Modulare Virtualisierungs- und Clusterlösung für Solaris und Linux

OSL Storage Cluster 4.0

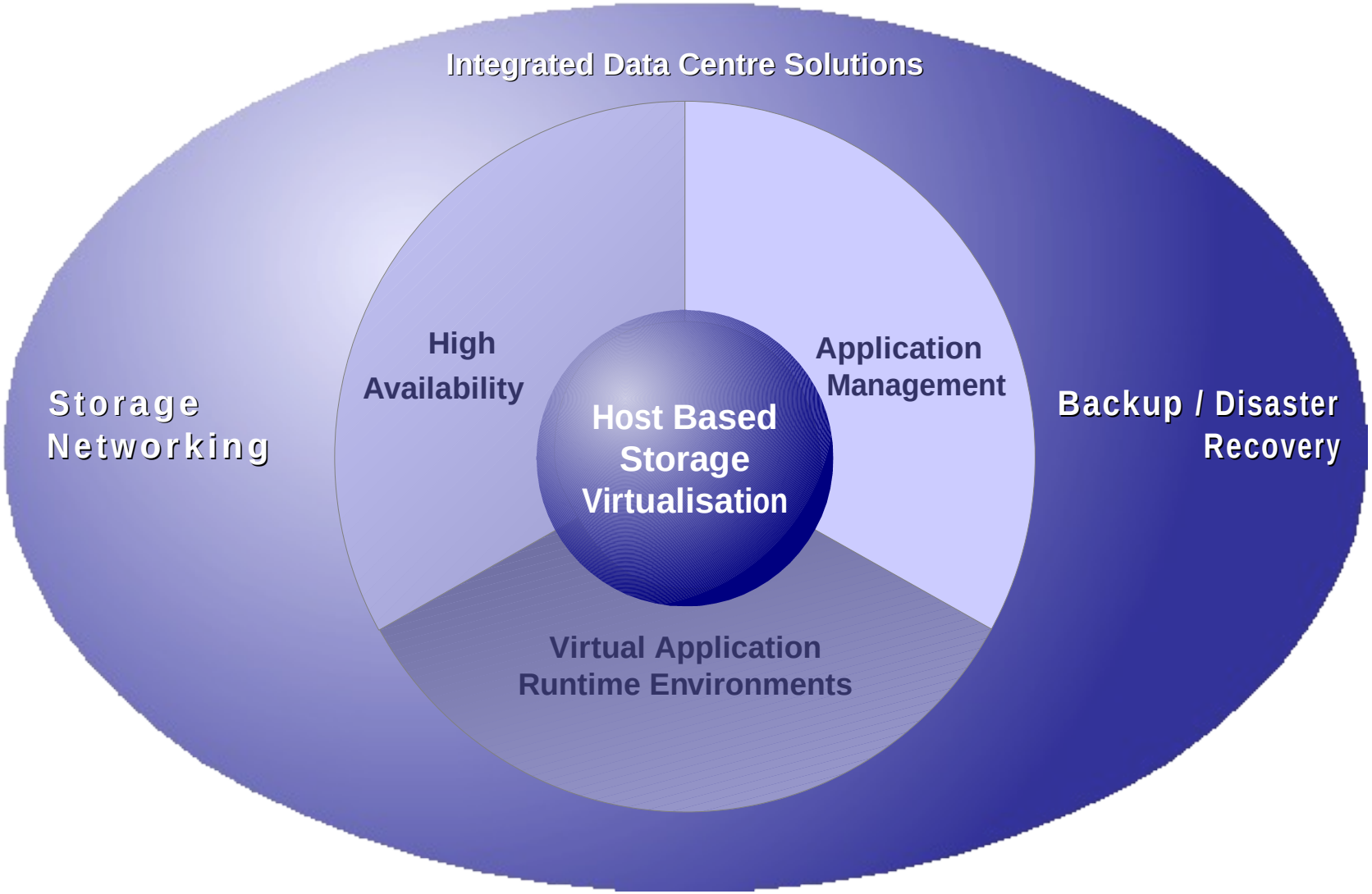
11. OSL Technologietage
Berlin 24./25. September 2013

bert.miemietz@osl.eu

Überblick

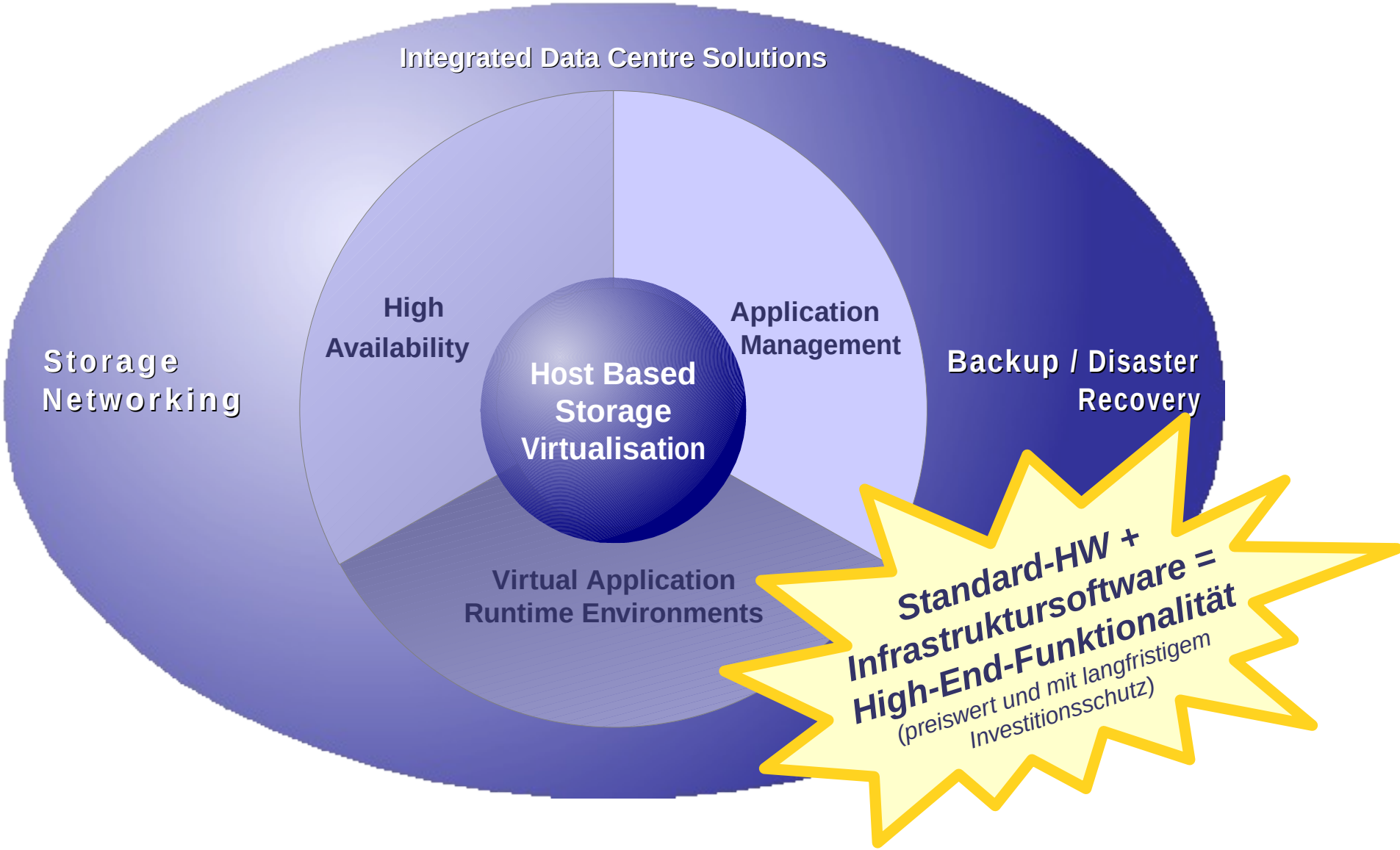
OSL entwickelt Infrastruktur-Software

Storage Virtualisation • Volume Management • Converged Networking
Virtual Machines • Clustering • High Availability • Disaster Protection



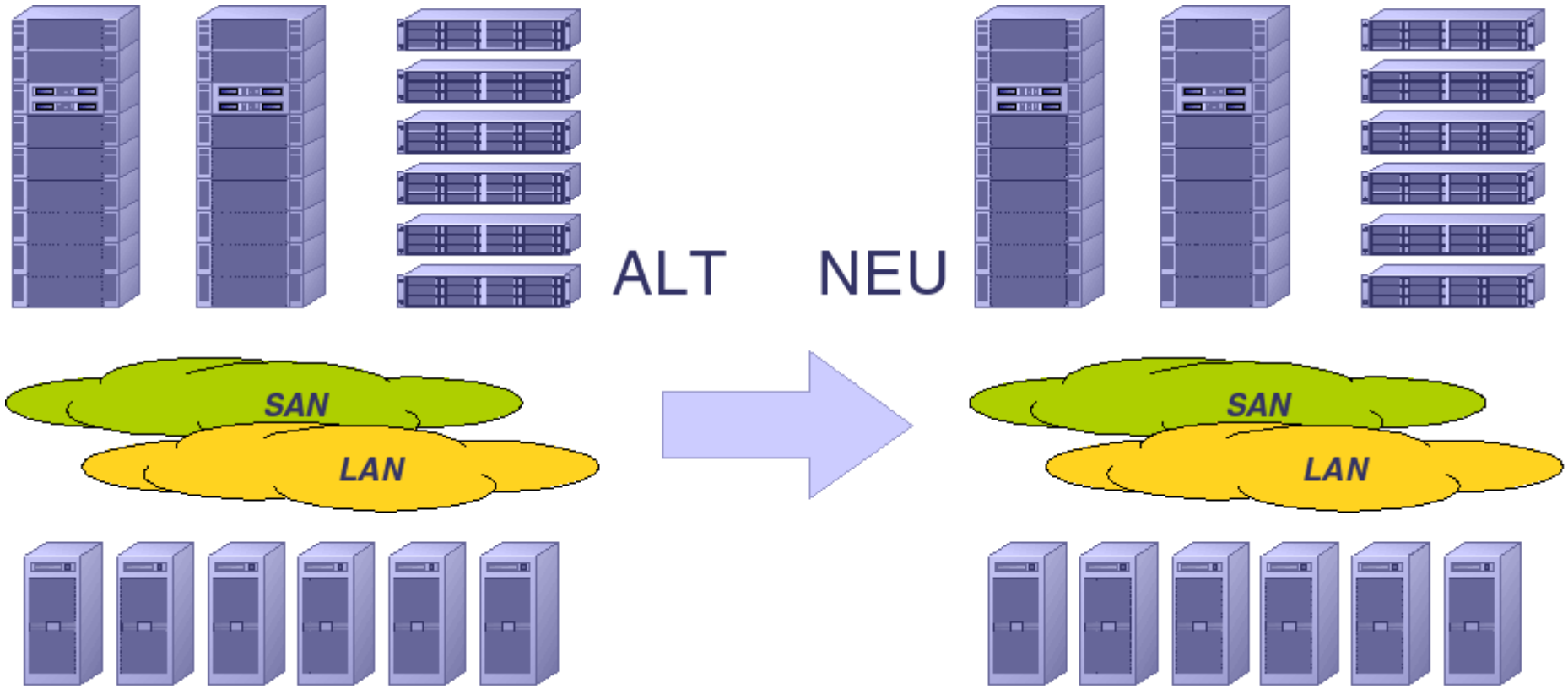
OSL entwickelt Infrastruktur-Software

Storage Virtualisation • Volume Management • Converged Networking
Virtual Machines • Clustering • High Availability • Disaster Protection



Warum sollte ich so etwas einsetzen?

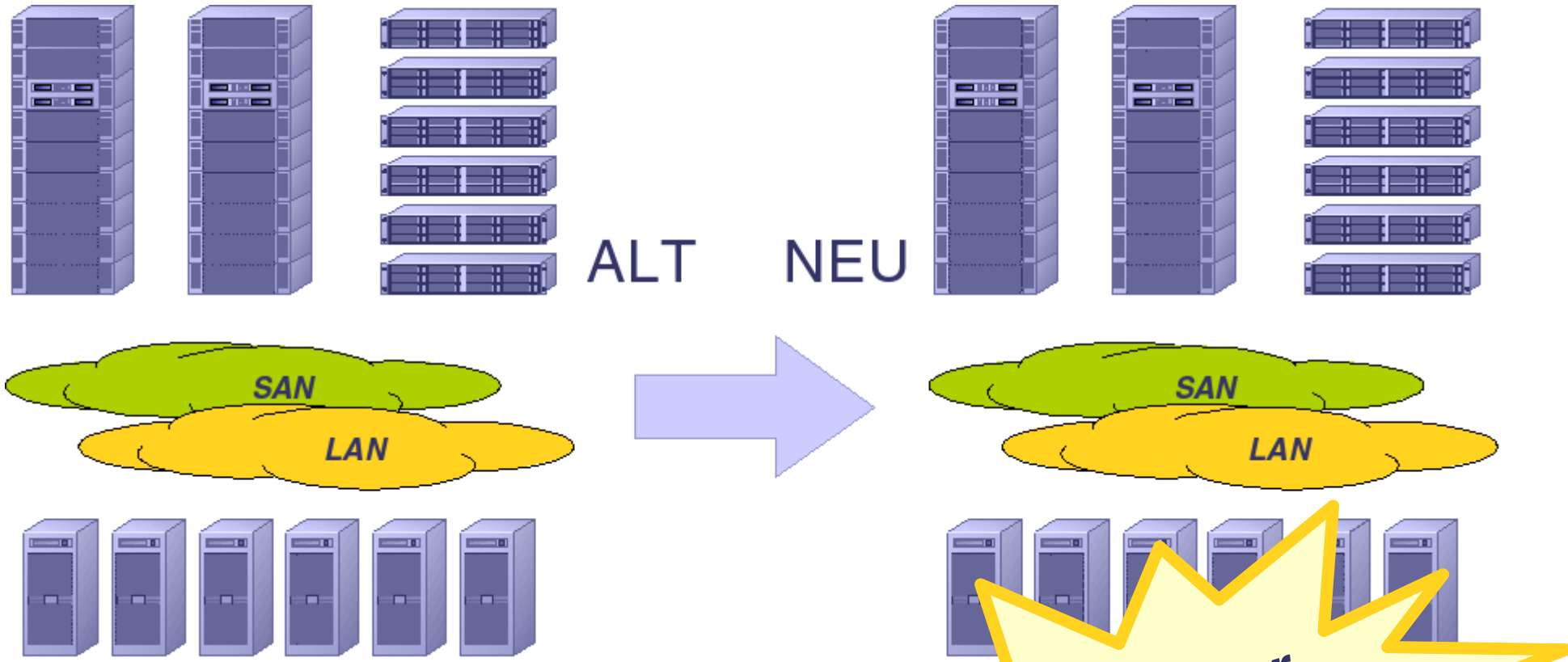
Ein Beispiel



- einfacheres, sicheres Management, schlankere Prozesse
- mehr Verfügbarkeit und Flexibilität
- Hardwareabstraktion ⇒ keine Angst vor Systemwechseln
- bessere Auslastung der Systemressourcen ⇒ mit weniger mehr erreichen

Warum sollte ich so etwas einsetzen?

Ein Beispiel



- einfacheres, sicheres Management, schlankere
- mehr Verfügbarkeit und Flexibilität
- Hardwareabstraktion ⇒ keine Angst vor Syste
- bessere Auslastung der Systemressourcen ⇒ mit weniger mehr erreichen

Es ist nur SOFTWARE !

Und noch etwas

Mit "dummer" Hardware zur cleveren Lösung

- Applikationen
- Middleware
- Resource Management
- Nutzer / Berechtigungen
- Betriebssystem
- Server-Hardware
- LAN
- SAN
- Storage
- Backup

Und noch etwas

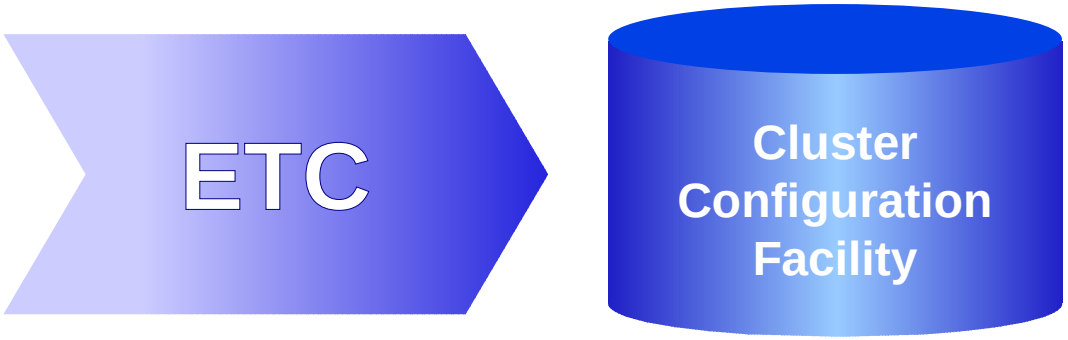
Mit "dummer" Hardware zur cleveren Lösung

Applikationen	ETC
Middleware	ETC
Resource Management	ETC
Nutzer / Berechtigungen	ETC
Betriebssystem	ETC
Server-Hardware	ETC
LAN	ETC
SAN	ETC
Storage	ETC
Backup	ETC

Und noch etwas

Mit "dummer" Hardware zur cleveren Lösung

Applikationen	ETC
Middleware	ETC
Resource Management	ETC
Nutzer / Berechtigungen	ETC
Betriebssystem	ETC
Server-Hardware	ETC
LAN	ETC
SAN	ETC
Storage	ETC
Backup	ETC

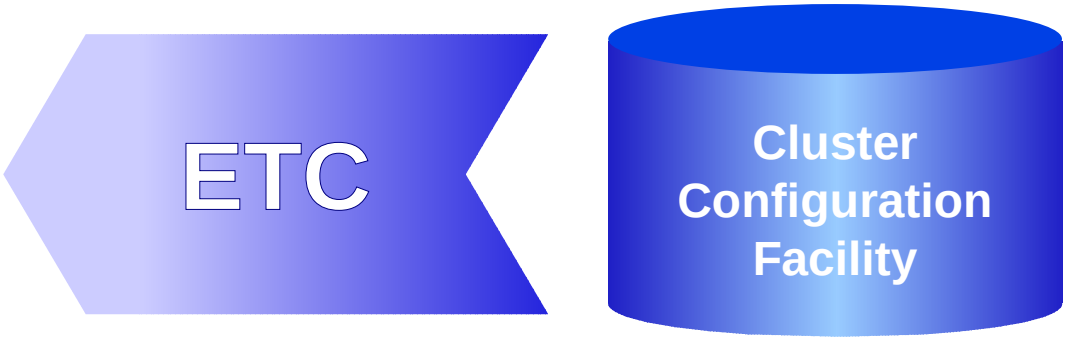


- hardwareabstrakt / plattformneutral
- permanent
- clusterweit

Und noch etwas

Mit "dummer" Hardware zur cleveren Lösung

Applikationen	ETC
Middleware	ETC
Resource Management	ETC
Nutzer / Berechtigungen	ETC
Betriebssystem	ETC
Server-Hardware	ETC
LAN	ETC
SAN	ETC
Storage	ETC
Backup	ETC



- hardwareabstrakt / plattformneutral
- permanent
- clusterweit

OSL SC 4.0 - Ausgereiftes Portfolio

Widerspiegelung der Gesamtarchitektur im Produktpaket

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

Cluster- Speichervirtualisierung	**RSIO**	**Extended Data Management**
globale Storage-Pools (rechnerübergreifend)	Datacenter Block-I/O z. B. via Ethernet	Integration RAID-basierter Datenkopien & Snapshots
globales Disk-Inventory	Integration mit Speichervirtualisierung	hostbasierte Spiegelung
globale Geräte / globaler Namensraum	Multipathing + Trunking	online Datenmigration
Clustervol-Manager/automatische Allokation	Cross-Platform	Daten klonen
Disk Access Management		
I/O-Multipathing		

OSL SC 4.0

Das Rechenzentrum ist immer ein Cluster

- Globale Administration und Prozesse (Rechte, Anwendunssteuerung, Backup ...)
- Global Storage Pool
- Ggf. auch HA usw.

Wir machen ernst:

- Globale hostbasierte Speichervirtualisierung bis in die VM / zur Applikation
- Applikationsbewußtsein in allen Ebenen
- Vollsymmetrische Funktion und Administration ohne Split Brain

In einem Cluster:

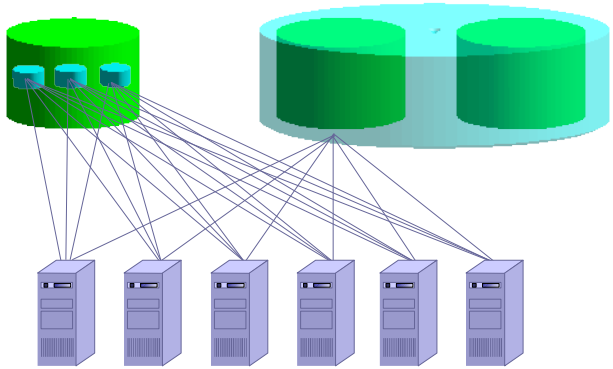
- bis zu 128 Knoten
- einzigartiges Virtual-Node-Konzept
- bis zu 8192 Application Volumes auf bis zu 2048 Physical Volumes mit jeweils bis zu 32 Datenpfaden
- bis zu 1024 Applikationen oder Virtual Machines
- Cross-Platform (Solaris und Linux / verschiedene OS-Ständen und Architekturen)

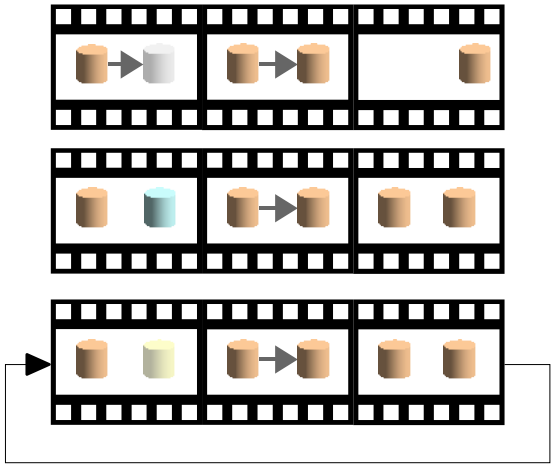
Speichervirtualisierung

OSL (Disk Virtualizing) Storage Cluster

Speichervirtualisierung: hostbasiert – clusterfähig - schnell

Speichervirtualisierung
clusterweit
globale Pools
Daten verschieben
Daten klonen
Daten spiegeln
Sonderfunktionen





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 laufenden Applikations-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

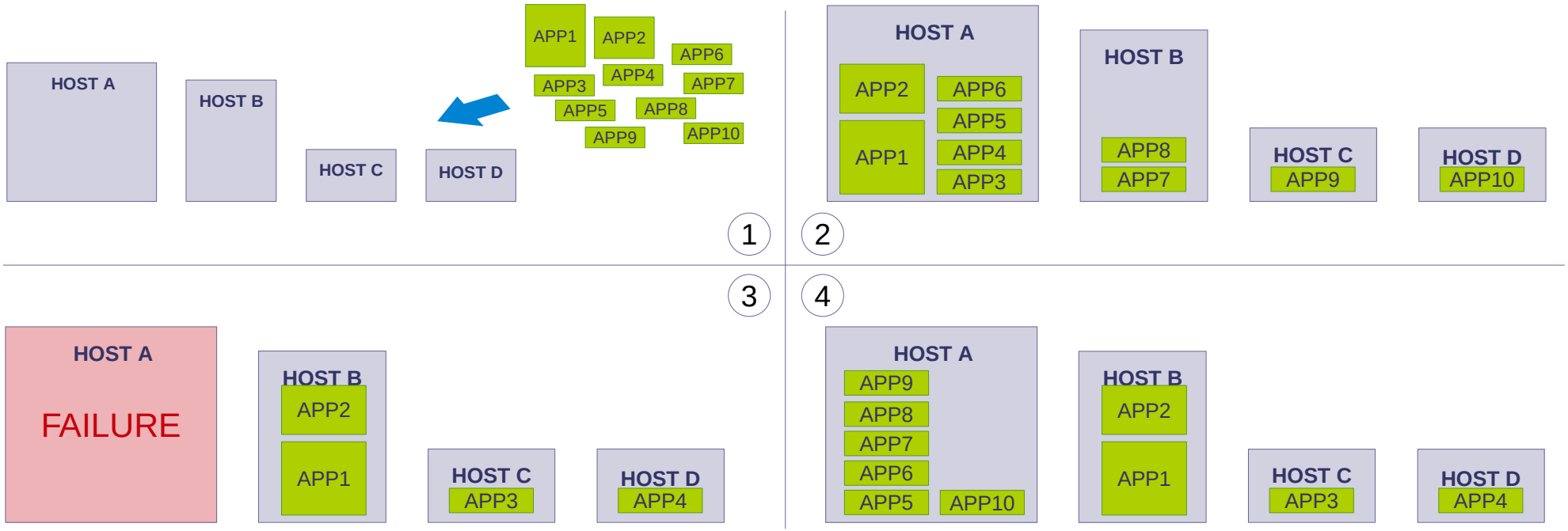
XVC (Extended Volume Controls)
z.B. Pause, Stop, Trigger, Aktionen
Bandbreitensteuerung
detaillierte Statistik

Clusterframework

Clusterframework

Herausragendes Design – überlegene Funtionalität

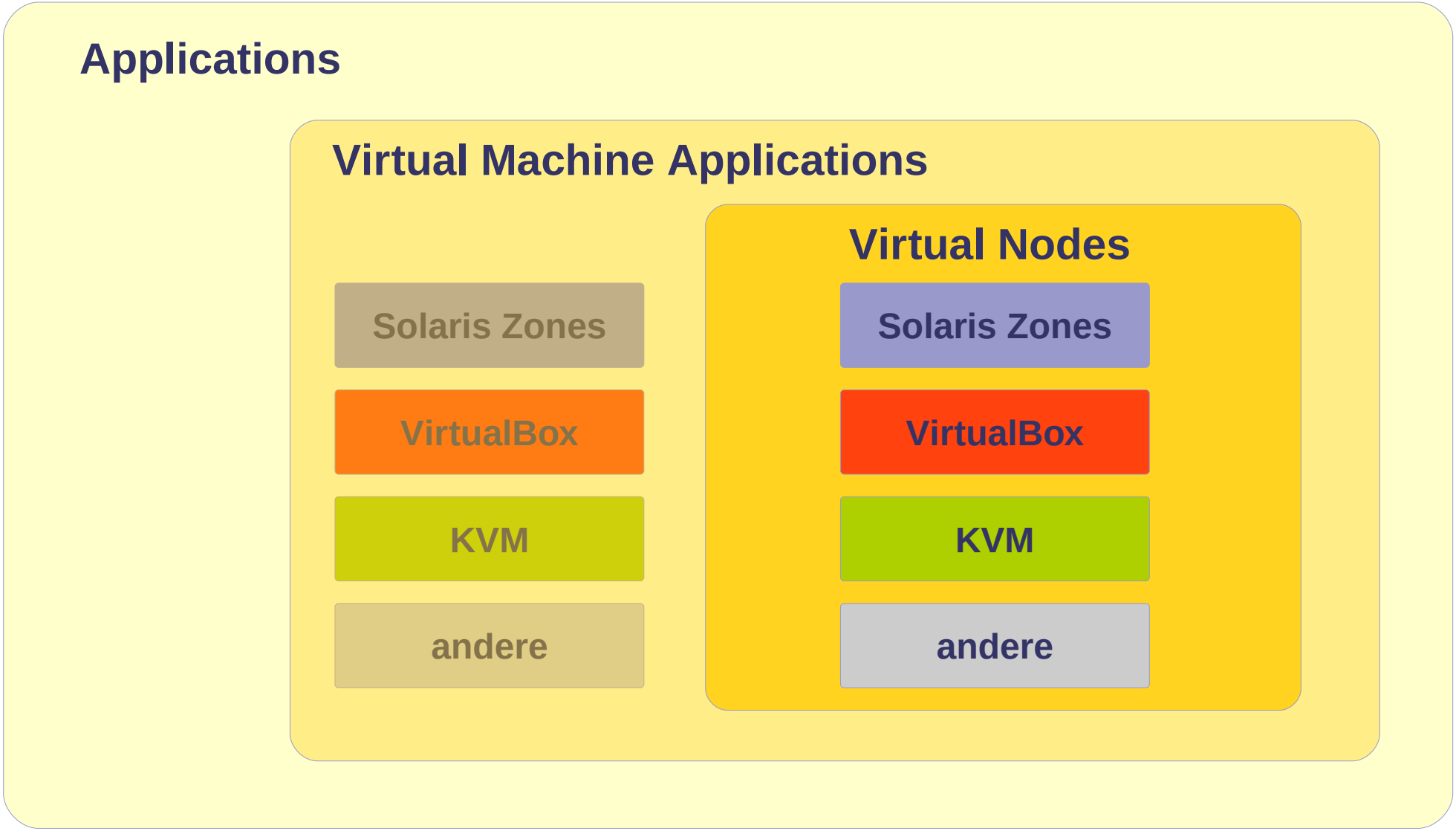
- selbstorganisierend – Berücksichtigung / Steuerung Ressourcen
- vollkommen symmetrisch
- herausragende Robustheit - kein Split Brain
- zentrale Administration von jedem Knoten aus
- Cross-Platform (Solaris 7-11, x86 + SPARC, diverse Linux-Plattformen)
- verknüpft mit Speichervirtualisierung (Applikationsbewußtsein, Backup & DR ...)



Virtual Nodes

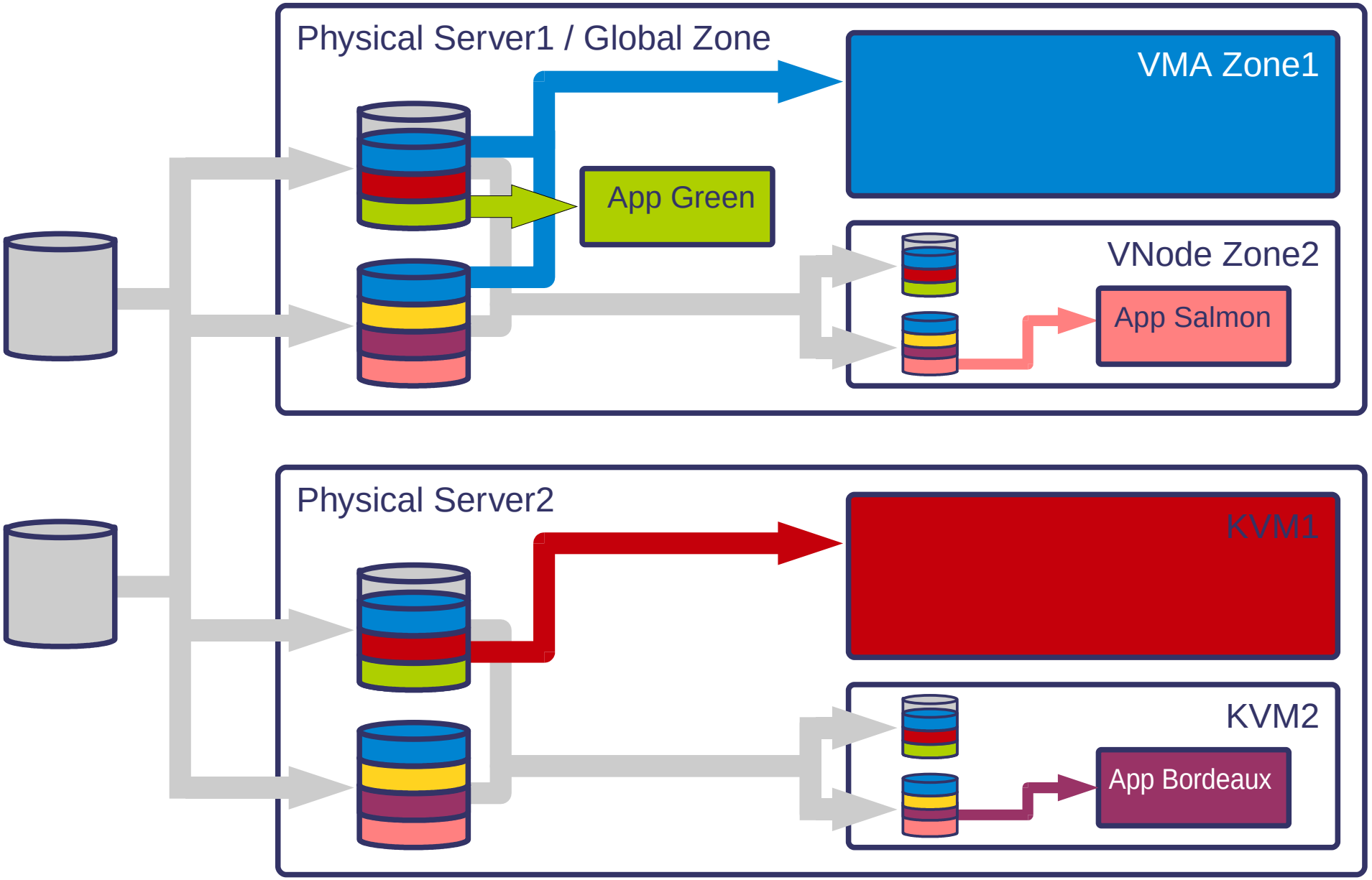
Zur Systematik von Applikationen

Applications – Virtual Machine Applications – Virtual Nodes



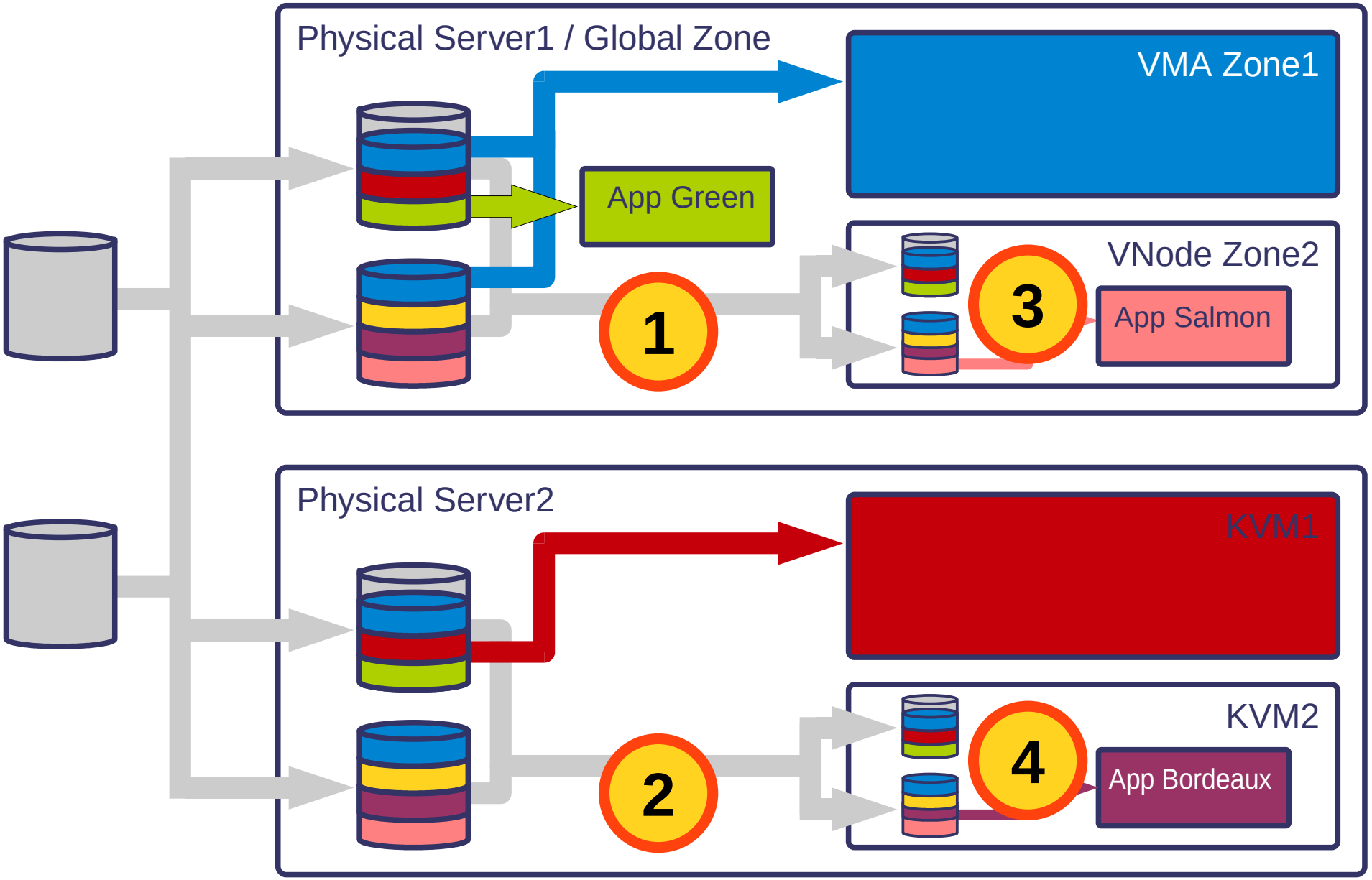
VMA und VNode im Clusterframework

Bedeutung für Clusterframework und Speichervirtualisierung



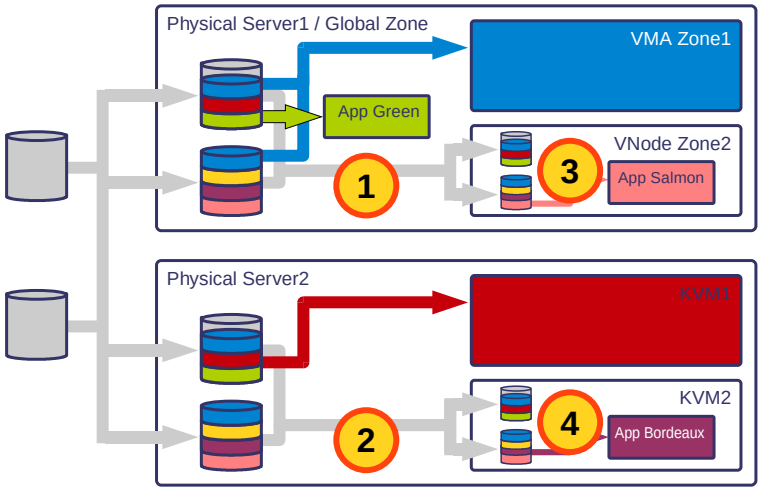
VMA und VNode im Clusterframework

Bedeutung für Clusterframework und Speichervirtualisierung - Clusterknoten



Physical Nodes und Virtual Nodes

Bedeutung für Clusterframework und Speichervirtualisierung - Clusterknoten



- Physical Nodes sind reale Hardware
- Virtual Nodes werden per SW erzeugt
- beide erscheinen in der Knotenliste
- beide haben Zugriff auf Speichervirtualisierung
- beide unterliegen Disk Access Management
- beide können Applikationen ablaufen lassen
  ⇒ einfachste P2P-/P2V-/V2V-/V2P-Migration

```
# ndadmin -lvvv
sc-nodename       id    state tf os          cpu-isa        vcpu  clock  memory

server1            1   ONLINE p  SunOS 5.10  amd64             8   3000   16384
server2            2   ONLINE p  Linux 3.x   amd64             8   3000   16384
zone2@0            3   ONLINE v  SunOS 5.10  amd64             2   3000    3964
kvm2@0             4   ONLINE v  Linux 3.x   amd64             2   3000    3964
```

physical node

virtual node

Durch Zuordnung der VNode-Capability und Installation von OSL SC kann eine VM zum Virtual Node werden.

VMA und VNode im Clusterframework

Noch einmal das Wichtigste in aller Kürze

Gemeinsamkeiten aller VMs einschl. VNodes auf Solaris und Linux

- Erzeugen/Löschen/Cluster Control/Start/Stop/Migration/CPU/MEM: **`vmadmin(1m)`**
- Detailkonfiguration und Zuordnung I/O-Geräte: **`vmconfig(1m)`**

VMA – Virtual Machine Application

- nur Zugriff auf zugewiesene (VM-spezifische) Geräte
- kein Zugriff auf Steuerung der Speichervirtualisierung
- Solaris Zone Storage mit allen Solaris-typischen Einschränkungen
- Steuerung Applikationen nur in Eigenregie der VM-Instanz
- keine eigene Node-ID / Disk Access Management hängt am Hypervisor-Node

Virtual Node

- voller Zugriff auf Clusterengine und Speichervirtualisierung
- extrem einfaches Storage-Handling
- eigene Node-ID / individuelle Zugriffsteuerung im Disk Access Manager
- Cluster kann Applikationen in VNodes steuern
- Failover/Migration von Applikationen zw. VNodes u. VNodes bzw. PNodes

Neuigkeiten im Device-Handling

Application Specific Devices (ASD)

Geräteknoten je Applikation

- Zuordnung der klassischen Geräte zu Applikationen bei der Beschreibung der Applikation bzw. der VM:
  `# vmconfig(1m)` - alle Arten von virtuellen Maschinen (auch Zonen)
  `# ardadmin(1m)` - alle anderen Applikationen

- Detaillierte Prüfungen der gesamten Konfiguration (alle Applikationen/VMs !!!), Syntax etc. bereits bei Erstellen der Konfigurationsbeschreibung

- ASDs werden erst beim Start der VM/Applikation erstellt,
  beim Stop wieder entfernt

- erhöhte Sicherheit

- erheblicher Geschwindigkeitszuwachs beim Zpool-Import in großen Konfigurationen / Vermeiden von Zugriffskollisionen

- Verfügbarkeit ASD prüfen: **`# appdevs -l app_name`**

- alle Geräte mit Applikations-/VM-Zuordnung sind erfaßt
  ⇒ schnelle Übersicht mit: **`# smgr -qa`**

Multi-Volume Filesysteme

Auch hier Zuordnung nach eindeutigen Regeln

- klassische Unix-Lösung vfstab
  - eineindeutige Zuordnung Dateisystem <-> Gerät
  - scheitert bei zfs und btrfs
- Erweiterung für klassische Applikationen um **`sfstab`** (special filesystem table) in neuem Resource Description Processor-Format (rdproc):

```
#type      id          parameter      dev-args
zpool      gurke       altroot=/      gurke-a@0 gurke-b@0 gurke-c@0
```

- Für VMs werden analoge rdproc-Sätze mit **`vmconfig(1m)`** erstellt, dabei besteht eine Rückholmöglichkeit für bis zu 10 vorherige Konfigurationen
- dies funktioniert exakt identisch auch unter Linux und für btrfs
- für Sondernutzungen wie ASM oder Raw-Device-Datenbankinstallationen wurde der Ressourcentyp "**rpool**" neu eingeführt.
- theoretisch lassen sich auch normale FS-Einträge mit sfstab statt vfstab erstellen.

O S L R S I O

Remote Storage I/O

“Eine wirklich gute Idee erkennt man daran,
daß ihre Verwirklichung von vornherein ausgeschlossen erscheint.”
- Albert Einstein -

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)
- eigene Frames ⇒ prinzipiell nicht auf Ethernet / IP angewiesen
- 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
- In Verbindung mit OSL SC ist LAN-free Backup möglich

Natürlich ist I/O über Netz vernünftig!

Netzwerkkonvergenz, Vereinfachung und Rightsizing sind im SDDC zwingend

- absolut ausreichende Performance
  - 2 x 1Gbit LAN ⇒ bis zu 230 MiB/s **pro Node**
  - 4 x 1Gbit LAN ⇒ bis zu 460 MiB/s **pro Node**
  - 2 x 10Gbit LAN ⇒ über 500 MiB/s **pro Node**
  - 2 x 40Gbit LAN ⇒ >> 1000 MiB/s **pro Node**
  - Backup muß heute nicht mehr über's Netz
- keine Hotspots / Überlastungen
- höhere Portdichten, ggf. Senkung Energieverbrauch
- niedrigere Ausfallraten
- einfachere Handhabung
- alles über ein Netz!
- enorme Kostensenkungen in der Infrastruktur

Natürlich ist I/O über Netz vernünftig!

Netzwerkkonvergenz, Vereinfachung und Rightsizing sind im SDDC zwingend

- absolut ausreichende Performance
  - 2 x 1Gbit LAN ⇒ bis zu 230 MiB/s **pro Node**
  - 4 x 1Gbit LAN ⇒ bis zu 460 MiB/s **pro Node**
  - 2 x 10Gbit LAN ⇒ über 500 MiB/s **pro Node**
  - 2 x 40Gbit LAN ⇒ >> 1000 MiB/s **pro Node**
  - Backup muß heute nicht mehr über's Netz
- keine Hotspots / Überlastungen
- höhere Portdichten, ggf. Senkung Energieverbrauch
- niedrigere Ausfallraten
- einfachere Handhabung
- alles über ein Netz!
- enorme Kostensenkungen in der Infrastruktur

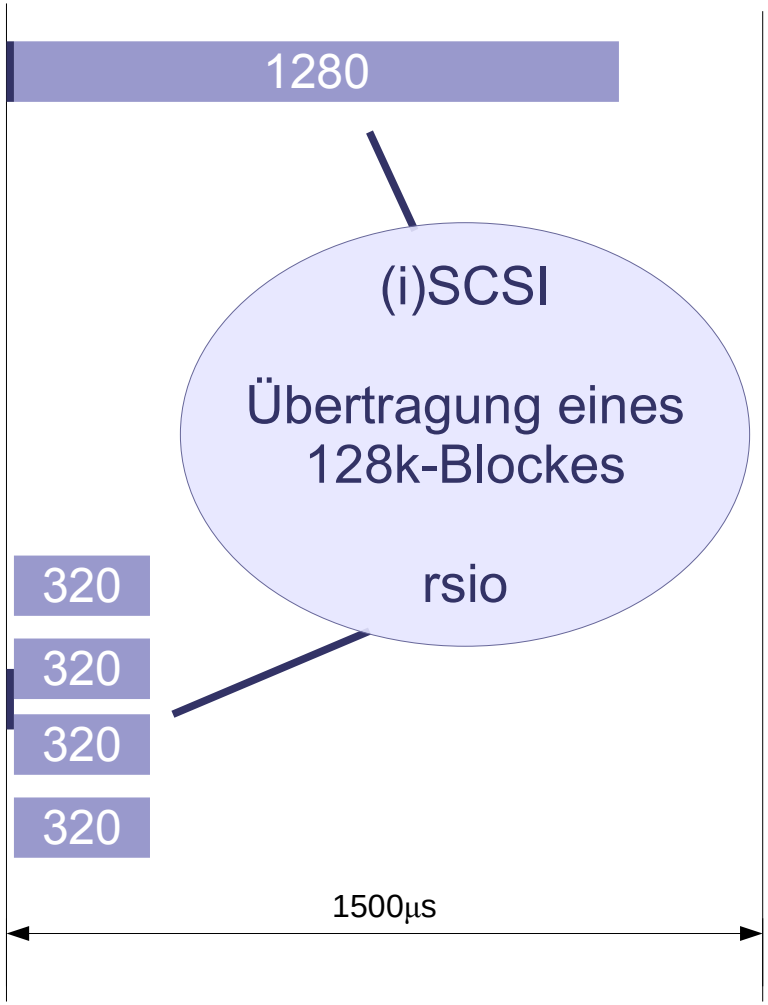
. . . wenn ich das richtige Protokoll habe!

Block-I/O neu definiert

RSIO – Remote Storage I/O

Vorteile der RSIO - Technologie

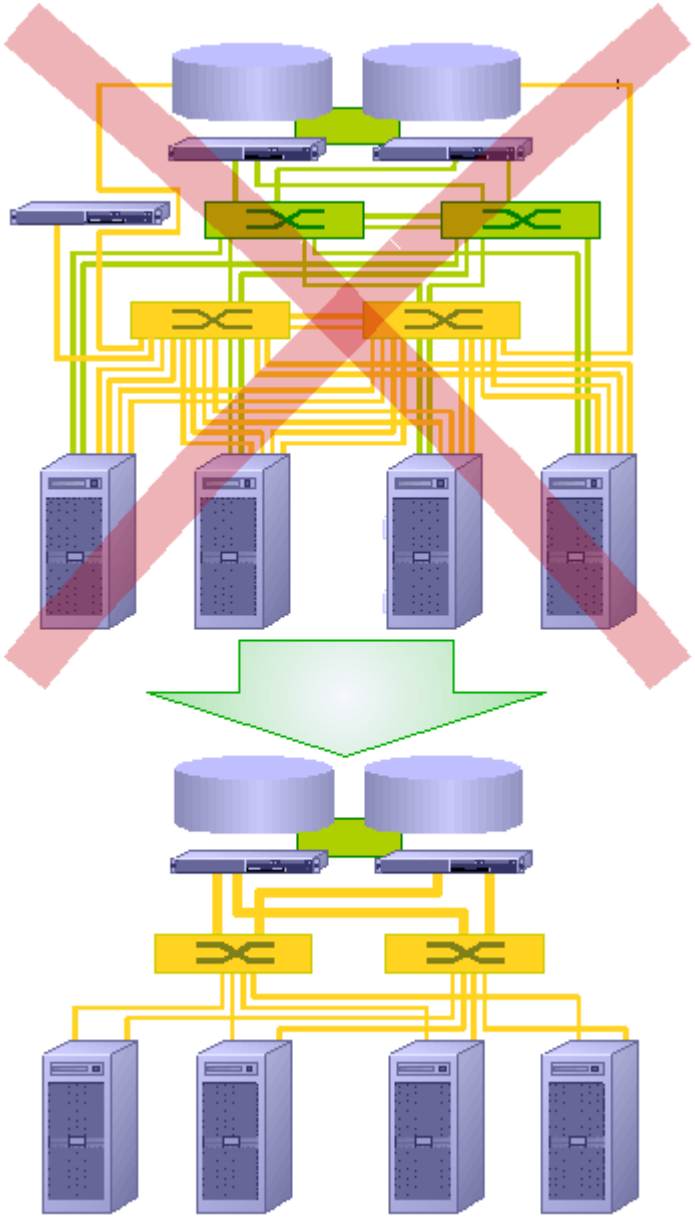
- Definition eigener Frames
  - Unabhängigkeit von TCP
  - ermöglicht erst **Scatter-Gather-Multipathing**
  - ermöglicht Zusatzfunktionen wie Checksum / Encryption
  - Frames mit variabler Größe
  - Overhead per Frame nur 16 Byte
- Trennung von Treiber und Transport
  - größerer Funktionsumfang bei hoher Portabilität
  - besseres Error-Handling
  - Performance offensichtlich kein Problem
  - hochflexibler Multithreading-Support
  - bessere Abschirmung des Kerns
- Integriertes Multipathing und Trunking
- Selbstkonfiguration und Error Recovery
- Unterstützung geclusterter Server
- vollständige Abbildung relevanter Schnittstellen (dkio)
- Performance-"Nachteile" (Latenz 250μs 10GE vs. 130μs FC) werden je nach Kontext kompensiert oder überkompensiert



. . . RSIO kann über verschiedene Netztechnologien laufen und ermöglicht SAN-Funktionalität über Standard-Ethernet!

Bedeutung von RSIO

Ermöglicht Durchbruch beim SDDC-Gedanken

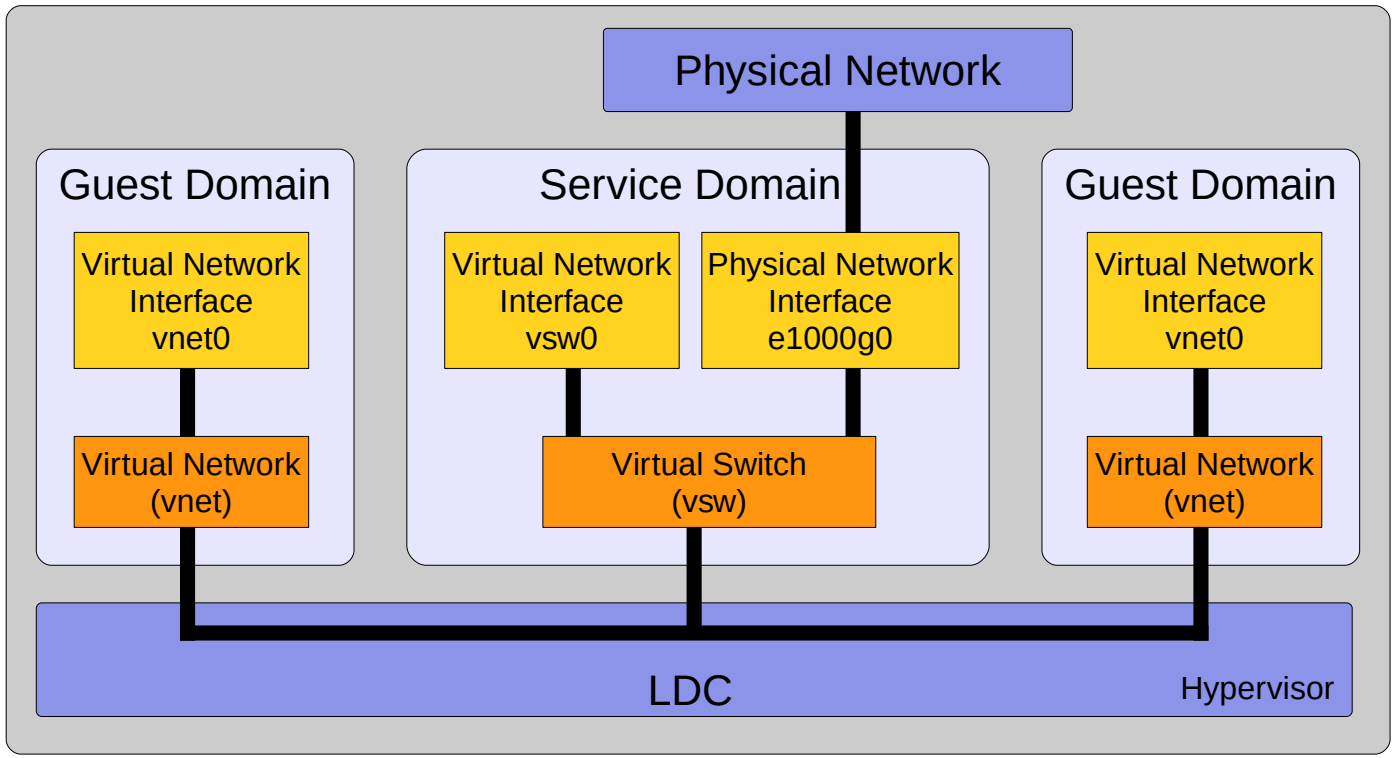


- **Netzwerkkonvergenz** für klassische Systeme
- FC-Port-**Konsolidierung**
- vereinfachte Infrastruktur
- höhere Zuverlässigkeit
- enorme **Skalierbarkeit** z. B. für Provider:
  - Anzahl Clients (Initiators)
  - Anzahl Server (Targets)
  - Durchsatz
  - Clusterkonzepte
- ultraschneller I/O mit Infiniband
- ermöglicht den Aufbau von Shared-Storage-Netzen (Voraussetzung für HA u. ä.) mit einfachen Ethernet-Komponenten
  ⇒ erhebliche **Senkung Einstiegsschwelle HA**

Bedeutung von RSIO

Unverzichtbar für das "SDDC in a Server"

- "Superdicke" Server erlauben, das ganze RZ in einer Maschine darzustellen
- Beispiel mit LDOMs (gilt analog für KVM/XEN):



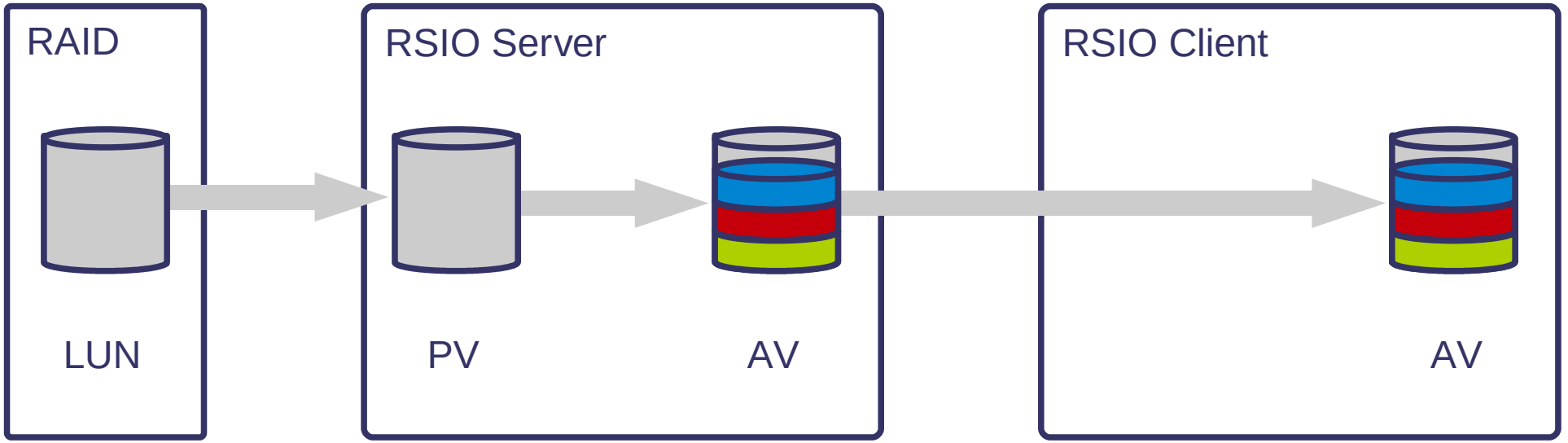
- wo sehen Sie hier ⇧ ein SAN bzw. I/O-Netzwerk?
- virtuelle Devices sind Bedingung für einfache Handhabung und Live-Migration
- RSIO löst dieses Problem mit der überlegenen Nutzung einer virtuellen Ethernet-Infrastruktur und der Brückenfunktion zu FC: einfach – zuverlässig – skalierbar

OSL Physical Device Server

on demand – hochperformant - flexibel

Klassische RSIO-Nutzung

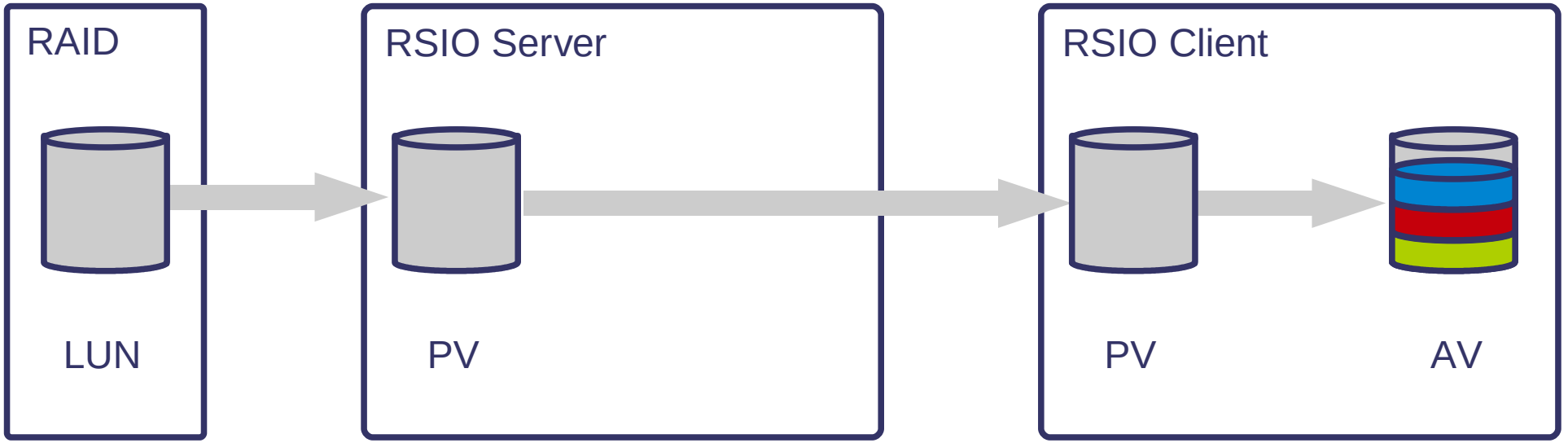
Vollvirtualisierte Blockgeräte für RSIO-Clients



- Application Volumes (AV) werden direkt an den RSIO-Client hinausgereicht
- Der Client benötigt nur einen sehr kleinen Treiber
- Speichervirtualisierung läuft "remote" auf dem RSIO-Server
- zusätzliche Entlastung des RSIO-Clients
- Nachteile: - VNode-Konzept in Zonen auf RSIO-Clients nicht nutzbar
  - RSIO und FC nicht ohne weiteres gemeinsam nutzbar, z. B. für Online Data Mobility oder Spiegel

OSL Physical Device Server

Physical Volumes für RSIO-Clients mit eigener Speichervirtualisierung



- Physical Volumes (PV) werden direkt an den RSIO-Client hinausgereicht dabei FC-seitiges Multipathing bereits auf RSIO-Server
- Der Client muß seine eigene Speichervirtualisierung nutzen
- Vorteile:
  - Zonen werden als VNodes nutzbar
  - Speichervirtualisierung / FC / RSIO auf einem Knoten
  - Data Mobility und Spiegelung zw. FC und RSIO
- On-demand-Einbindung ultraschneller Devices über vorhandene Infrastruktur

OSL Physical Device Server - Beispiel

Beschleunigung von Batchläufen / DB-Importen etc. mit hohem Random-I/O

- Auch Enterprise-RAID-Systeme stoßen bei hoher Random-I/O-Last und großen Daten-Volumina schnell an die Grenzen (ca. 80-160 MByte/s)
- Einbindung eines RSIO-Servers mit internen SSD, Durchsatz bei beliebigem Random-I/O > 1GByte/s
- Einbindung kann z. B. ad hoc über vorhandene 10GE-Karten erfolgen

Mögliches Szenario:

- Online-Migration auf OSL Physical Device Server
- Durchführung lastintensive Aufgaben
- Erwarteter Durchsatz > 500MByte/s bei 2 x 10GE
- Rückmigration auf konventionellen Storage auch online

Migrationen

Solaris – Linux - VMs

OSL Storage Cluster 4.0

Herausragende Cross-Plattform-Funktionalitäten

- Alle freigegebenen Platformen auch im Mix zu fahren:
  - Solaris 7 bis 11
  - Linux (SLES/RHEL, div. Releases)
  - SPARC und x86
- Keine Änderung der Prozesse bei Plattformwechseln
- Spezielle Lösungen z. B. für Migration von Solaris 10 nach Solaris 11
  -> extrem geringer Aufwand z. B. für Zonen
- Storage-Cluster-Anwender führen Wechsel der Server-Hardware, des Betriebssystemstandes, der Speichersysteme oder der SAN-Infrastruktur seit Jahren mit extrem geringem Aufwand durch

virtualization and clustering – made simple