Flexibel und hochverfügbar mit Oracle Solaris

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

bert.miemietz@osl.eu

Virtual Storage & Clustering

Speichervirtualisierung

Worum geht es bei der Speichervirtualisierung?

- Hardwareabstraktion
- Verschieben physikalischer oder logischer Limits (Größen, Volumezahl)
- verbesserte I/O-Performance
- verbesserte Verfügbarkeit
- Herstellen einer Eignung für Disaster Recovery
- Daten online verschieben und reorganisieren
- Zusatzfunktionen wie
  - Bandbreitensteuerung
  - Backup to Disk
  - permanent Backup u./o. Snapshots
- entscheidend: intelligenter Cluster-Support (s. folgende Folien)

Speichervirtualisierung im Cluster

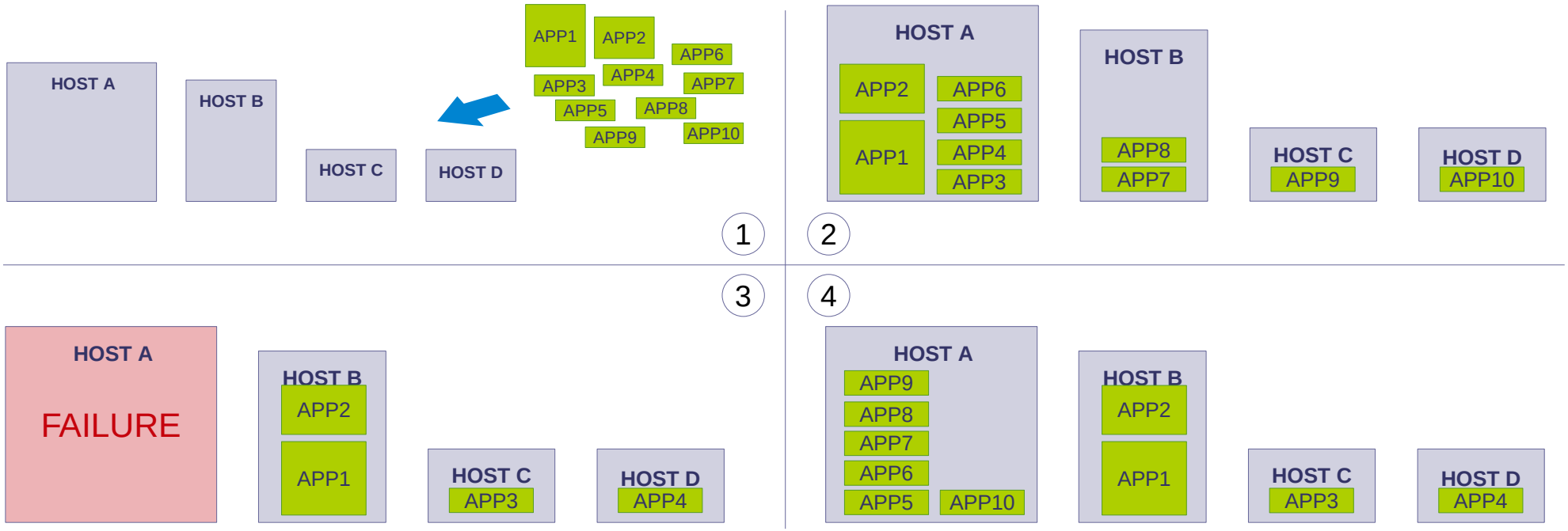
Vorteile mit der Speichervirtualisierung des OSL Storage Clusters u. a.:

- Speichervirtualisierung und Cluster in Einem -> überlegenes Design
- globaler Storage-Pool – Enterprise Storage Directory
- enorm vereinfachtes Gerätehandling
  - globale Geräte / globaler Namensraum / wahlfreie Gerätenamen
  - alle Anschlußtechniken von SCSI und iSCSI über FC und Infiniband nach gleichem Schema
  - integriertes, leicht verständliches Multipathing, Dynamic Hardware Reconfiguration Capabilities
  - identische Handhabung von Solaris 7 bis Solaris 11, Sparc, x86 und sogar Linux
  - **steht auch für Zonen und LDOMs zur Verfügung**
- automatisiertes Disk-Access-Management
  - allgemein enormer Sicherheitsgewinn in Clusterumgebungen bei NULL-Administration
  - ZFS sicher auf Shared Storage / im Cluster betreiben
- Applikations- / VM-Bewußtsein
  - applikationsbezogene, automatisierte Aktionen (Spiegeln, Backup-to-Disk, Backup-to-Tape, DR)
  - Übersicht Nutzung Storage Pool nach Applikationen / VMs
  - Allokation und Bandbreitensteuerung nach Applikationen
- herausragende Performance und Bandbreitensteuerung
  - keine Appliance, kein Flaschenhals – beliebige Verfügbarkeits- und Durchsatzskalierung
  - Fähigeit zur Bandbreitensteuerung (per Volume und per Applikation/VM)

Merkmale der OSL-Clustertechnologie

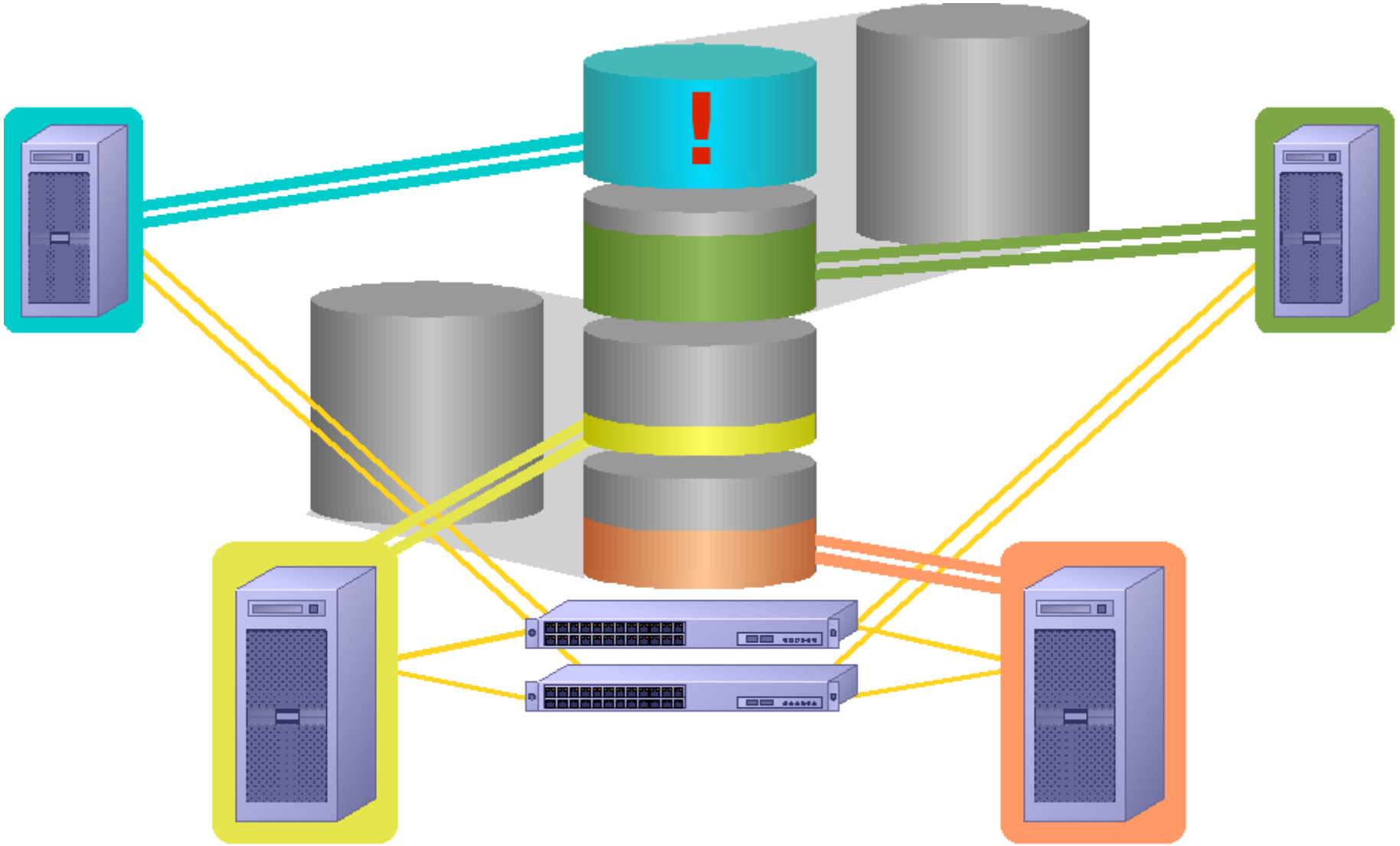
Nicht nur enge Verknüpfung mit der Speichervirtualisierung

- selbstorganisierend – Berücksichtigung / Steuerung Ressourcen
- vollkommen symmetrisch
- herausragende Robustheit - kein Split Brain
- zentrale Administration von jedem Knoten aus
- Cross-Platform
- verknüpft mit Speichervirtualisierung (Applikationsbewußtsein, Backup & DR ...)



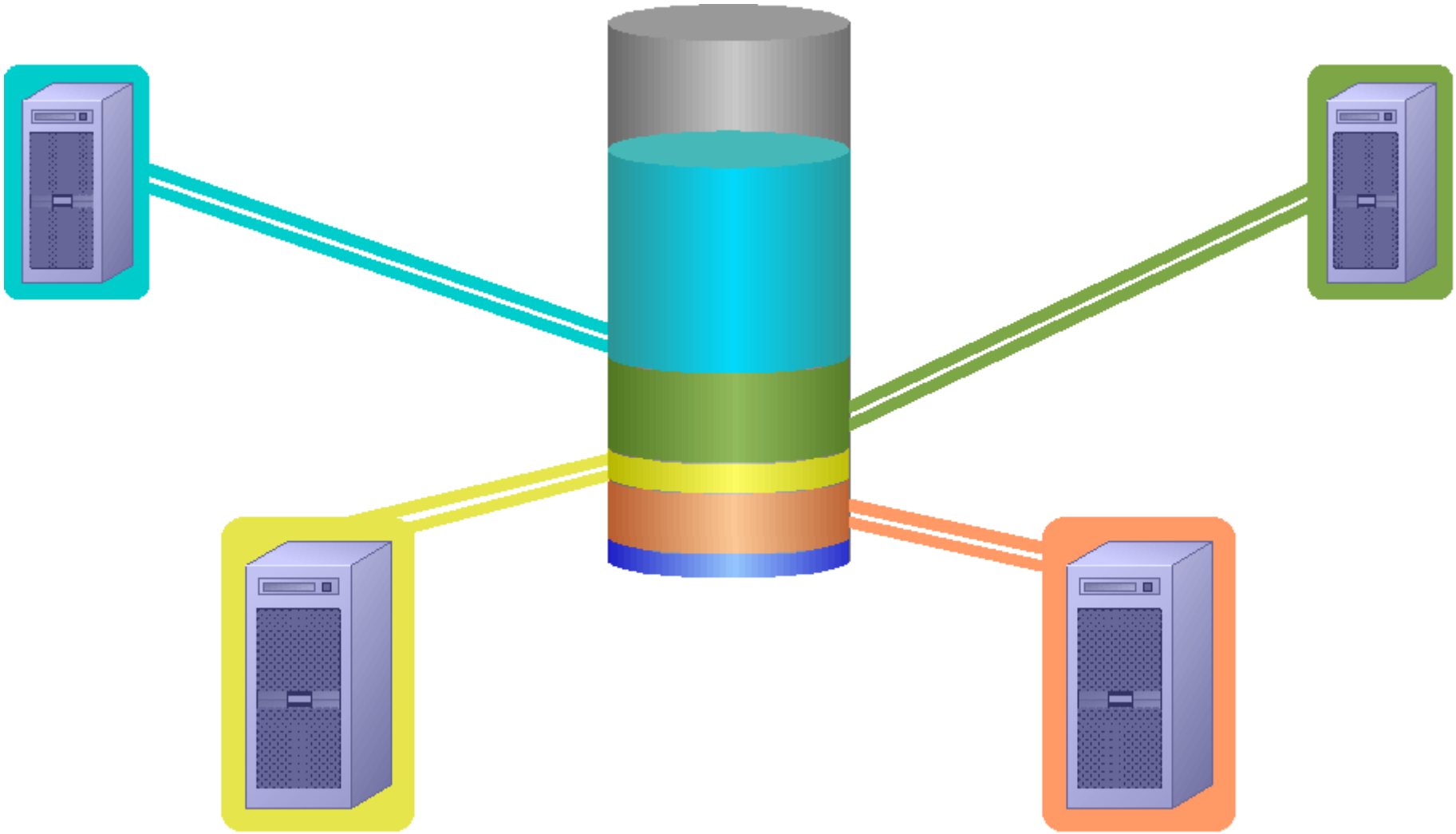
Virtual Storage + Cluster als Einzellösungen

Komplexe Ansammlung von Einzelbausteinen



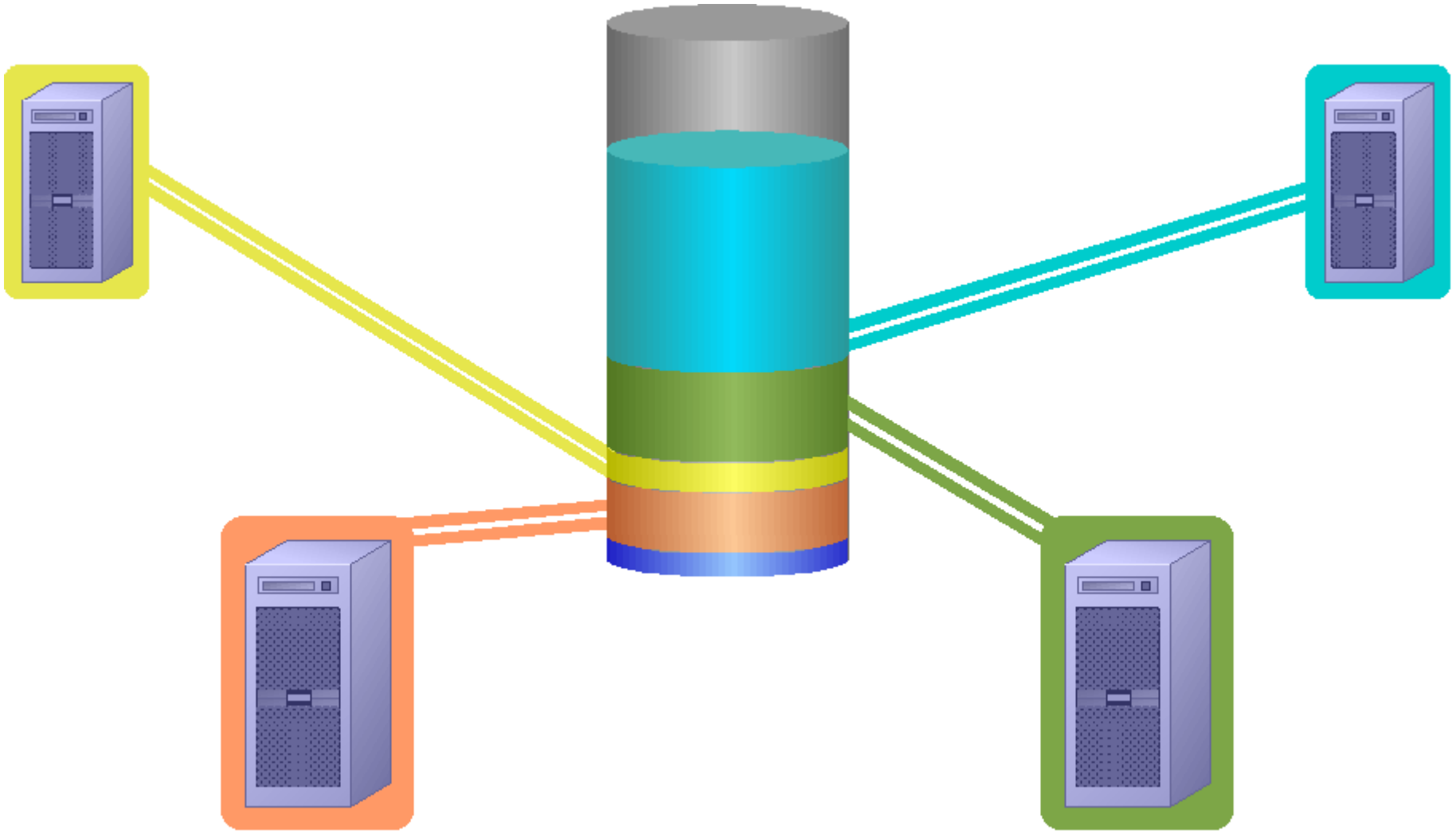
Disk Virtualizing Storage Cluster

Integrierte Sichtweise



Disk Virtualizing Storage Cluster

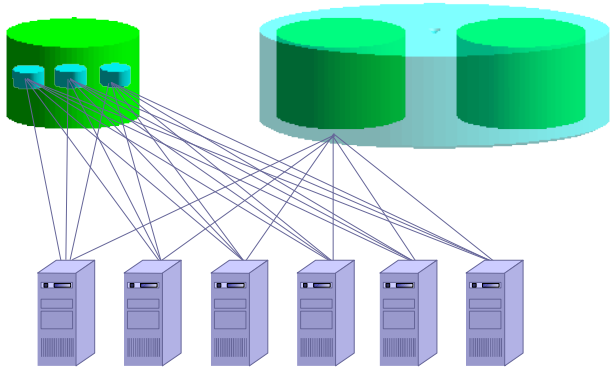
hostbasiert, hardwareabstrakt, flexibel, skalierbar, austauschbare Komponenten

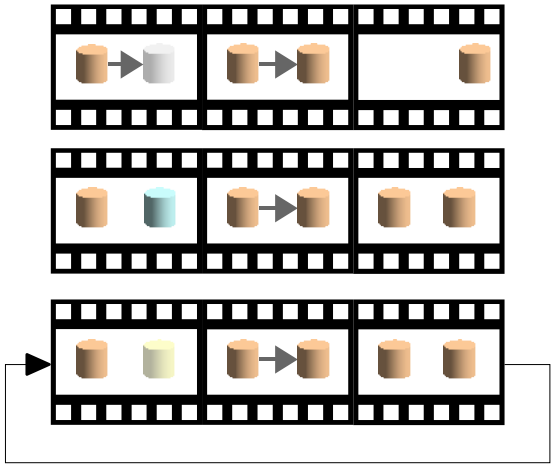


OSL (Disk Virtualizing) Storage Cluster

Überzeugende Funktionalität

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

Neues 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):

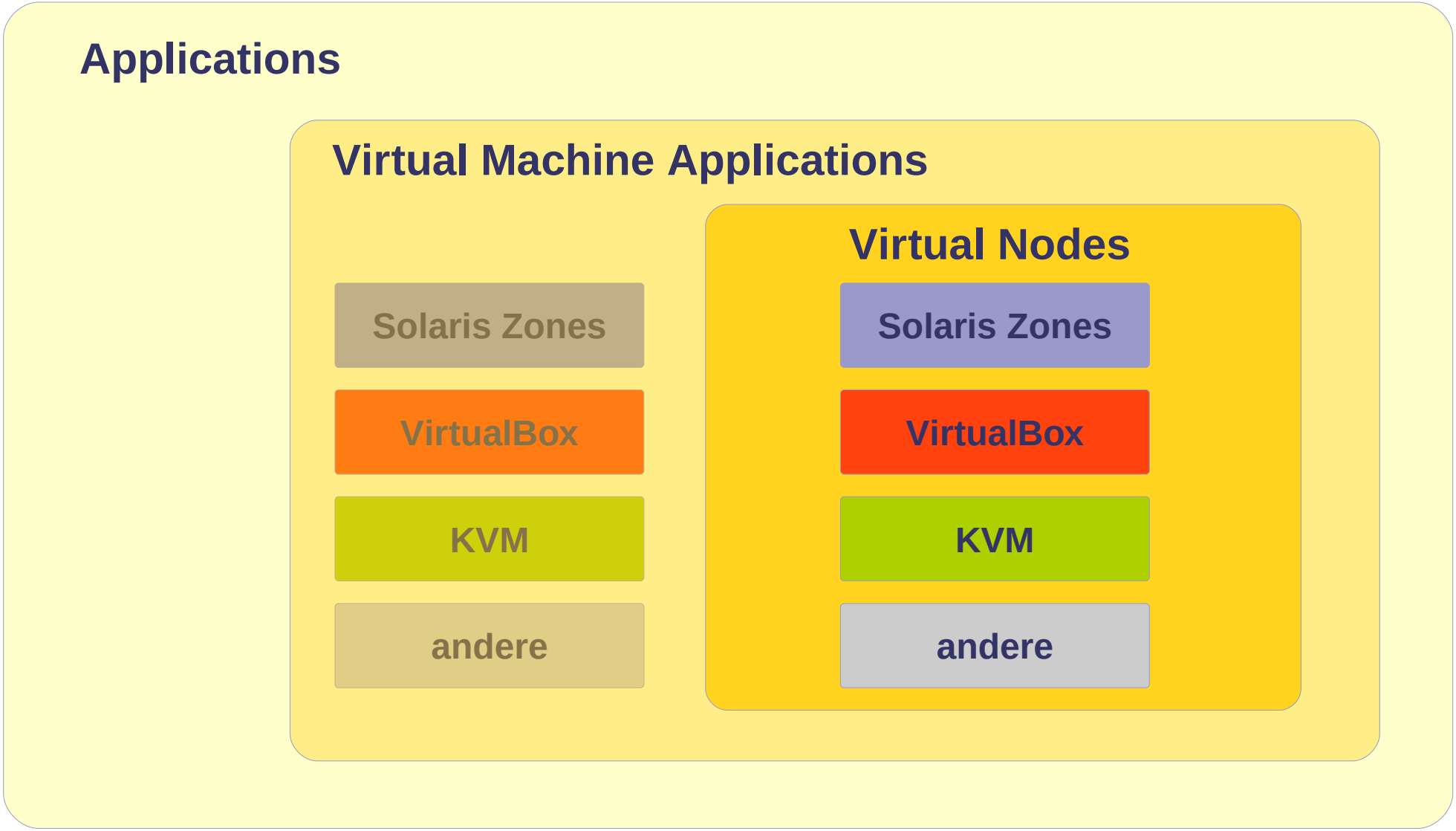
```
#type      id          parameter      dev-args
zpool      gurke       altroot=/      gurke-a@0 gurke-b@0 gurke-c@0
```

- 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 anstelle vfstab erstellen.

Das neue Virtual-Node-Konzept

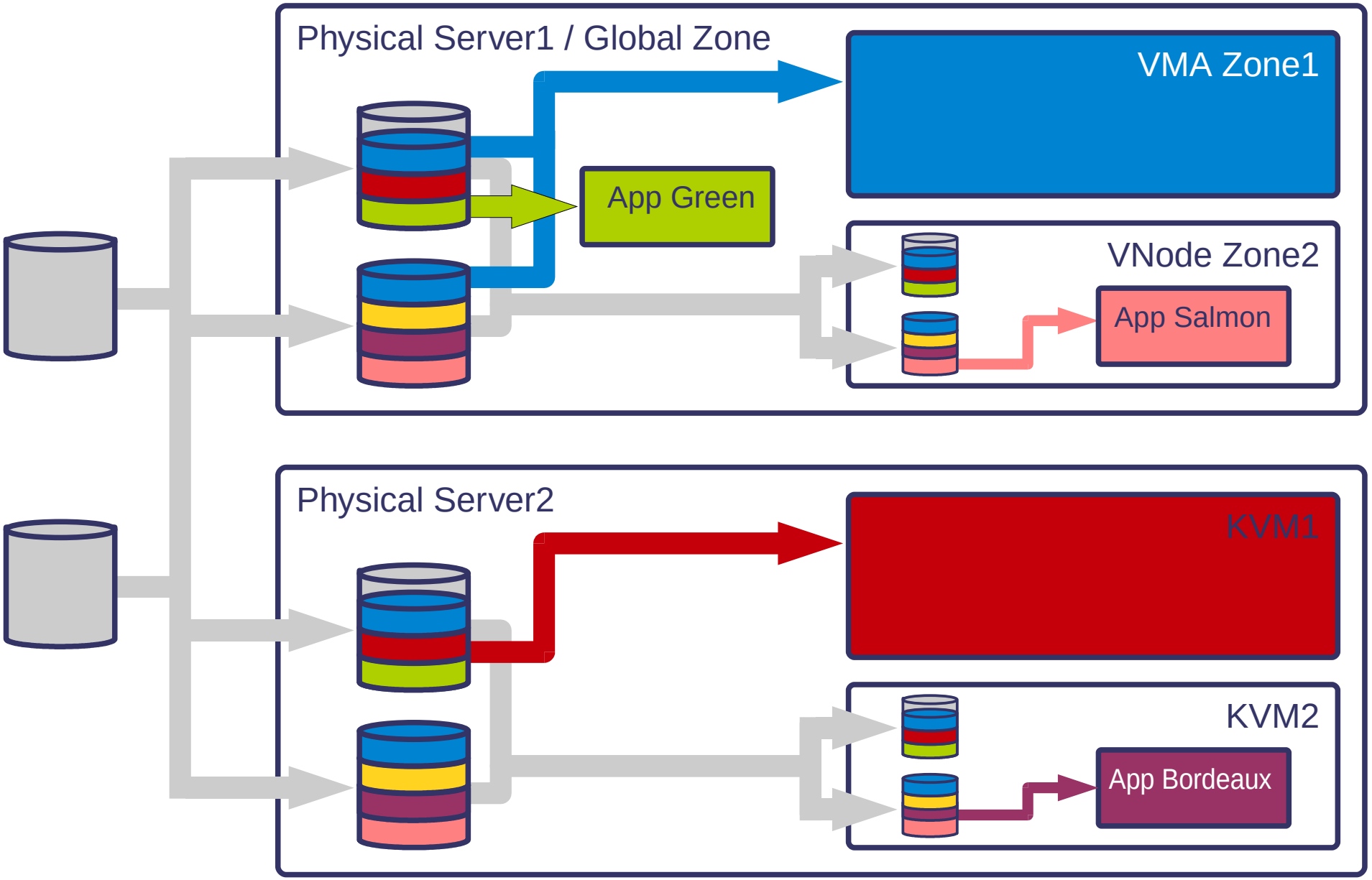
Zur Systematik von Applikationen

Applications – Virtual Machine Applications – Virtual Nodes



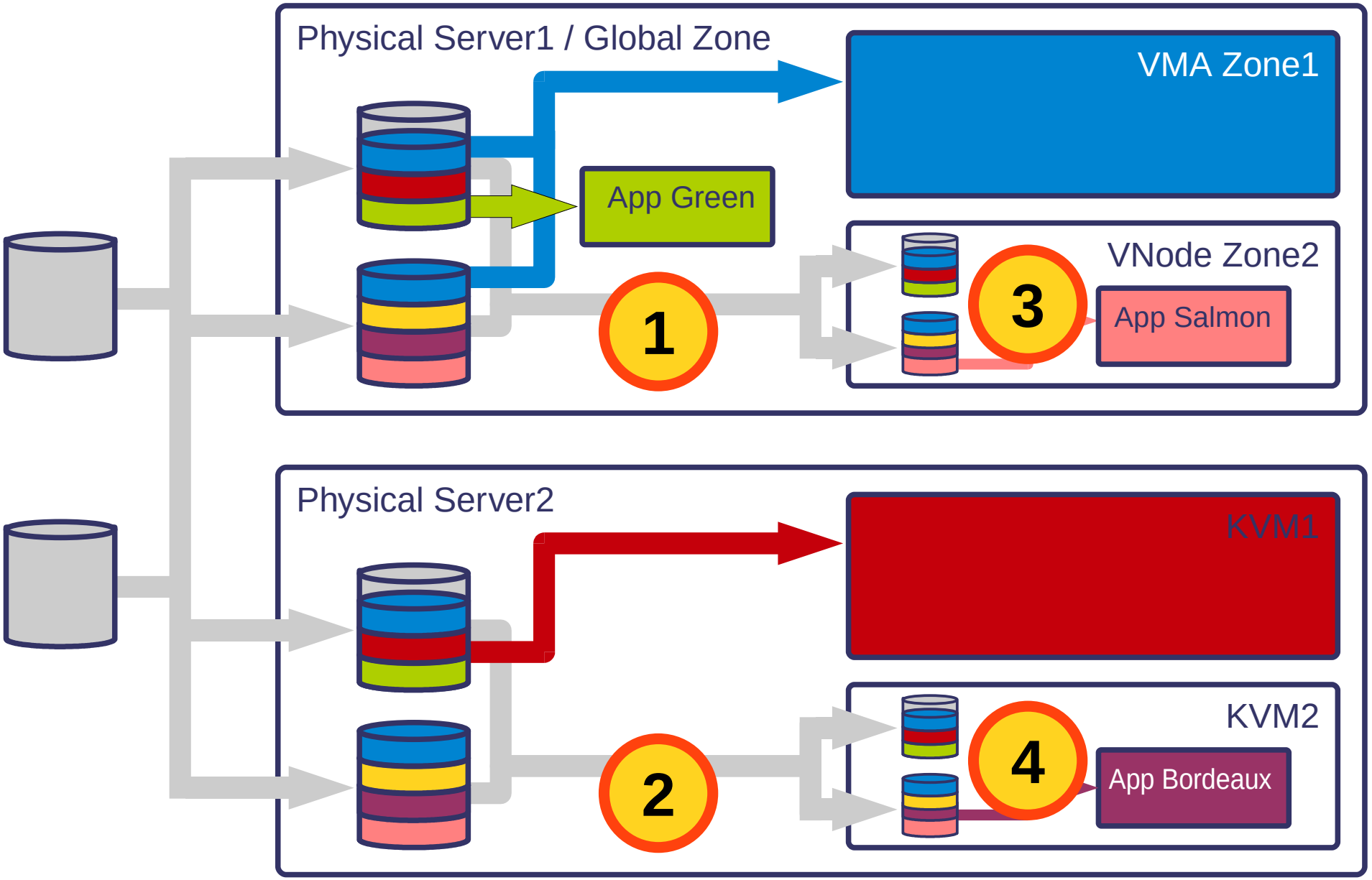
VMA und VNode im Clusterframework

Bedeutung für Clusterframework und Speichervirtualisierung



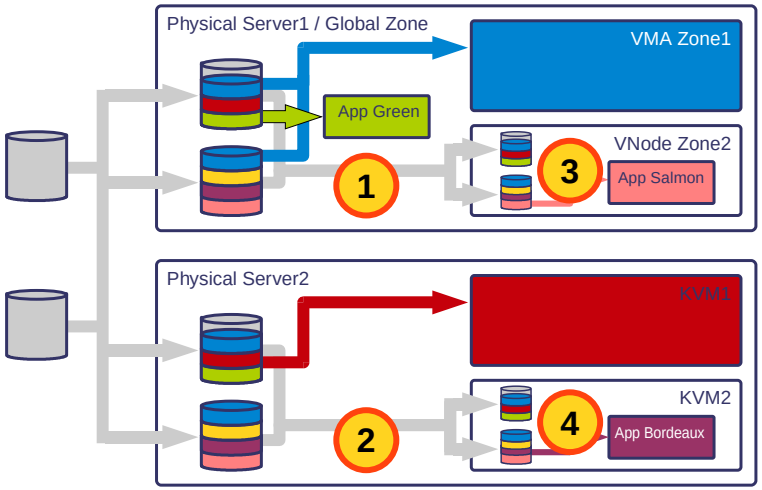
VMA und VNode im Clusterframework

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.

Gemeinsamkeiten von VMs

Einbindung in ganzheitliches Konzept

- Anlegen / Ressourcensteuerung über vmadmin

```
# vmadmin -c vm_name -F {lkvm | lfxen | lpxen | vbox | szone | ...}
```

- wesentliche Definitionen sind zentral: CPU / Memory / Storage
- Start / Stop / Failover über normale SC-Mechanismen
- Installation ggf. automatisierbar (typspezifisches Verfahren)
- typabstraktes Beschreibungsformat in **`vmconfig(1m)`** mit permanentem globalem Zugriff und Versionshistorie
- VM wird systemspezifisch beim Start erzeugt / beim Stop gelöscht
- selbstverständlich failoverfähig / Livemigration dort wo möglich
- automatischer Aufbau der notwendigen Netzkonfiguration (hardwareabstrakt)

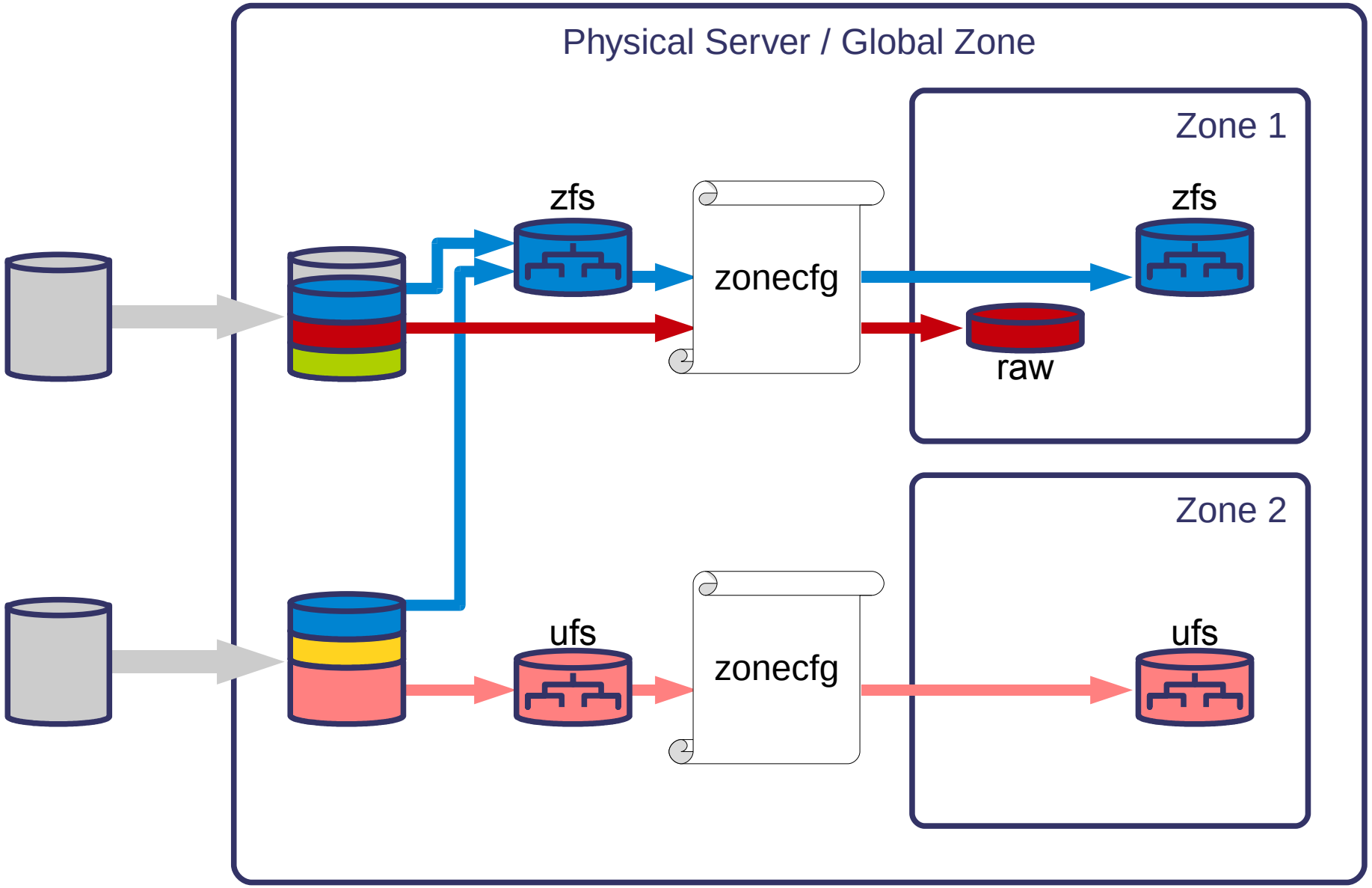
Zonen im OSL SC – eine normale VM

Einbindung in ganzheitliches Konzept

- Anlegen / Ressourcensteuerung über vmadmin
  - # vmadmin -c vm_name -F {lkvm | lfxen | **szone** | ...}
  - Zuweisung CPU / Memory / Storage
- Start / Stop / Failover über normale SC-Mechanismen
- automatisiertes Anlegen über Menüsystem oder zone_install
- Detail-Konfiguration ab 4.0 über Solaris-Standard-Werkzeuge
- Solaris Zones waren schon immer Zonen auf Shared Storage und schon immer failoverfähig
- Backup to Disk / Tape integriert (dvam-Tools)
- selbstverständlich failoverfähig
- automatischer Aufbau der notwendigen Netzkonfiguration (hardwareabstrakt)
- Fähigkeit zur Migration von Solaris 10 nach Solaris 11

Zonen im OSL SC als normale VM

Ein genauerer Blick auf Disk Devices und Dateisysteme



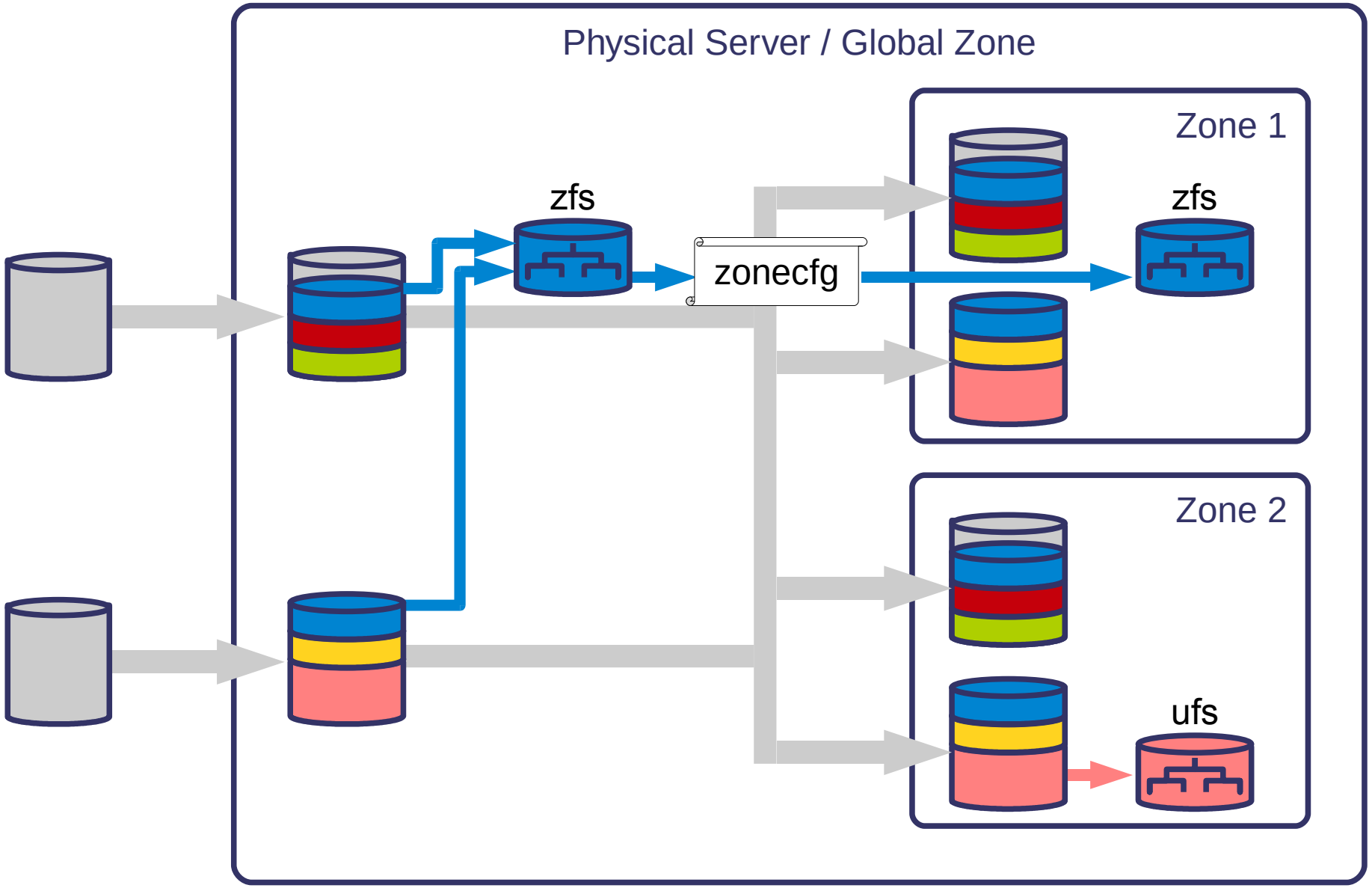
Zonen als Virtual Node

Die Zone als Cluster-Node

- Storage Cluster ist ab Version 4.0 in der Zone installierbar und lauffähig
- Zone wird zum Clusterknoten
- Anwendungssteuerung damit bis in die Zone hinein
- Anwendungen können zwischen Hosts und Zonen beliebig migrieren
- Zone-Failover auf dem gleichen Host, zwischen Domainen oder Hosts
- voller Zugriff auf Speichervirtualisierung in der Zone
- Application Specific Devices + Disk Access Management für Zonen (ASM!)
- Applikation kann aus der Zone heraus Spiegel steuern,
  Multipathing administrieren usw.
- Zone kann sich selbst spiegeln
- Zone kann (fast) wie ein normaler Host betrieben werden (vfstab ...)

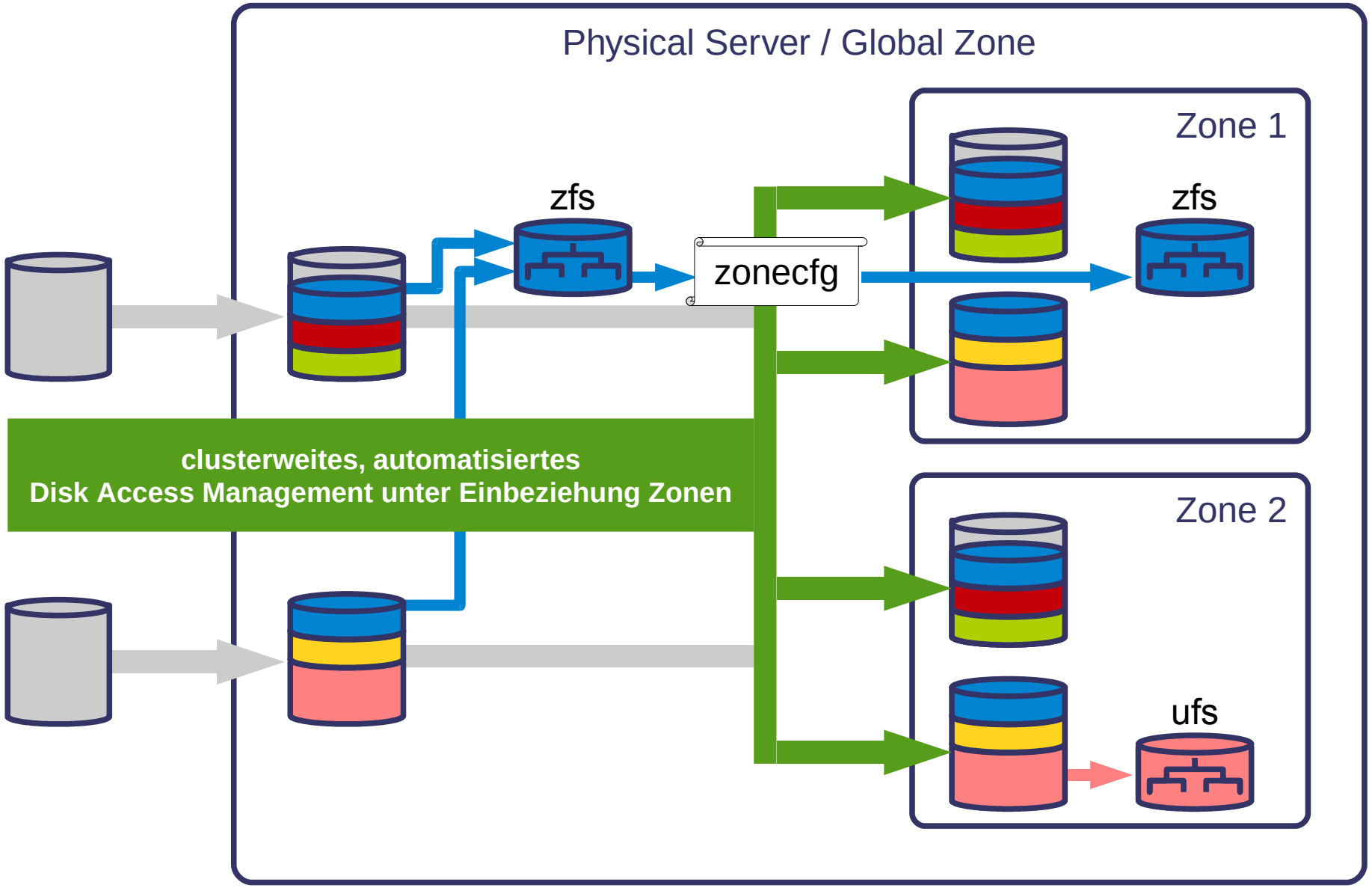
Zonen als Virtual Node

Ein genauerer Blick auf Disk Devices und Dateisysteme



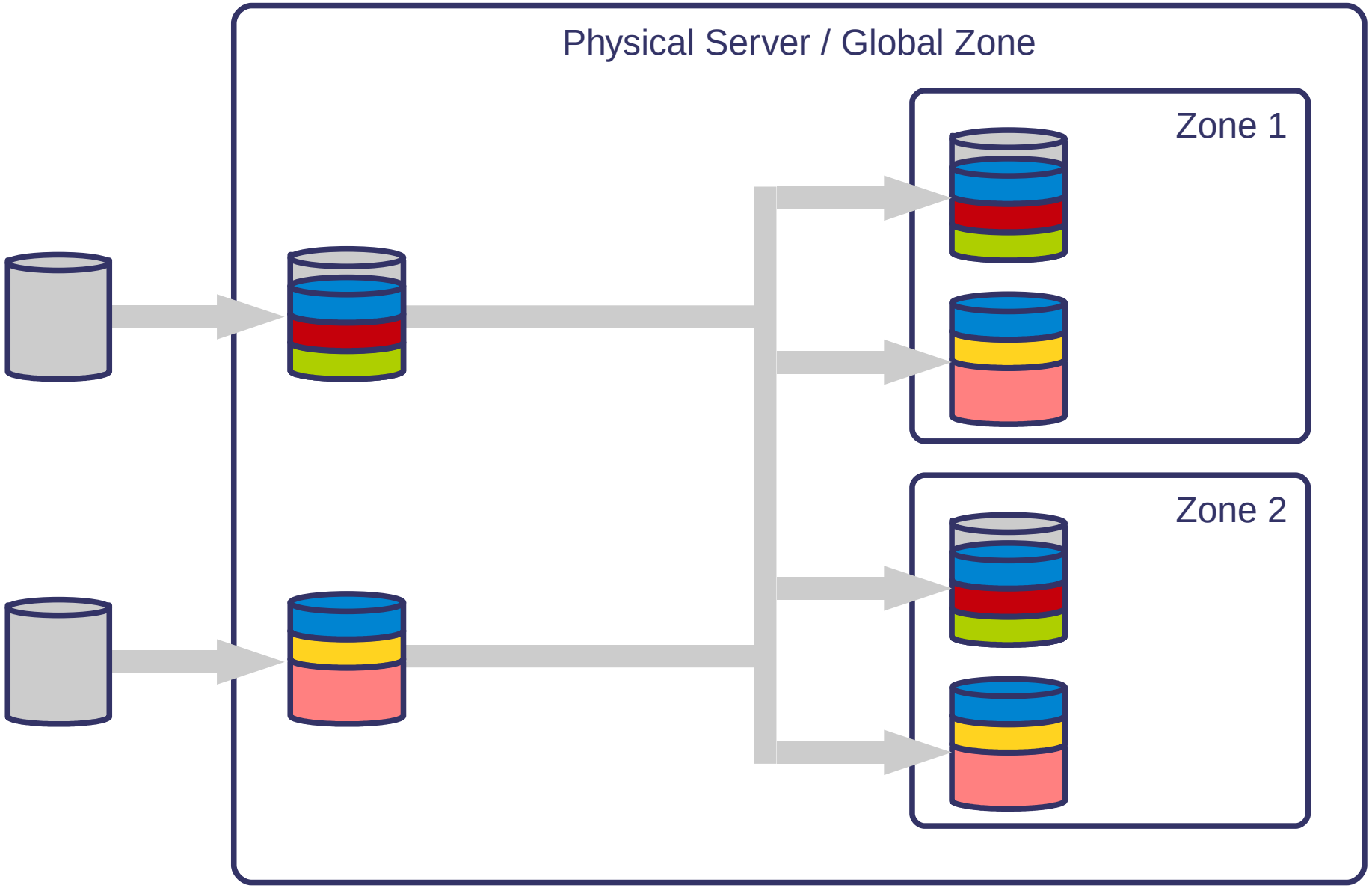
Zonen als Virtual Node

Vereinfachung und mehr Sicherheit nicht nur mit Raw Devices



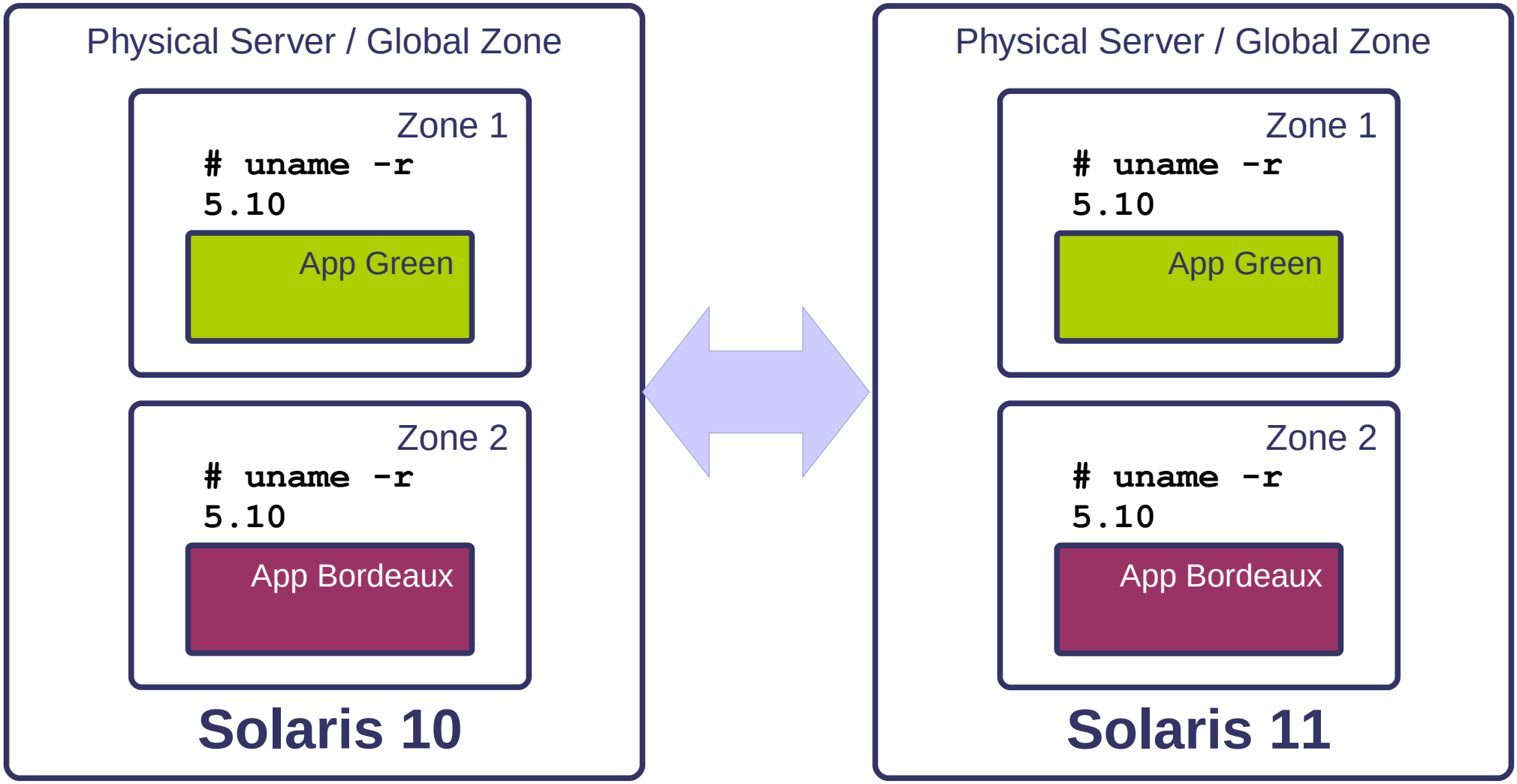
Zonen als Virtual Node

Einheitliche Sichtweise auf allen Ebenen



Zonen in der Migration auf Solaris 11

Branded Zones für eine Migration in kleineren Schritten



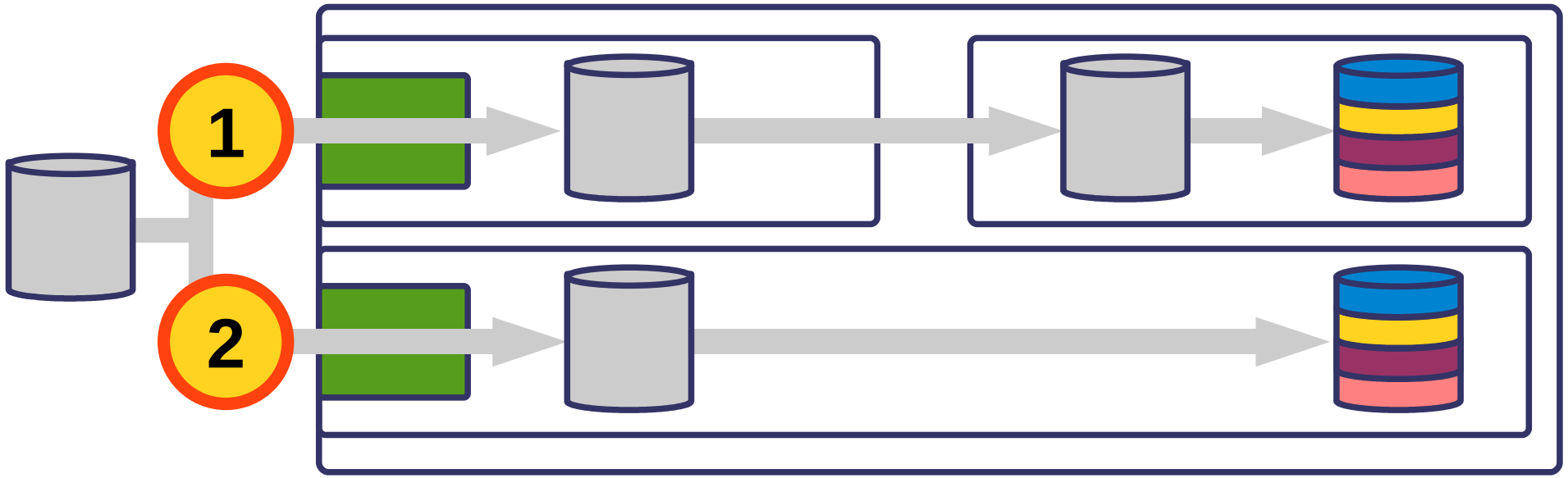
- **Vorteile von Solaris 11 nutzen**
- **keine Änderung an der Anwendungsumgebung**

LDOMs
im OSL Storage Cluster

LDOM – Physical oder Virtual Node?

Derzeit noch Physical Node – Variante mit FC-Access

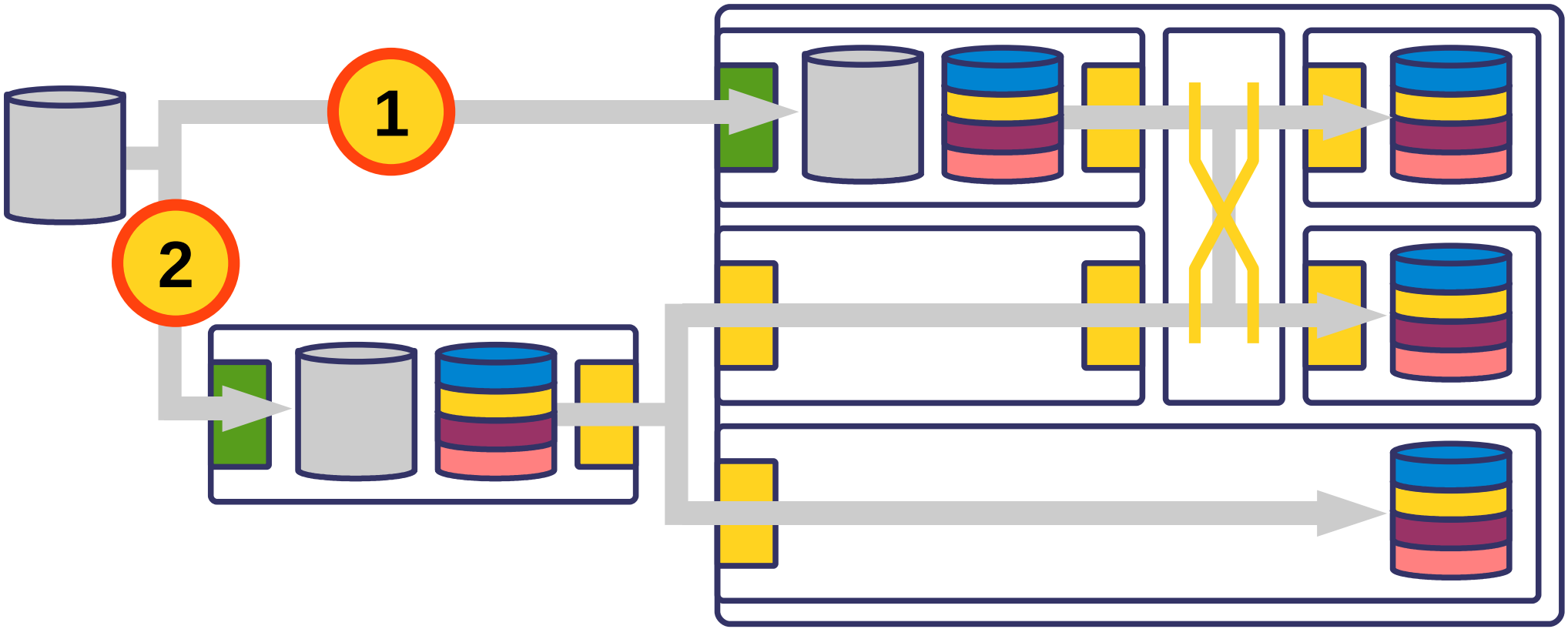
- LUNs bereitgestellt über:
  1. Service-Domain -> Guest Domain
  2. native I/O-Domain
- OSL Storage Cluster wird in der Ziel-Domain installiert, Virtualisierung setzt auf LUNs auf
- Nur Variante 1 liefert Live-Migration, ist aber sehr limitiert
  - aufwendige Gerätekonfiguration
  - Limitierung LDCs
- Variante 2 limitiert die Zahl möglicher Domainen erheblich



LDOM in Verbindung mit RSIO

Die LDOM als RSIO-Client / Virtual Node

- V-Storage via RSIO:
  1. FC-to-Ethernet in Service-Domain
  2. über externen RSIO-Server und Service-Domain
  3. über externen RSIO-Server und Ethernet-I/O-Domain
- Live-Migration immer möglich
- Zahl LDOMs sehr groß
- flexibles und schlankes Device-Handling selbst für tausende Geräte



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

Zusammenfassung

Solaris – warum mit OSL?

Ausgeklügelte, integrierte Gesamtarchitektur – das steckt dahinter

- Anwendungsbeschreibung für alle Funktionen sichtbar
  - Speichervirtualisierung mit "Applikationsbewußtsein"
  - ⇒ enormer Funktionsgewinn
  - Selbstkonfiguration (z. B. Backup für komplexe Applikationen mit 1 Kommando)
- Trennung OS - Applikationen
  - wichtige Informationen leben unabhängig von Host und Speichersystem weiter
  - Eliminierung von Migrationsaufwänden
- Plattformunabhängigkeit
  - Mischung verschiedener Rechnerarchitekturen
  - Mischung verschiedener Betriebssysteme
  - Mischung verschiedener Connectivity-Lösungen
- Intelligenz aus der Infrastruktur zum Host zurückholen
  - einfache Handhabung (Schulungsaufwände!)
  - höhere Verfügbarkeit
  - leichtere Migration
  - funktionale Überlegenheit
    (s. direkter Zugriff auf Anwendungen und Speichervirtualisierung, Disk Access Manager ...)

virtualization and clustering – made simple