# Solaris Container

## Grundlagen und Best Practices

**Detlef Drewanz**
Senior Systems Engineer, Operating Systems Ambassador
**Sun Microsystems GmbH**

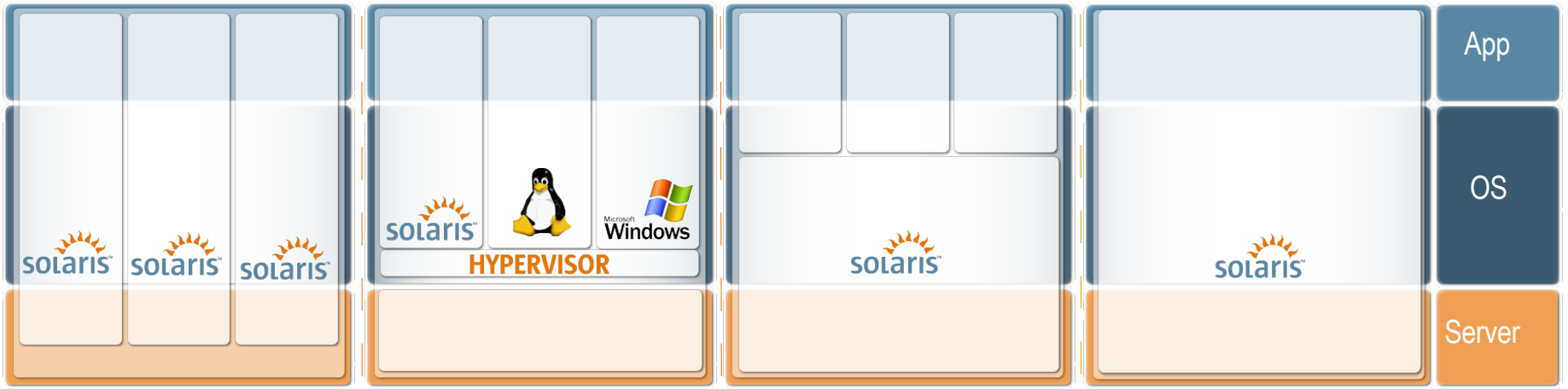
# Virtualisierung mit Solaris

**Physikalische Virtualisierung**
**(Domains/ Physikalische Partitionen)**

**Logische Virtualisierung**
**(LDom, xVM*, VMware)**

**OS Virtualisierung**
**(Solaris Container, Solaris Trusted Extensions, BrandZ)**

**Ressource Management**
**(Solaris Ressource Manager)**



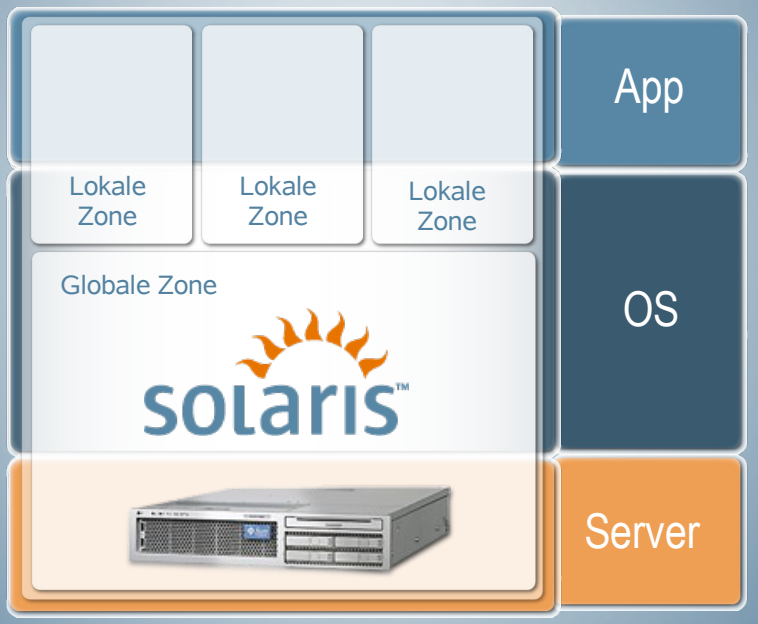

HW-Fehler Abgrenzung

Eigenes OS/ Separater Speicher

Gemeinsames OS/ Gemeinsamer Speicher

Isolation der OS-Umgebung

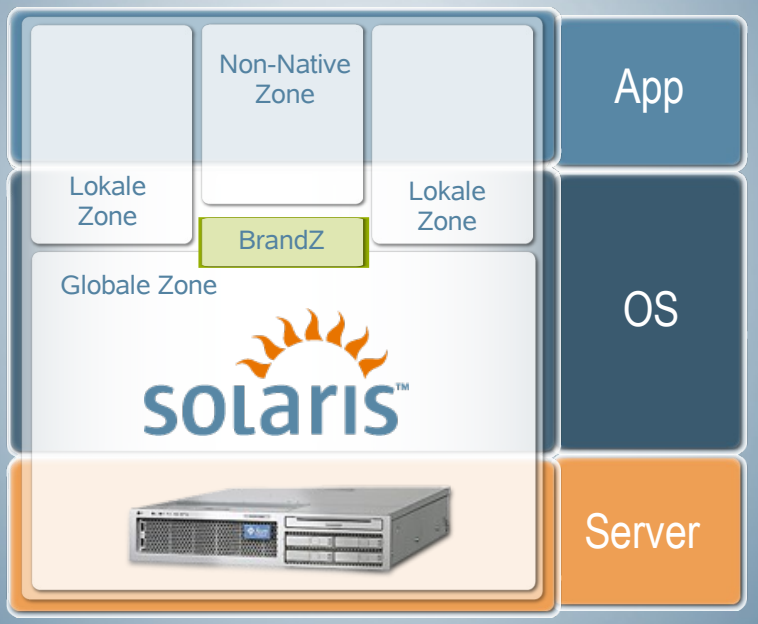
Gemeinsame OS-Umgebung




- OS-Virtualisierung
- Eine OS-Instanz
  - Tausende Umgebungen
- Partitionierung von Anwendungen
- Konsolidierung
- Begrenzung von Ressourcen
- Schneller Neustart
- Clonen
- Migration

# Solaris Container

## Solaris Zonen und Ressource Management

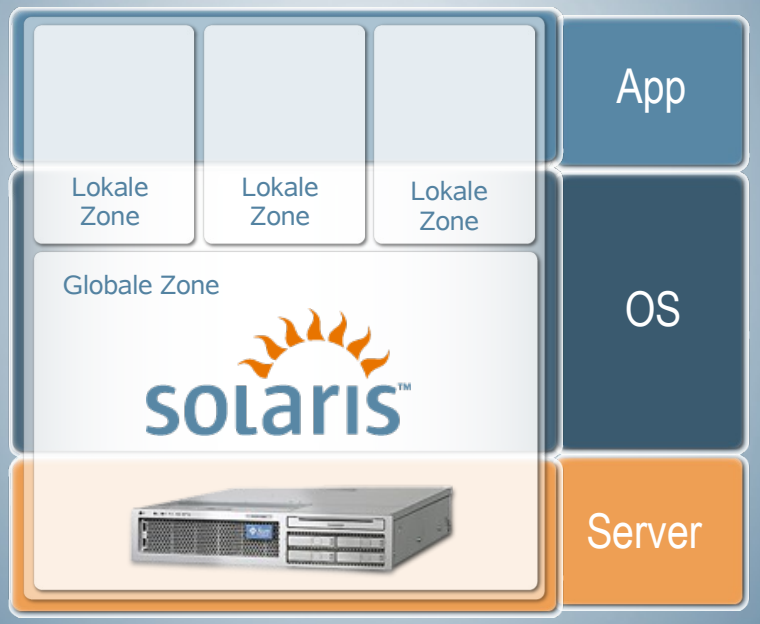



- Unterstützung Nicht-nativer Zonen
- Laufzeitumgebung ohne Kernel
  - wie Solaris Zonen
- Solaris Container für Linux Anwendungen
  - lx-Brand (nur x86/x64)
  - 32-Bit RedHat 3.5 - 3.8
  - 2.6 Kernel in Nevada
- Solaris 8 Migration Assistant
  - Solaris 8-Brand (nur SPARC)
  -
  -

# Branded Zones

# Der lx Brand

- Erzeugung einer lx Zone
  - zonecfg -z lx-zone "create -t SUNWlx; set zonepath=/export/home/lx-zone"
  - Installation von Linux Software in die Zone
    - zoneadm -z lx-zone install -d archive_path (CD/DVD,ISO-Image, tarball)
  - Läuft auf dem Solaris Kernel
- Linux Software selbst wird nicht durch die lx-Zone mit Solaris distributiert




- OS-Virtualisierung
- eine OS-Instanz
  - > viele Umgebungen
- Partitionierung von Anwendungen
- Konsolidierung
- Begrenzung von Ressourcen
- Schneller Neustart
- Cloning und Migration

# Solaris Container

## Solaris Zonen und Ressource Management

# Allgemeine Anwendungsfälle für Solaris Container(1)

- Konsolidierung
  - Lastoptimierung von Anwendungen
  - Zusammenfassung von Netzwerken
  - Monitoring
  - Backup
- Hosting
  - Webserver, Mailserver

# Allgemeine Anwendungsfälle für Solaris Container(2)

- Softwareentwicklung
  - Trennung von Entwicklung/Test/ Qualitätskontrolle/Produktion
  - Konsolidierung von Testsystemen
- Isolation
  - Grid Computing
  - Security Bereiche für Anwendungen
  - Schulungssysteme

# Virtualisierung mit Containern

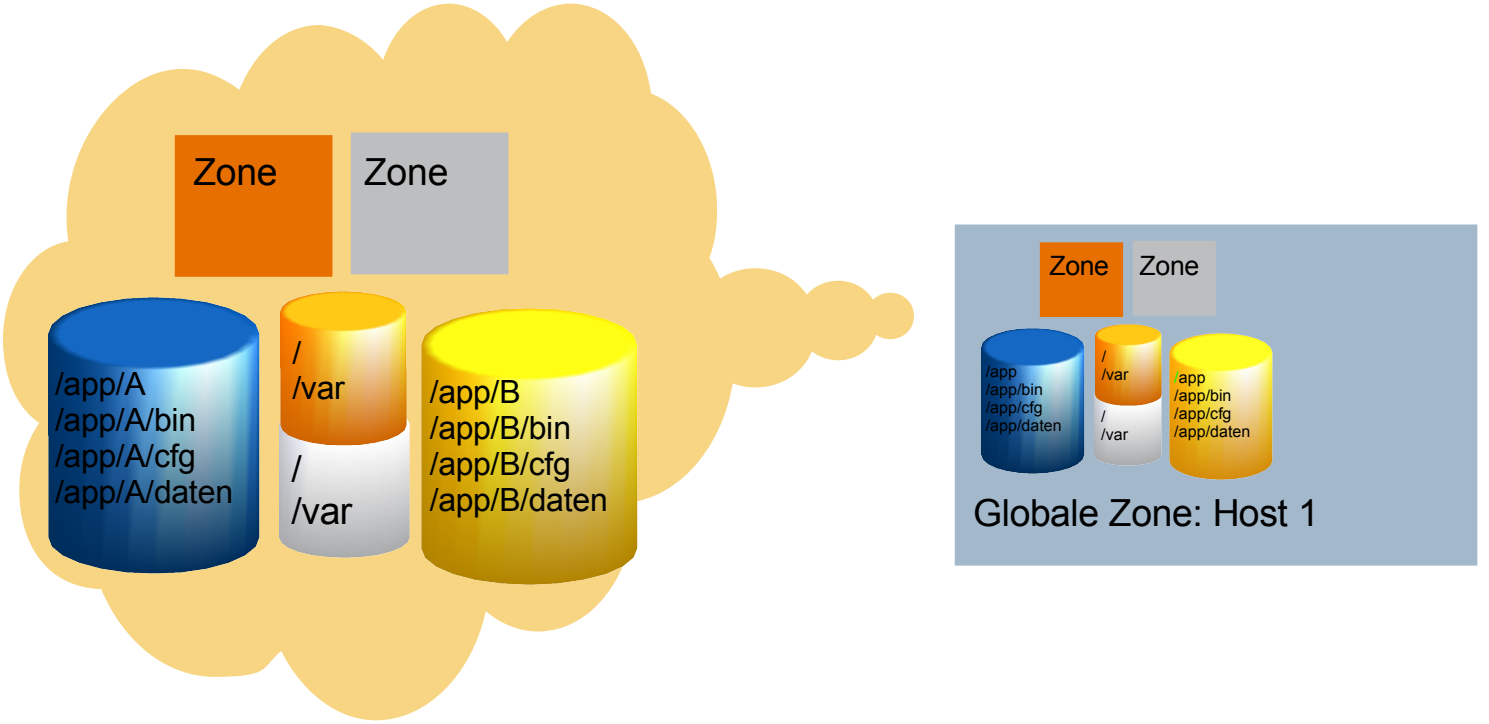
- Abbildung von Services in Containern
  - Eine Anwendung = Ein Container
- Anwendungen erfordern einfache und umfassende Isolation der Systemressourcen
  - Netzwerk, Disk, Memory, CPU, Prozesse, Umgebung
- Herausforderungen für das Lifecycle Management von Containern
  - Installation, Patch, Upgrade

# Virtualisierung mit Containern

- Abbildung von Services in Containern
  - > Eine Anwendung = Ein Container
- Anwendungen erfordern einfache und umfassende Isolation der Systemressourcen
  - > Netzwerk, Disk, Memory, CPU, Prozesse, Umgebung
- Herausforderungen für das Lifecycle Management von Containern
  - > Installation, Patch, Upgrade

# Anwendungen unabhängig von Zonen

- Zone als neutrale Laufzeitumgebung
  - > Ein Service *appA* pro Zone (serviceorientierte Zone)
  - > Service wird durch Filesystem bestimmt
  - > Service ist unabhängig von der Zoneninstallation
  - > Service kann zwischen Zonen bewegt werden



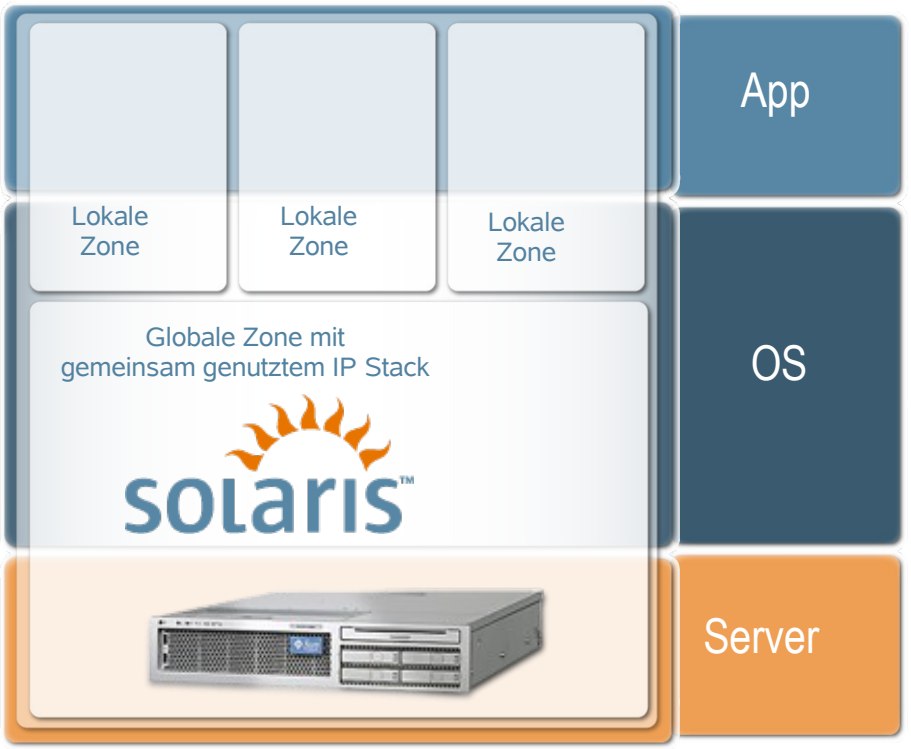
# SunCluster und Solaris Containers

- Failover Zone / HA Container Agent
  - > Zone als Ressource ansehen
  - > Monitoring der Zone
  - > Auslösung:
    restart einer Zone
    oder
    failover
- Zone Node
  - > Zone als virtuellen Node ansehen
  - > Monitoring der Anwendung in der Zone
  - > Auslösung:
    restart der Anwendung in der Zone
    oder
    failover in eine Zone in einem anderen System

# Virtualisierung mit Containern

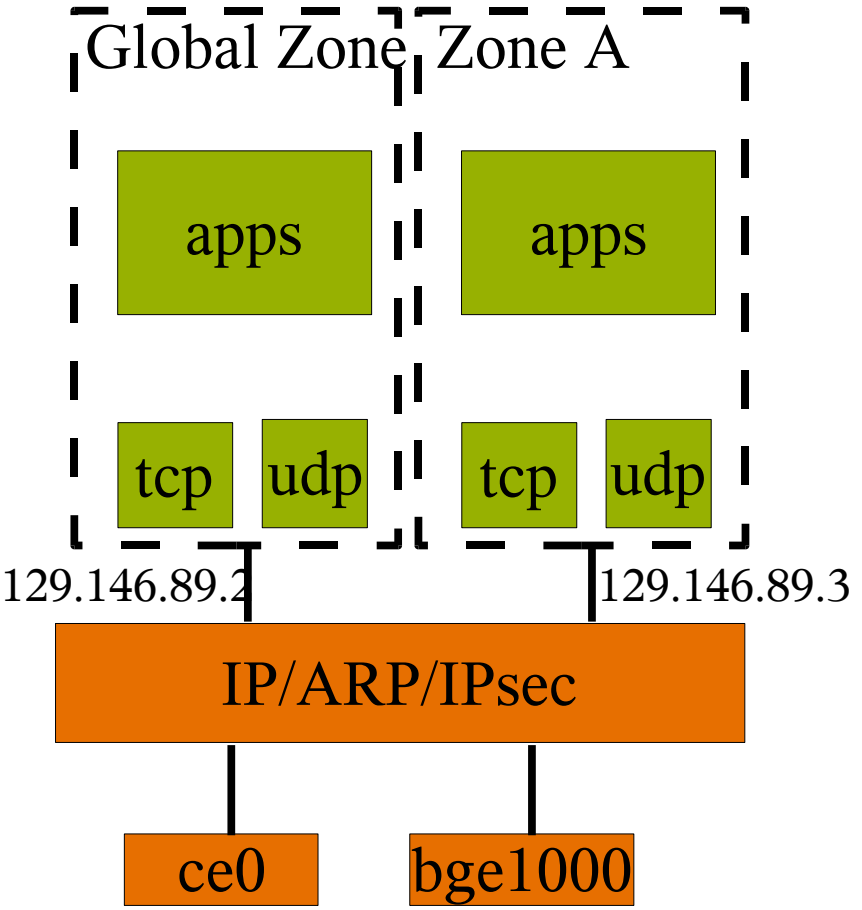
- Abbildung von Services in Containern
  - > Eine Anwendung = Ein Container
- Anwendungen erfordern einfache und umfassende Isolation der Systemressourcen
  - > Netzwerk, Disk, Memory, CPU, Prozesse, Umgebung
- Herausforderungen für das Lifecycle Management von Containern
  - > Installation, Patch, Upgrade

# Netzwerke und Solaris Zonen



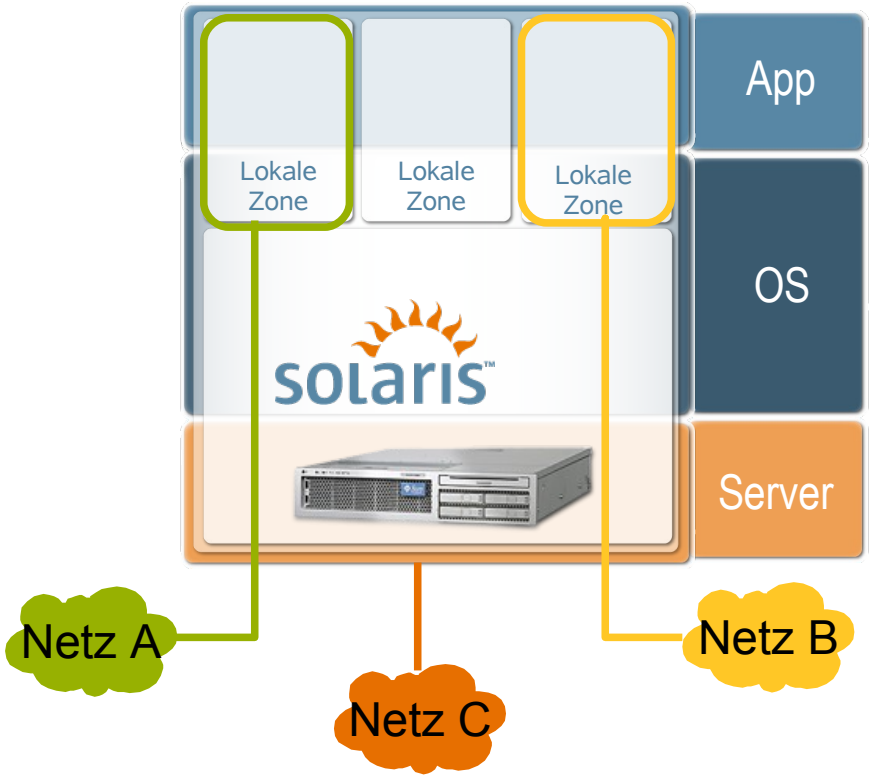
# Stand heute: Shared-IP Zonen

- Anwendungen sind separiert
- Zonen nutzen die zugewiesenen IP-Adressen
- IP/ARP/IPsec für alle Zonen gemeinsam genutzt
  - > Routing, ARP, Konfiguration
- TCP, UDP, SCTP für jede Zone separiert



# Isolierte Netzwerke und Solaris Zonen

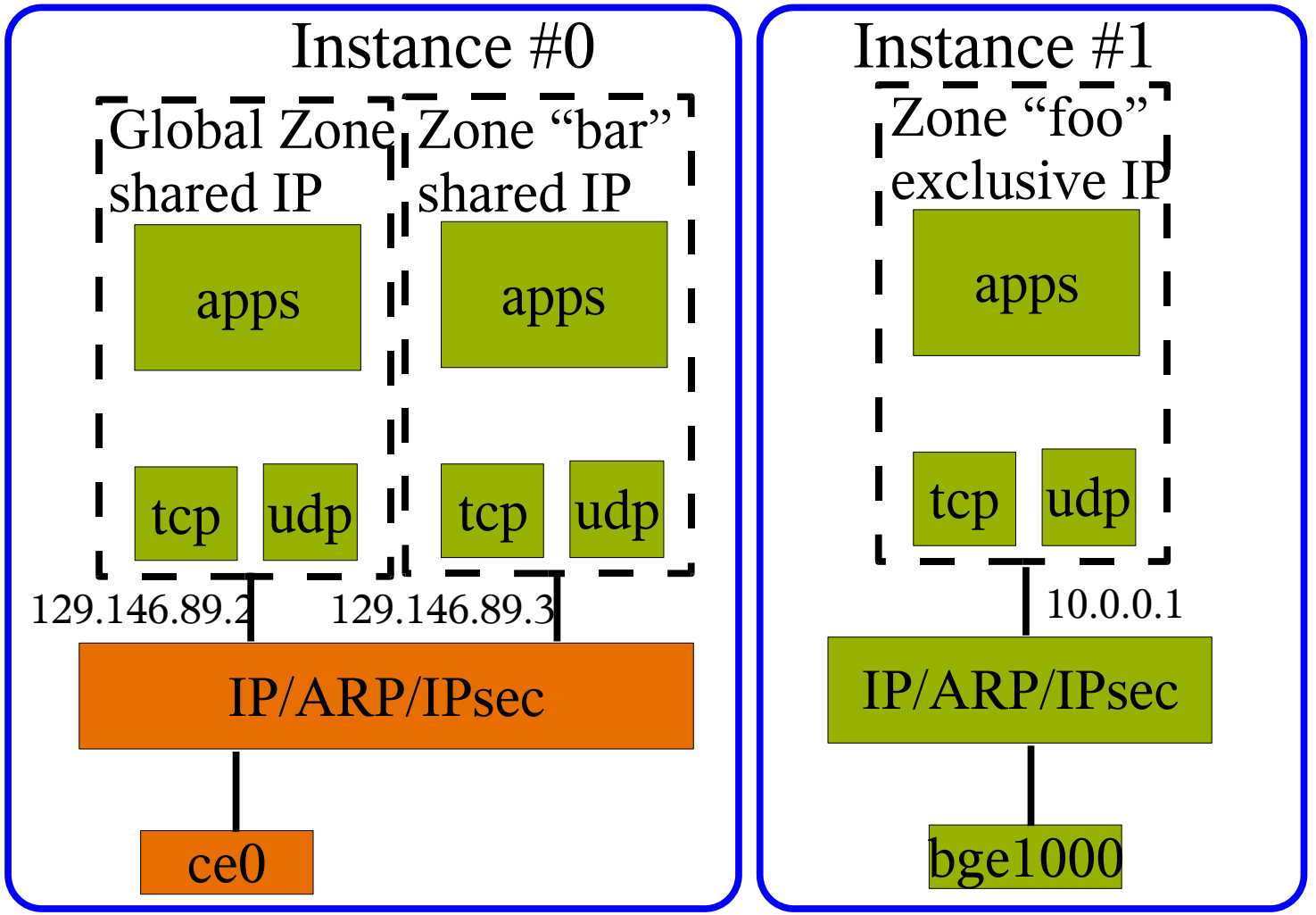
- Unterschiedliche Netzwerke für einzelne Zonen ?
  - > Mit vollständiger sicherer physikalischer Isolation
  - > Separate Interfaces oder Vlans
  - > Separates Routing und/oder IP-Parameter



# IP Isolation: Mehrere IP Instanzen

## Shared-IP und Exclusive-IP

- Eine shared-IP und mehrere exclusive IP-Instanzen



# IP-Isolation: Was ist separiert ?

- IP Routing Tabelle
- ARP Tabelle
- IPsec Konfiguration
- IP Filter Konfiguration und Statistik
  - > Konfiguration in der lokalen Zone
  - > Separates Projekt in S10 7/07
    - > IP Filter zwischen shared-IP Zonen durch die globale Zone
- TCP/IP ndd Variablen
  - > Nicht für ndd-Variablen der Datalink Device Treiber
- snoop ist begrenzt für das (V)LAN der lokalen Zone

# IP-Instanzen Limitierungen (1)

- Nur für GLDv3 Treiber Interfaces (Projekt Nemo)
  - z.Zt.: bge, e1000g, ixge, xge, nge, rge
  - dladm show-link darf nicht “legacy” zeigen
- IP Adresse der exclusive-IP Zone nicht in zonecfg konfigurierbar
  - /etc/sysidcfg vor dem ersten Boot der Zone installieren
  - Aufwändigere Änderung der IP-Adresse eine Zone durch die globale Zone

# IP-Instanzen Limitierungen (2)

- Kaum Netzwerklimitierungen durch Globale Zone durchsetzbar
  - > Limitierung welches LAN/VLAN die exclusive-IP Zone nutzen darf
  - > Keine Limitierung der Nutzung des Netzwerkes
  - > IP Filter in der exclusive-IP Zone
  - > Zone admin hat alle Möglichkeiten der Nutzung des Netzwerkes: IP Adresse ändern, ARP, Raw-Access, Snoop

# Exclusive-IP Stack Konfiguration

```
# dladm show-link
 bge0        type: non-vlan  mtu: 1500          device: bge0
 bge1        type: non-vlan  mtu: 1500          device: bge1
 # zonecfg -z excl
 zonecfg:excl> set ip-type=exclusive
 zonecfg:excl> add net
 zonecfg:excl:net> set physical=bge1
 zonecfg:excl:net> end
 zonecfg:excl>
```

- VLAN-Konfig: vlanid * 1000 + NIC instance Nummer
  - > z.B. vlan 3 auf bge1 = 3 * 1000 + 1 = bge3001
- Wichtig: N exclusive-IP Zonen im gleichen (V)LAN benötigen N Interfaces

# Netzwerkvirtualisierung: Ausblick

- Projekt Crossbow
  - VNICs
    - Mehrere Datalink Interfaces auf einem NIC
    - DHCP ohne separate (V)LANs
  - Resource controls
    - Einfache Bandbreitenbegrenzungen per Zone/domU

# Virtualisierung mit Containern

- Abbildung von Services in Containern
  - > Eine Anwendung = Ein Container
- Anwendungen erfordern einfache und umfassende Isolation der Systemressourcen
  - > Netzwerk, Disk, Memory, CPU, Prozesse, Umgebung
- Herausforderungen für das Lifecycle Management von Containern
  - > Installation, Patch, Upgrade

# Ressourcen-Limitierung für Zonen

## Heute Häufig genutzte Möglichkeiten

- CPU
  - > Prozessor Sets (mit Ressource Pools)
    - zonecfg pool
  - > Garantierte CPU-Zeit für die Zone
    - zonecfg rctl: zone.cpu-shares
  - > Garantierte CPU-Zeit für ein Projekt
    - /etc/project: project.cpu-shares
- Begrenzung der max. LWP in einem Container
  - zonecfg rctl: zone.max-lwps (fork-Schleifen vermeiden)
- Hauptspeicher
  - Capping des phys.Hauptspeicherverbrauches (rcapd) in der Zone
  - IPC rctls per /etc/project gesetzt
- Begrenzung von /tmp beim mount in /etc/vfstab
  - mount_tmpfs -o size=xxx

# Ressource Management und Container

- Stand Solaris 10 11/06
  - Resource Controls (rctladm, prctl)
  - Resource Pools (poolcfg) und Prozessorsets (psrset)
  - Resource Capping (rcapd) in lokaler Zone
  - Fair Share Scheduler (FSS) - cpu-shares, tasks, projects
- Komplexe Syntax
- Umfangreiches Handbuch
- Aufwändig zu konfigurieren
  - wenig Nutzung in aktuellen Projekten

# Neue Ressource Controls ab S10 7/07

```
# prctl -i zone <zonename>

NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.max-swap
        system          16.0EB   max    deny                         -
zone.max-locked-memory
        system          16.0EB   max    deny                         -
zone.max-shm-memory
        system          16.0EB   max    deny                         -
zone.max-shm-ids
        system          16.8M    max    deny                         -
zone.max-sem-ids
        system          16.8M    max    deny                         -
zone.max-msg-ids
        system          16.8M    max    deny                         -
zone.max-lwps
        system          2.15G    max    deny                         -
zone.cpu-cap
        system          4.29G    inf    deny                         -
zone.cpu-shares
        privileged      1        -      none                         -
        system          65.5K    max    none                         -
```

# Ressource Manager Integration
## (ab Solaris 10 7/07)

- Einfache Benutzung von Ressource Controls
- Automatische Konfiguration von Infrastruktur
  - z.B. FSS durch die Benutzung von cpu-shares einschalten
- Persistente Konfiguration für die globale Zone
- Konfiguration gehört zur Zone (zonecfg)
- Capped Ressourcen
  - Absolutes Verbrauchslimit für Ressourcen
- Dedicated Ressourcen
  - Direkt zugewiesene exclusive Ressourcen

# Neue Ressource Controls

| | Dedicated | Capped |
|---|---|---|
| **CPU** | Temporärer Pool (pset) | cpu-caps* |
| **Memory** | Temporärer Pool (mset*) | rcapd & rctls |

# Temporäre Pools

- Konfiguration per zonecfg
  - > Konfiguration zieht bei Migrationen mit
  - > Wenn die Anzahl cpu nicht verfügbar ist, bootet die Zone nicht --> Wichtig bei Migrationen
- Pools werden beim Boot der Zone erzeugt
  - > Name SUNWtmp_<zonename>
- Pool ist genau einer Zone zugewiesen (dedicated)
- poolstat für Kontrolle

```
zonecfg:keetonga> add dedicated-cpu
zonecfg:keetonga:dedicated-cpu> set ncpus=1-4
zonecfg:keetonga:dedicated-cpu> set importance=30
zonecfg:keetonga:dedicated-cpu> end
```

# Temporäre Pools - Anpassungen

- Anpassungen der laufenden Zone
  - > Hinzufügen einer CPU

```
poolcfg -dc 'transfer 1 from pset pset_default to SUNWtmp_keetonga'
```

  - > Entfernen einer CPU

```
poolcfg -dc 'transfer 1 from pset SUNWtmp_keetonga to pset_default'
```

  - > CPU Range verändern

```
poolcfg -dc 'modify pset SUNWtmp_keetonga
    (uint pset.min=1; uint pset.max=3)'
```

# Capped Memory(1)

- Neue Ressource capped-memory per Zone
  - > Begrenzung des phys. Memory einer Zone
  - > Konfiguration und Kontrolle aus der globalen Zone

```
zonecfg:keetonga> add capped-memory
zonecfg:keetonga:capped-memory> set physical=1g
zonecfg:keetonga:capped-memory> end
```

- Anpassung der laufenden Zone

```
# rcapadm -z keetonga -m 2g
```

- rcapstat(1M) zeigt phys./virt. Memory und caps

# Capped Memory(2)

- Zu kleine memory caps führen zu Memory Paging
- Paging Limitieren durch Begrenzung des swap

```
zonecfg:keetonga> add capped-memory
zonecfg:keetonga:capped-memory> set swap=1g
zonecfg:keetonga:capped-memory> end
```

- Anpassung der laufenden Zone

```
# prctl -n zone.max-swap -v 2g -t privileged -r -e deny -i zone keetonga
```

# Capped Memory(3)

- Begrenzung des non-pageable Memory
  - > device-locked memory, ISM, mlock(3C) memory (DISM)
  - > proc_lock_memory Privileg ist nun in jeder Zone

```
zonecfg:keetonga> add capped-memory
zonecfg:keetonga:capped-memory> set locked-memory=100m
zonecfg:keetonga:capped-memory> end
```

- Anpassung der laufenden Zone

```
# prctl -n zone.max-locked-memory -v 200m -t privileged
        -r -e deny -i zone keetonga
```

# Ressource Controls und Scheduler Klasse für Zonen

- IPC Ressource Controls für Zonen
  - max-shm-memory max shared memory
  - max-shm-ids max number of shared memory IDs
  - max-msg-ids max number of message queue IDs
  - max-sem-ids max semaphore IDs

```
zonecfg:keetonga> set max-shm-memory=20m
```

- Scheduling Klasse für Zonen setzen

```
zonecfg:keetonga> set scheduling-class=fss
```

# Konfiguration der globalen Zone

- Persistente Konfiguration für die globale Zone
  - pool
  - cpu-shares
  - capped-memory
    - physical
    - swap
    - locked
  - dedicated-cpu
    - ncpus

# Ressourcemanagement: Ausblick

- CPU Caps
  - > CPU Verbrauch begrenzen - Sub-CPU
- Memory Sets
  - > Einen festen Hauptspeicherbereich einer Zone zuweisen

# Virtualisierung mit Containern

- Abbildung von Services in Containern
  - > Eine Anwendung = Ein Container
- Anwendungen erfordern einfache und umfassende Isolation der Systemressourcen
  - > Netzwerk, Disk, Memory, CPU, Prozesse, Umgebung
- Herausforderungen für das Lifecycle Management von Containern
  - > Installation, Patch, Upgrade

# Erzeugung von Zonen

- Schnelle Erzeugung von Zonen
  - JET
  - N1SPS
  - zoneadm clone Feature (mit ZFS* noch schneller)
  - Skriptbasiert mit sog. Template Mastern
- zoneroot in ZFS noch nicht offiziell supported
  - Upgrade/Life Upgrade funktioniert noch nicht mir Zonen im ZFS
  - pkg-/patch-Tools können den freien Platz im ZFS nicht sicher bestimmen

# Bewegen einer Zone (Migration)

- Seit Solaris 10 11/06
  - zoneadm detach und attach
    - detach: Status wechselt von *installed* in *configured*
  - erzeugt <*zonepath*>/*SUNWdetached.xml* mit
    - config, Packagelist, Patchlist
- Wichtig:
  - Package- und Patch-Stand in globaler Zone zwischen beiden Systemen müssen gleich sein
  - HW Environment
    - gemountete Devices
    - Interface-Namen
- keine Live Migration !

# Patchen von Zonen

- Zonen werden sequentiell gepatched
- Alle Zonen sind zusammen betroffen
- Lange Downtime durch Patching möglich
- Deshalb:
  - Life Upgrade zum Patchen von Zonen benutzen
    - Verfügbar für Zonen mit S10 7/07
  - Patching möglichst vermeiden
  - ggf. Anwendung migrieren und Zonen neu erzeugen

# Upgrade von Zonen

- Mit S10 7/07 wird Life Upgrade auch für Zonen funktionieren
  - > Details zu Life Upgrade siehe PTE 11/2006
- Life Upgrade kopiert mit lucreate die globale Zone und alle lokalen Zonen

# Funktionsweise Live Upgrade

- Upgrade einer Kopie der System-Installation
  - OS, Einstellungen, Anwendungssoftware, Daten
  - Boot Environment (BE)
- Erfordernisse
  - Packages SUNWluu, SUNWlur, SUNWlucfg)
  - aktuell gepatchtes OS (vor allem pkg- und patch-Tools)
    - Siehe SunSolve Infodoc 72099
  - Plattengröße: aktuelle Installation + 10%
- Unterstützte Solaris Versionen
  - SPARC: ab Solaris 2.6
  - x86: ab Solaris 7

# Ablauf Live Upgrade – auf einen Blick

- Kopieren des aktuellen Boot Environments
  - > lucreate
- Upgrade Boot Environment mit neuem OS-Image
  - > luupgrade
- Ggf. Patches oder Packages löschen/installieren
  - > luupgrade -T/-t bzw. luupgrade -P/-p
- Aktivieren des Boot Environment
  - > luactivate
- Reboot (init 6)
  - > bei Fehler: boot -s, luactivate, fix, ...

# Zusammenfassung

- Anwendungen in Zonen möglichst “beweglich” halten
- IP-Instanzen für GLDv3 Interfaces
  - NIC's direkt Zonen zuordnen
- Vereinfachtes Ressource Management für Zonen
  - dedicated CPU, Capped memory
  - Einfacher benutzbar
  - Globale Zone persistent konfigurierbar
- Life Upgrade für Zonen
  - Für Patching und Upgrade von Zonen nutzen

# Vielen Dank !

Detlef.Drewanz@sun.com

http://blogs.sun.com/solarium

# Wie Container einführen (1)

- Ziele für die Einführung klar machen
  - > Wo können Zonen helfen ?
  - > Wo erhöhen Zonen Komplexität ?
- Anwendung auf Solaris 10 verfügbar ?
  - > Besonderheiten der Anwendung in der Zone bekannt ?
- Erfahrungen mit Solaris 10 verfügbar ?
  - > Neue Features (SMF)
  - > Veränderte Betriebskonzepte

- Der 1. Schritt: Einführung von Solaris 10 !
- Der 2. Schritt: Einführung von Zonen

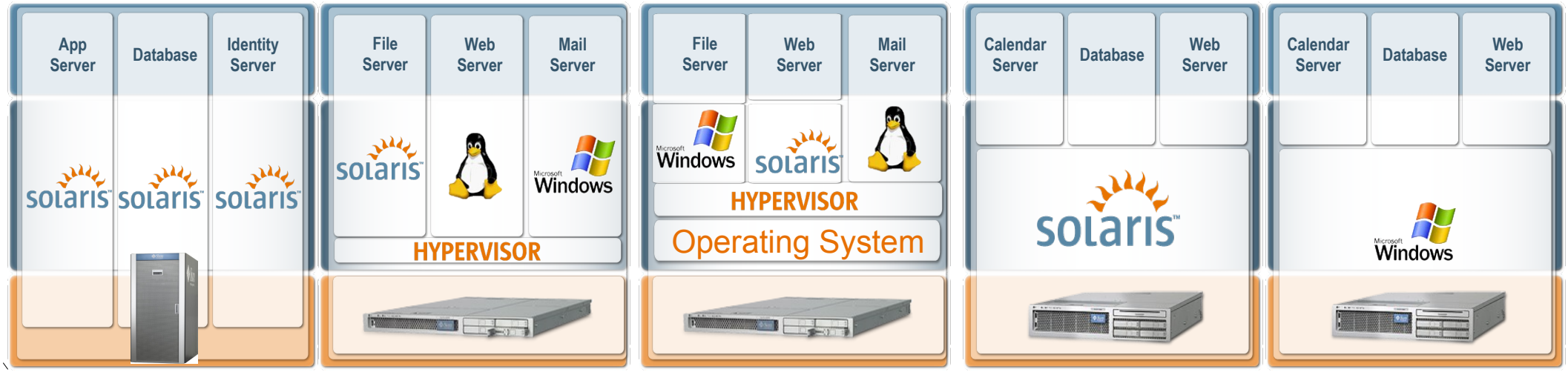
# Wie Container einführen (2)

- Zonentyp festlegen
  - Sparse-root Zone vs. Whole-root Zone
- Root-Plattenlayout
  - Zusätzliche Filesysteme
- Architektur vonZonen mit Anwendungen
  - Zonenorientiert oder Serviceorientiert
- Veränderte Betriebskonzepte mit Zonen
  - Automatisierte Installation von Zonen und Anwendungen
  - Zonen in Netzwerken
  - Installation, Abnahme, Monitoring, Backup, Patching, ...

# Agenda

- Solaris Container in der Entwicklung
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Virtualisierung für Server und Anwendungen




## Hard Partitions

Dynamic System Domains

## Virtual machines

Hypervisor: Type 1

Logical Domains

xVM Server

Xen

VMware

Microsoft Virtual Server

## Virtual machines

Hypervisor: Type 2

Virtual Box

VMware Workstation

VMware Server

Microsoft Virtual Server

## OS Virtualization

Solaris Containers (Zones + SRM)

## Application Virtualization

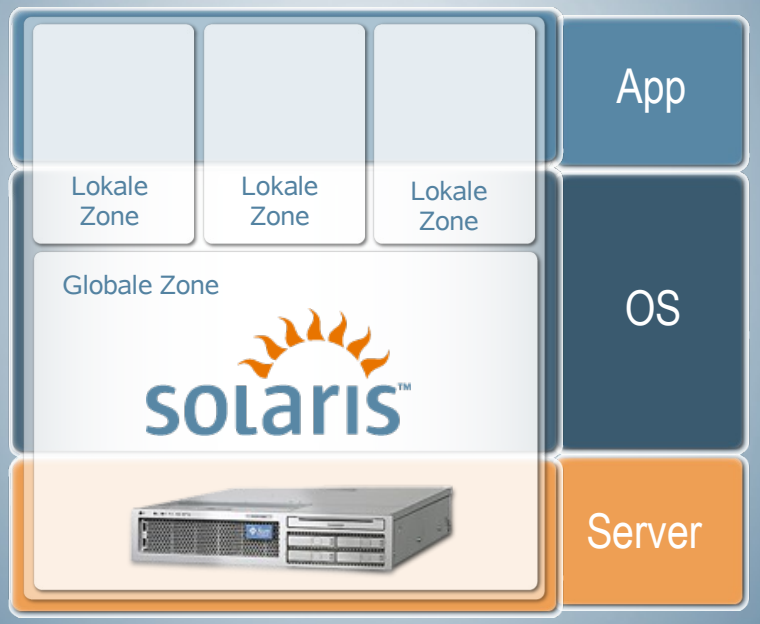
Solaris 8 & 9 Containers

Solaris Containers for Linux Applications

Sun xVM VDI

Microsoft SoftGrid

VMware ThinApp




## OS-Virtualisierung

- Eine OS-Instanz
- Viele Ausführungsumgebungen
- Wenig Overhead für die Virtualisierung

## Isolation

- CPU, Memory, Disk, Netzwerk, Prozesse, Umgebung

## Ressource Management

- CPU, Speicher, Prozesse

Sicher

Kostenfrei im Solaris enthalten

# Solaris Container

## Solaris Zonen und Ressource Management

# Ausblick: Zonen in OpenSolaris

- Grundlegende Funktionsweise wie bisher
- Zonen werden als Brand ipkg installiert
  - > Bei der Installation ist Internetzugang erforderlich
  - > Packages werden aus einem OpenSolaris Package Repository über das Internet installiert
- Die Standard-Zone zur Zeit ist „whole-root“
  - > „Sparse-root“ ist durch Konfiguration von inherit-pkg-dir erzeugbar, aber noch nicht offiziell unterstützt
  - > Information über installierte Packages in inherit-pkg-dir in der Zone nicht verfügbar (pkg info <Package>)

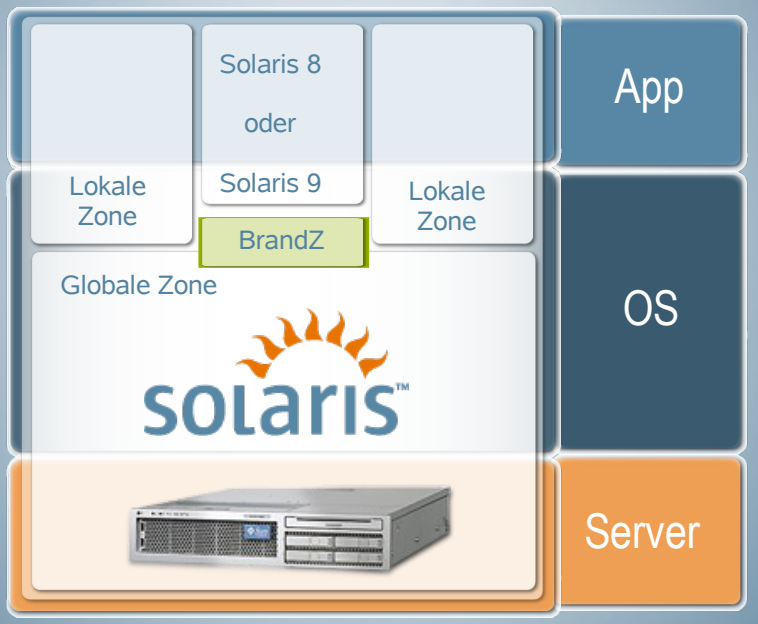
# Agenda

- Solaris Container in der Entwicklung
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Solaris 8/9 Container: Hilfe zur Migration

- Vorhandene alte Systeme laufen aus der Wartung
- Anwendung noch auf Solaris 8 oder Solaris 9
  - > Alte Anwendungen ...
  - > Umstellung auf Solaris 10 noch nicht erfolgt
  - > Zeitmangel, Aufwand zur Zeit nicht möglich, ISV nicht bereit, ...
- Neue Systeme (T.../M...) sollen eingesetzt werden

- Lösung mit Solaris 8/Solaris 9 Containern




- Solaris 8/9-Anwendungen laufen in Solaris Containern
- Nutzung “alter” Anwendungen auf neuen Systemen
- Volle Eigenschaften von Zonen
  - Performance
  - Ressource Management
  - Security
- Service Subscription erforderlich

# Solaris 8/9 Container

# Solaris 8/9 Container: Anwendung

- Archivieren Solaris 8 System (P2V Tool, flar, ...)
- Konfigurieren Solaris 8 Zone (zonecfg)
- Installieren des Archivs (zoneadm install ...)
- Ggf. Nacharbeiten
  - > Volume Manager Konfiguration entfernen
  - > /etc/system-Werte anpassen
  - > hostid anpassen (zonecfg Paramter)
- Fertig !

- Ggf. anschliessend Migration in eine native Zone

# Solaris 8/9 Container: Support

- Kostenpflichtig
  - Frei Downloadbar!
  - Lizensiert: Sockets des Zielsystems
    - 1 Jahr Subscription mit RTU und Support
    - Basis-System braucht Gold oder Platin
  - Nicht Bestandteil von Solaris 10
    => keine Verlängerung von Solaris 8 !!!
    - Vintage Support Phase 1 endet am 31. März 2009
      - Patches + Hotline
    - Vintage Support Phase 2 endet am 31. März 2012
      - Patches + Hotline
      - **Kostenpflichtige** neue Patches
      - Spezieller Vintage Support Vertrag möglich
- Anwendungs-Support:
  - ISV Support? In-House Applikationen!

# Agenda

- Solaris Container in der Entwicklung
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Ressource Management und Zonen

- Drei Arten des Ressourcen Managemenets
  - Garantierte Ressourcen: Garantierung des Mindesten
    - Ungenutzte Ressourcen können durch andere Zonen benutzt werden
      - Fair share scheduler: CPU Shares
  - Capped Ressourcen: Festlegung des oberen Limits
    - Ungenutzte Ressourcen können durch andere Zonen benutzt werden
      - CPU caps, Memory caps, swap caps
  - Zugewiesene Resource: Exclusive Zuteilung
    - Ressource wird nur durch die Zone genutzt
      - Dynamic Processor Pools, Zugewiesenes NIC

# CPU Caps
## (Solaris 10 5/08)

- Kappung des CPU-Verbrauchs in 1/100 Teilen
  - > Feiner Granulierbar als dedicated cpu-sets
  - > Limitierung, wieviel CPU eine Zone nutzen darf - auch wenn mehr vorhanden ist

```
zonecfg:keetonga> add capped-cpu
zonecfg:keetonga:capped-cpu> set set ncpus=0.12
zonecfg:keetonga:capped-cpu> end
```

# CPU Sichtbarkeit in Zonen

- Standard: alle Zonen sehen alle CPU
- Nutzung von Ressource Pools mit CPU-Zuweisung limitiert die Sichtbarkeit von CPU in Zonen
- Monitoring- und Status-Tools in Zonen zeigen nur die im Pool verfügbaren CPU
- Ermöglicht veränderte Lizensierung von Anwendungen

# Monitoring von Zonen

- prstat -Z Zusammenfassung per Zone
- mpstat -p, poolstat Prozessorpools und Auslastung
- rcapstat -z  Speicherverbrauch und Capping
- DTrace Toolkit
  - http://opensolaris.org/os/community/dtrace/dtracetoolkit/
  - zvmstat  per-zone VM Statistiken
- zonestat CPU und Speicherverbrauch von Zonen
  - http://opensolaris.org/os/project/zonestat

# Agenda

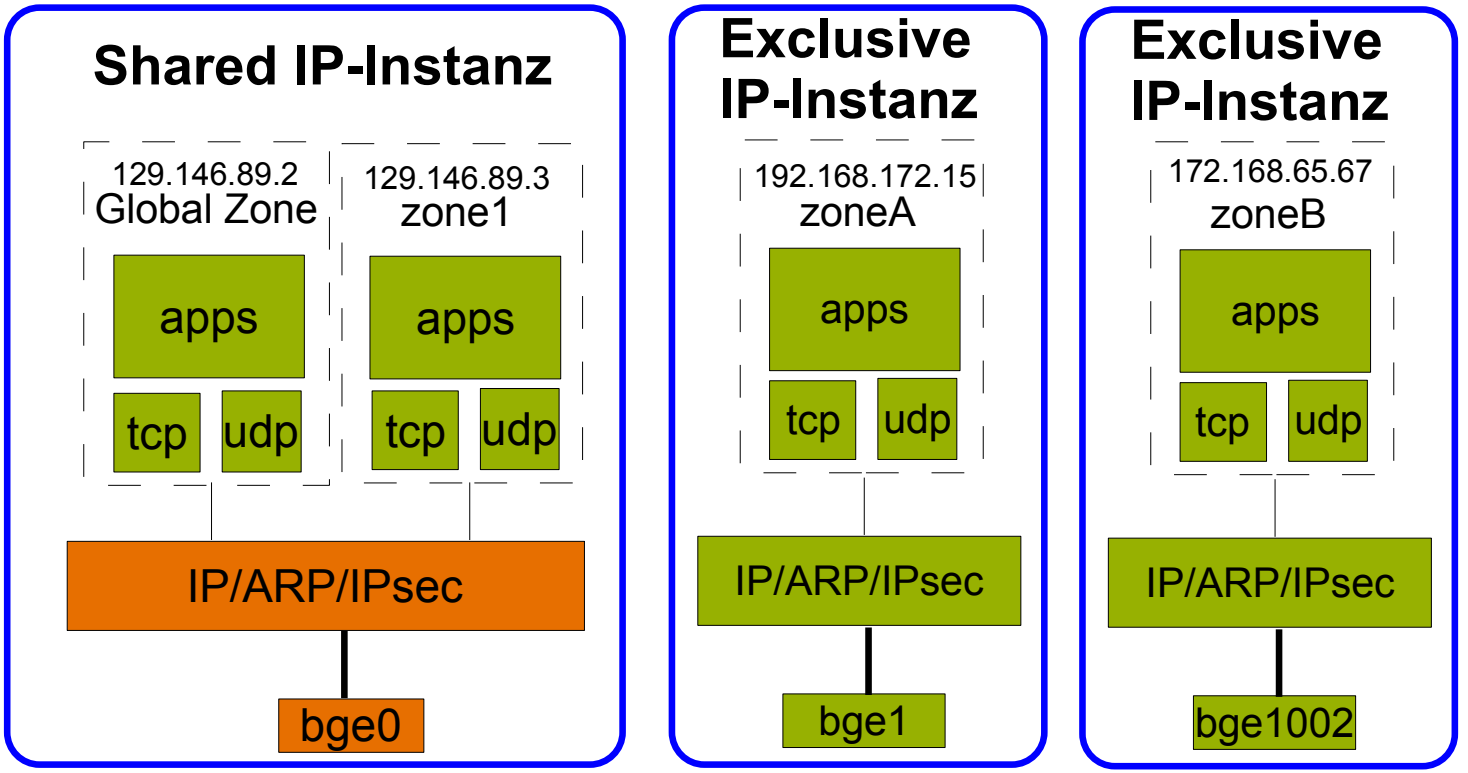
- SunCluster und Solaris Container
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Netzwerkkonzepte und Zonen

- Zonen benötigen in den meisten Fällen eine Netzwerkkonfiguration für ihre Aufgabe
- Zu beachten hier:
  - Trennungsvorschriften der Netzwerke zwischen Zonen
  - Kommunikationswege der Anwendungen in den Zonen
  - Erforderliche Anzahl von Netzwerken pro Zone und im Gesamtsystem
  - Anzahl verfügbarer Netzwerkkarten
  - Genau eine shared-IP Instanz
- Wieviele Adressen sind per Interface möglich ?
  - ndd -get /dev/ip ip_addrs_per_if

# Netzwerkkonzepte: IP-Instanzen

- Genau eine shared-IP Instanz
- Mehrere exclusive IP-Instanzen
  - > gldv3-Interfaces oder tagged-VLAN notwendig
    - > Siehe http://opensolaris.org/os/project/crossbow/faq/#ipinst_which_nic
  - > Voller Zugriff auf das physikalische Interface



# Zones Default Router (Shared-IP Zone)
## (Solaris 10 10/08)

- Festlegung des Default Routers einer Zone
  - > Sorgt für die Existenz der Route in der globalen zone
  - > Route wird nicht durch Shutdown der Zone gelöscht

```
zonecfg:keetonga> add net
zonecfg:keetonga:net> set defrouter=192.168.1.1
zonecfg:keetonga:net> end
```

# Ausblick: Projekt Crossbow
## (Nächste Version von OpenSolaris)

- VNICs - Virtuelle Netzwerkadapter
  - > Mit wenigen Adaptern viele Netzwerkports abbilden
- FlowControl
  - > Einfache Bandbreitenkontrolle für VNICs
- Etherstubs
  - > Der Software-Netzwerkswitch zum Bündeln von VNIC's

# Agenda

- Solaris Container in der Entwicklung
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Storagekonzepte für Zonen

- Die GZ kann kein NFS-Server für die NGZ sein
  - > loopback-mounts benutzen
- Die NGZ kann kein NFS-Server sein
  - > ggf. unfsd3 von sourceforge.net als userland NFS-Server nutzen
- Anwendungs-Daten
  - > Auf allem, was Daten speichert und wiedergibt
    - > UFS, ZFS, NFS, iSCSI, RAW
  - > Zugriff auf den Speicher durch die globale Zone
  - > Bereitstellung an die lokale Zone
- Zoneroot
  - > Ab Solaris 10 10/08: UFS oder ZFS supported

# LiveUpgrade von Zonen und ZFS
## (S10 10/08)

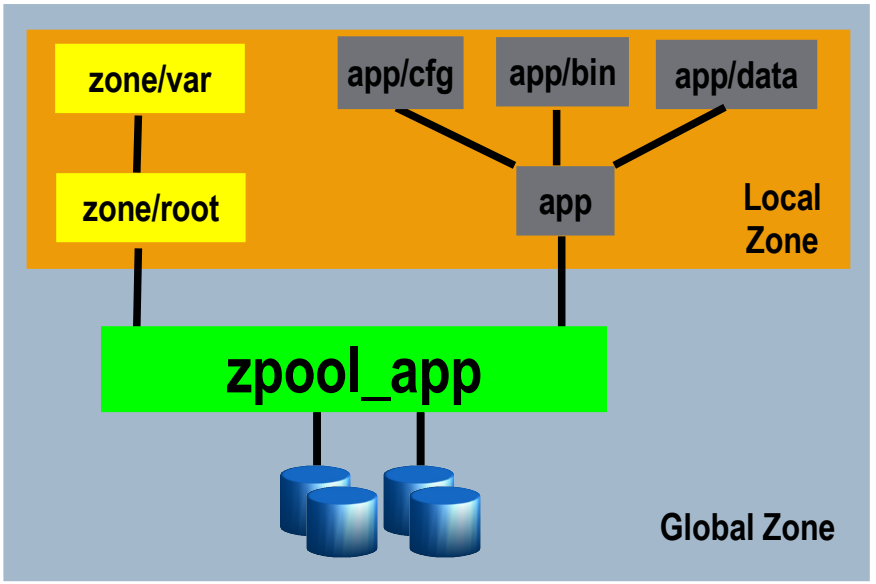
- lucreate -> luupgrade -> luactivate -> init 6
- Release Notes S10 10/08 beachten !
- Erzeugung neuer Boot Environements (BE) durch snapshot+clone im rpool
- zoneroot sollte im rpool liegen
  - Wenn nicht, wird durch lucreate zoneroot in rpool verlagert
    - dann ggf. nach luactivate und reboot mit zoneadm -z zone move <neues-zoneroot> wieder auf eigenen zpool bewegen
- Während lucreate keine Änderungen an Zustand der Zone vornehmen !

# $Z^2$ : Zonen und ZFS

- Ein großer zpool für Zonen und Daten ?
- Wie dann einzelne Zonen migrieren ?
- Also ?
  - > Je Anwendung ein zpool ?
  - > zpool export/import zur Zonenmigration ?
  - > Alternative: send/receive für kleine zpools ?
- Welche Möglichkeiten ergeben sich noch ?

# $Z^2$ : Der kombinierte zpool

- Anwendung und Zone teilen sich einen zpool (zpool_app)
  - > Upgrade/Live Upgrade kopiert hier auch die Applikationsdaten
  - > zpool, zone udn Anwendung sind bei einer Migration verbunden



# ZFS Dataset Delegation

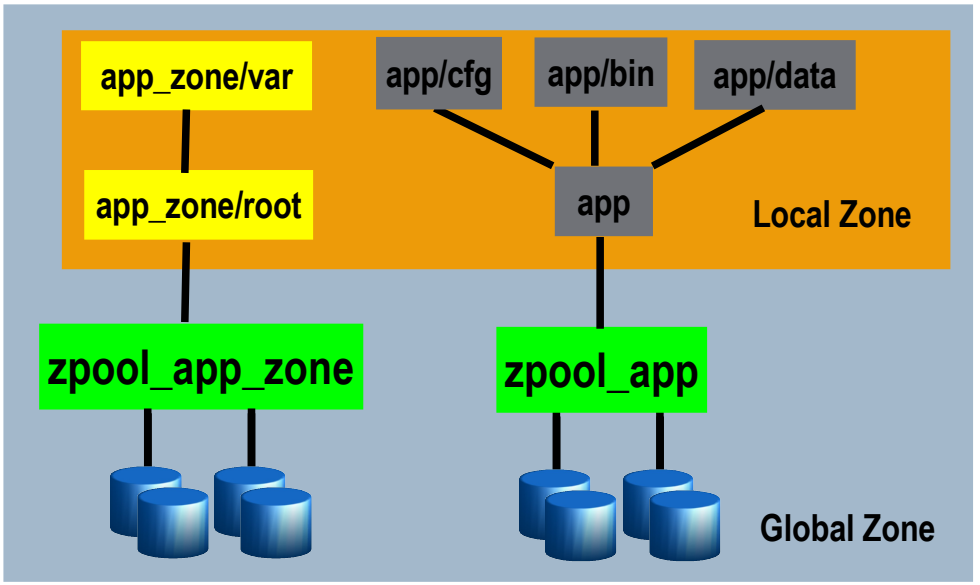
- ZFS dataset an lokale Zone delegieren
- Mount in GZ durch zoneadmd beim Boot
- ZFS Operationen in der Zone möglich
  - zfs erzeugen, snapshots, Properties ändern

```
zonecfg:keetonga> add dataset
zonecfg:keetonga:dataset> set name=zpool/keetonga
zonecfg:keetonga:dataset> end
```

- Wenn ZFS-Verzeichnisse per lofs lokale Zonen übergeben werden
  - Keine ZFS Operationen in der lokalen zone möglich

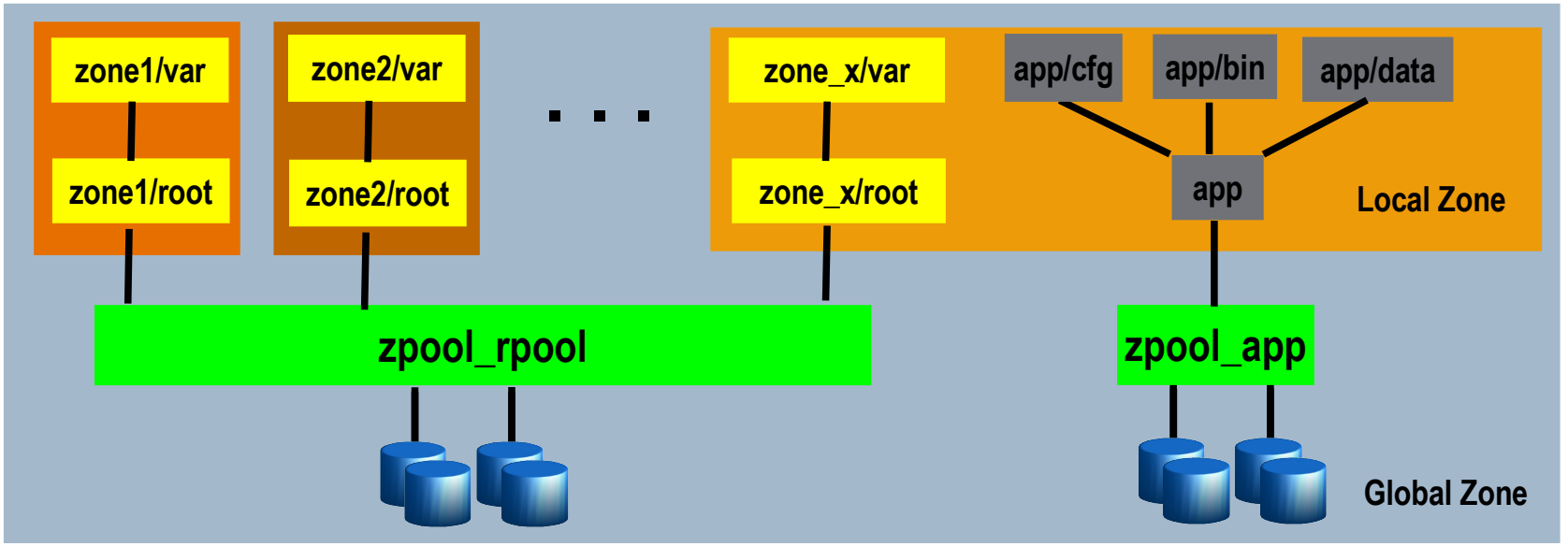
# $Z^2$ :Der unabhängige Service-zpool

- Anwendung und Zone haben je einen eigenen zpool
  - > Zone ist Laufzeitumgebung der Anwendung
  - > Jede Anwendung hat ihren eigenen zpool
- Unabhängige Migration und Upgrade wird möglich



# $Z^2$ : Der zones-zpool (rpool)

- Jede Anwendung hat einen eigenen zpool
- Die Zonen teilen sich den zpool für root und var
  - > Kann der rpool der GZ sein (für LiveUpgrade erforderlich)
- Nutzung von zfs quotas zur Limitierung des Platzverbrauches der Zonen

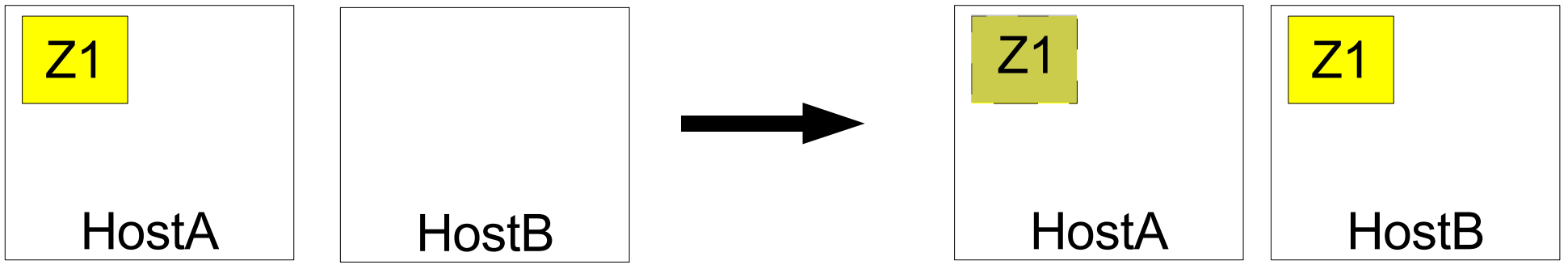


# Agenda

- Solaris Container in der Entwicklung
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Migration von Zonen

- Bewegen einer Zone von HostA zu HostB
- Zonen müssen angehalten sein
- Keine Live Migration
- Nutzbar für Systemupgrade und Patching



# Migration einer Zone: detach und attach

- 1. zoneadm -z zone detach
  - „Abhängen“ von der GZ
    - Zonenstatus wechselt von „installed“ in „configured“
- 2. Zonepath kann als Verzeichnis bewegt werden
  - Einfach per SAN oder iSCSI machbar
  - Sonst: zfs send/receive, ufsdump/ufsrestore, etc.
- 3. Eine neue Zone erzeugen
  - zonecfg -z <zone> create -a /zonepath
    - Alte Zonenkonfiguration bestimmt neue Zone
- 4. zoneadm -z zone attach
  - Zone und Host werden auf Kompatibilität geprüft (Patches und Packages)
    - Zonestatus wechselt von „configured“ in „installed“

# Migration von Zonen: Update-on-Attach

- zoneadm -z <zone> attach -u
  - Update einer Zone bei attach
  - Nur Hinzufügen von Patches und Packages
  - Bzw. Versionsnummern der Packages und Patches
    - GZ >= NGZ
- Zur Migration zwischen sun4u und sun4v
- Für „aufwärtiges“ Patchen
- Leider noch nicht für alle Patches geeignet (z.B. IDR's)
  - Siehe Patch Readmes und Release Notes
  - ggf. Nachtrag in /usr/lib/brand/native/bad_patches
- Bei Problemen mit attach -u
  - Ggf. SUNWdetached.xml löschen und attachen

# Agenda

- SunCluster und Solaris Container
- Solaris 8/9-Container
- Ressource Management und Monitoring von Zonen
- Netzwerke und Zonen
- Storage und Zonen
- Migration von Zonen
- Patch und Upgrade von Zonen

# Zones und Patching

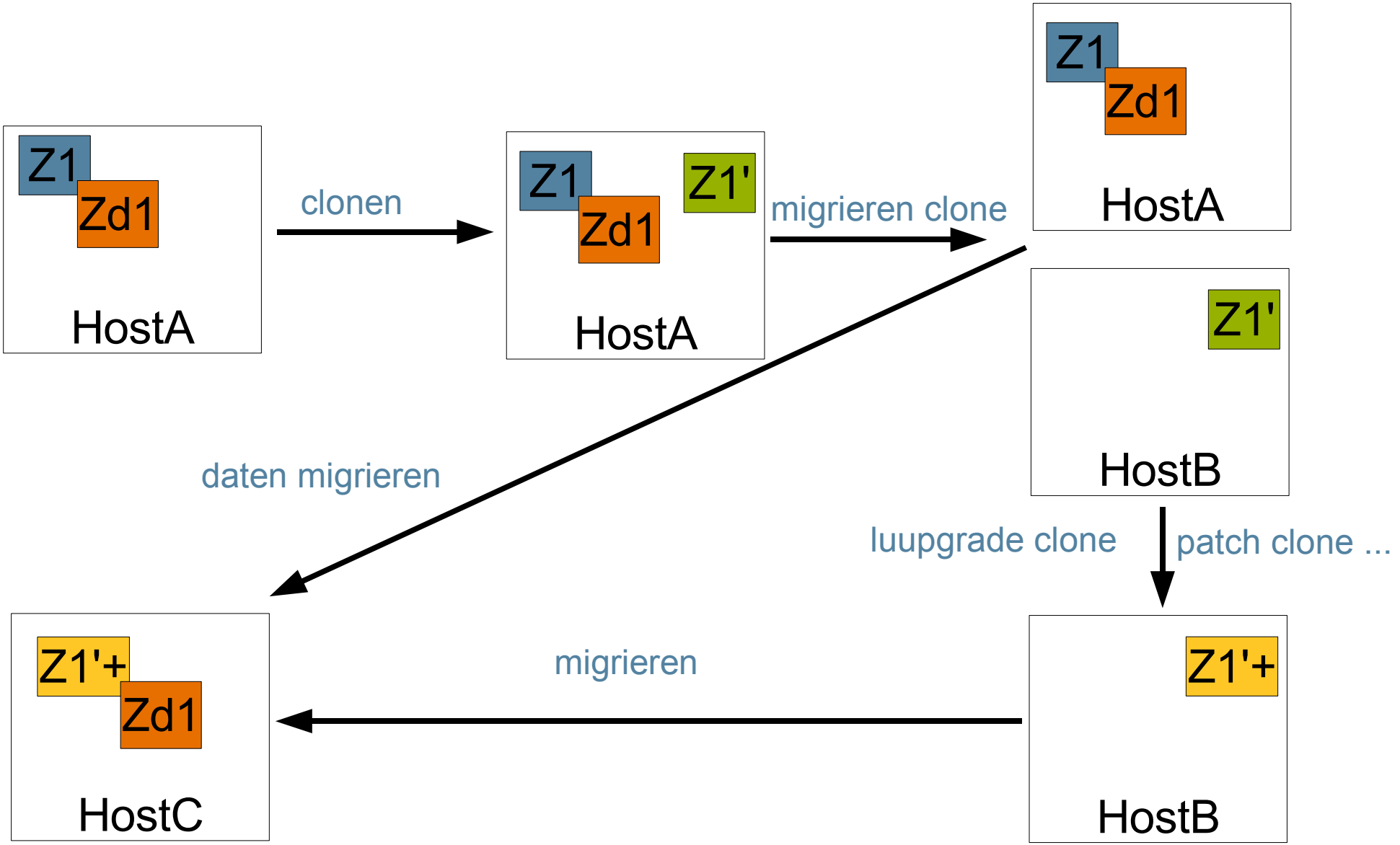
- Lokale Zonen werden über die globale Zone gepatched
- Der Zeitaufwand zum Patchen bei vielen Zonen kann erheblich sein.
- Paralleles Patchen von Zonen teilweise im Einsatz aber instabil
  - > Kann zu inkonsistenten Patchständen zwischen GZ und NGZ führen

# Zones und Patching: Varianten

- Neuinstallation anstelle von Patchen
  - Zonen automatisiert erzeugen (incl. lokaler Anpassungen)
  - Anwendungsdaten aus separaten Volumes dazumounten
  - Patchen durch:
    - Zonen löschen
    - Globale Zone patchen
    - Zonen neu erzeugen
    - “Alten” Anwendungsdaten hinzufügen
- LiveUpgrade nutzen
  - Alle Zonen gleichzeitig patchen

    oder
  - Zonen separat clonen, migrieren und auf separatem System patchen/upgraden

# Patchen durch LiveUpgrade




Zone (Z1), Kopie von Z1, Kopie von Z1 mit neuem Patchstand, Daten und Anwendung (Zd1)

# Zusammenfassung

- Das Ressource Management schafft flexible Einsatzmöglichkeiten
- Solaris Container mit ZFS erlauben die Gestaltung neuer Konzepte
- Netzwerkkonzepte sind ein wichtiger Bestandteil bei der Planung von Solaris Containern
- Patch- und Upgrademechanismen von Solaris Containern sollten im Vorfeld gewissenhaft geplant werden
- Solaris Container stellen nach wie vor die effizienteste Art der Virtualisierung dar