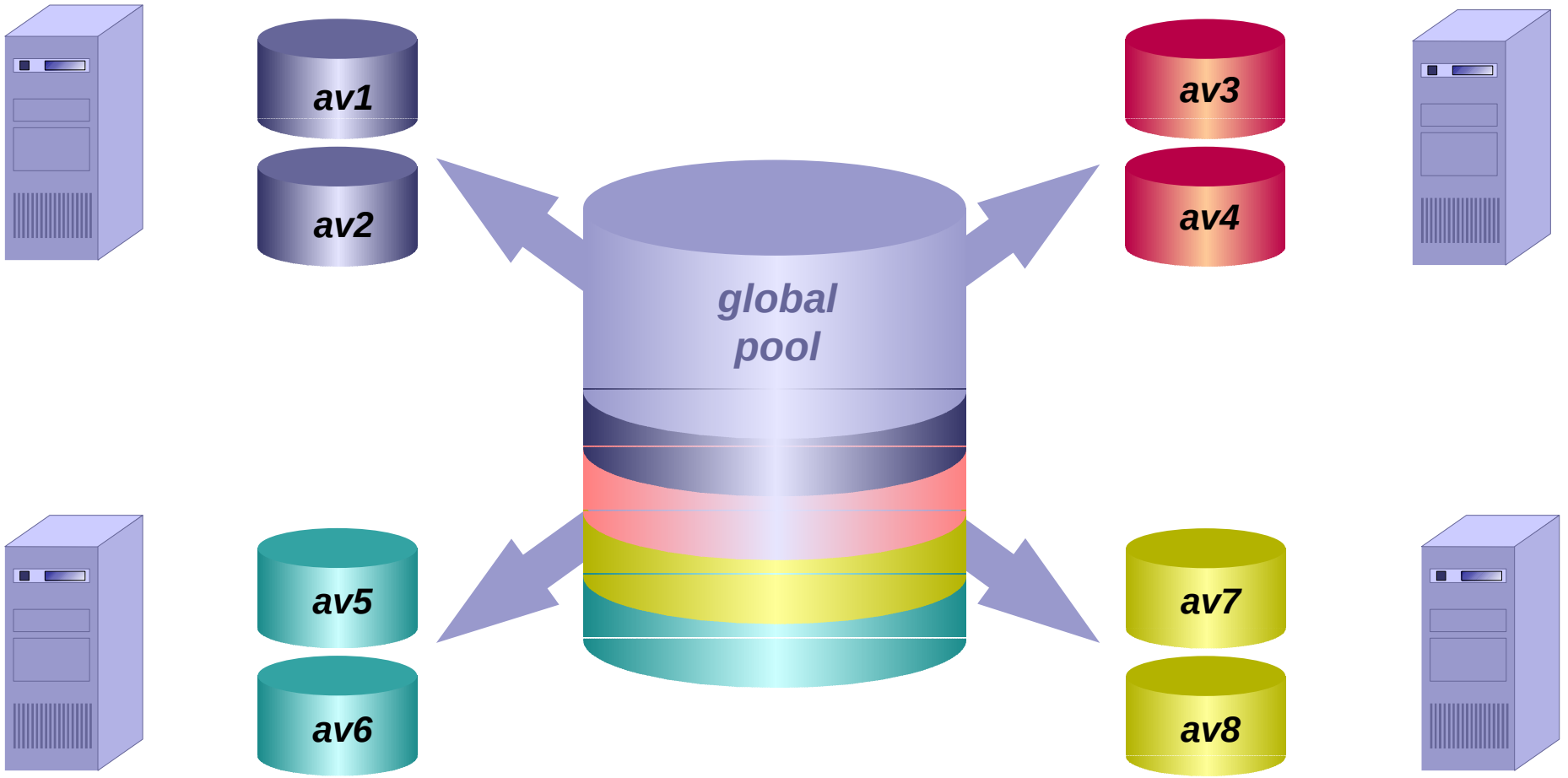
Und was noch?

Verschnittfreie Nutzung der Ressourcen



Und was noch?

Verschnittfreie Nutzung der Ressourcen



Praxisteil 2

- *smgr und Global Pool*
- *Anlegen von Volumes*
- *Nutzungsübersichten*

OSL Storage Cluster
Praxisteil 2: Von der LUN zum Application Volume

- **Application Volumes beinhalten die Daten der Applikationen**
  - **Daten gehören auf Application Volumes**
  - **Größe ist frei definierbar – Es wird sich am Storage Pool bedient**
  - **Volumetypen**
    - *Simple, Concat, Stripe*
- **Anlegen eines Volumes**
  - **`smgr`**
    - *Bedient sich am Pool*
  - **`avadmin`**
    - *Genaue Positionierung auf den Physical Volumes*

- **Anlegen eines Volumes**
  - **Typ simple:**

```
[root@sol-1(iscsi)] smgr -c mein_vol -S 1g
[root@sol-1(iscsi)] avadmin -lvv mein_vol
0 mein_vol 2097152 of 2097216 blocks "simple" in 1 pieces, 32 block clusters
  [ 1] iscsi_d1-1 [10485824...12583039]
```

  - **Typ stripe:**

```
[root@sol-1(iscsi)] smgr -c stripe_vol -S 1g -F stripe -o pieces=2
[root@sol-1(iscsi)] avadmin -lvv stripe_vol
0 stripe_vol 2097152 of 2097216 blocks "stripe" in 2 pieces, 32 block clusters
  [ 1] iscsi_d1-1 [12583040...13631647]
  [ 2] iscsi_d2-1 [10485824...11534431]
```

OSL Storage Cluster
Praxisteil 2: Von der LUN zum Application Volume

- **Überblick über den Storagpool**
  - `avadmin -l` - lokale Sicht
  - `avadmin -q` - Sicht des Clusters
  - `smgr -q` - Query Abfragen: summary, used_ext, free_ext, used_ext_all

Ende Praxisteil 2

Was mir noch fehlt....

Es geht nicht nur um Speichervirtualisierung allein

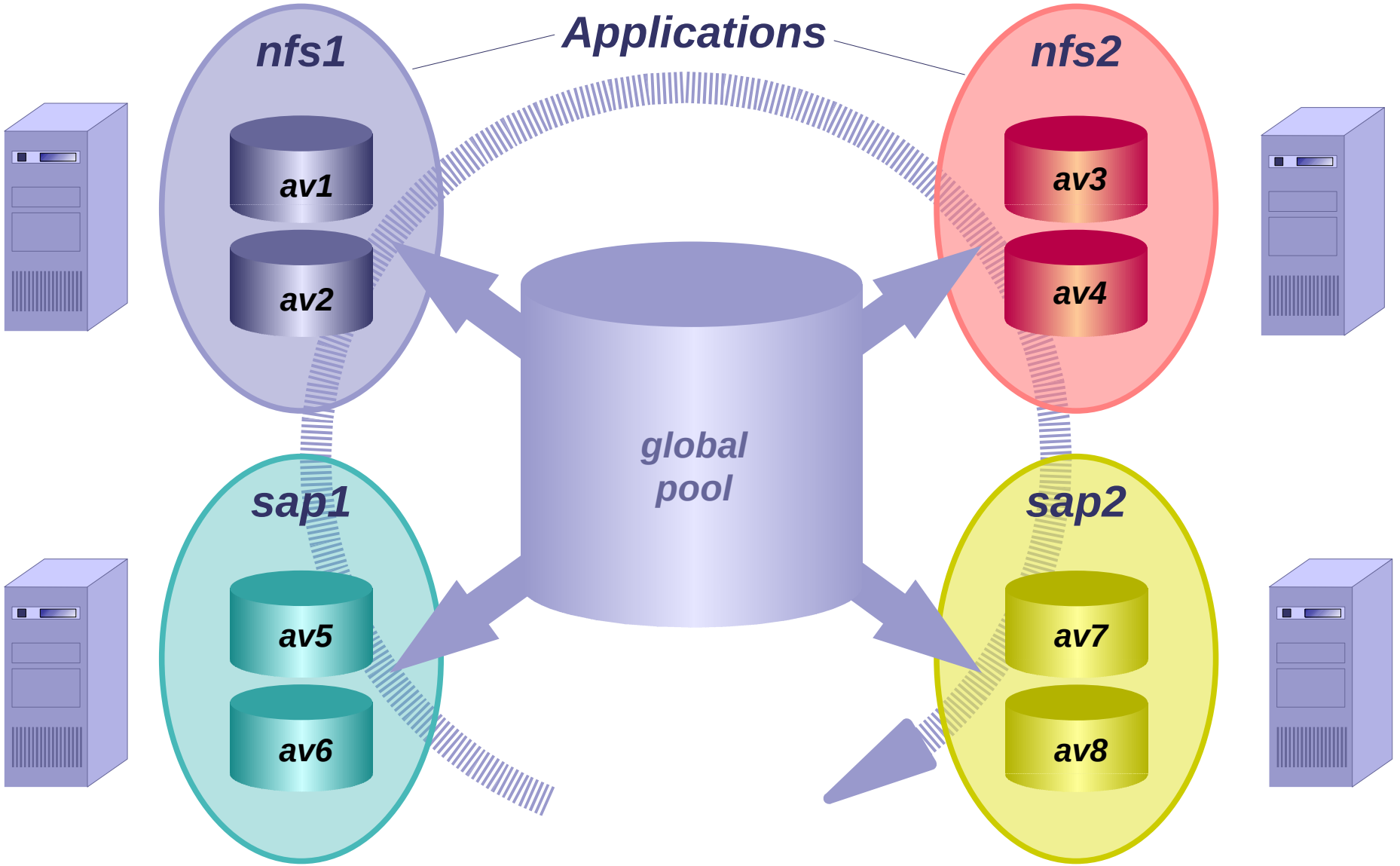
- *ich habe: - virtuelle Geräte*
  - clusterweit Zugriff auf diese Geräte
  - keinerlei Hardwarebezug in meiner Konfiguration

- *aber: - am Ende muß ich den Anwendern Applikationen bereitstellen, flexibel, mit hoher Performance, gesicherten Daten und hoher Verfügbarkeit*

Was ich noch brauche, ...

ist die Integration mit den Anwendungen

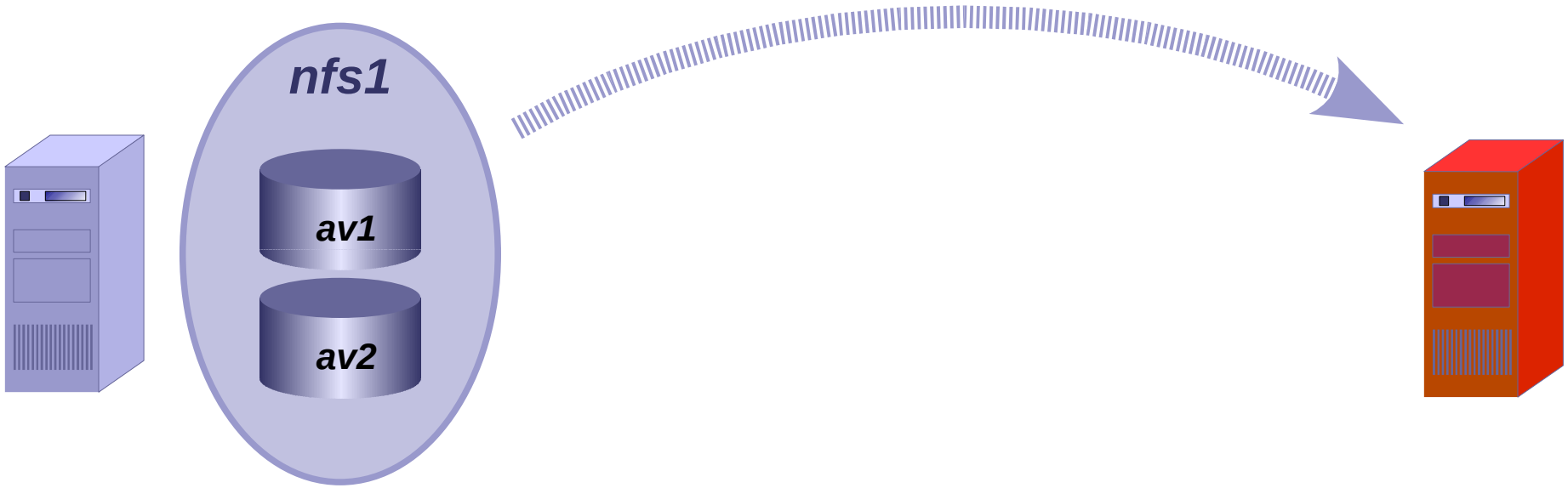
Speichervirtualisierung, Clustering. Applikationsmanagement gehören zusammen

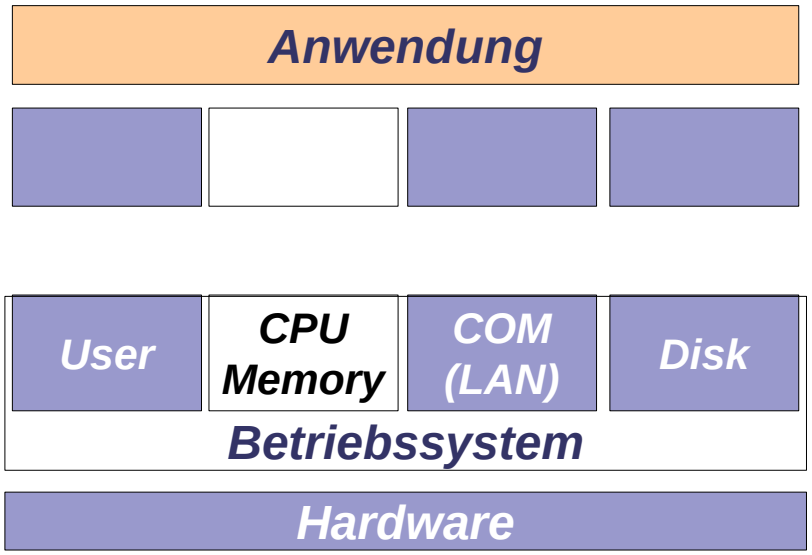


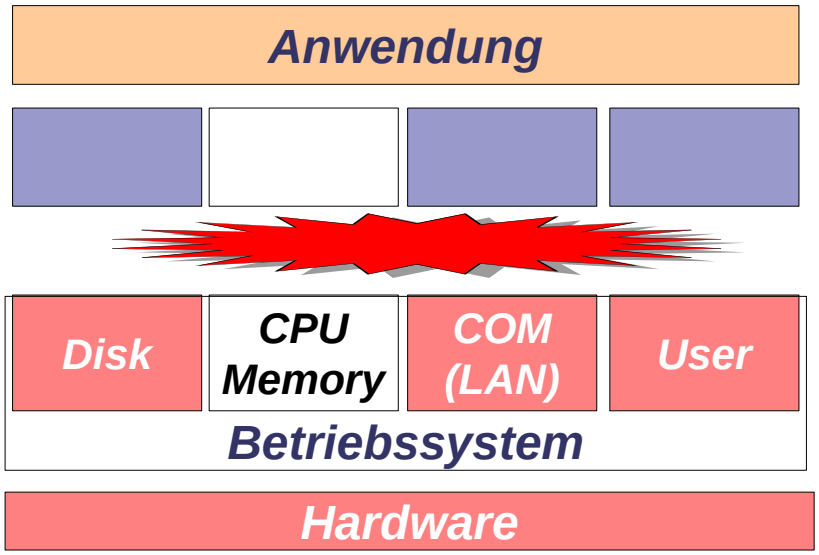
Anwendungen im Cluster organisieren + steuern

Das Problem beim Verschieben von Anwendungen

Ohne Virtualisierung wenig Aussicht auf Erfolg

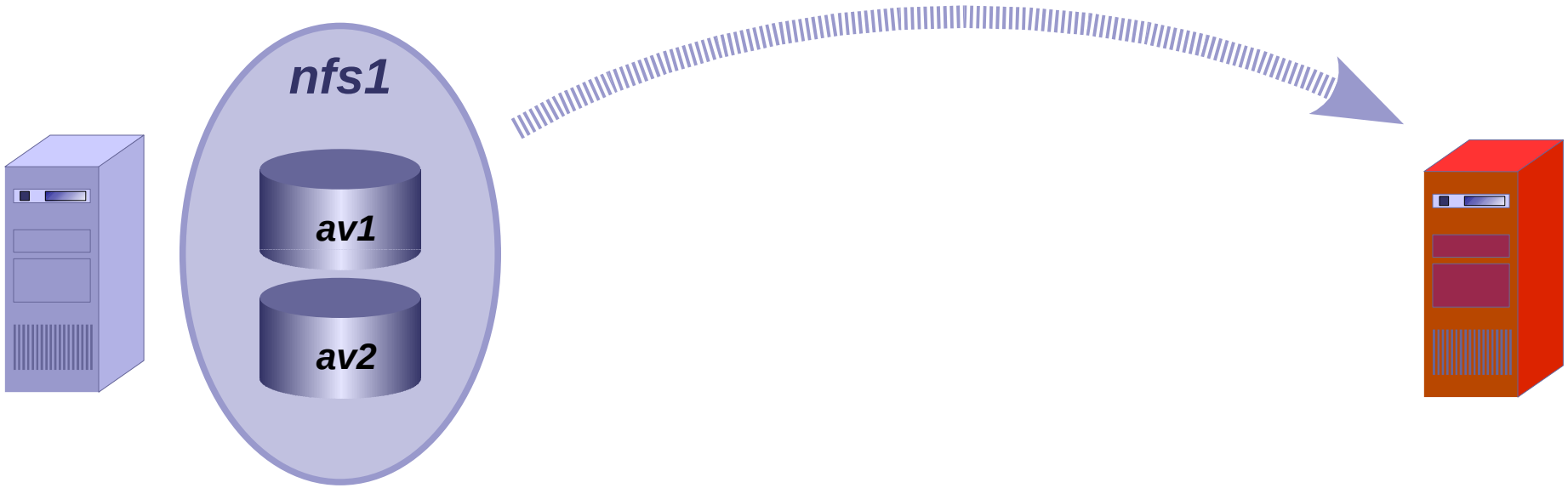


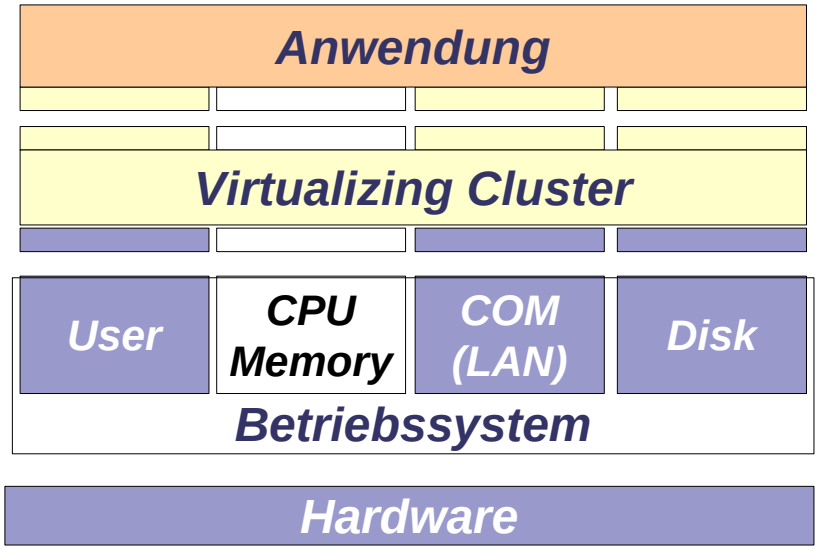


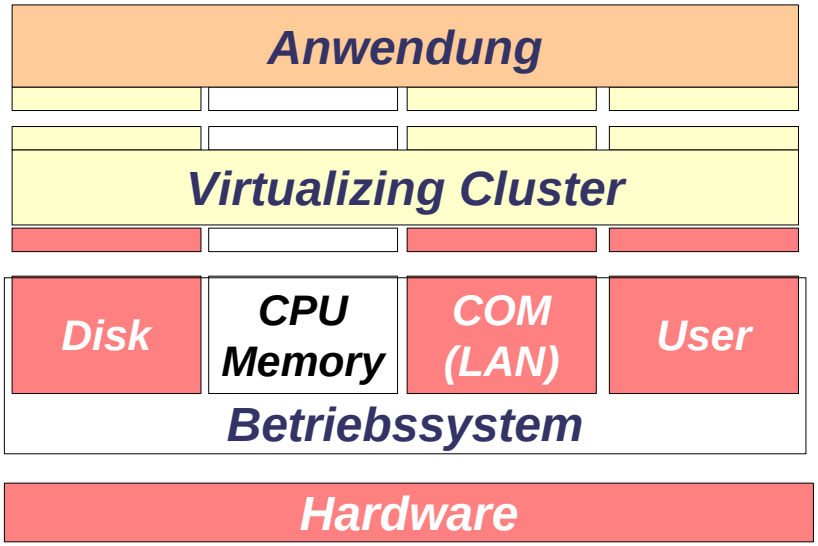


Kein Problem mit Virtualisierung

Hardwareabstraktion ist der Schlüssel zum Erfolg

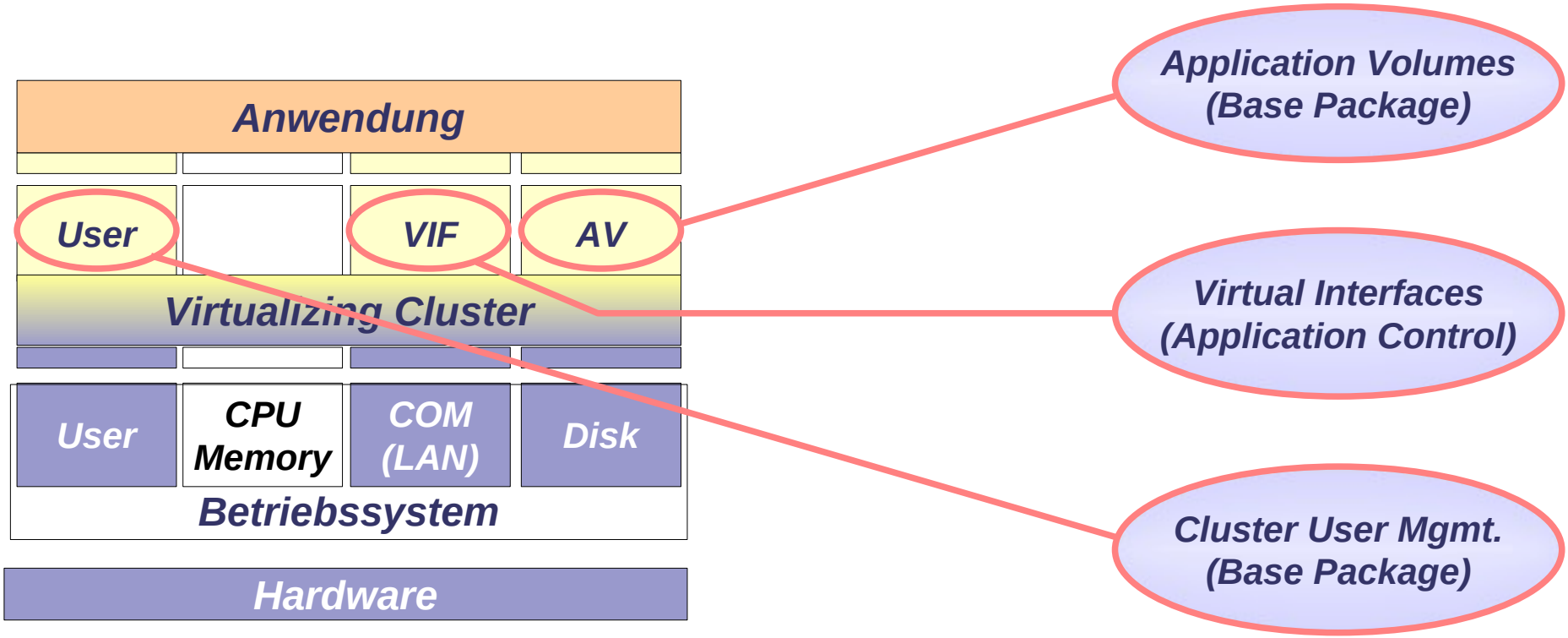






Virtualisierte Ablaufumgebungen für Applikationen

Sind ganz ohne Virtuelle Maschinen oder Zonen möglich



Clusterfähige, virtualisierte Ablaufumgebungen bestehen aus:

- *Anonymen und Virtualisierten Ressourcen des Betriebssystems (RAM, CPU, VFS)*
- *aufsetzenden Virtualisierungs- und Clusterfunktionen (OSL Storage Cluster)*

Wem das zu einfach ist, dem bleiben zusätzlich Zonen und virtuelle Maschinen

Applikationen anlegen und beschreiben

OSL Storage Cluster -Applikationen im Überblick

Anwendungsdeklaration / -definition

o Priorität / Verdrängungsmöglichkeiten
o Migrationsstrategie und Execution Mode
o Ressourcensteuerung (IO-Bandbreite)
o Verknüpfung von Usern mit Applikationen

- *für jede Art von Applikationen*
- *applikationsabstrakt*
- *dienen der Steuerung durch die Cluster Engine*

```
# appadmin
```

Application Resource Description

clusterweit verfügbare Beschreibung von:
o genutzten Volumes, Filesystemen, IP-Adr. etc.
o Start- und Stopmethoden
o Methoden zum Abbruch einer Applikation
o Methoden zum Recover einer Applikation
o Methoden zum Monitoring / Auto-Restart

- *applikationsspezifisch*
- *einheitliches Schema*
- *freies Format*
- *dient der spezifischen Steuerung der einzelnen Applikation*

```
# ardadmin
```

Die Anwendungs-Resourcenbeschreibung (ARD)

Der Inhalt der Application Resource Description im Überblick

Nach Aufruf des Editiermodus findet sich die ARD unter /dvsc/ard/app_name.

Sie enthält folgende Verzeichnisse:

etc	*Eventual Technical Configuration* *Alle spezifischen Konfigurationsdaten zu Ihrer Anwendungen wie die dazugehörige vfstab, die Konfiguration virtueller IP-Adressen, der zugehörige Monitor oder anwendungsspezifische Konfigurationsparameter.*
start	*Application Start Scripts*
stop	*Application Stop Scripts*
break	*Application Break Scripts*

Applikationsbezogene Konfigurationsdaten gehören fortan nicht mehr in die Konfigurationsdateien des Betriebssystems, sondern in die der Applikation!

Die ARD hält dafür mindestens folgende Files bereit:

dfstab	*beim Start der Anwendung bereitzustellende NFS-Shares*
vfstab	*beim Start der Anwendung zu montierende Filesysteme auf Application Volumes bzw. NFS-Shares*
vif	*beim Start der Anwendung zu konfigurierende virtuelle IP-Adressen*

Was eine "richtige" Applikation noch braucht

Applikationsbezogene Start- und Stopp-Prozeduren

Über applikationsbezogene "User"-Start- und Stopp-Prozeduren können wir weitere Aktionen einbinden.

- *Die Scripts befinden sich in den ARD im Unterverzeichnis »start«, »stop«, »break«,*
- *Scripts werden beim Applikationsstart in folgender Reihenfolge ausgeführt:*
  1. *Built-in*
  2. ***Benutzerdefinierte Scripts** (S01-S99) **ähnlich** den **RC-Scripts** (/etc/rc2.d/S...)*
  3. *Built-in*

  Stop- und Break-Prozeduren analog mit K01-K99
- *Die Scripts werden mit 2 Argumenten aufgerufen:*

  $1 = "start" oder "stop"

  $2 = nickname
- *Die Scripts müssen einen definierten Return-Code liefern:*

  0 bei fehlerfreier Beendigung

  >0 bei aufgetretenen Fehlern, die weiteren App.-Start unmöglich machen
- *Gravierende Fehler (RC != 0) führen beim Start sofort zur Ausführung der Stop-Prozeduren.*

Praxisteil 3

- *Applikation anlegen*
- *Start- und Stop-Prozeduren*
- *Beispiel Oracle*

- *Für eine Oracle Datenbank benötigen wir beispielsweise*
  - *Serveradresse (IP): ./etc/vif*
  - *Dateisysteme: ./etc/vfstab*
  - *Stoppskripte: ./stop/K**
  - *Oracle Konfiguration: ./etc/oracle*
  - *Startskripte: ./start/S**
- *Konfigurationsdateien in der ARD der Applikation:*

```
[root@sol-1] ardadmin -e oras10
NOTICE (ardadmin): application "oras10" is now in edit mode.
[root@sol-1] cd /dvsc/ard/oras10
[root@sol-1] ls
break  etc    start  stop
[root@sol-1] ## EINSTELLUNGEN VORNEHMEN
[root@sol-1] cd /
[root@sol-1] ardadmin -c oras10
NOTICE (ardadmin): Changes to ARD of "oras10" committed. Edit mode quit.
```

Applikationen im OSL Storage Cluster

Praxisteil 3: Applikationen Starten und Stoppen

- **Starten der Applikation**

```
[root@sol-1] appstart oras10
```

- **Startlog anzeigen**

```
[root@sol-1] applogcat start oras10
```

- **Applikation stoppen**

```
[root@sol-1] appstop oras10
```

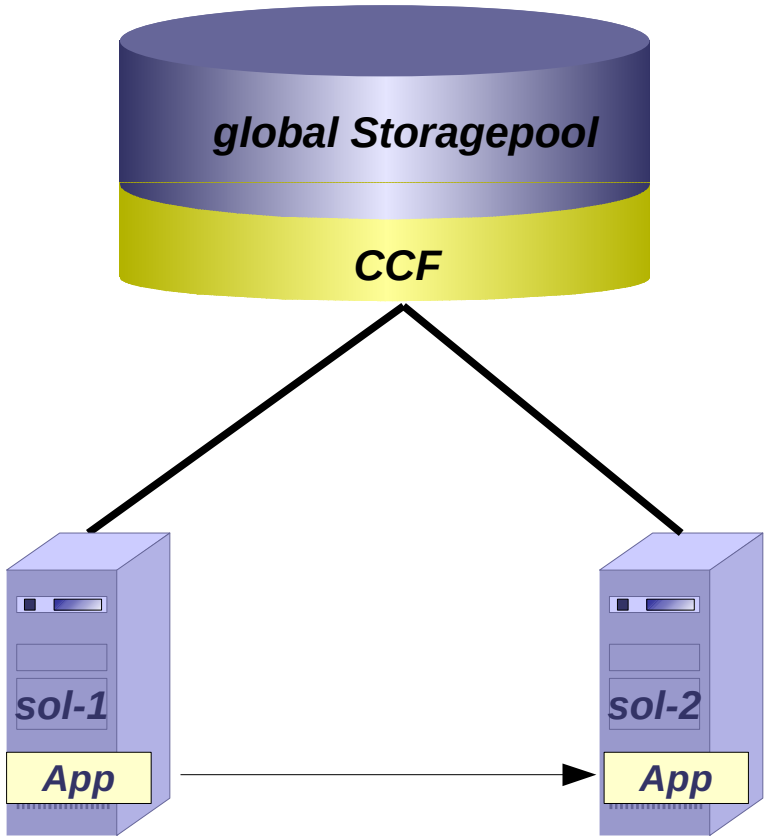
- **Stopplog anzeigen**

```
[root@sol-1] applogcat stop oras10
```

Applikationen im OSL Storage Cluster

Praxisteil 3: Applikationen im Cluster

- **Applikationen können auf allen Nodes im Cluster gestartet werden, denn ...**
  - ... die Konfiguration ist global verfügbar
  - ... die User sind Clusterweit identisch
  - ... der Storage ist auf allen Nodes im selben Namensraum vorhanden
  - ... die Applikationen sind gekapselt installiert



- ***Management der Applikationen – `appadmin`***
  - ***Wie ist der Status meiner Applikation***
  - ***Auf welchem Node läuft meine Applikation***
  - ***Auf welchen Nodes könnte meine Applikation gestartet werden***
  - ***Welche Ressourcen benötigt meine Applikation***

`appadmin -qo` ***- Kurzer Überblick über den Status der Applikationen***

`appadmin -qv` ***- Detaillierte Übersicht über die Applikationen***

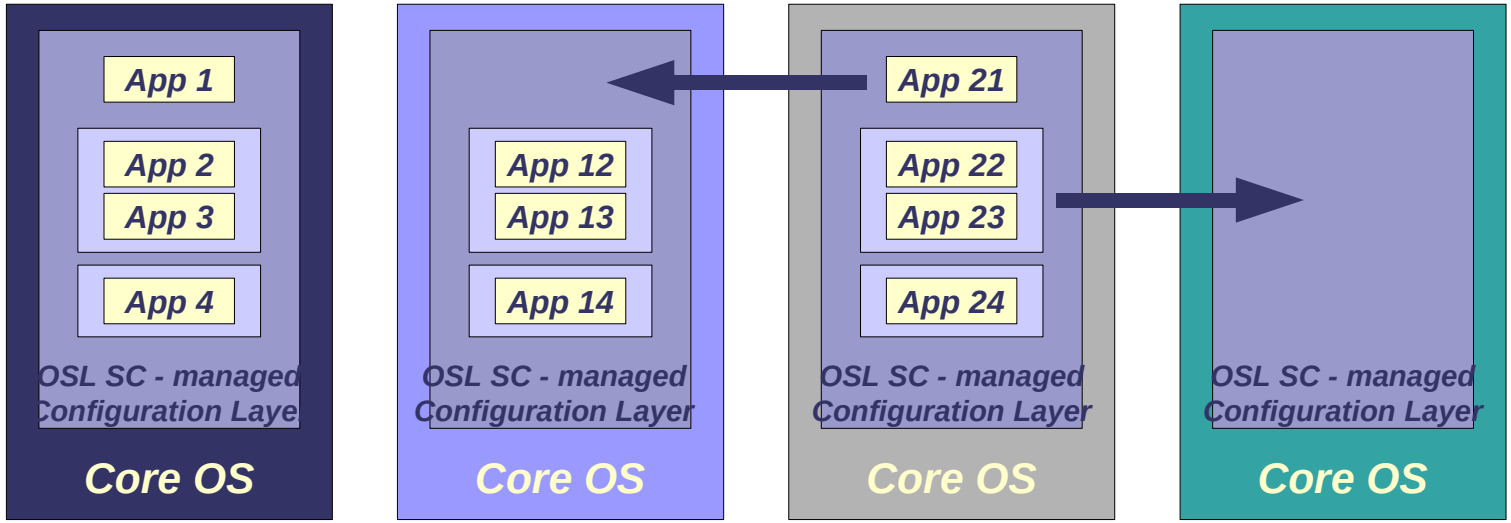
Ende Praxisteil 3

Was haben wir gewonnen?

Organisation in Applikationen ermöglicht Virtualisierte Ablaufumgebungen

- *Global Devices und Application Resource Description*
  - *Raw- und Blockdevices + Dateisysteme*
  - *ZFS*
  - *IP-Adressen und NFS*
- *Globales Nutzer- und Gruppenmanagement*
- *Automatische Adaption ASCII-Konfigurationsdateien*
- *Globales Management und Migrationsdienste für Zonen*

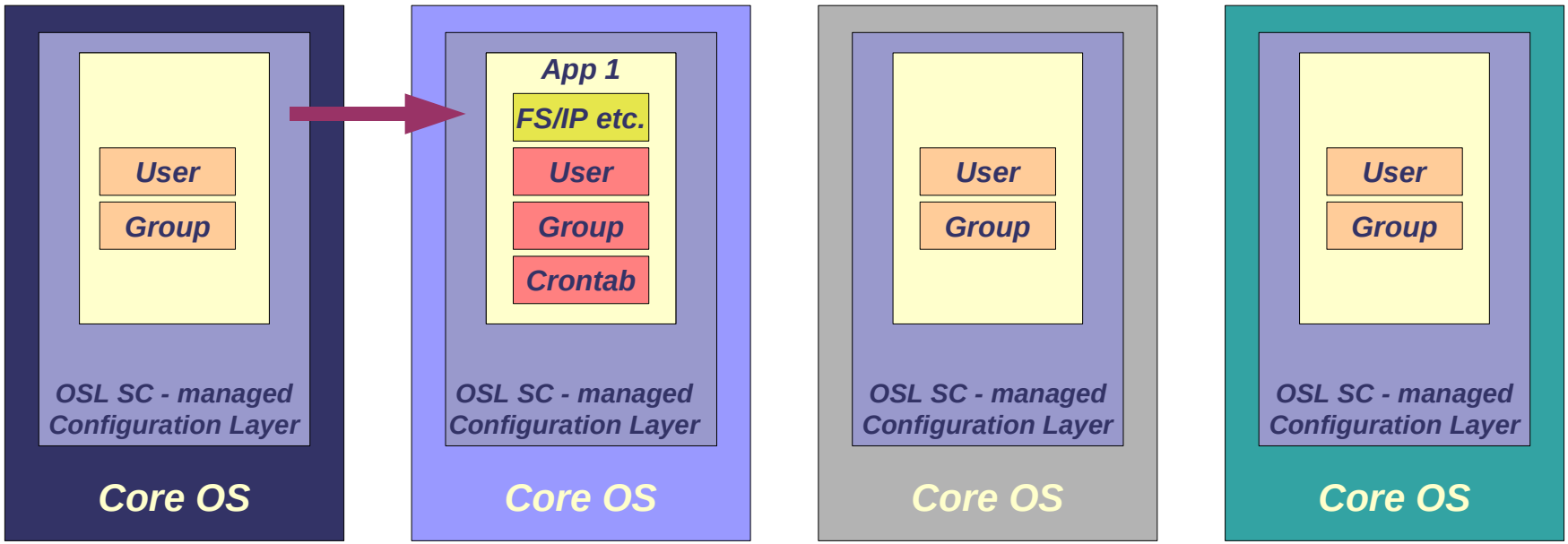
Umgebung lebt außerhalb der Maschine weiter



Was fehlt uns noch – Global User Management

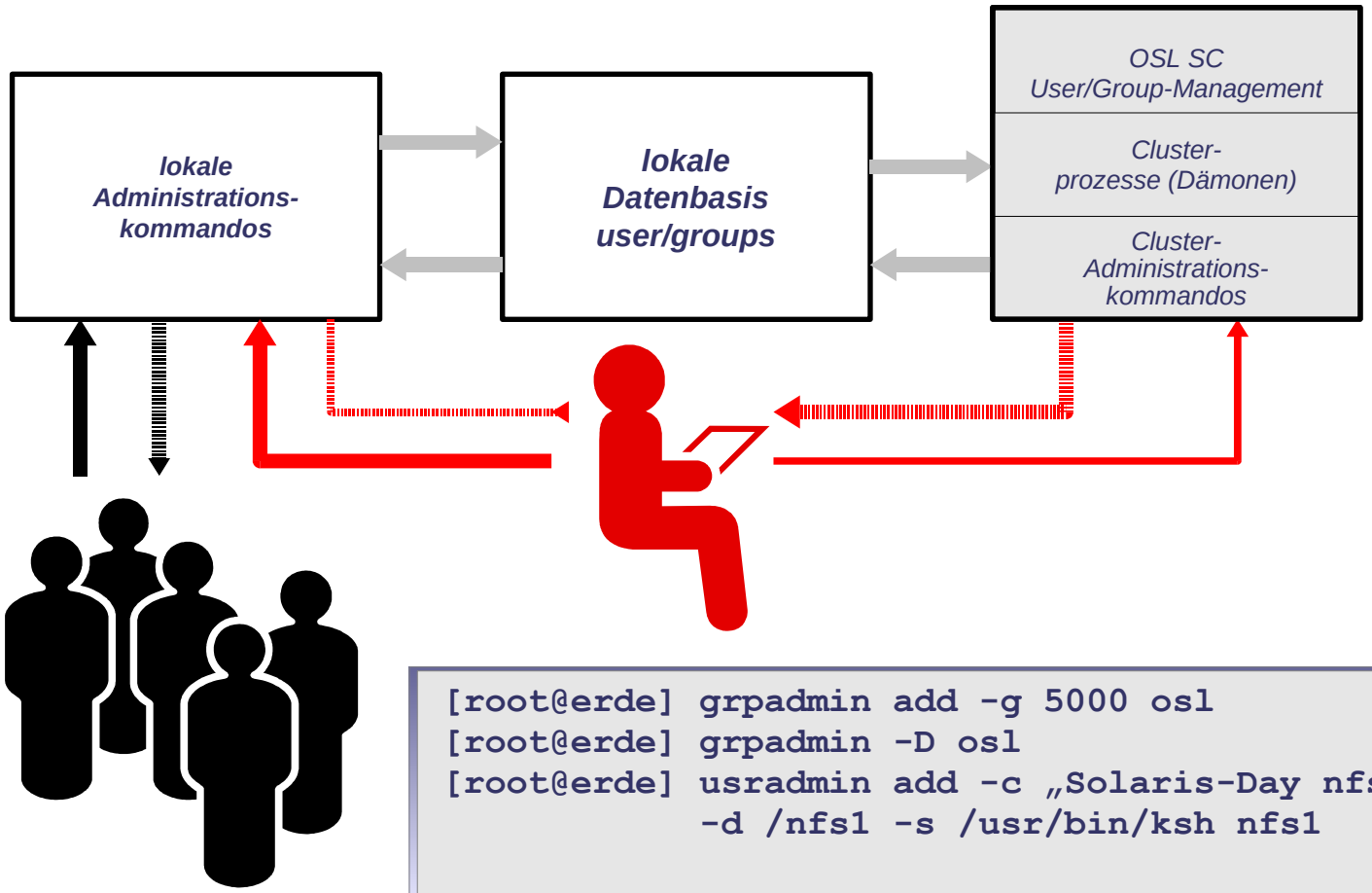
Globale Nutzerverwaltung komplettiert virtualisierte Ablaufumgebungen

- *geeignet für Server / Application Service User*
- *Unabhängig von externen Services wie NIS/LDAP/ADS*
- *Vermeidung von Konflikten, Synchronisation, automatische Reparatur*
- *User kann einer Applikationen zugeordnet werden*
- *Crontab und Login-Möglichkeit wandern mit der Applikation*
- *auch nach Neuinstallation sofort wieder verfügbar*



Global User Management – so funktioniert es

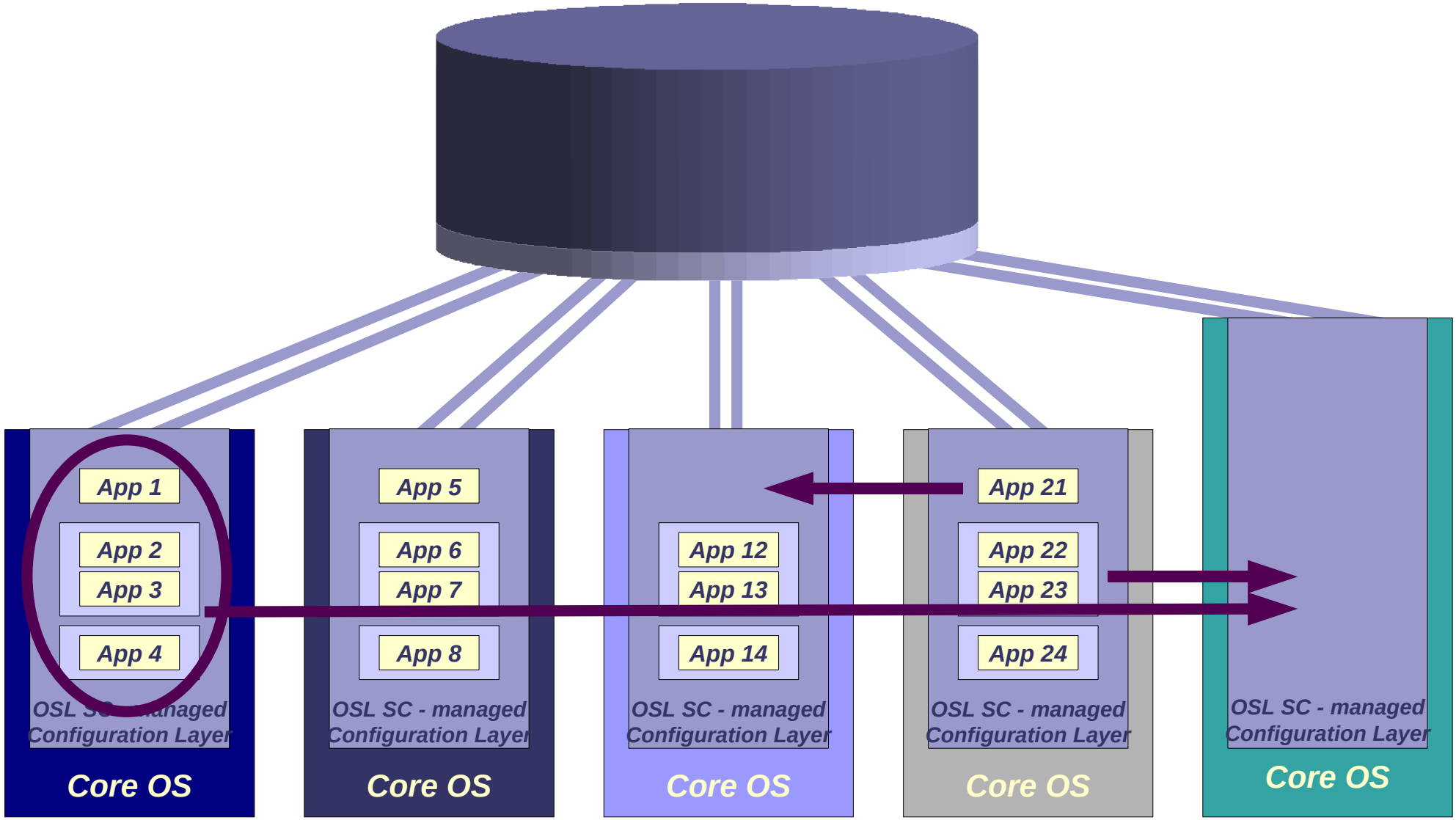
Wir legen beispielhaft eine Gruppe und einen Nutzer global an




```
[root@erde] grpadmin add -g 5000 osl
[root@erde] grpadmin -D osl
[root@erde] usradmin add -c „Solaris-Day nfs1“ -u 5000 -g 5000 \
            -d /nfs1 -s /usr/bin/ksh nfs1
```

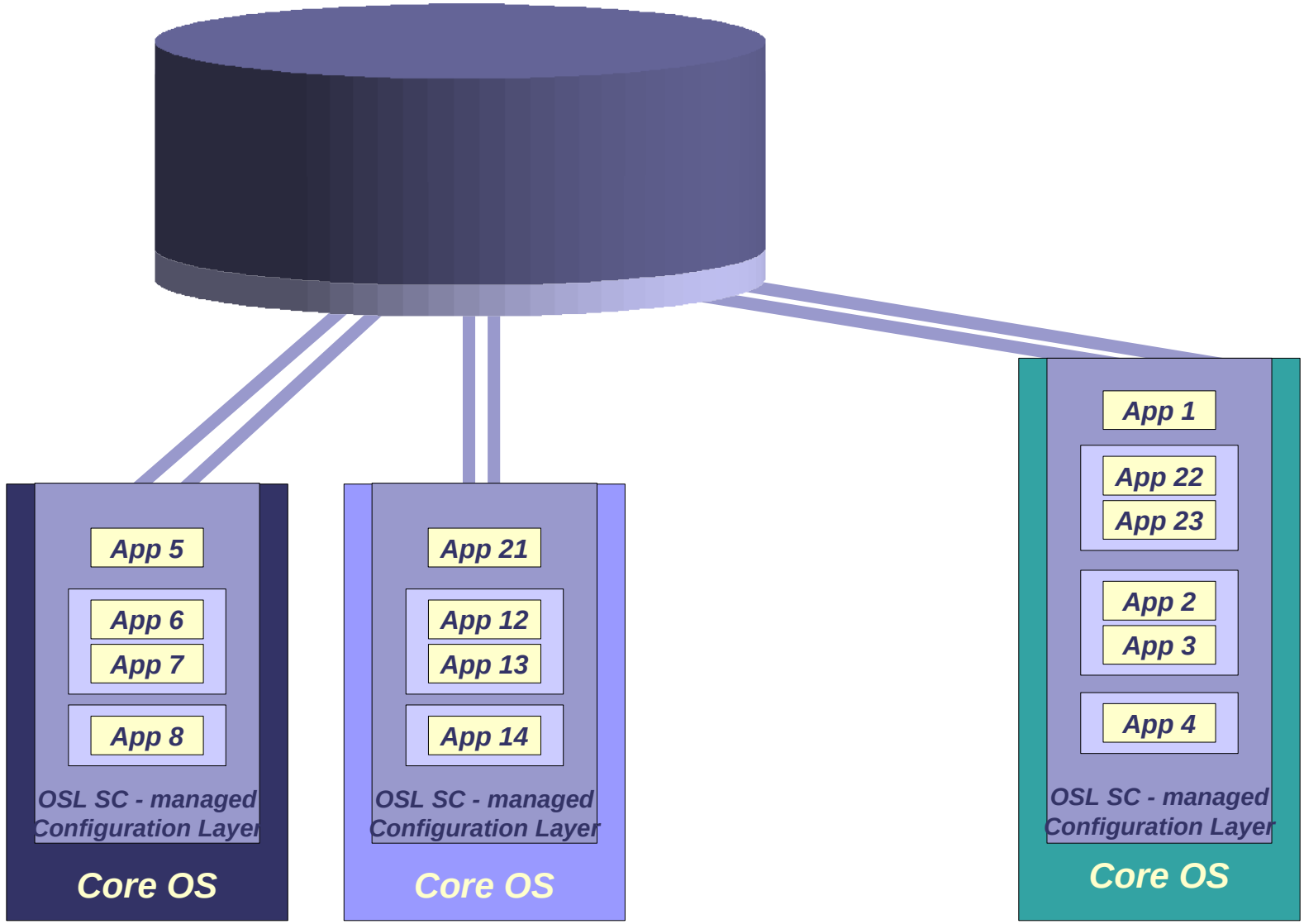
Was sind dynamische Knotengruppen?

Hardware tauschen und Cluster verändern ohne Konfigurationen anzupassen



Was sind dynamische Knotengruppen?

Hardware tauschen und Cluster verändern ohne Konfigurationen anzupassen



Automatisierung und Hochverfügbarkeit

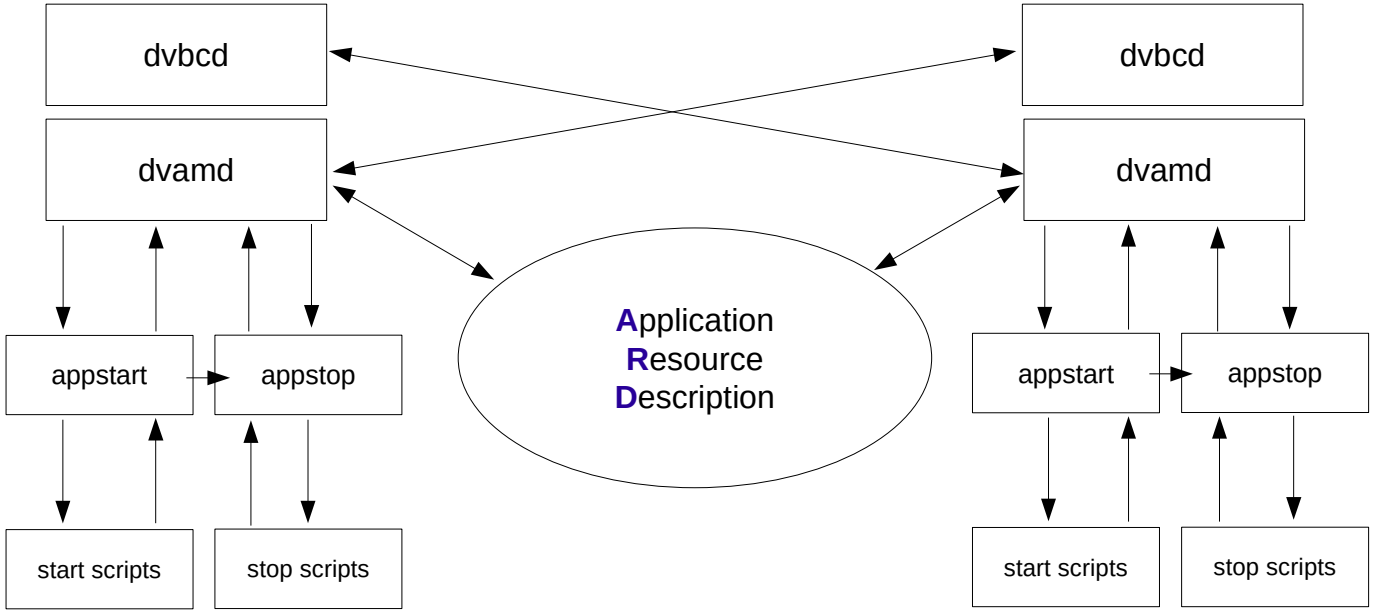
Nächster Schritt: Hochverfügbarkeit

Von manuellen zu automatischen Abläufen

*Anstelle der expliziten manuellen Steuerung können wir die Steuerung auch dem Cluster überlassen. Dazu gibt es drei Direktiven (= **target state**) für die Behandlung einer Applikation durch die Cluster Engine:*

no control	*keine Steuerung durch den Cluster, wohl aber Überwachung*
up	*Cluster versucht, Anwendung am Laufen zu halten*
down	*Cluster beendet Anwendung, falls nötig.*

Damit ist zugleich die Hochverfügbarkeit implementiert.



Was man noch wissen könnte

Prioritäten und Knotenlisten

prio 1	node 1	node 2	node 3	node 4	node 5	node 6	node 7	node 8
prio 2	node 4	node 3	node 1	node 6				
prio 3								
prio 4								
prio 5								
prio 6								
prio 7								
prio 8								
prio 9								
prio 10								
prio 11								
prio 12								
prio 13								
prio 14								
prio 15								
prio 16								

- *Anwendungen sind eineindeutig clusterweite Prioritäten zugeordnet*
  - *welche Anwendung wird zuerst gestartet?*
  - *Anwendungen höherer Priorität können bei Bedarf solche mit niederer Priorität verdrängen*
- *die Position eines Knotens in der Knotenliste einer Applikation bestimmt die Affinität der Applikation zum jeweiligen Knoten*
- *Es sind exklusive (default) und parallele Ausführungsmodi möglich*
- *Jeder Knoten kann mehreren Applikationen zugeordnet sein*
- *Bei der Auswahl des Zielknotens können Performanceaspekte berücksichtigt werden*
- *Es sind dynamische Knotengruppen möglich*

Praxisteil 4

- *Monitoring*
- *automatische Umschaltung und Knoteneliminierung*

Applikationen im Cluster

Praxiteil 4: Gestaltung der Hochverfügbarkeit

- **Was soll bei einem Ausfall im Rechenzentrum passieren**
  - Die Anwendung soll schnell wieder laufen
  - Es dürfen keine Daten verloren gehen
- **Wie erreicht man diese Anforderungen in einem Cluster?**
  - Die Daten und die Applikationen liegen im shared Storage
  - Die Applikationsbeschreibung ist global verfügbar
  - Es sind ausreichend freie Ressourcen vorhanden
  - Ein anderer Host bemerkt den Ausfall
  - Der fehlerhafte Knoten ist mit Sicherheit beendet

Anstelle der expliziten manuellen Steuerung können wir die Steuerung auch dem Cluster überlassen. Dazu gibt es drei Direktiven (= target state) für die Behandlung einer Applikation durch die Cluster Engine

- ***no control*** ***keine Steuerung durch den Cluster, wohl aber Überwachung***
- ***up*** ***Cluster versucht, Anwendung am Laufen zu halten***
- ***down*** ***Cluster beendet Anwendung, falls nötig.***

Damit ist zugleich die Hochverfügbarkeit implementiert.

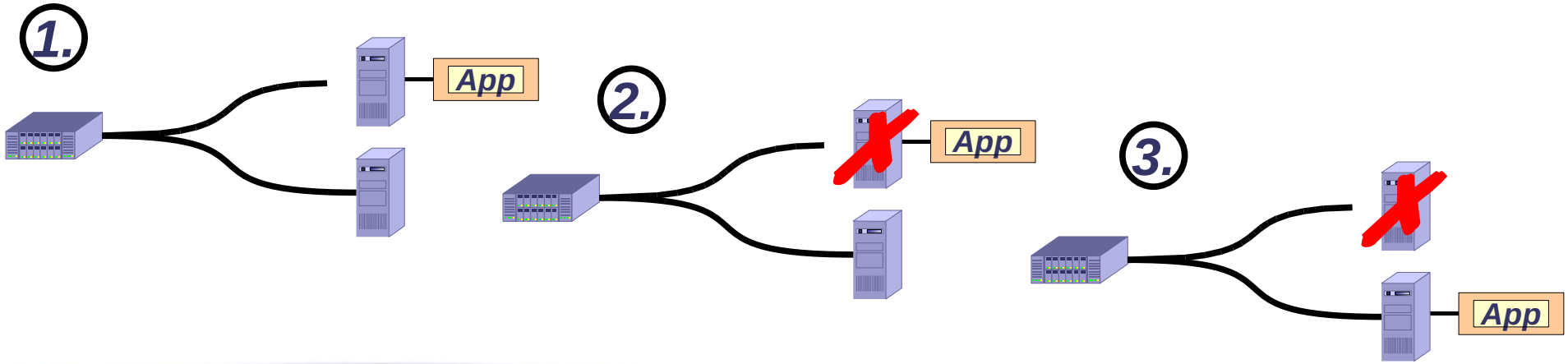
Applikationen im Cluster

Praxiteil 4: Gestaltung der Hochverfügbarkeit

- **Über den Status einer Applikation kann festgelegt werden, dass diese im Cluster hochverfügbar sein soll.**
  - **Möglicher Status: no_control, up, down**
  - **Setzen des Applikationsstatus**

```
#> appadmin -s up clock
```

  - **Applikationen mit Status "up" werden im Cluster verfügbar gehalten**
    - *Eliminierung fehlerhafter Knoten*
    - *Übernahme der Applikationen*



Applikationen im Cluster

Praxiteil 4: Gestaltung der Hochverfügbarkeit

- **Setzen des Status einer Applikation auf "up"**

  Applikationsstatus setzen: `appadmin -s up clock`

- **Überprüfen ob mind. 1 Node "ready for application" ist**

  Applikationsinformationen anzeigen: `appadmin -qv clock`

- **Überprüfen ob die Node-Power-Control Routinen definiert sind**

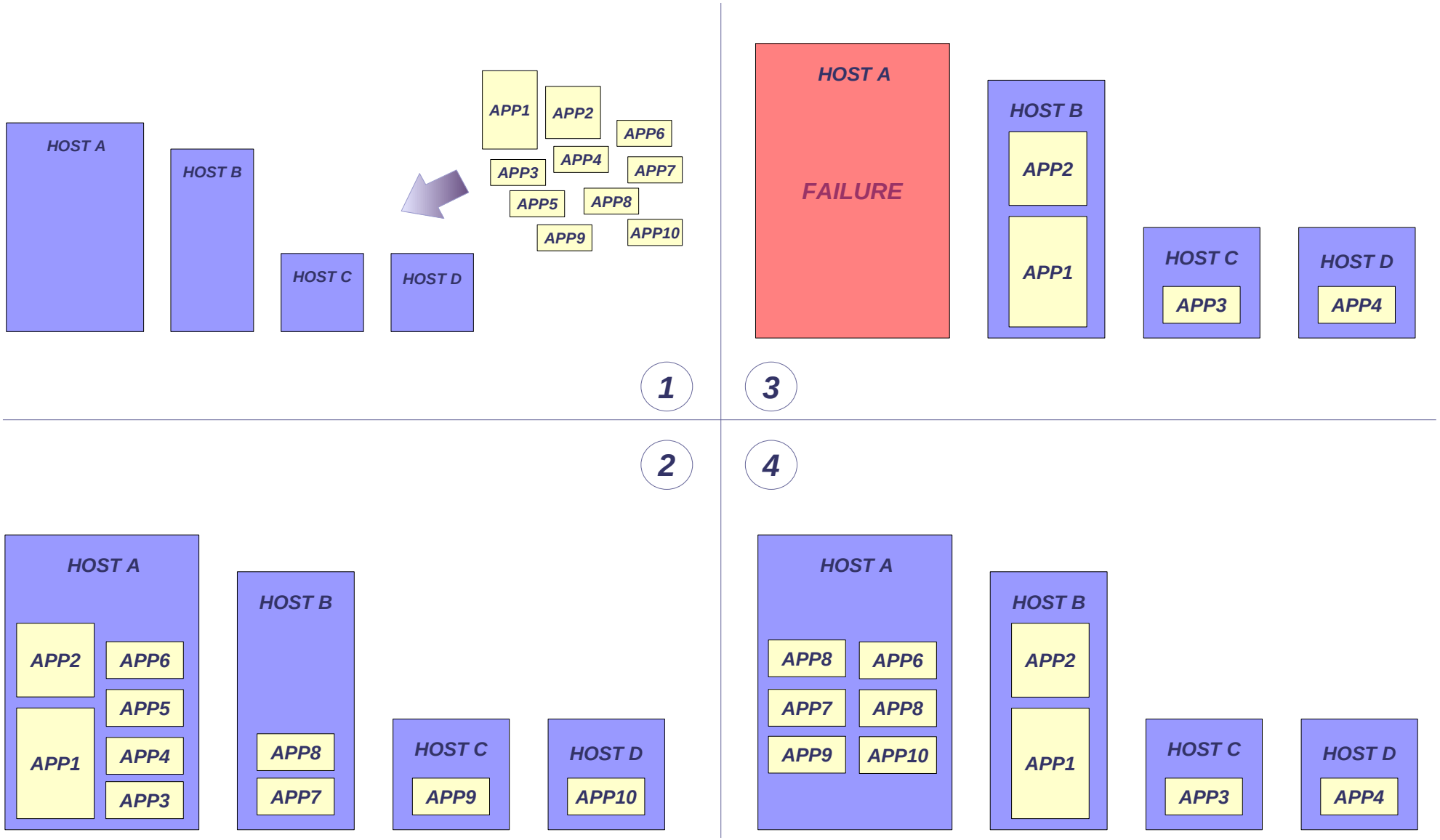
  Detailierte Nodeinformationen anzeigen: `ndadmin -lvvvv`

- **Was passiert nun, wenn der produktive Server einen Panic hat?**
  - **Ein verbleibender Node registriert, dass der Knoten Status "FAULTY" wird**
  - **Der Node wird mit eliminiert und das Ergebnis wird verifiziert**
  - **Die Applikation wird von einem verbliebenen Node übernommen**

Ende Praxisteil 4

Und das geht natürlich auch

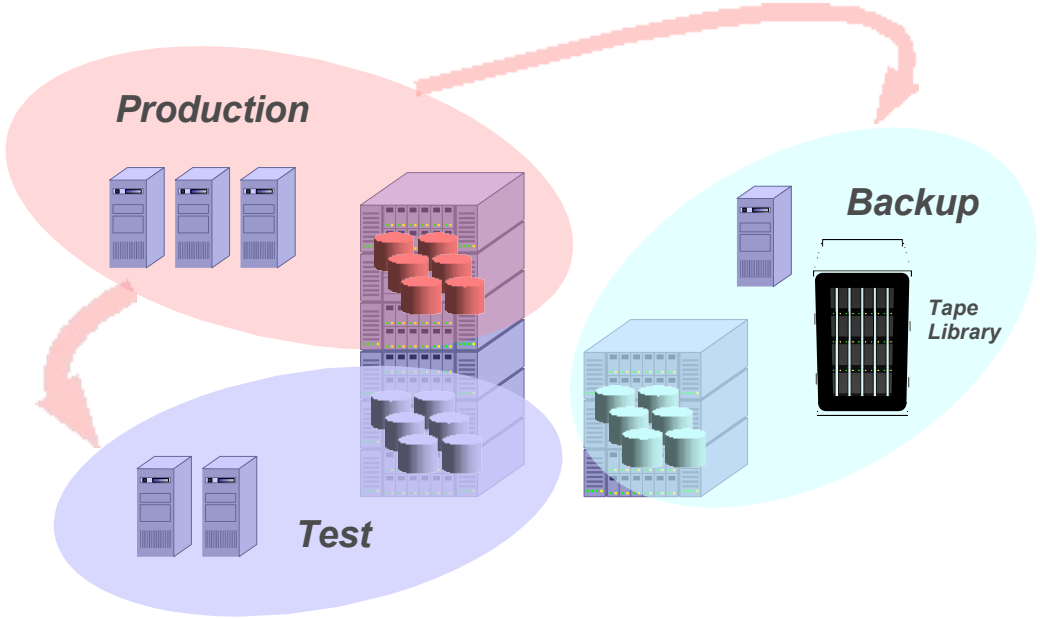
High Availability und Adaptive Computing



Extended Data Management
&
Application Aware Storage Management

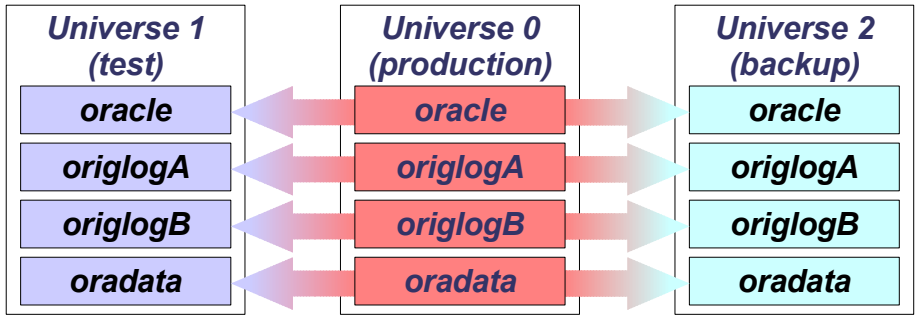
Storage-Universen im OSL-Storage-Cluster

Abbildung logischer Beziehungen



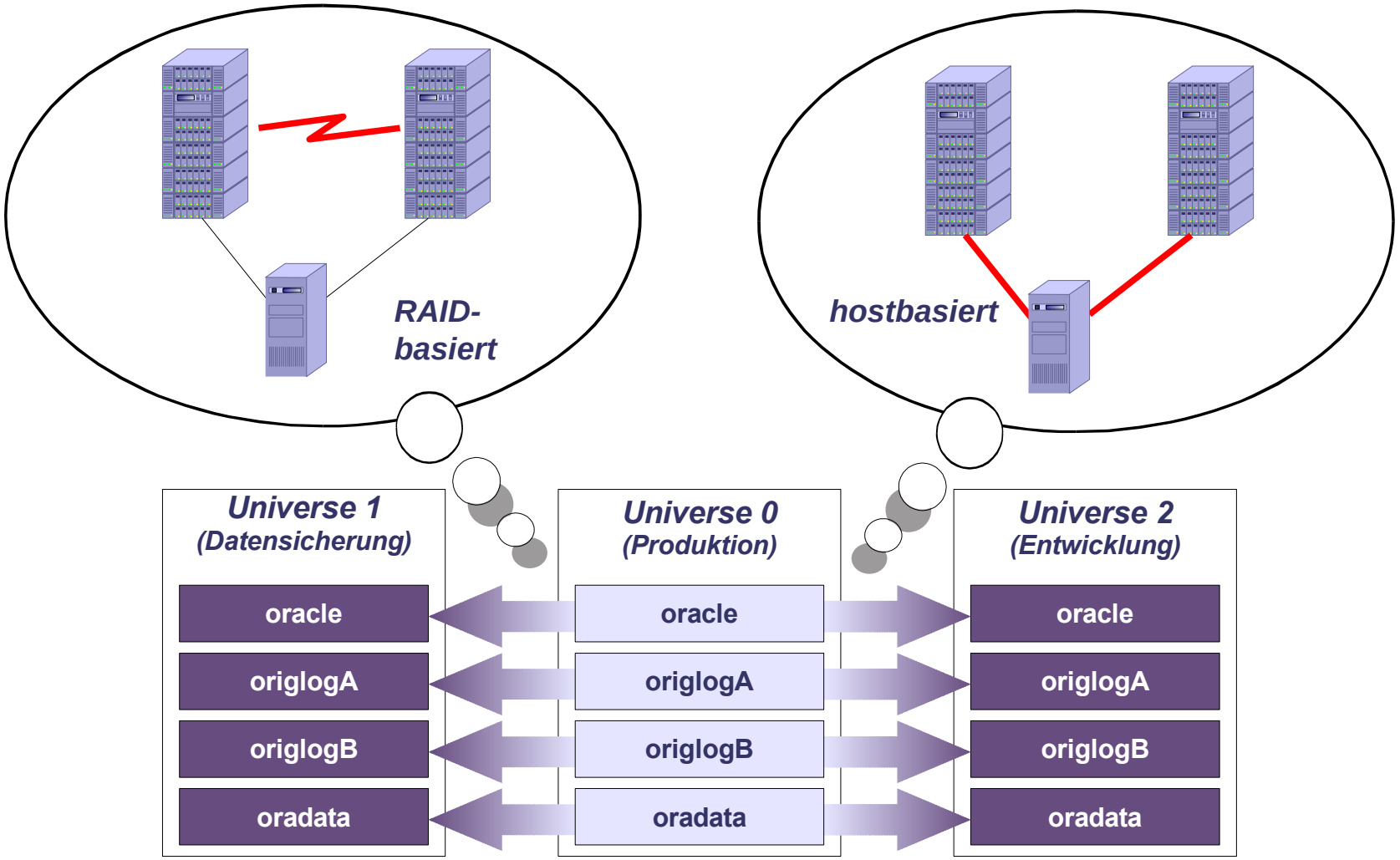
- *OSL Storage Universen bilden die Aufteilung der Ressourcen nach der Art der Nutzung ab*
- *Kopien eines Originals können jederzeit erstellt werden, auch unter Beibehaltung des Namens*
- *OSL Storage Cluster besitzt Informationen über die logischen Beziehungen zwischen den Universen*

- *Volle Integration der Universen in das Betriebssystem*
- *leichte Identifikation anhand des Namens*
- *Zugriff auf jede Instanz*
  - jederzeit
  - von jedem Host aus



Erzeugung von Storage-Universen

Zwei Möglichkeiten

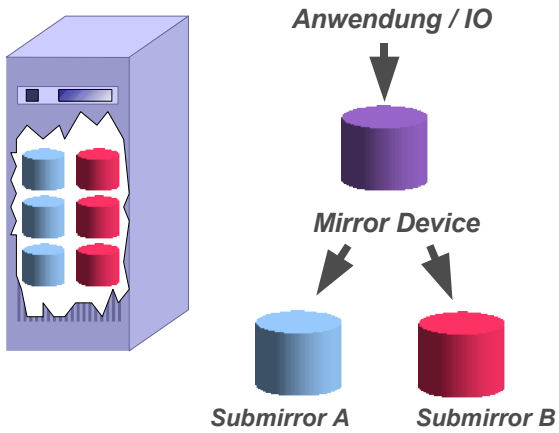


Warum noch eine Spiegel-Software?

Das gab es doch schon vor 30 Jahren

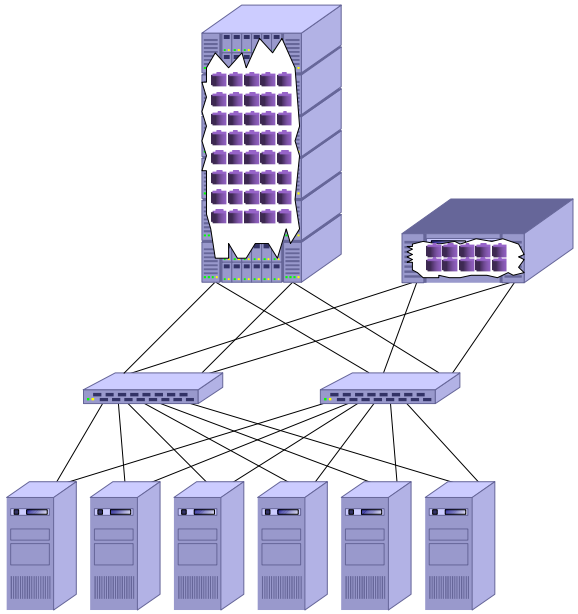
Stimmt! Und seitdem sah das so aus:

- *Designschwerpunkt:*
  Schutz vor Plattenausfällen
- *statische Konfiguration*
- *Implementierung meist über hierarchisch organisierte Geräteknoten*
- *geringe Zahl von Geräten*
- *i. d. R. Administration für einen Rechner*
- *aufwendige Administration / OLR-Operationen*



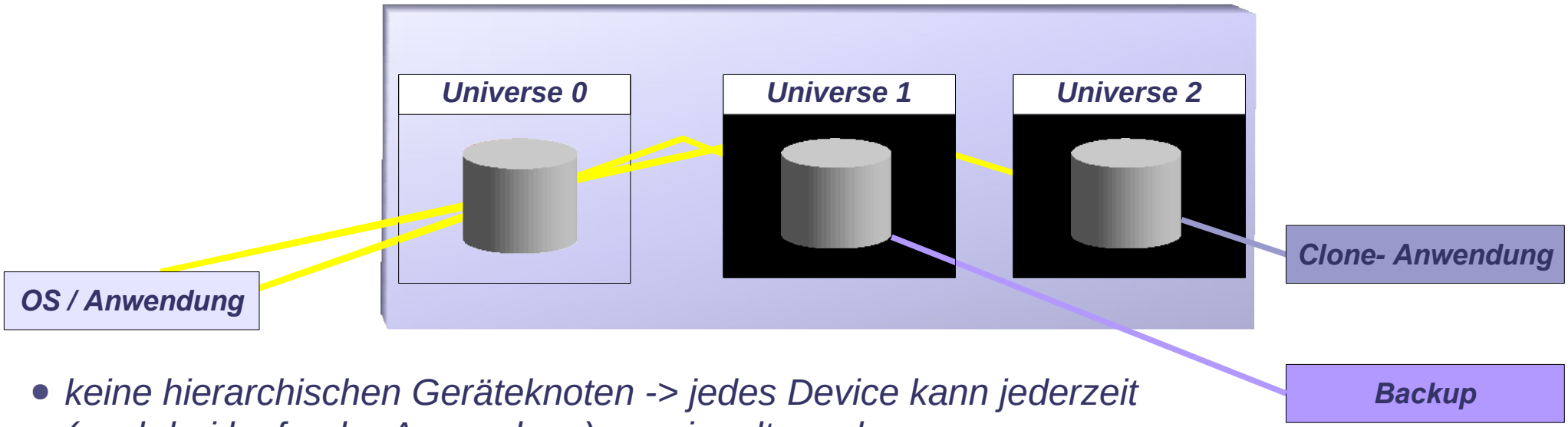
Aber die Anforderungen haben sich geändert:

- *Schutz vor Plattenausfällen sollte keine Rolle mehr spielen*
- *Hostbasierte Spiegelung heute für:*
  - *Backup und Backup to Disk*
  - *schnellen Wiederanlauf nach logischen Fehlern*
  - *Clonen von Anwendungen (etwa Produktion auf Test)*
  - *ggf. „Mißbrauch“ für Disaster Precaution*
- *große Zahl von Geräten*
- *enge Verknüpfung mit der Anwendung*
- *Clustertauglichkeit*



Datenspiegelung mit XDM

ist auf RAID-Systeme, heutige RZ-Infrastrukturen und Anforderungen zugeschnitten

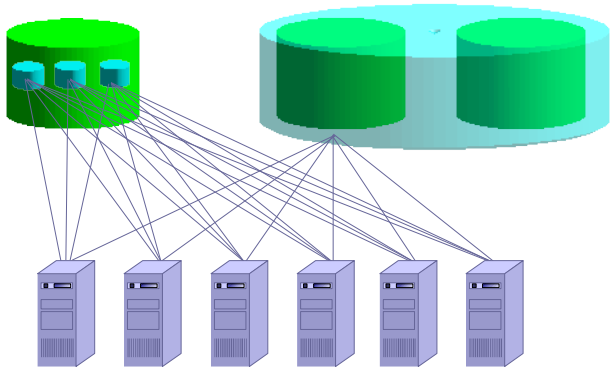


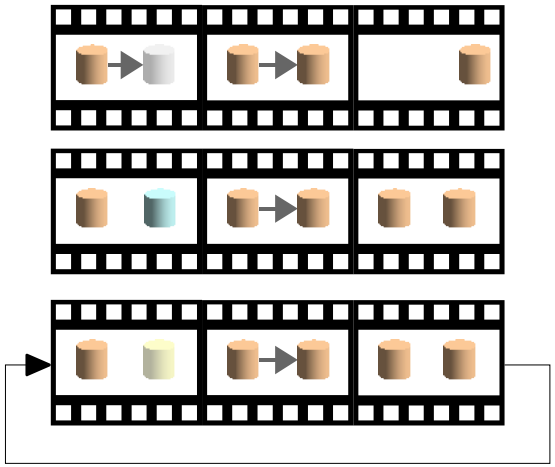
- *keine hierarchischen Geräteknoten -> jedes Device kann jederzeit (auch bei laufender Anwendung) gespiegelt werden*
- *Identische Gerätenamen für Master und Image dank OSL Storage Universen*
- *Überbrückung von Ausfällen des Masters (wenn Images im Status „connected“)*
- *Nach Disconnect der Images Zugriff auf diese vom selben oder von anderen Clusternodes*
- *spezielles Sync-Verfahren für maximale Performance bei minimalem Einfluß auf Applikation*
- *Atomic Disconnect für beliebig zusammenstellbare Volumes und Volume-Gruppen*
- *synchrone, asymmetrische IO-Strategie mit Berücksichtigung wahrscheinlicher Anwendungsumgebungen: RAID-to-RAID Kopie, niedrigere Performance des Image-RAIDs*
- *Master und Images können unterschiedliche Volume-Typen und -Größen haben*

Speichermanagement mit OSL Storage Cluster

Blockbasierte Virtualisierung – bedarfsgerecht und zuverlässig

- 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

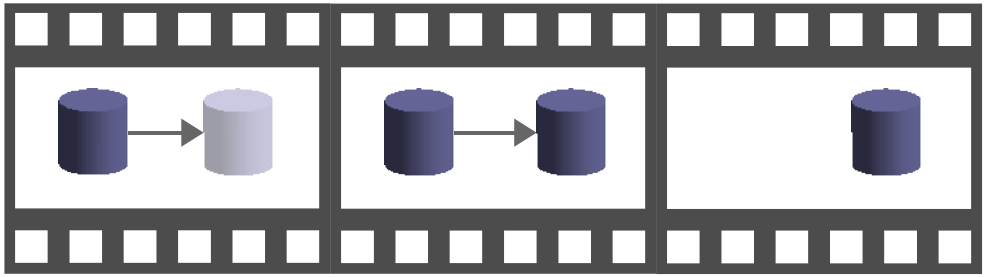
Online-Datenkopien auf wahlfreie Ziele
atomare Operationen für mehrere Volumes

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

Ernst gemeint: Flexibilität im Storage-Management

Beispiel: Online-Migration von Daten zwischen RAID-Systemen



online Daten verschieben / reorganisieren
automatische Priorisierung Anwendungs-IO

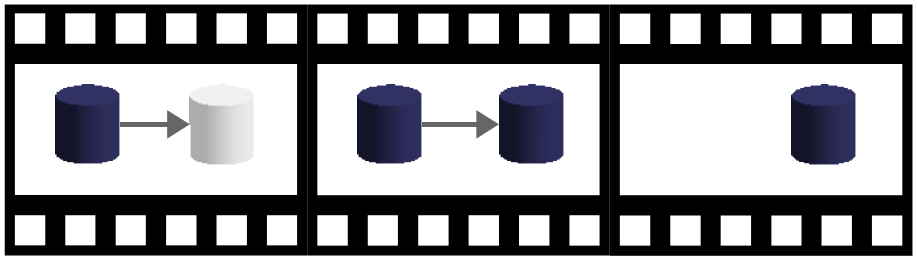
- *Zielkonfiguration der Migration als Shadow-Volume konfigurieren*
- *Move anstarten*
- *Fertig*

```
[root@erde] avadmin -lvv nfs1
0 nfs1 2097152 of 2097216 blocks "simple" in 1 pieces, 32 block clusters
  [ 1] new1 [0...2097215]
[root@erde] smgr -c ziel -S 1g old1
[root@erde] avmove nfs1 ziel
[root@erde] avadmin -lvv nfs1
0 nfs1 2097152 of 2097216 blocks "simple" in 1 pieces, 32 block clusters
  [ 1] old1 [0...2097215]
```

Praxisteil 5

- *Move*
- *Clone*
- *Mirror*
- *applikationsbezogene Aktionen*

OSL Storage Cluster
Praxisteil 5: Spiegeln, Clonen, Verschieben von Speicher

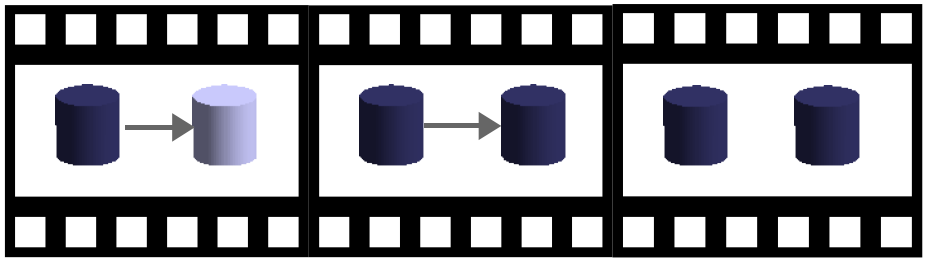


- **Verschieben von Volumes (avmove)**
  - Vorgehensweise
    - Erstellen eines Shadow-Volumes mit gleicher Größe
    - Synchronisation mit dem Quellvolume
    - Shadow-Volume schlüpft in die Rolle des Quellvolumes
  - Möglichkeiten
    - Online Umzug einiger Hot-Spots auf schnellen Speicher
    - Speichermigration bei Neuanschaffung von RAID-Systemen

- ***Verschieben von Volumes (avmove)***

```
[root@sol-1] avadmin -lv mein_vol
0 mein_vol 2097152 of 2097216 blocks "simple" in 1 pieces, 32 block clusters
[root@sol-1] smgr -c shadow -S 2097152 -F stripe -o pieces=2
[root@sol-1] avmove mein_vol shadow
INFO (avmove): moving in universe 0 from mein_vol to shadow
[root@sol-1] avmove -l
0 mein_vol moving to shadow (3.53% done)
[root@sol-1] avadmin -lv shadow
[root@sol-1] avadmin -lv mein_vol
0 mein_vol 2097152 of 2097216 blocks "stripe" in 2 pieces, 32 block clusters
[root@sol-1] avmove -l
```

OSL Storage Cluster
Praxisteil 5: Spiegeln, Clonen, Verschieben von Speicher



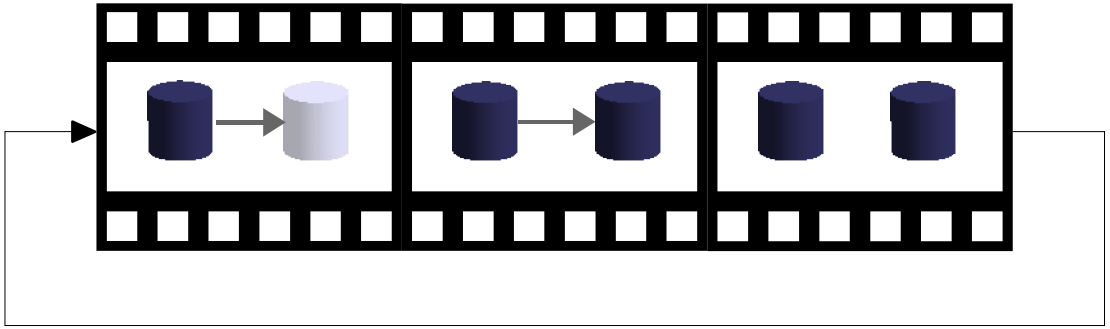
- **Duplizieren von Volumes (avclone)**
  - Vorgehensweise
    - Erstellen eine Cloneziels
    - Synchronisation mit dem Quellvolume
    - Zeitkonsistentes absplitten von einem oder mehrern Clone-Zielen
  - Möglichkeiten
    - Systemkopien für Tests oder QA

- ***Duplizieren von Volumes (avclone)***

```
[root@sol-1(iscsi)] avadmin -lv mein_vol
0 mein_vol 2097152 of 2097216 blocks "stripe" in 2 pieces, 32 block clusters
[root@sol-1(iscsi)] smgr -c dein_vol -S 2097152
[root@sol-1(iscsi)] avclone mein_vol dein_vol
INFO (avclone): cloning from mein_vol@0 to dein_vol@0
[root@sol-1(iscsi)] avclone -l
mein_vol@0 cloning to dein_vol@0 (3.39% done)
[root@sol-1(iscsi)] avclone -l
mein_vol@0 cloning to dein_vol@0 (completed)
[root@sol-1(iscsi)] avclone -d dein_vol@0
```

  - *Das Abtrennen des Speichers ist auch Zeitkonsistent für mehrere Volumes möglich*
    - *Startfähige Clones von kompletten Applikationen*
    - *Einfache und vollständig integrierte Systemkopie von Oracle Datenbanken oder SAP Systemen*

OSL Storage Cluster
Praxisteil 5: Spiegeln, Clonen, Verschieben von Speicher



- **Spiegeln von Volumes (avmirror)**
  - Vorgehensweise
    - Anlegen eines Spiegels (selber Name – anderes Universum)
    - Synchronisation mit dem Mastervolume
    - Auftrennen und benutzen des Spiegels
    - Inkrementelle Nachsynchronisation des Spiegels
  - Möglichkeiten
    - Systemkopien für Tests oder QA
    - Backup
    - Hochverfügbarkeit

- ***Spiegeln von Volumes (avmirror)***

```
[root@sol-1] avadmin -e set_role master mein_vol
[root@sol-1] avadmin -lv mein_vol
0 mein_vol 2097152 of 2097216 blocks "stripe,master" in 2 pieces, 32 block clusters
[root@sol-1] smgr -c mein_vol@1 -S 2097152 -R image
[root@sol-1] avmirror -l mein_vol
0       mein_vol ( stripe, 2pc, 1024m) MASTER SOURCE -----  synchronized
1       mein_vol ( simple, 1pc, 1024m) image    -    -----  disconnected
[root@sol-1] avmirror -c mein_vol@1
INFO (avmirror): "1 mein_vol" connected as mirror instance, starting sync.
[root@sol-1] avmirror -l mein_vol
0       mein_vol ( stripe, 2pc, 1024m) MASTER SOURCE s----  synchronized
1       mein_vol ( simple, 1pc, 1024m) image  target s----  synchronized
[root@sol-1] avmirror -d mein_vol@1
INFO (avmirror): "1 mein_vol" disconnected from mirror (io logging).
[root@sol-1] avmirror -l mein_vol
0       mein_vol ( stripe, 2pc, 1024m) MASTER SOURCE s-1--  synchronized
1       mein_vol ( simple, 1pc, 1024m) image    -    s0123  disconnected
```

- **Spiegeln von Volumes (avmirror)**
  - Bis zu 3 Spiegelinstanzen pro Mastervolume
  - automatisches Failover zu synchronisierten Images bei Ausfall des Masters
  - Zeitkonsistentes Abtrennen von mehreren Images
    - Wiederanlauffähiges Backup
  - inkrementelle Synchronisation
  - Source Flag kann auf jeden abgetrennten Spiegel gesetzt werden
    - keine Änderungen an der Applikation notwendig, falls von einem Spiegel gestartet wird

Im OSL Storage Cluster 3.1 stehen die Applikationen im Vordergrund.

Ein Kunde meinte:

“Die Version 3.1 ist kaum noch ein Storage Cluster sondern schon ein Application Cluster”

Bei fast allen Clusteranfragen und Kommandos kann ein Applikationsname übergeben werden, um somit einen Bezug zu einer bestimmten Anwendung zu erhalten.

Im Rechenzentrum stellen sich viele Fragen...

Wieviel Speicherplatz belegt meine Applikation bp1@0?

#> smgr -qa bp1@0

```
#> smgr -qa bp1@0
used by bp1@0       :                  33544 MB          33 GB      0.032 TB

-------------------------------------------------------------------------------
TOTAL STORAGE POOL SUMMARY
free:           8603311612 bl          4200836 MB        4102 GB      4.006 TB
totl:          10359112860 bl          5058161 MB        4940 GB      4.824 TB
-------------------------------------------------------------------------------
```

Im Rechenzentrum stellen sich viele Fragen...

Ist meine Applikation bp1@0 vom derzeitigen Pfadausfall betroffen?

#> pvadmin -lvva bp1@0

```
#> pvadmin -lvva bp1@0
0 p01 (ok) 585920024 blocks over 1 path(s)
  >[ 1] (ok) /dev/rdsk/c1t5000402001EC04F4d1s1
0 p04 (ok) 585920024 blocks over 1 path(s)
  >[ 1] (ok) /dev/rdsk/c1t5000402001EC04F4d4s1
0 p02 (ok) 585920024 blocks over 1 path(s)
  >[ 1] (ok) /dev/rdsk/c1t5000402001EC04F4d2s1
0 p03 (ok) 585920024 blocks over 1 path(s)
  >[ 1] (ok) /dev/rdsk/c1t5000402001EC04F4d3s1
INFO (pvadmin): all physical volumes for application bp1@0 attached
```

Im Rechenzentrum stellen sich viele Fragen...

Welche Application Volumes gehören zur Applikation bp1@0?

#> avadmin -la bp1@0

```
#> avadmin -la bp1@0
0 ora_BP1
0 sapmnt_BP1
0 usrsap_BP1
0 origlogA_BP1
0 origlogB_BP1
0 mirrlogA_BP1
0 mirrlogB_BP1
0 saparch_BP1
0 sapdata1_BP1
0 sapdata2_BP1
INFO (avadmin): all volumes of application bp1@0 attached
```

Im Rechenzentrum stellen sich viele Fragen...

Welche Physical Volumes sind von bp1@0 belegt worden?

#> pvadmin -qa bp1@0

```
#> pvadmin -qa bp1@0
0 p01            with   bp1@0      using   4096m+  of    279g+  (  1.4%)
0 p04            with   bp1@0      using     20g+  of    279g+  (  7.4%)
0 p02            with   bp1@0      using   1924m+  of    279g+  (  0.7%)
0 p03            with   bp1@0      using   6344m+  of    279g+  (  2.2%)
```

Im Rechenzentrum stellen sich viele Fragen...

Wie erzeuge ich ein neues Application Volume für bp1@0?

#> smgr -c volume -S 1g -a bp1@0

```
#> smgr -c volume -S 1g -a bp1@0
#> avadmin -lvvh volume
0 volume 1024m of 1024m+ "simple" in 1 pieces, 16k clusters
  [ 1] p01 [24658432...26755647]
```

Im Rechenzentrum stellen sich viele Fragen...

Sind alle Spiegelvolumes von bp1@2 disconnected?

#> avmirror -qa bp1@2

```
#> avmirror -qa bp1@2
2        ora_BP1 ( simple, 1pc, 4096m) image    -    s0123  disconnected
2     sapmnt_BP1 ( simple, 1pc,  500m) image    -    s0123  disconnected
2     usrsap_BP1 ( simple, 1pc,  500m) image    -    s0123  disconnected
2   origlogA_BP1 ( simple, 1pc,  200m) image    -    s0123  disconnected
2   origlogB_BP1 ( simple, 1pc,  200m) image    -    s0123  disconnected
2   mirrlogA_BP1 ( simple, 1pc,  200m) image    -    s0123  disconnected
2   mirrlogB_BP1 ( simple, 1pc,  200m) image    -    s0123  disconnected
2    saparch_BP1 ( simple, 1pc, 6144m) image    -    s0123  disconnected
2   sapdata1_BP1 ( simple, 1pc,   20g) image    -    s0123  disconnected
9 mirror instances (0 av missing), 0 synced, 5 active, 0 need maintenance
```

OSL Storage Cluster
Praxisteil 5: Applikationsbezogene Speicherverwaltung

- *Die applikationsbezogene Speicherverwaltung entlastet den Administrator*
  - *Es ist nicht mehr nötig bestimmte Volumes von Hand zusammenzusuchen*
  - *SLA können besser eingehalten werden*
  - *Der Überblick über die Storage Infrastruktur bleibt auch bei großen Installationen gegeben und das von jedem Clusternode aus*
  - *Durch die einheitlichen Kommandosyntax erhält man schnell die gewünschten Ergebnisse*

#> avadmin -lva <APP>

#> avmirror -qa <APP>

#> smgr -qa <APP>

#> smgr -c volume -S 1g -a <APP>

#> pvadmin -lvva <APP>

Eigenes Tool für Spiegeloperationen ganzer Applikationen

appmirror

Funktionen:

- *Anlegen neuer Applikationsspiegel*
- *Synchronisation und Abtrennen von Spiegeln*
- *Update der Spiegelapplikationen*
- *Sperren und entsperren von Spiegeln*
- *Setzen des Source-Flags auf alle Volumes der Spiegelapplikation*
- *Unterstützung bei der Rückwärtssynchronisation*

Einfache Syntax von appmirror:

```
#> appmirror -?
Usage: appmirror -t tgt_app -a action,...

valid actions are:
  create          - create mirror application including required volumes
  update          - update mirror application and create required volumes
  nolog           - create or disconnect images without io-logging
  connect         - connect images of mirror target application
  disconnect      - disconnect images of mirror target application
  lock            - lock application images in current state
  unlock          - remove state locks from applications volumes
  delete          - delete application including volumes
  setsource       - set source to volumes of target application
  priority=n      - create application with priority n
  replace         - replace all ard data and create required volumes
```

Beispiele:

Anlegen einer Spiegelapplikation im Universum 2

```
#> appmirror -t clock@2 -a create
```

Synchronisation und Logging-Disconnect einer Spiegelapplikation mit vorhergehndem Update

```
#> appmirror -t clock@2 -a update,connect,disconnect
```

Sperren einer Spiegelapplikation

```
#> appmirror -t clock@2 -a lock
```

Entsperren einer Spiegelapplikation

```
#> appmirror -t clock@2 -a unlock
```

Setzen des Source Flags auf eine Spiegelapplikation

```
#> appmirror -t clock@2 -a setsource
```

Vorteile von appmirror:

- *Sofort startfähige Applikationsspiegel*
- *Kein umständliches hantieren mit XVCs*
- *Atomarer Disconnect aller Volumes einer Applikation*
- *Möglichkeit des Updates von Spiegelapplikationen*
- *1. Schritt zum erfolgreichen On-Disk-Backup*
- *Einfaches Erstellen von Systemkopien mit inkrementeller Nachsynchronisation*

Einsatz als Backupwerkzeug:

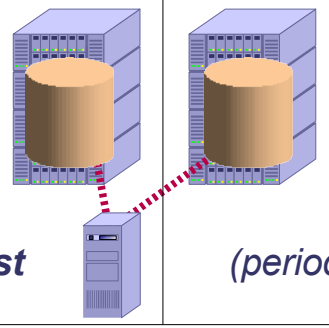
- *dvamb2d – Backup-to-Disk mit Pre- und Postprocessing*
- *dvamsave – Backup auf Tape ohne Belastung der produktiven Applikation*

Ende Praxisteil 5

Und warum überhaupt noch Spiegel?

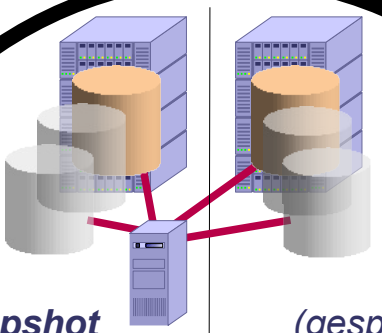
Spiegel vs. Snapshots – l… …n

Snapshots und Datenkopien im Vergleich



incr. per Host (periodischer Sync)

Speicherbedarf	200-300%
simultaner Zugriff Original + Kopie von mehreren Hosts	ja
Performanceanforderung Original-Speichersystem	hoch



ZFS-Snapshot (gespiegelt)

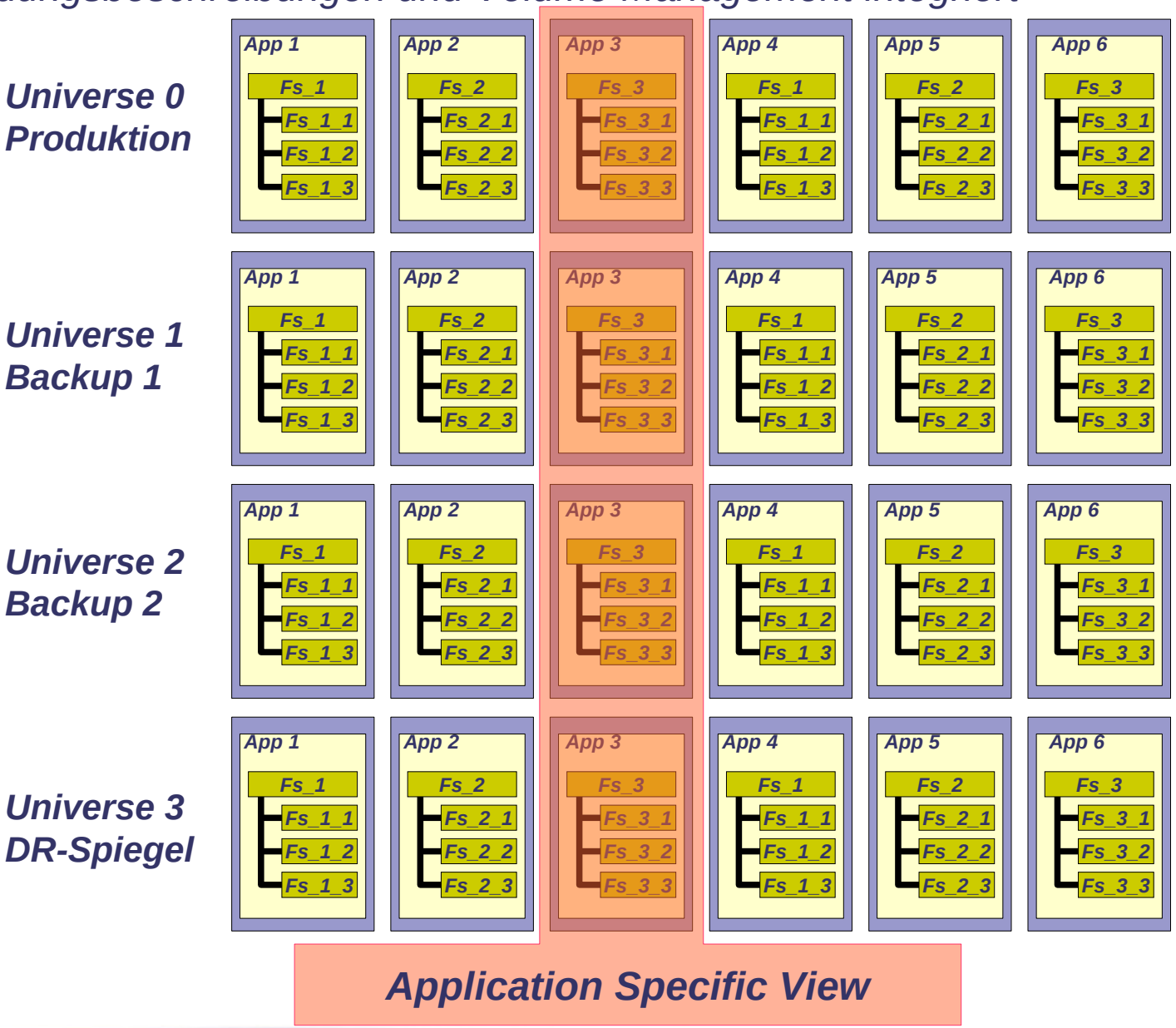
Speicherbedarf	250-400%
simultaner Zugriff Original + Kopie von mehreren	nein
Performanceanforderung Original-Speichersystem	sehr hoch

Speicherbedarf							
mit o.g. Speicherbedarf mögliche Kopien bzw. Snapshots							
OLTP-Performance Original				++			
OLTP-Performance bei gleichzeitig Zugriff auf Original und Kopie/Snap				+			
simultaner Zugriff auf Original und Kopie/Snap vom gleichen Host				ja			ja
simultaner Zugriff auf Original und Kopie/Snap von verschiedenen Hosts				ja	ja		nein
Backup-Performance Kopie/Snap (bei simultanem OLTP-Betrieb Original)				o	o	o	-
Integration mit Host-OS + Applikationen	o		++	-	-	+	+
Handhabung Komplettlösung	o	o	+	-	-	o/+	o/+
Performanceanforderung Original-Speichersystem	hoch	hoch	hoch	sehr hoch	sehr hoch	sehr hoch	sehr hoch
Performanceanforderungen an das Spiegel-Speichersystem (remote)	hoch	mäßig	mäßig	X	sehr hoch	X	sehr hoch
Verfügbarkeit Kopie/Snap nach User- oder SW-Fehler	X	++	+/++	++	++	+	+
Schutz gegen Ausfall Ausfall Orignal-Speichersystem	++	+	+	X	++	X	+/++
Brauchbarkeit Snap/Kopie als Sicherung	X	++	++	--	++	--/-	-/o
Belastung Host	sehr gering	sehr gering	gering	sehr gering	sehr gering	mäßig	mäßig

++ sehr gut + gut o mäßig - schwächer -- schlecht X entfällt/nicht vorhanden

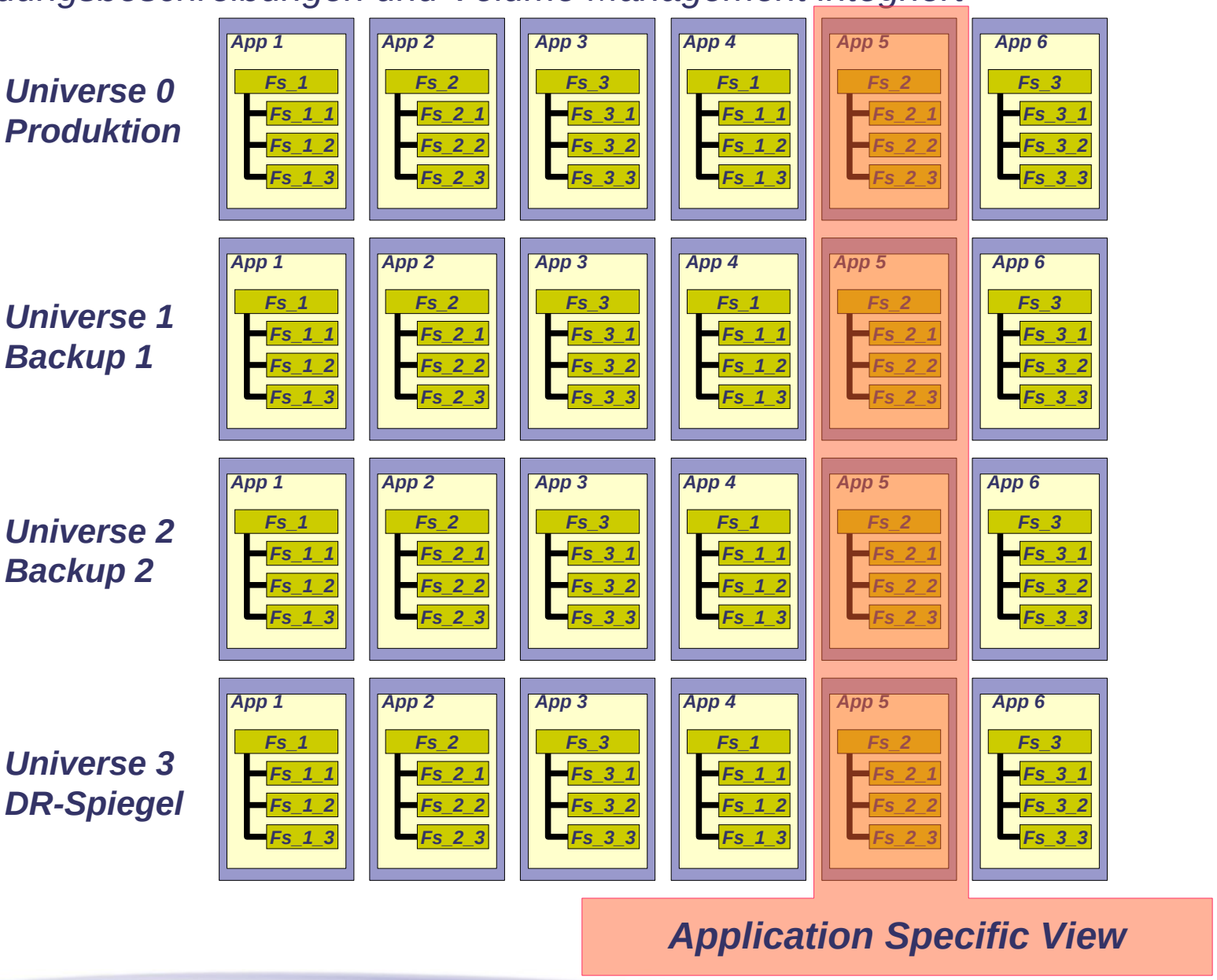
Application Aware Storage Management

Anwendungsbeschreibungen und Volume Management integriert



Application Aware Storage Management

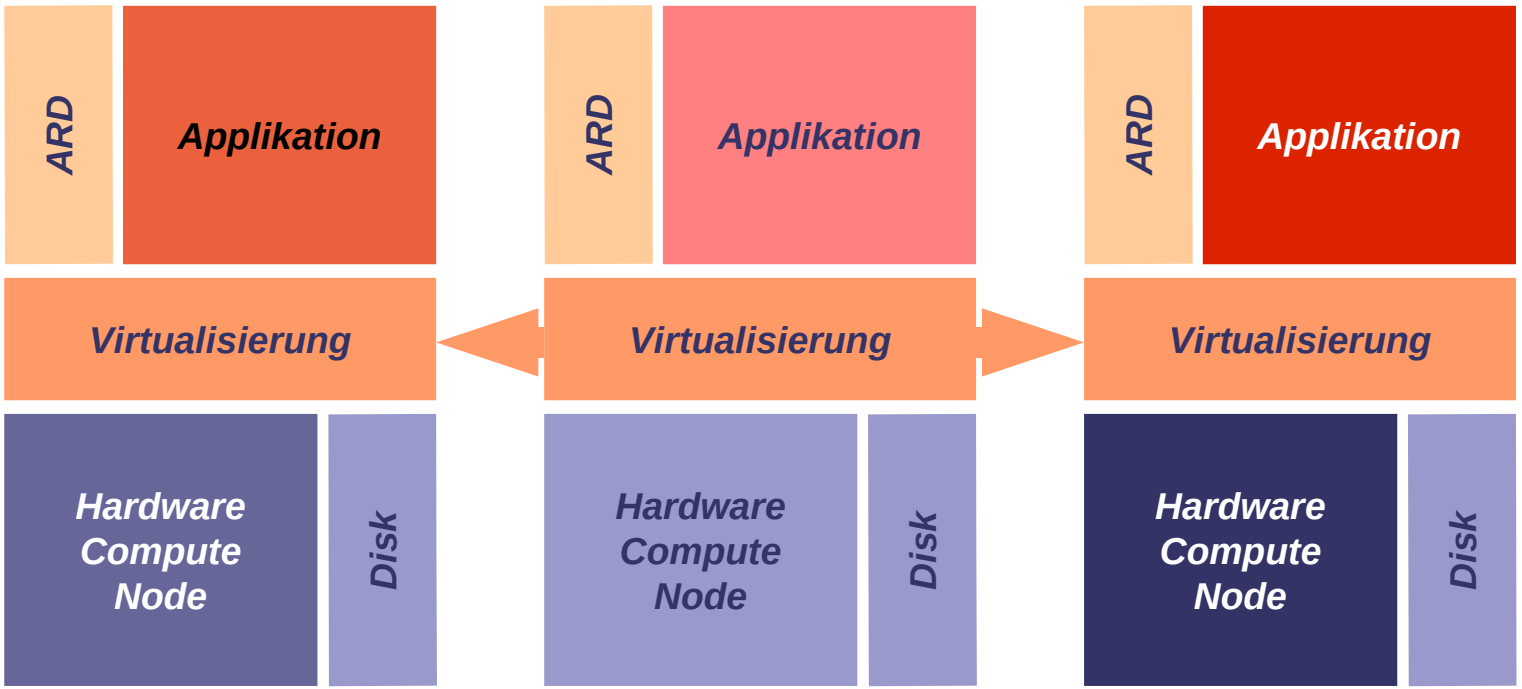
Anwendungsbeschreibungen und Volume Management integriert



EAS – die „technologiefreie“ Virtualisierung

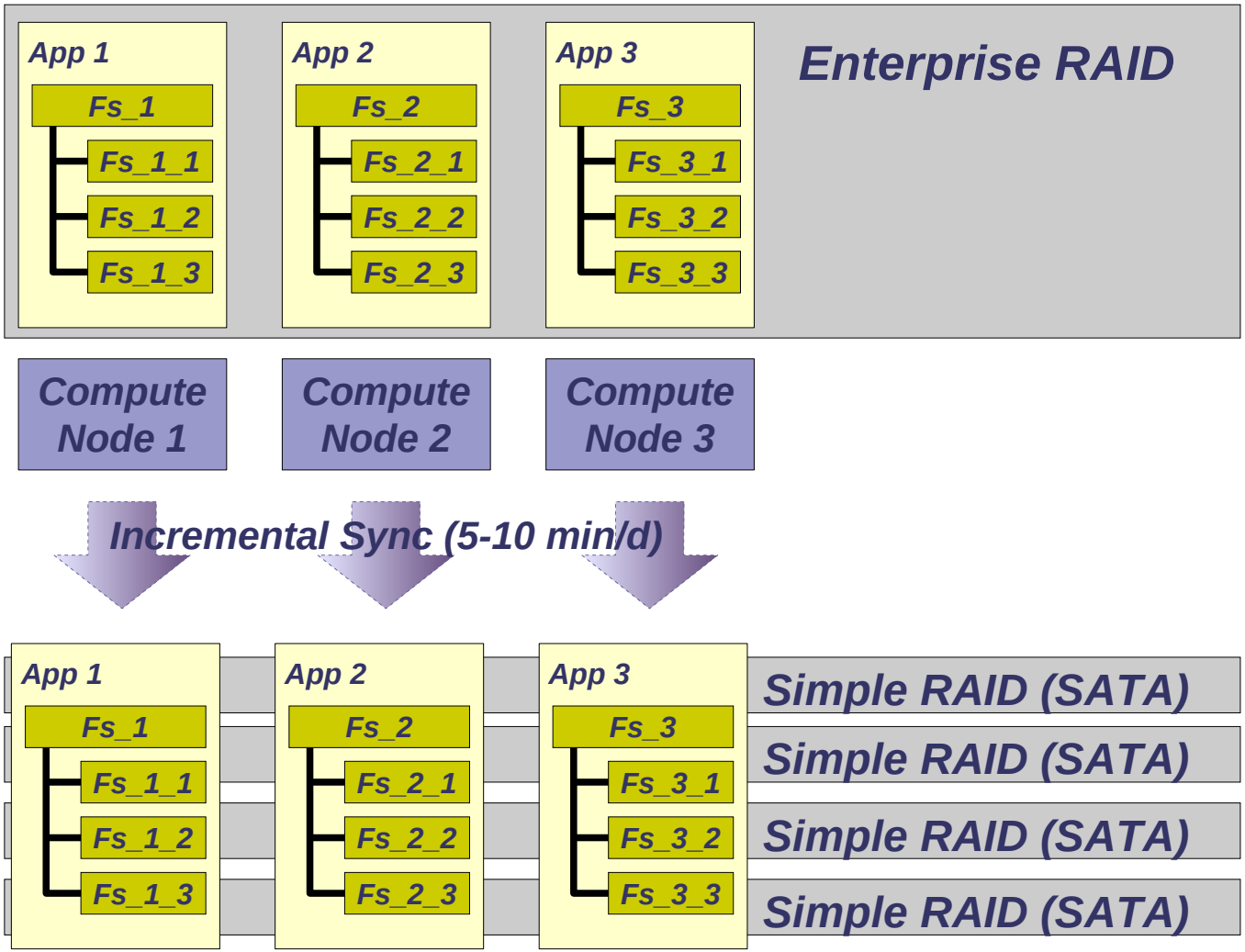
Systemkopien mit XDM

- *Gemeinsam mit Kunden erarbeitete EAS-Guides und Beispiele*
- *Tools für interaktive oder Batch-Bearbeitung*
- *Weitgehend automatisierte Abläufe*
- *Intelligente Nachsynchronisation*



High Speed Backup mit XDM

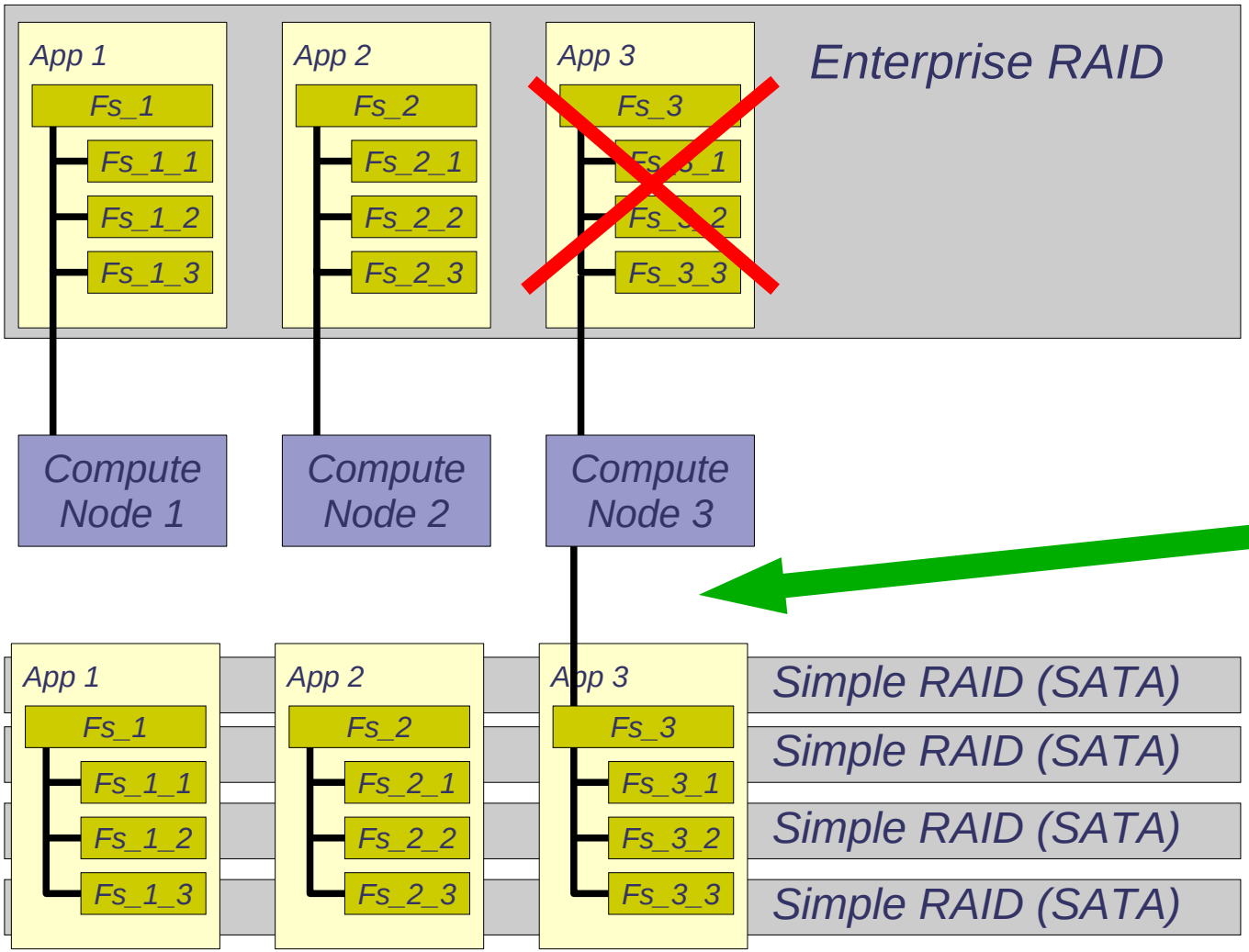
"Abfallprodukt" existierender Anwendungsbeschreibungen



- *Extrem kurzes Backup für Compute-Nodes*
- *High-Speed Streaming to Tape*
- *Keine Belastung der Compute-Nodes während Backup to Tape*
- *Restart-fähige Images der Applikation im Backup-Universum*

Restorefreies Recovery mit XDM

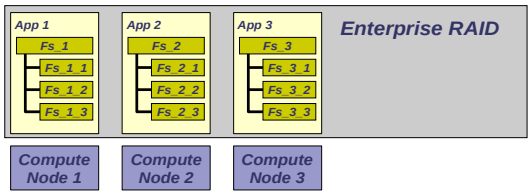
Tape-Backup wird nur im Ausnahmefall benötigt



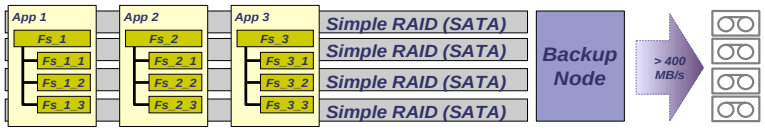
- *Sofortiger Wiederanlauf*
- *Kein Restore vom Tape*
- *Preview-Möglichkeit*
- *Bei Bedarf High-Speed Streaming from Tape*
- *Resync auf Enterprise-Storage bei bereits laufender Anwendung*

Differenzierung zu klassischen Backuplösungen

Möglichkeiten von Festplatten konsequent nutzen



Incremental Sync (5-10 min/d)

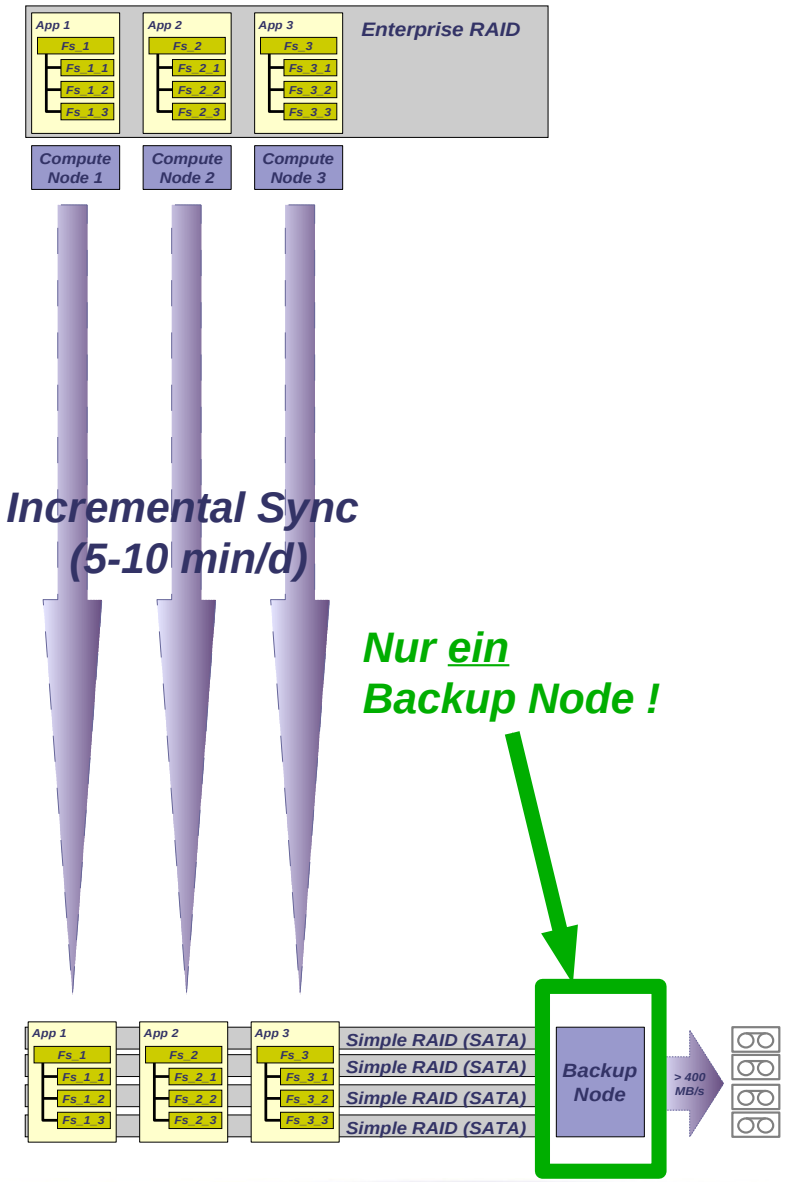


- *Extrem kurzes Backup für Application Nodes*
- *Minimale CPU-Belastung auf den Application Nodes (keine Verarbeitung der Daten)*
- *Nutzung SAN statt LAN*
- *kein Backup-Client auf Application Nodes (kein dezentrales Pflegen von Konfigurationen)*
- *Atomarer Backup – Dauer: NULL damit konsistenter Zustand*
- *Restartfähige Images der Applikation damit extrem schneller Wiederanlauf*
- *SW für Tape-Backup nur auf DASI-Server*
- *Zentrale Administration*
- *extreme Durchsätze bei Tape-Backup/-Restore möglich*
- *niedrige Anforderungen an Backup-RAID*
  - *ermöglicht SATA mit hoher Dichte*
  - *niedriger Platzbedarf*
  - *kürzere Backup-Zeiten*
  - *reduzierter Stromverbrauch / Wärmeabgabe*

- *adaptive Fahigkeiten bzw. "selbstlernend"*
- *Integration mit HV*
- *leicht zu DR-Umgebung ausbaubar*

Integration mit Backup to Tape

Beispielimplementierung für Legato Networker



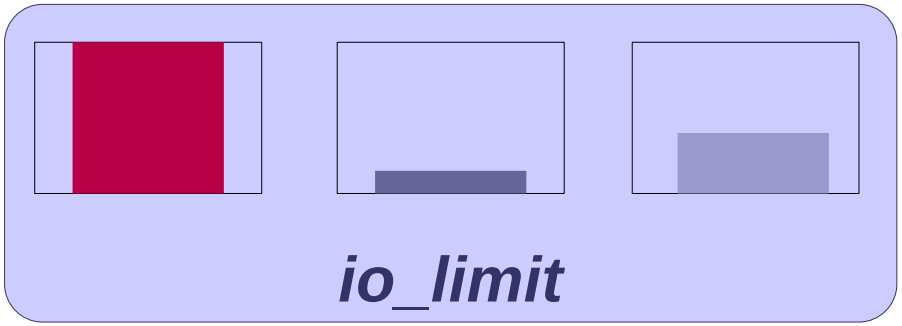
Was bietet die Integrationslösung von OSL?

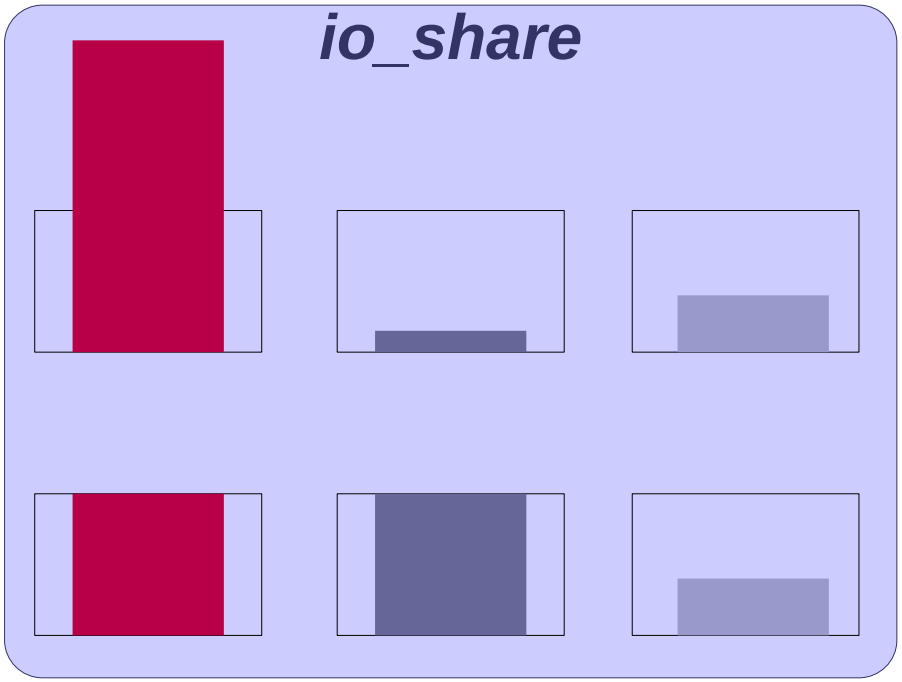
- *kombiniert B2D mit Bandsicherung*
- *applikationsorientiertes Verfahren*
- *sofortiger Neustart von Backup-Disk (kein Tape-Restore)*
- *Steuerung der Sicherungen über Networker-GUI oder CLI*
- *integriertes Pre- und Postprocessing*
- *mehrere Sicherungen pro Tag möglich*
- *differenzierte Erfolgskontrolle über die Networker-Indizes*
- *Aufzeichnung von Dateisystem-Informationen für Restore*
- *eindeutige Identifikation kompletter Sicherungen*
- *einfacher Restore kompletter Sicherungen*
- *weitere Funktionen für Oracle*
  - *Tool für Logrestore und Roll Forward*
  - *Archivierungslösung*
- *LAN-free Backup*
- *„cluster aware“, d.h. kein Eingriff bei Umschaltungen nötig*
- *automatische Gleichverteilung der Plattenlast*
- *hohe Durchsätze – VTL unter diesem Aspekt entbehrlich*

Bandbreitensteuerung

Vielfältige Möglichkeiten

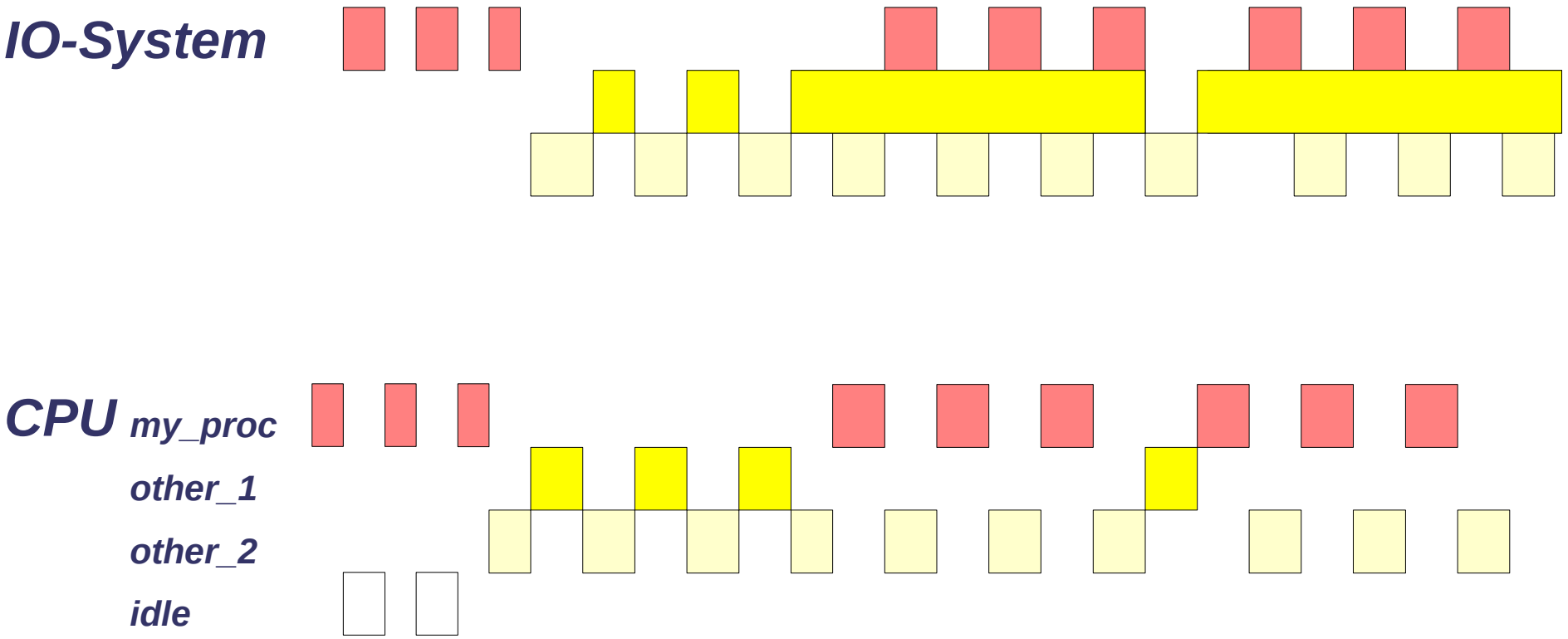
- *Warum?*
  - *Sättigung IO-Kanäle*
  - *Sättigung Speichersystem(e)*
  - *Konkurrenz Applikationen*
- *Was?*
  - *einzelne Volumes*
  - *Gruppen von Volumes*
  - *Applikationen*
- *Wie?*
  - *absolute Bandbreite (io_limit)*
  - *adaptives Konzept (io_share)*
  - *Limit für Synchronisationsvorgänge (sync_limit)*
- *Mit welchem Resultat?*
  - *verbessertes Antwortzeitverhalten*
  - *faire Verteilung von IO und CPU-Bandbreite*
  - *reduzierte CPU-Belastung*
  - *gesteigerter Gesamtdurchsatz*





IO- und CPU-Verteilung gemeinsam steuern

Über die Wechselwirkung von IO-Performance und CPU-Belastung



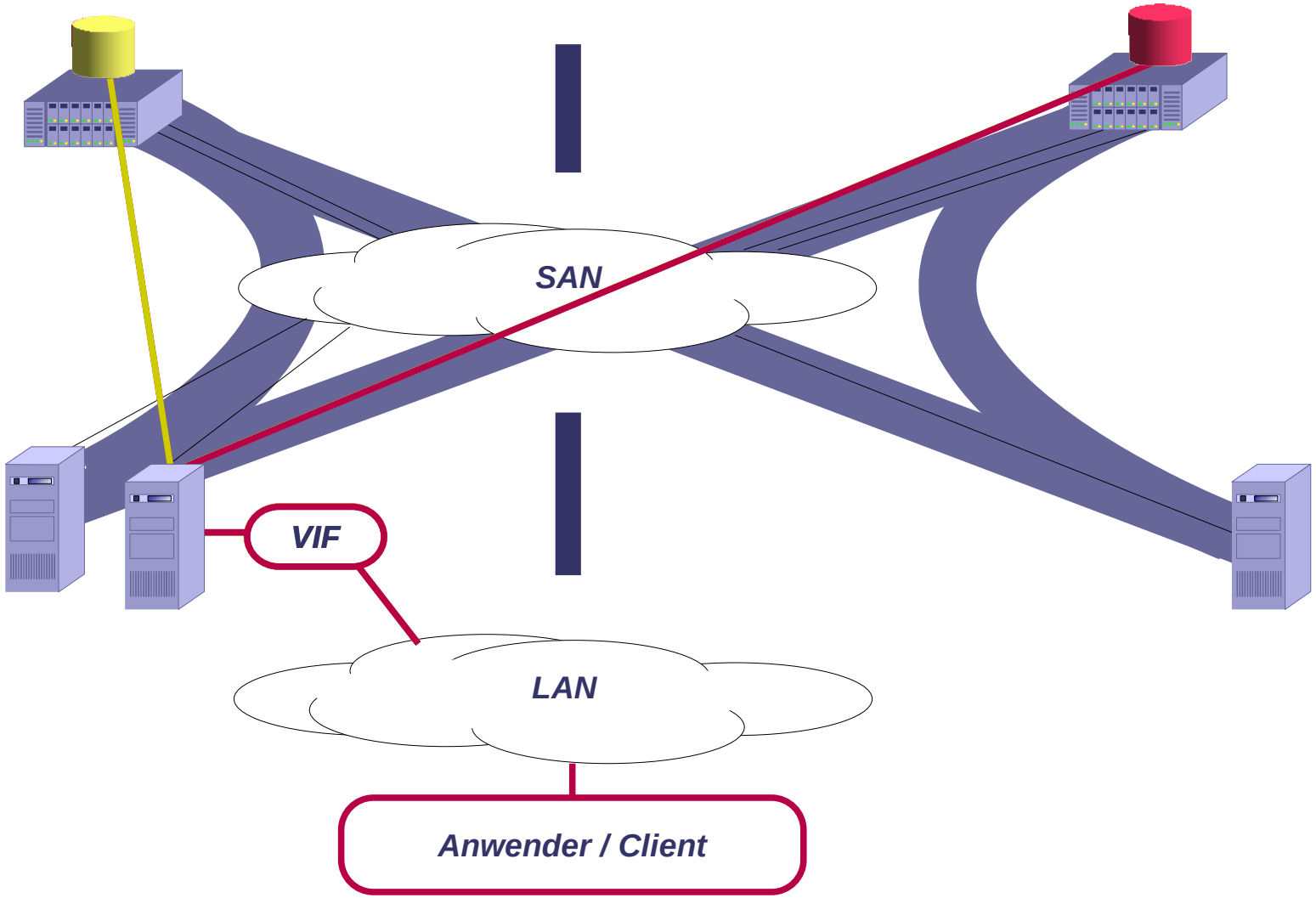
Σ Application Aware Storage Virtualization

Anwendungsbeschreibungen und Volume Management integriert

- *Konfiguration der Applikation ordnet Geräte Applikationen zu*
- *Übersicht zu Ressourcenverbrauch einzelner Applikationen*
- *Basis für Applikations-Spiegel /-Clones*
- *Applikationsbezogene Spiegelzustände*
- *Applikationsbezogene Steuerung von Aktionen (z. B. set source)*
- *Applikationsbezogene Bandbreitensteuerung*

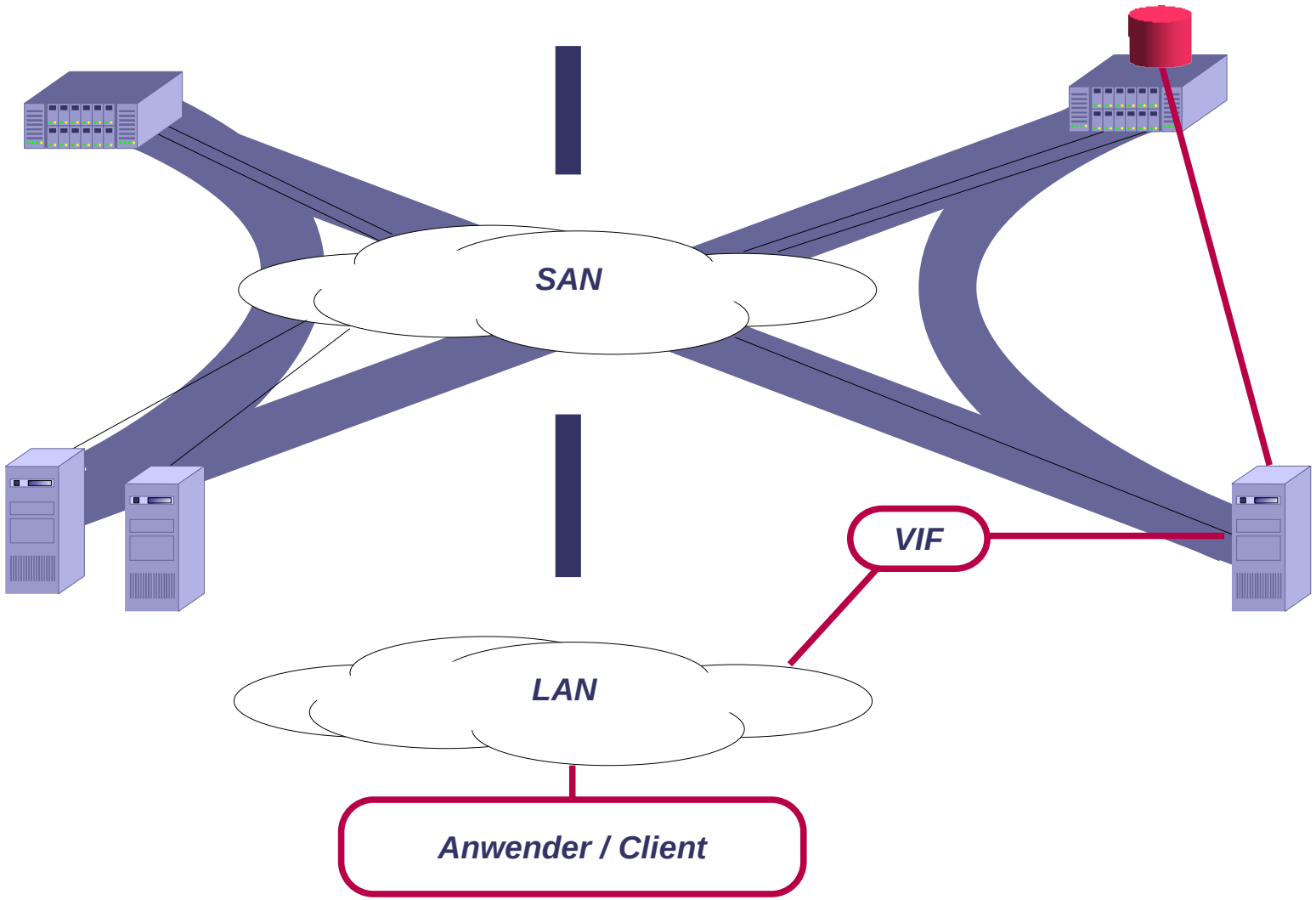
Alles zusammen

Clusterfähige Storage Virtualisierung, Backup, HV, Disaster Recovery



Alles zusammen

Clusterfähige Storage Virtualisierung, Backup, HV, Disaster Recovery



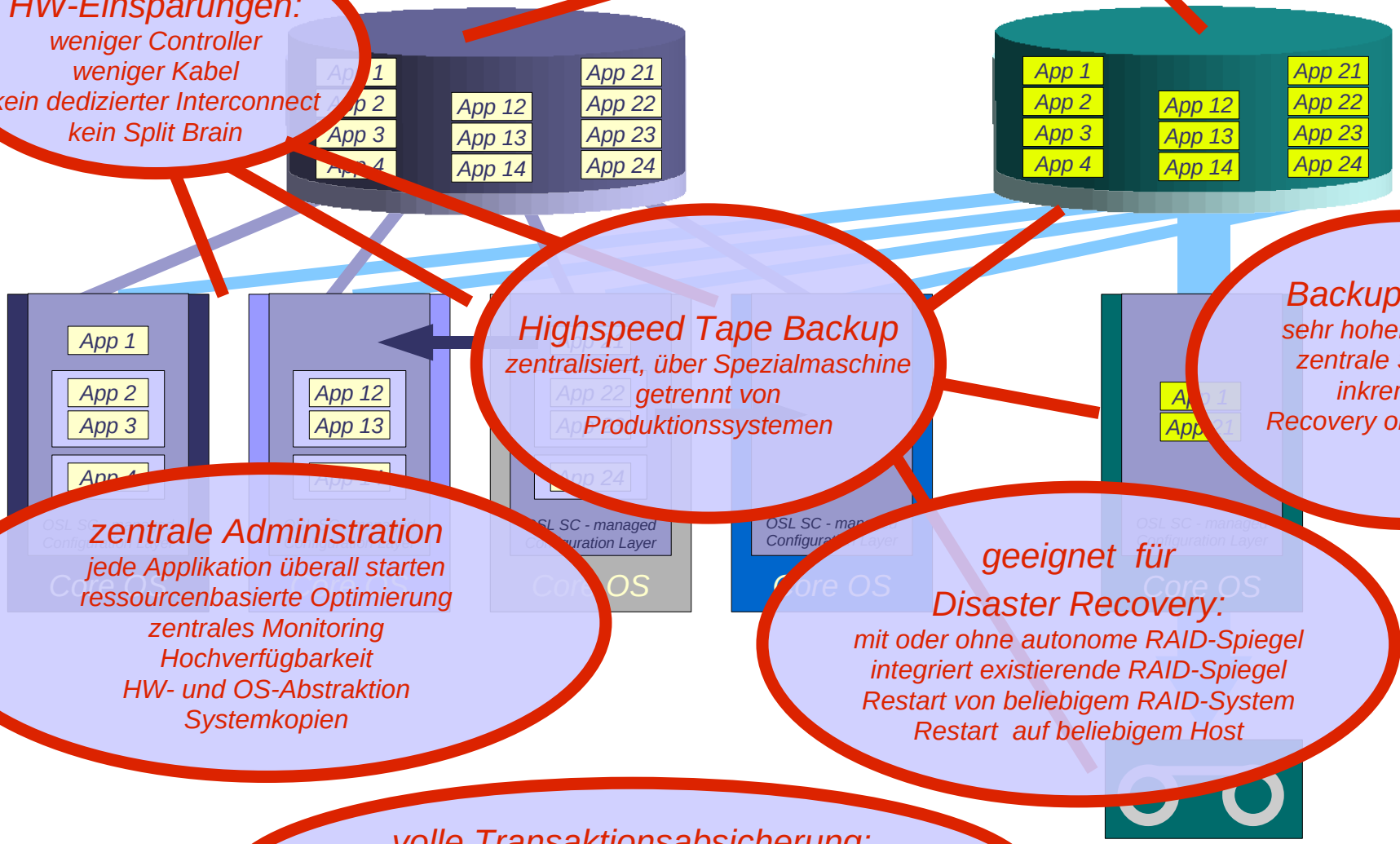
Zusammenfassung

Alles zusammen

Dynamische, flexible und hochverfügbare Infrastrukturen

Trennung I/O OLTP<>Backup:
spezialisierte Systeme ersetzen Universalsysteme
homogenes Lastprofil

HW-Einsparungen:
weniger Controller
weniger Kabel
kein dedizierter Interconnect
kein Split Brain



Highspeed Tape Backup
zentralisiert, über Spezialmaschine
getrennt von
Produktionssystemen

Backup to Disk:
sehr hoher Durchsatz
zentrale Steuerung
inkrementell
Recovery ohne Restore

zentrale Administration
jede Applikation überall starten
ressourcenbasierte Optimierung
zentrales Monitoring
Hochverfügbarkeit
HW- und OS-Abstraktion
Systemkopien

geeignet für Disaster Recovery:
mit oder ohne autonome RAID-Spiegel
integriert existierende RAID-Spiegel
Restart von beliebigem RAID-System
Restart auf beliebigem Host

volle Transaktionsabsicherung:
Kombination mit permanenten Spiegeln
Restart vom Disk-Backup (Spiegel)
Roll forward (full / until time / until cancel)