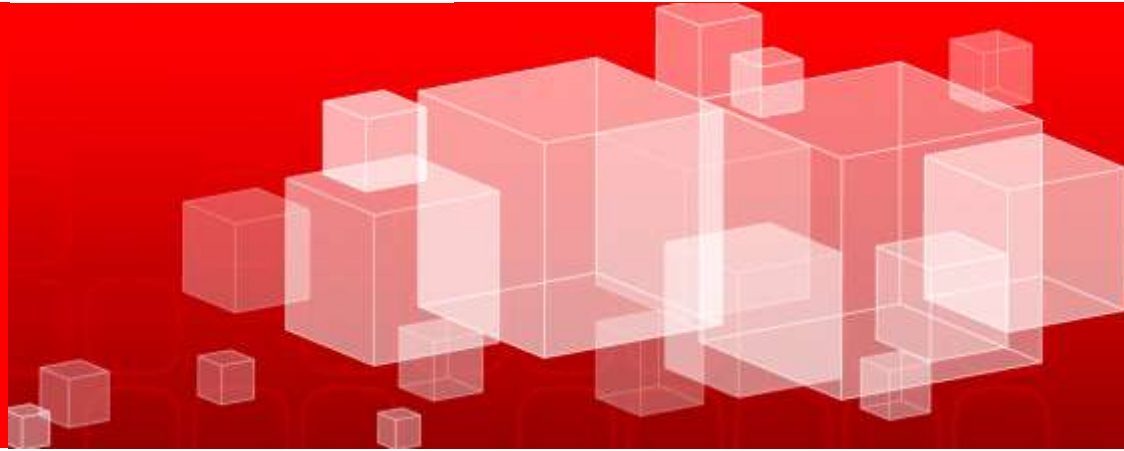


The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®

Oracle Solaris 11 Repositories Best Practices

Detlef Drewanz

Principal Sales Consultant, Server Architects Northern Europe

Agenda

- Solaris 11 IPS
- Solaris 11 Repositories
- Erstellung und Konfiguration von Repositories
- Deployment Strategien von Repositories

Oracle Solaris 11 Lifecycle Management

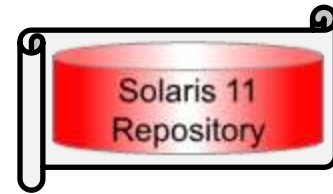
Kurzüberblick

- Kleinere Images für die OS-Installation
- Überarbeiteter OS-Installer
- ZFS als root-Filesystem
- Bootenvironments
- Neu: Image Packaging System (IPS)
- Neu: IPS-Repositories

Das Image Packaging System (IPS)

- Netzwerk-orientiertes Paketmanagement
 - Installation, Upgrade, Entfernen von Paketen
 - Pakete sind Sammlungen von Aktionen und Payload
- Abhängigkeiten zwischen Paketen
- Versionierung von Paketen
 - z.B. pkg://solaris/entire@0.5.11,5.11-0.175.1.0.0.24.2:20120919T19013
 - Package version numbering explained [MOS-ID 1378134.1]
- Paket-Updates anstelle von Patches

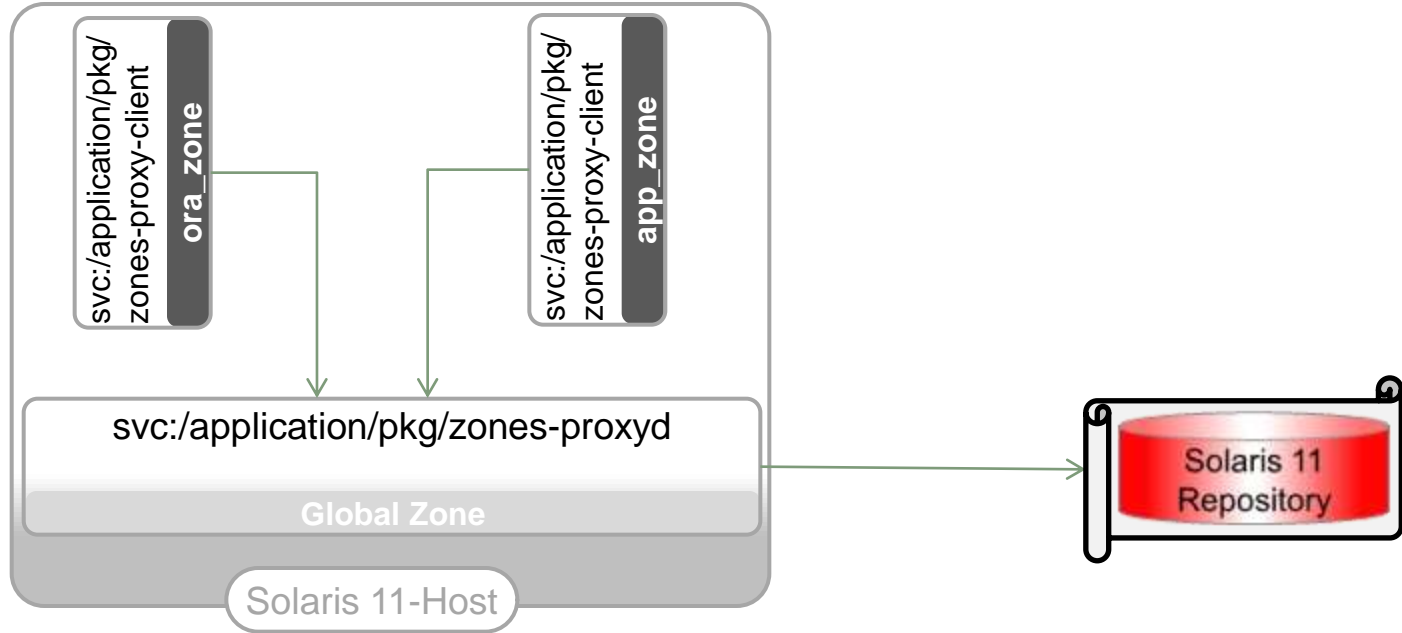
IPS-Repositories



- Verteilung von IPS-Pakete durch Publisher in Repositories oder Package-Archiven
 - Manifeste und Package Content (kombiniert x86 und SPARC)
- 2 Typen von Repositories
 - Origin: Enthält Metadaten und Package Content
 - Mirror: Enthält nur Package Content
- Zugriff: <http://...>, <https://...>, <file://...> (oder <nfs://...>)
- Das Repository für Oracle Solaris 11 (Baseline)
 - <http://pkg.oracle.com/solaris/release/en/index.shtml>

Integration von Zonen in IPS

IPS Proxy Service und System-Repository für Zonen



Repositories via Proxies

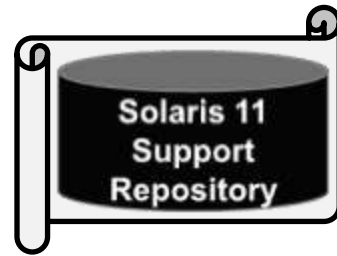
- Umgebungsvariable in der Shell setzen

```
# export http_proxy = http://<proxy>:<port>  
# export https_proxy = https://<proxy>:<port>
```

- Service Property für System-Repository setzen

```
# svccfg -s pkg/system-repository setprop config/http_proxy = \  
                                                                    astring:"http://<proxy>:<port>"  
# svcadm refresh pkg/system-repository
```

Das Solaris Support Repository



- Monatliches Update
 - Für Kunden mit Solaris Wartungsvertrag
- <https://pkg.oracle.com/solaris/support/>
 - Registrierung bei <https://pkg-register.oracle.com/>
 - Support Repositories Explained [ID 1021281.1]
- Support Repository Updates (SRU)
 - Download Offline von <http://support.oracle.com>
 - Oracle Solaris 11 SRU Index [ID 1372094.1]

Support Repository Updates (SRU)



- Monatliche Updates zum Release Repository
- Enthalten die neueste Version eines Paketes seit Release

Packagename	Erste Version	Version mit SRU 1	Version mit SRU 2	Version mit SRU 3	Version mit SRU 4
pkgA	1.0	1.1 (update)	1.1 (halten)	1.2 (update)	1.3 (update)
pkgB	2.7	-	2.7.1 (update)	2.7.2 (update)	2.7.2 (halten)
entire	1.0	1.0.1 (update)	1.0.2 (update)	1.0.3 (update)	1.0.4 (update)

Anlegen eines Solaris 11 IPS-Repositories

- Erstellung aus Baseline .iso
 - Oder per pkgrecv aus existierendem Baseline Repository
- Kompression für Metadaten der Pakete
 - Der Paket-Content liegt bereits komprimiert im Repository

```
# zpool create -O compression=on -O atime=off repo c3t2d0
# zfs create repo/s11.0
... mount Repository ISO Image ...
# rsync -aPq /media/SOL11REPO_FULL/repo /repo/s11.1
# pkgrepo -s /repo/s11.1 refresh
# pkg set-publisher -G 'http://pkg.oracle.com/solaris/release/' -g /repo/s11.0 solaris
```

Attribute eines Solaris 11 IPS-Repositories

- Abfrage der Attribute von Repositories

```
# pkgrepo get -s /repo/s11.1
SECTION  PROPERTY  VALUE
publisher      prefix    solaris
repository description This repository serves a copy of the Oracle Solaris 11.1 Build
24b Package Repository.
repository name  Oracle Solaris 11.1 Build 24b Package Repository
repository version 4
```

Konfiguration und Start des Depot Service

```
# svccfg -s pkg/server setprop pkg/inst_root=/repo/s11.0
# svccfg -s pkg/server setprop pkg/readonly=true
# svcadm refresh pkg/server
# svcadm enable pkg/server

# pkg set-publisher -G /repo/s11.0/ -g http://localhost solaris

# pkgrepo info -s http://localhost
PUBLISHER PACKAGES STATUS          UPDATED
solaris   4307      online      2012-10-14T21:58:19.920392Z
```

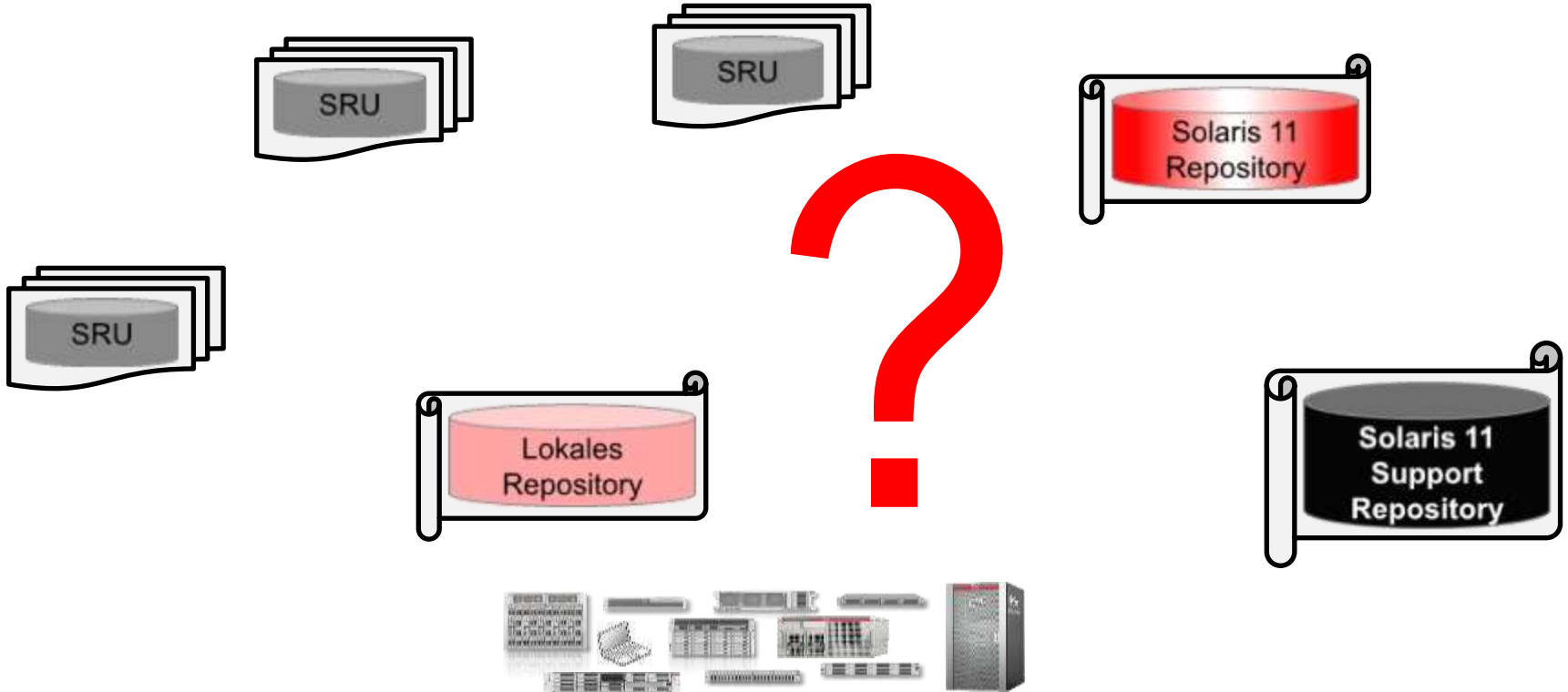
Integration eines SRU in ein Repository

- Zeitaufwand abhängig von der Größe des SRU ISO und der bereits eingespielten SRU
- Index-Rebuild nach SRU-Einspielen wichtig, damit die Suchfunktion funktioniert

... Download und mounten des Repository ISO Image ...

```
# pkgrecv -s /media/SOL11_1111_REPO_SRU4_INCR/repo -d /repo/s11.0 '*'  
# pkgrepo rebuild -s /repo/s11.0  
# svcadm restart pkg/server
```

Repositories und Updates



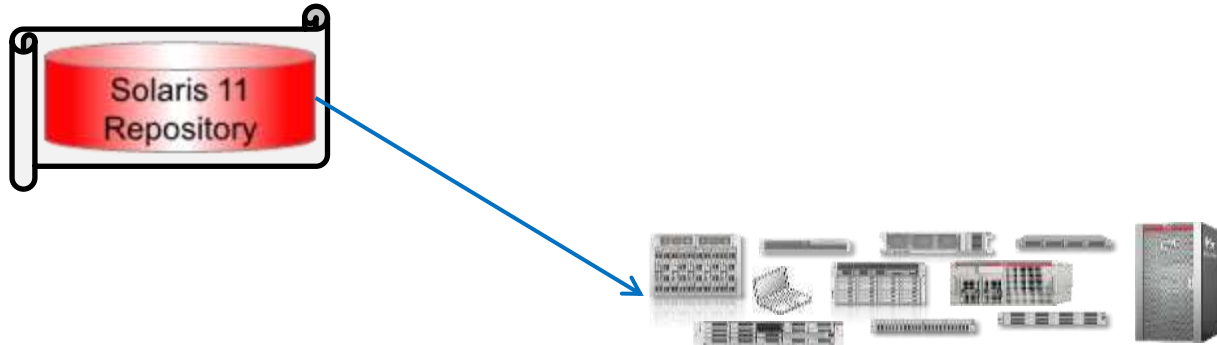
Agenda

- Solaris 11 IPS
- Solaris 11 Repositories
- Erstellung und Konfiguration von Repositories
- **Deployment Strategien von Repositories**

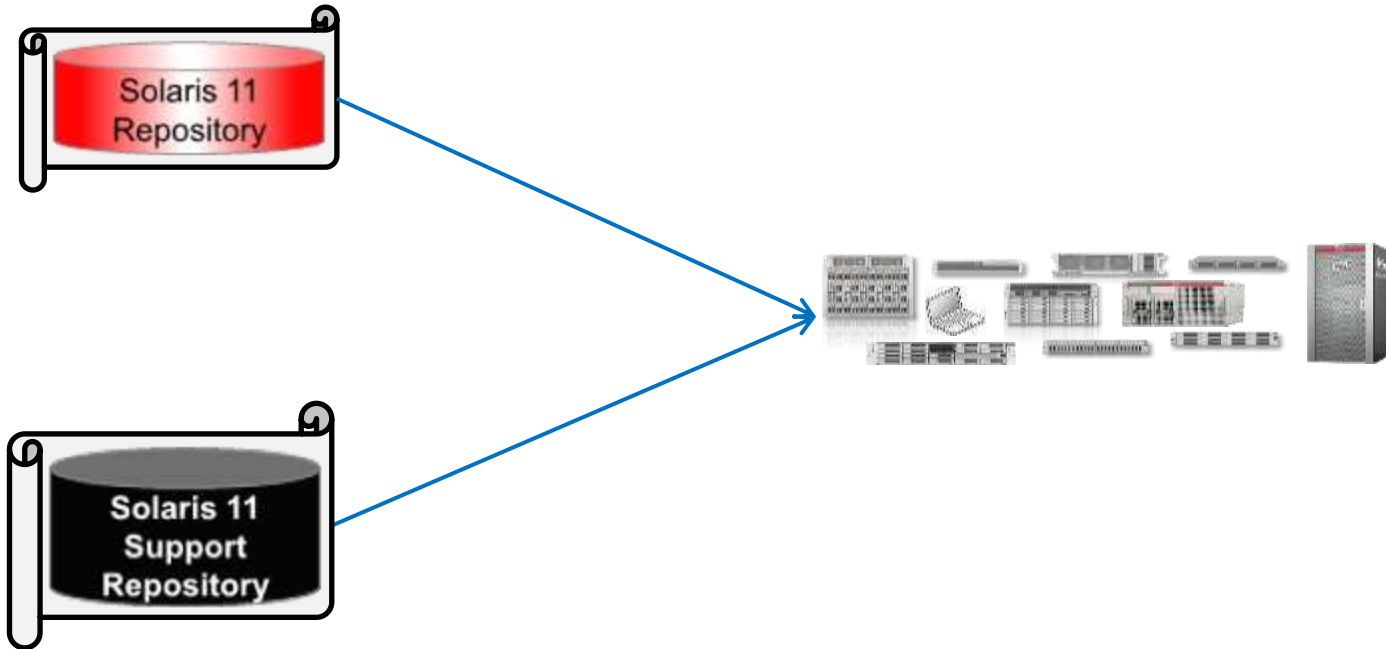
Deployment von Repositories

- Zentrales Repository oder lokale Repository-Kopie ?
- Je Release ein eigenes Repository oder Eines für alle ?
- Ein oder mehrere lokale Repositories ?
- Synchronisation mehrerer lokaler Repositories ?
- Zugriff lokal oder via NFS, HTTP oder HTTPS ?
- Integration von SRU's ?

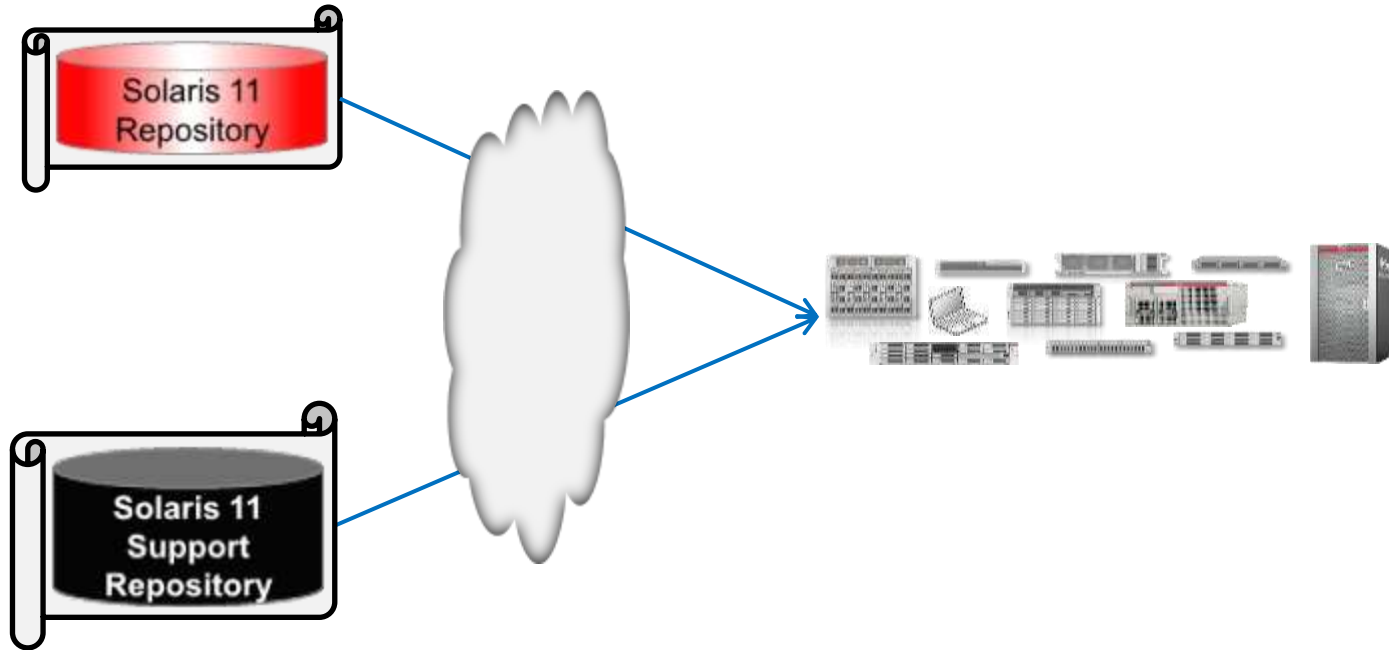
Zentrales Repository oder lokale Kopie ?



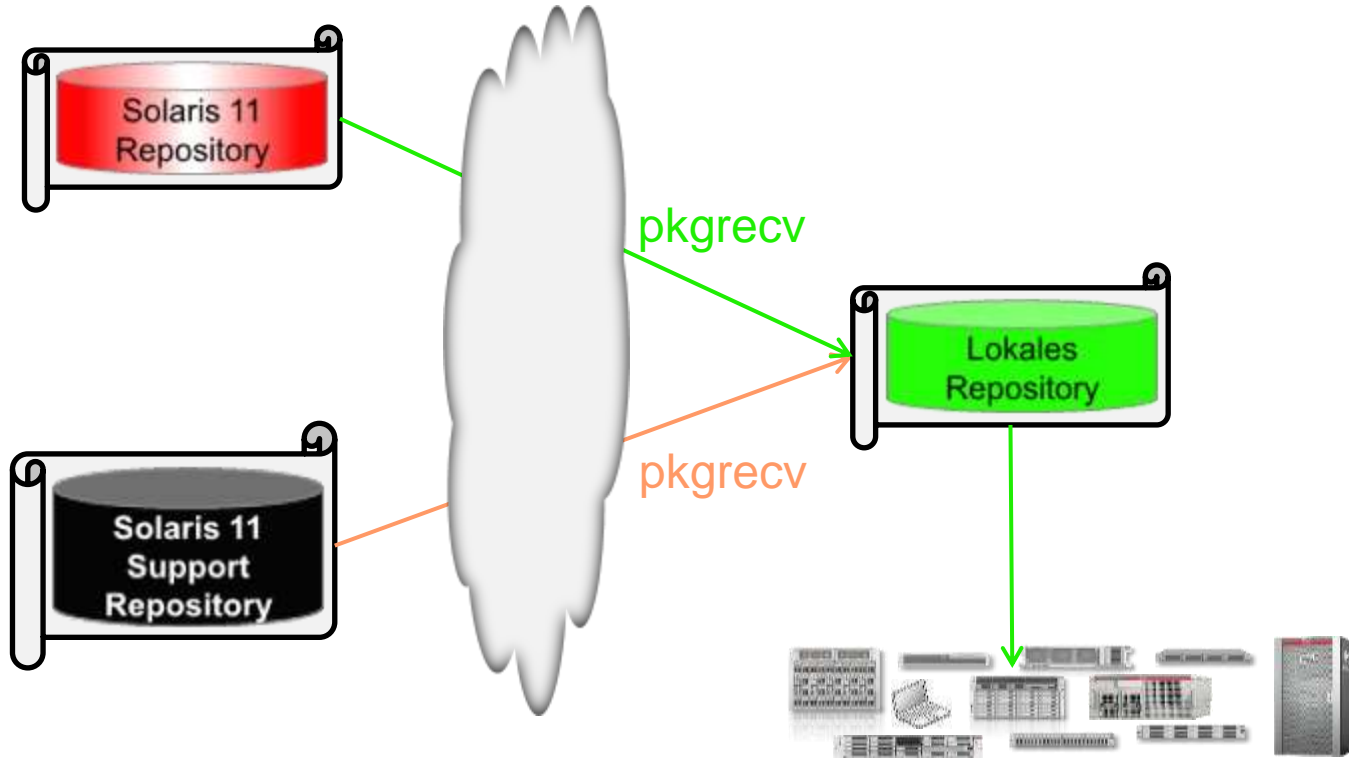
Zentrales Repository oder lokale Kopie ?



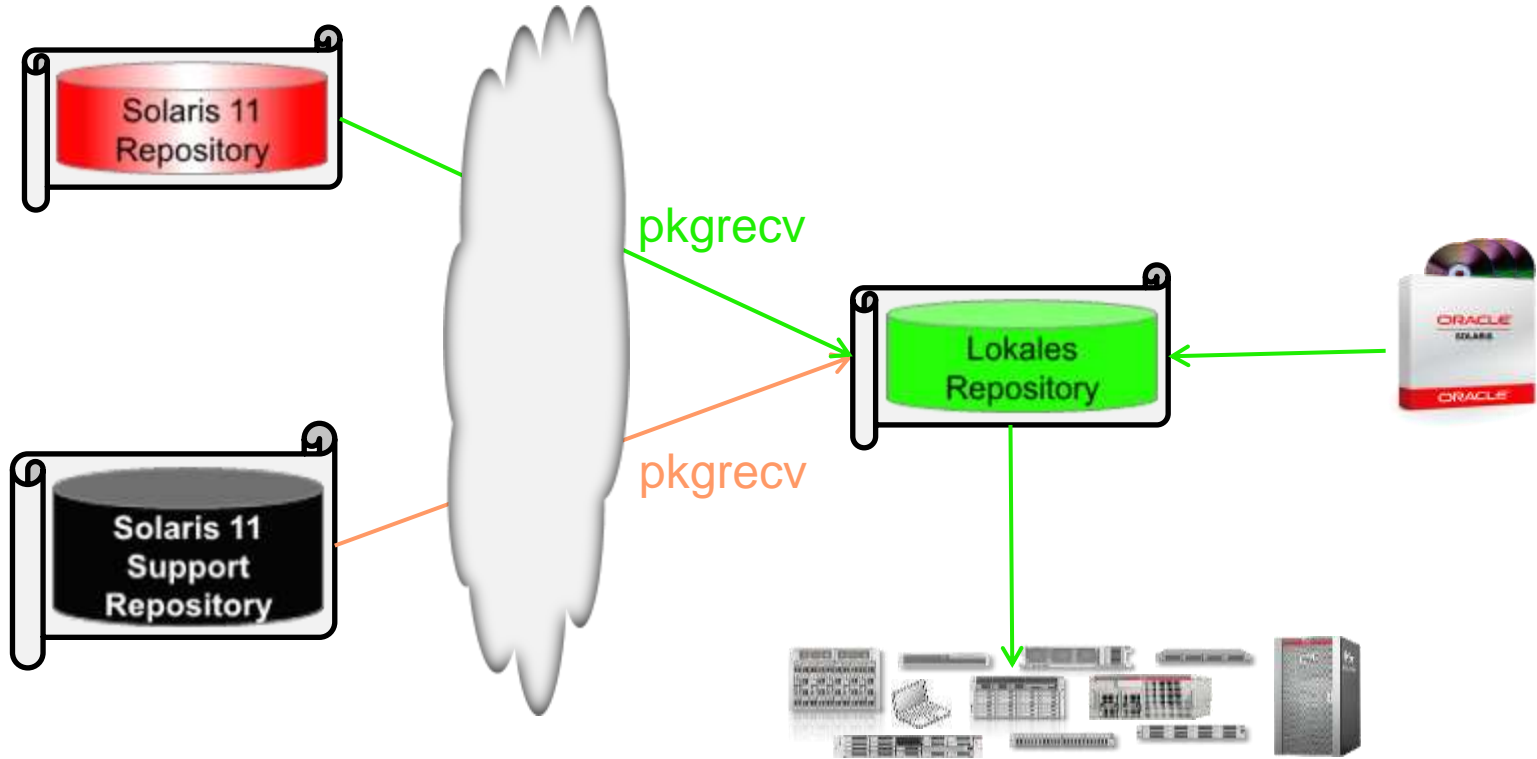
Zentrales Repository oder lokale Kopie ?



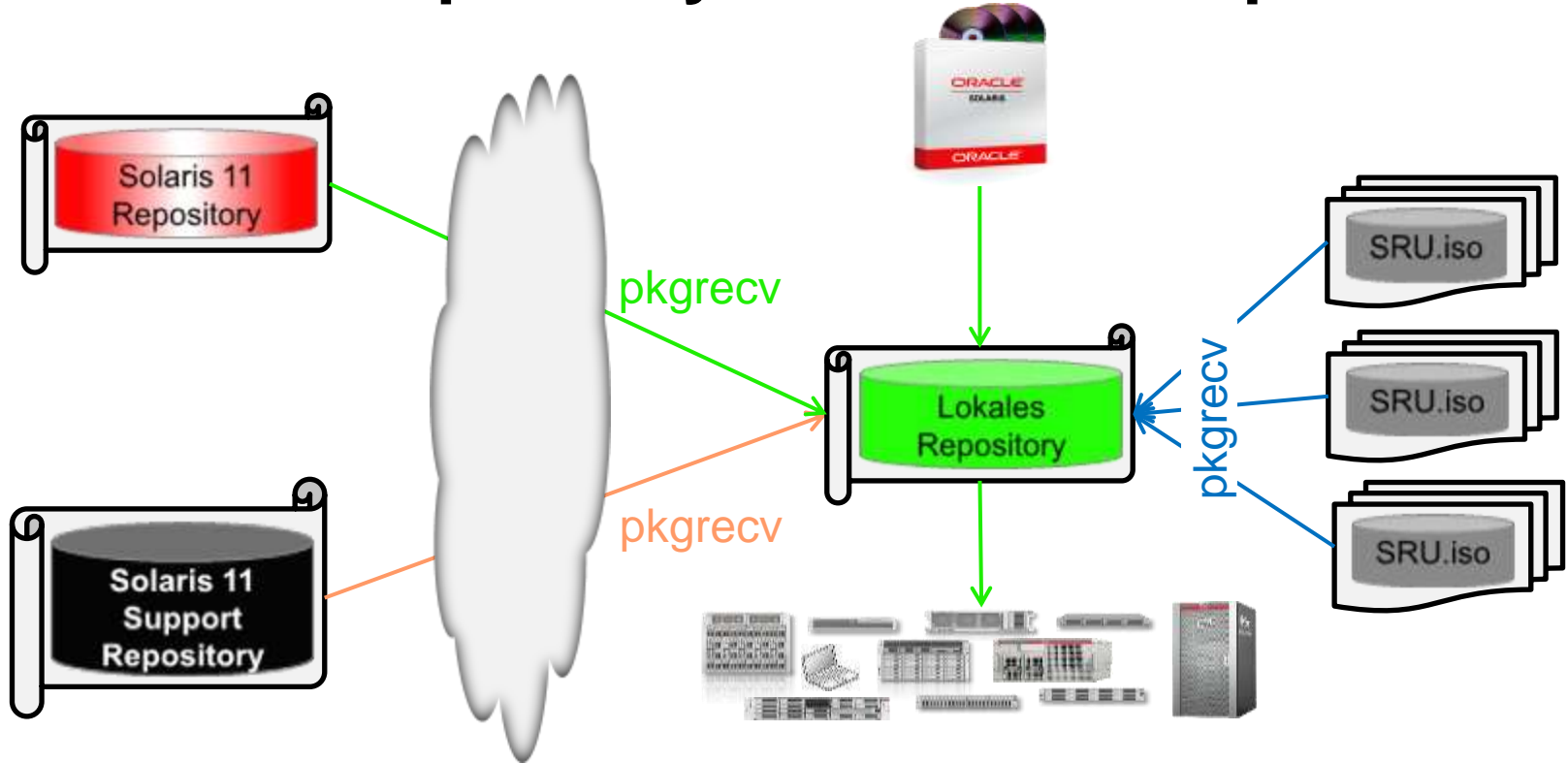
Zentrales Repository oder lokale Kopie ?



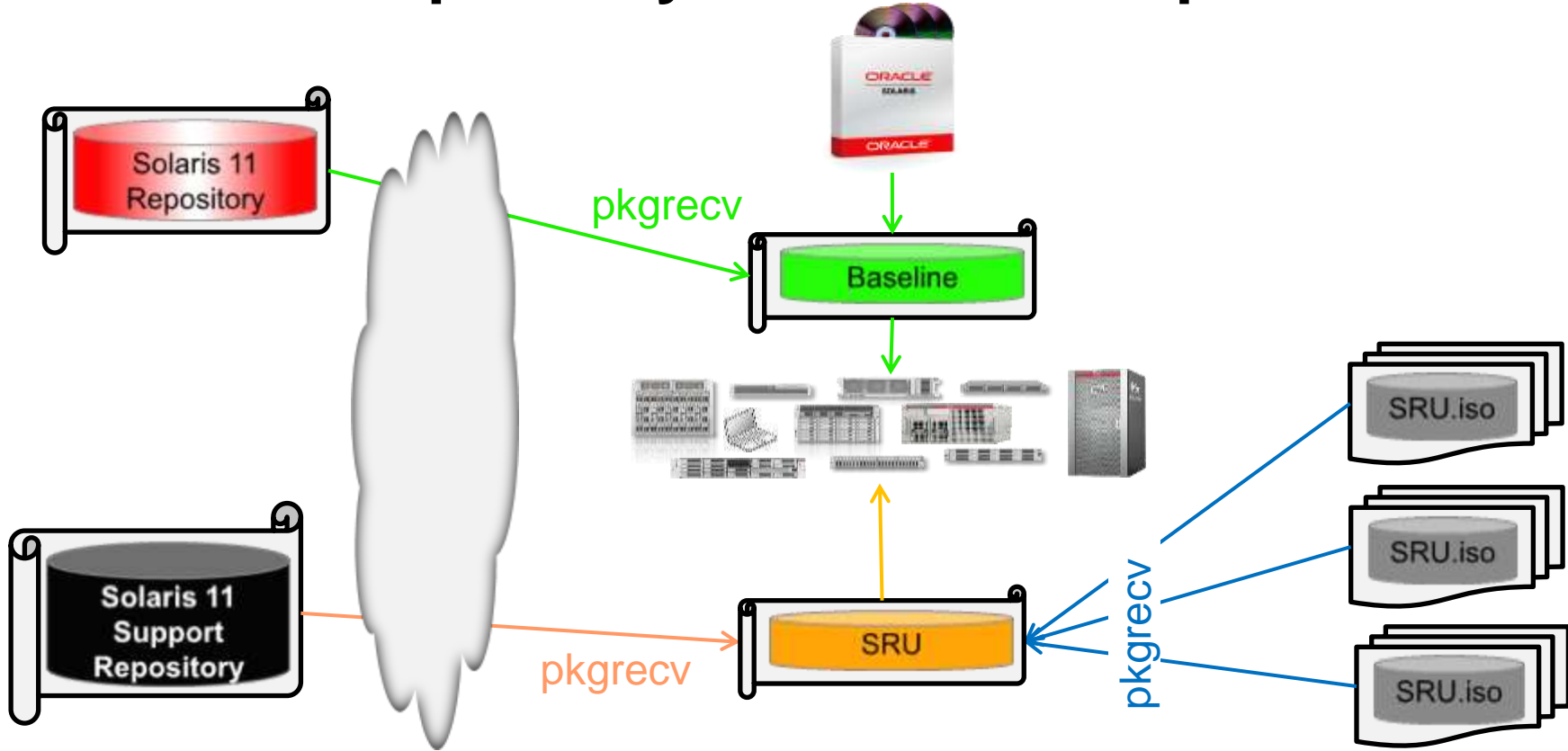
Zentrales Repository oder lokale Kopie ?



Zentrales Repository oder lokale Kopie ?



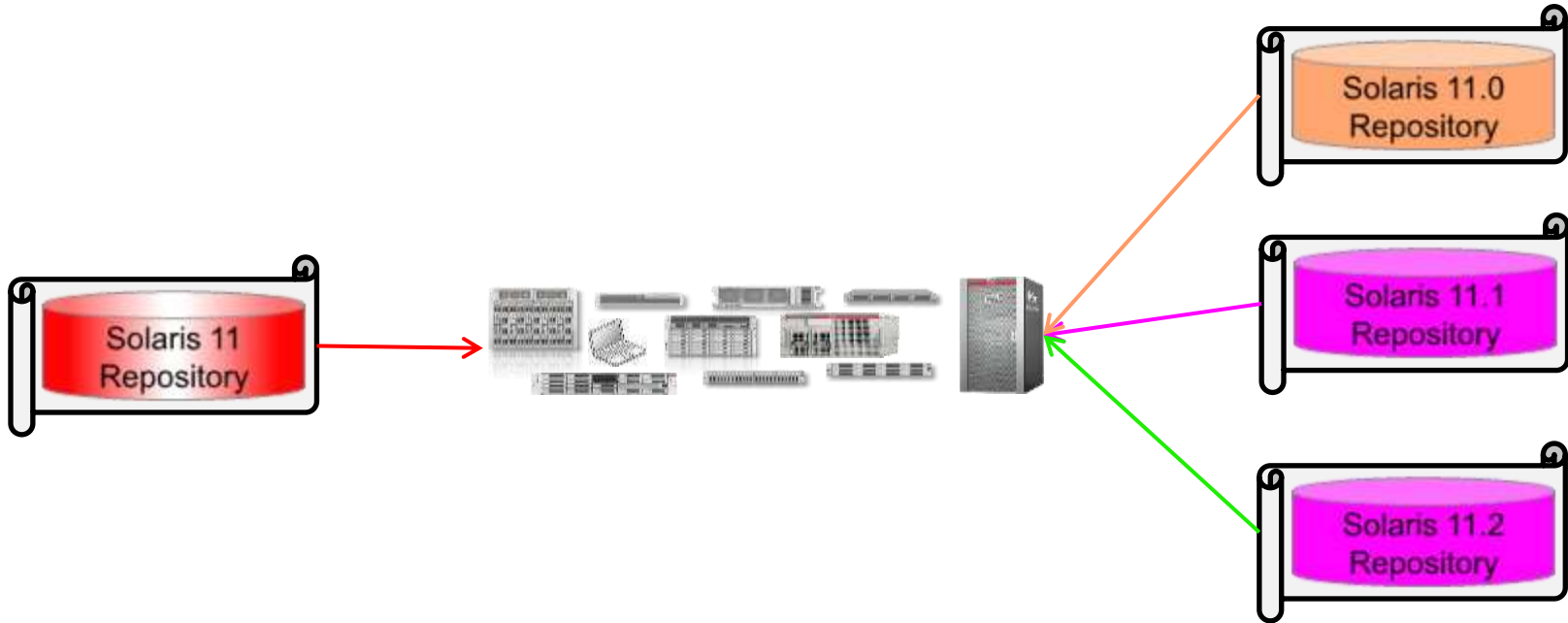
Zentrales Repository oder lokale Kopie ?



Zentrales Repository oder lokale Kopie ?

- Zentrales Repository
 - Kein Aufwand für Synchronisation
 - Verbindung zum Internet direkt oder via Proxy notwendig
- Lokale Repository-Kopie
 - Im lokalen Netzwerk verfügbar
 - Unabhängig von externem Netzwerk
 - Performance
 - Security
 - Synchronisation organisieren

Je Release ein eigenes Repository oder ein gemeinsames Repository ?



Je Release ein eigenes Repository oder ein gemeinsames Repository ?

- Strukturierung der Releases
- Aufwand zur Integration der Releases
- Pflege unterschiedlicher Publisher Einträge
- Nachtrag neuer Repositories bei neuen Releases
- Löschung alter Release-Stände
 - ZFS Snapshots nutzen

Ein oder mehrere lokale Repositories ?

- Aufwand zur Pflege mehrerer Repositories
- Synchronisation verteilter Repositories untereinander
- Skalierbarkeit eines einzelnen Repositories
- Netzwerkperformance zu einem zentralen oder zu verteilten Repositories

Synchronisation lokaler Repositories

- rsync
 - Schnelle Synchronisation auf Dateiebene
- zfs send/receive
 - Synchronisation auf ZFS Ebene
- pkgrecv
 - Synchronisation auf Repository-/Paket-Ebene
 - Empfohlener Prozess
 - "the right tool for the right job"

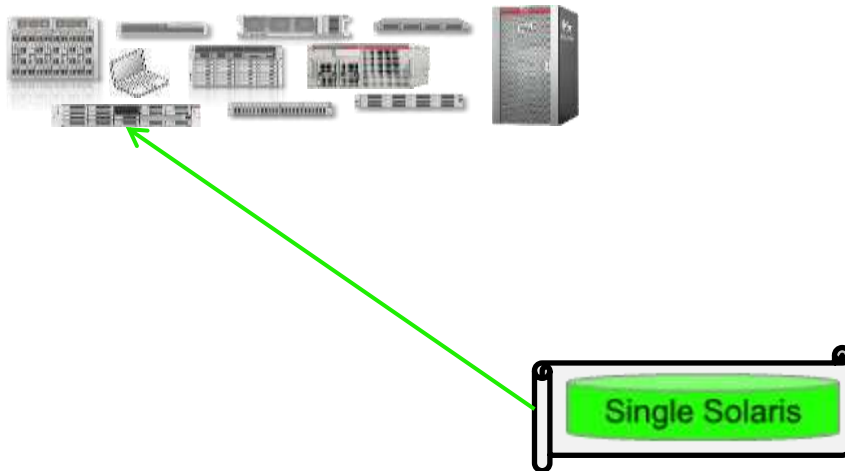
Lokales, NFS, HTTP oder HTTPS - Repository

- Lokal
 - Exklusiver, netzunabhängiger Zugriff
- NFS
 - NFS-Server hält das Repository, Zugang via NFS-Filesystem
- HTTP
 - Zentraler Zugang, Protokoll via Proxy und Cache möglich
- HTTPS
 - Wie HTTP, aber sicherer wenn erforderlich

Integration von SRU in lokale Repositories ?

- 4 Formen der Integration von SRU und Release Repository
 - Single Repo
 - Separated Repo
 - SRU Repo
 - Combined Repo

Single Repository

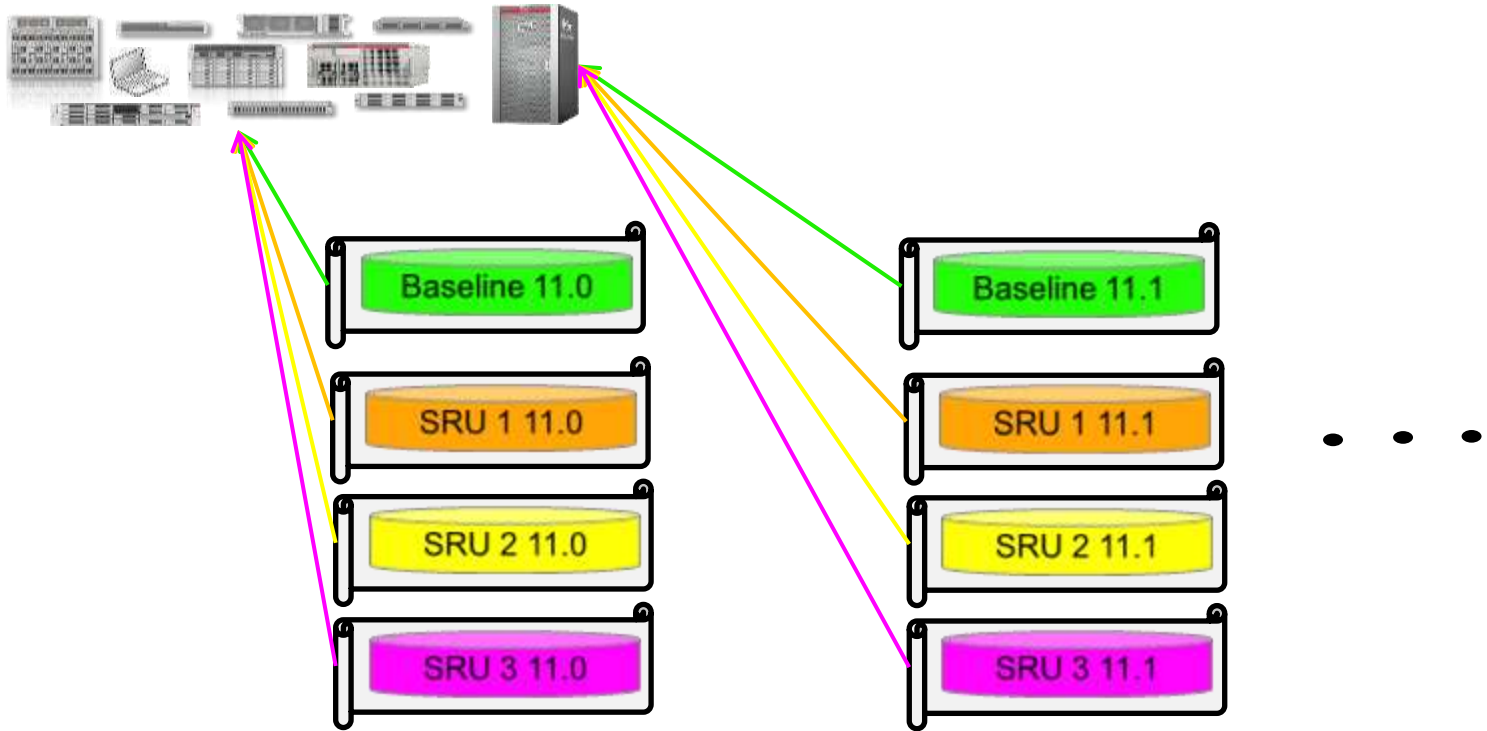


Single Repo

Alle SRU und alle Releases in einem Repository

- Nur 1 Repository-Eintrag für alle Updates zu verwalten
- Repository wächst mit jedem SRU und jedem Release
- "Aufräumen" der Repositories für alte Releases schwierig
- Keine Änderung des Publishers bei Upgrade notwendig
- **Gut einsetzbar, Anwendungsfall des Support Repositories**

Separated Repositories

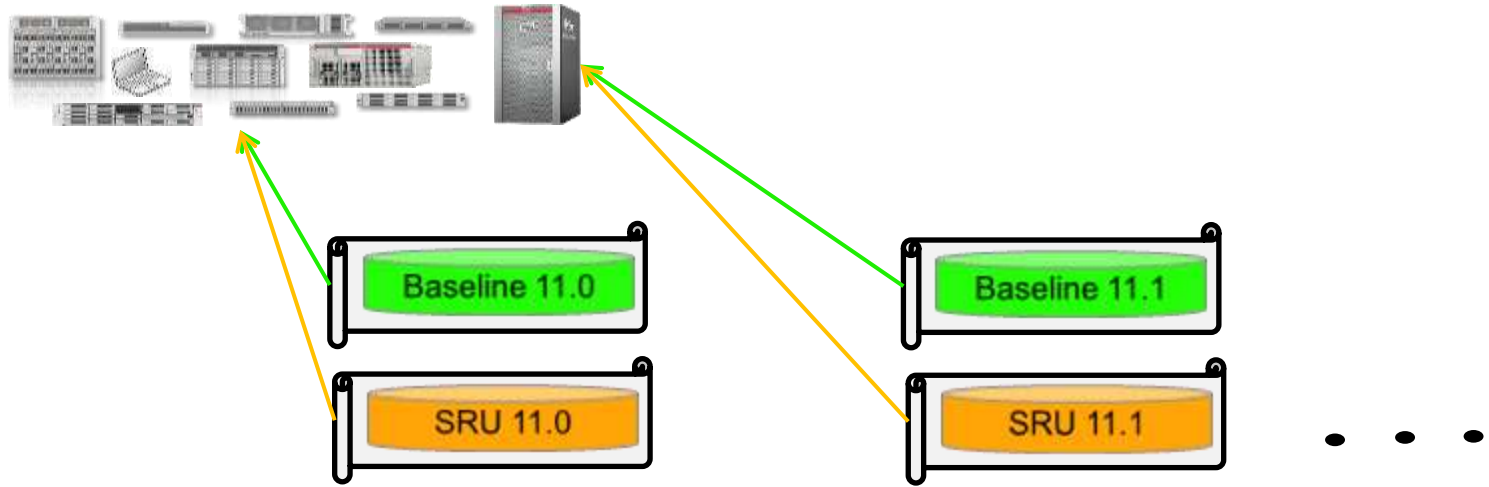


Separated Repo

Jedes SRU und jedes Release ein Repository

- Schnelles Auspacken und Erstellen der Repos
- Unabhängige Repos
- Erheblicher Pflegeaufwand der Publisher Einträge an jedem Client
- Nachtrag neuer Repositories bei bereits existierenden Clients
- Mehr Plattenplatz erforderlich
- **Nicht praxistauglich wegen Administrationsaufwand**

SRU Repositories

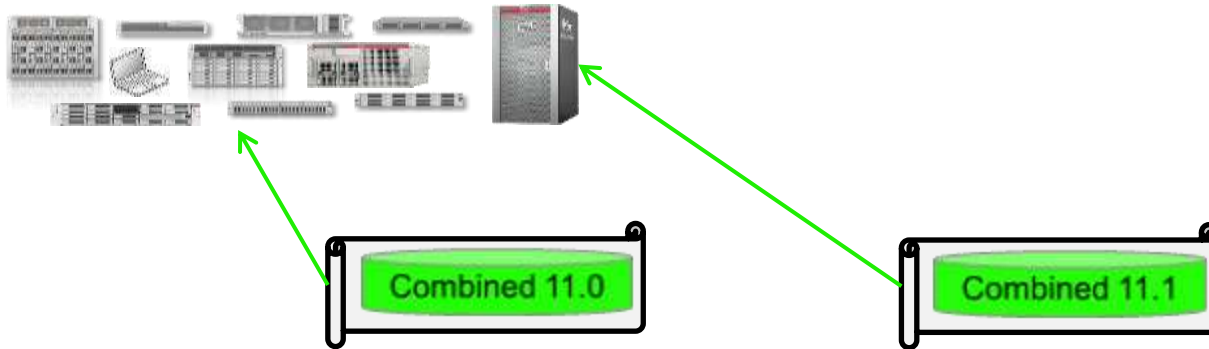


SRU Repo

Je ein Release- und ein SRU-Repository

- Schnelles Auspacken und Erstellen des Release-Repas
- Unabhängiges SRU-Repo
- Je Release 2 Repository-Einträge zu verwalten
- **Einsatzfall denkbar, aber ohne Mehrwert gegenüber Combined Repo**

Combined Repositories



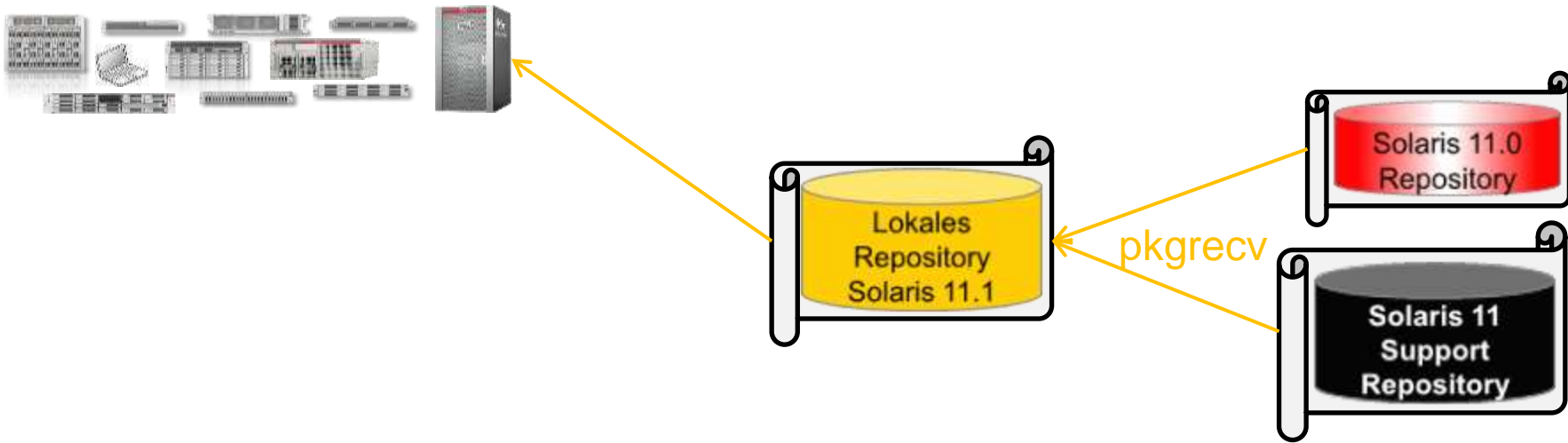
...

Combined Repo

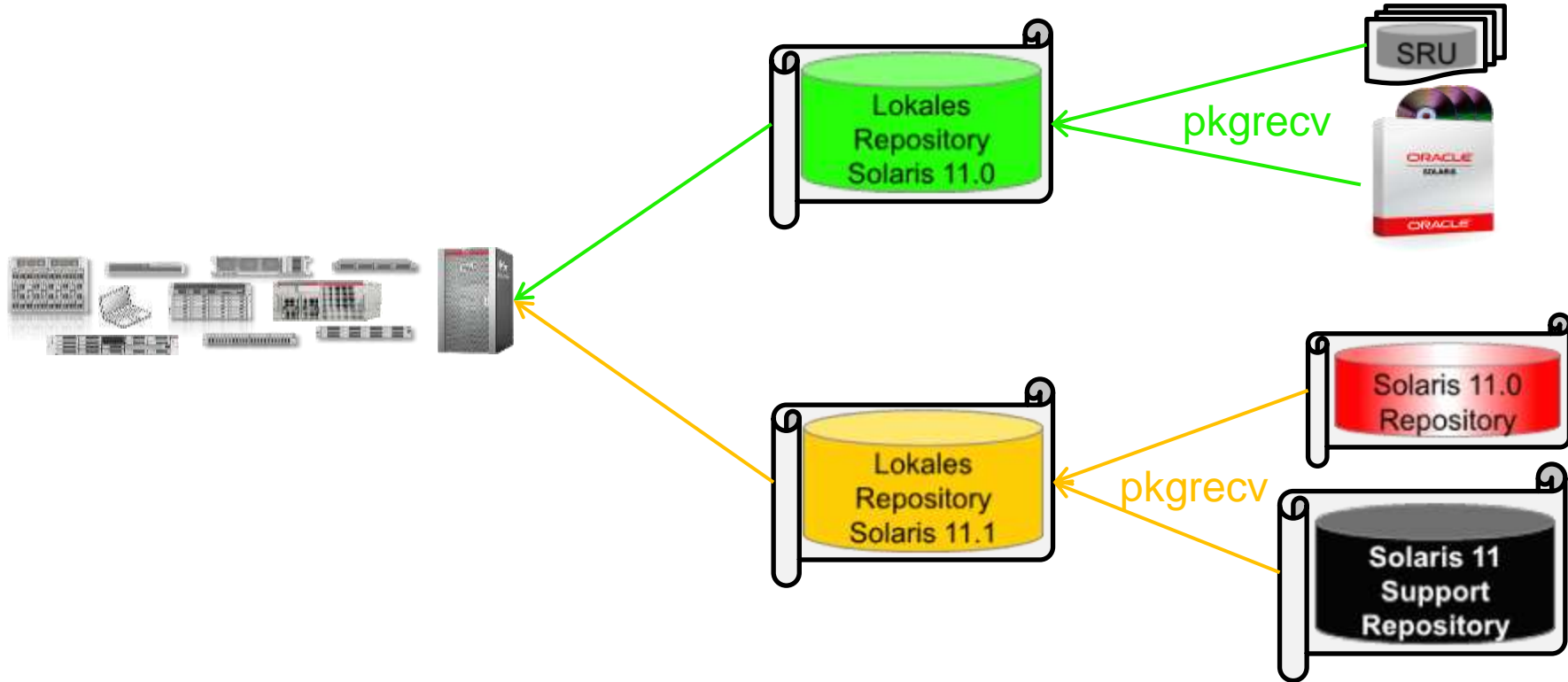
Je Release die SRU und das Release-Repository zusammen

- Je Release nur 1 Repository-Eintrag zu verwalten
- Repository wächst mit jedem SRU
- Je Release ein unabhängiges Repository
- Einfaches "Aufräumen" der Repositories für alte Releases
- Änderung des Publishers bei Upgrade notwendig
- **Gut einsetzbar**

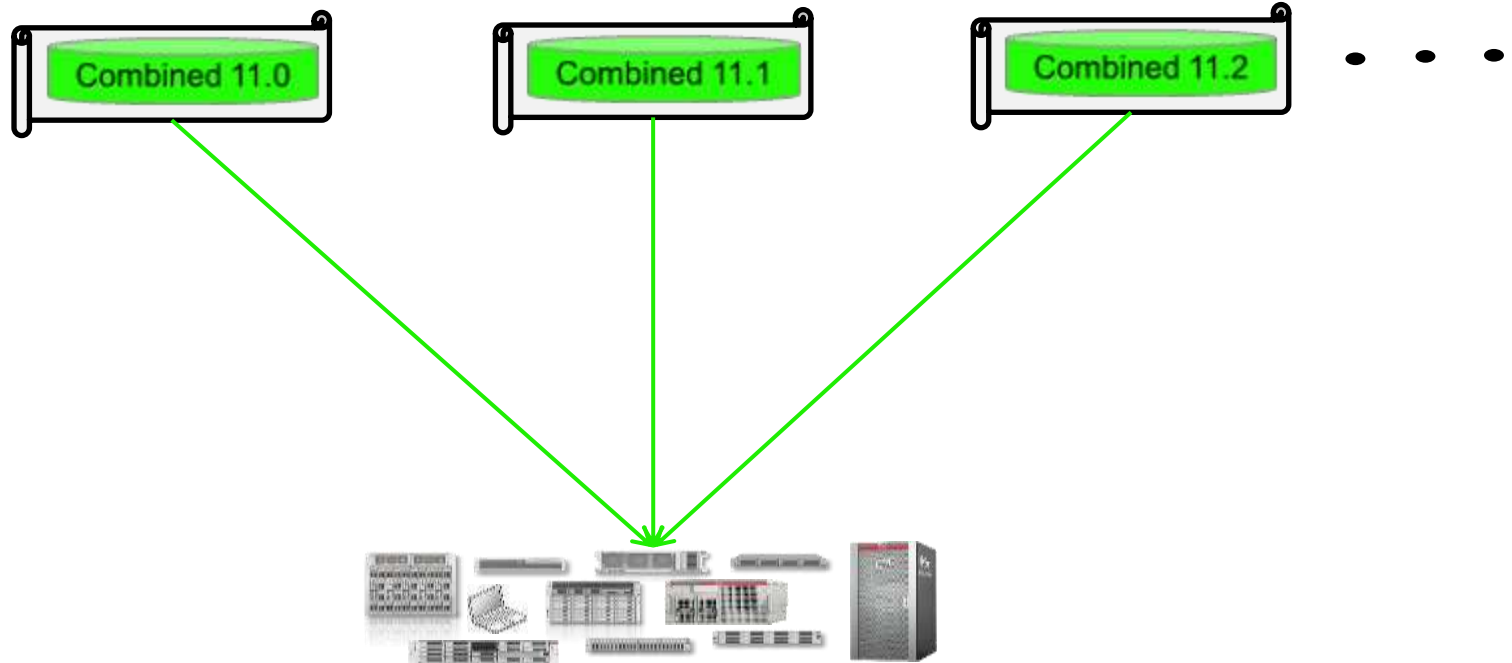
Das Combined Repository



Das Combined Repository



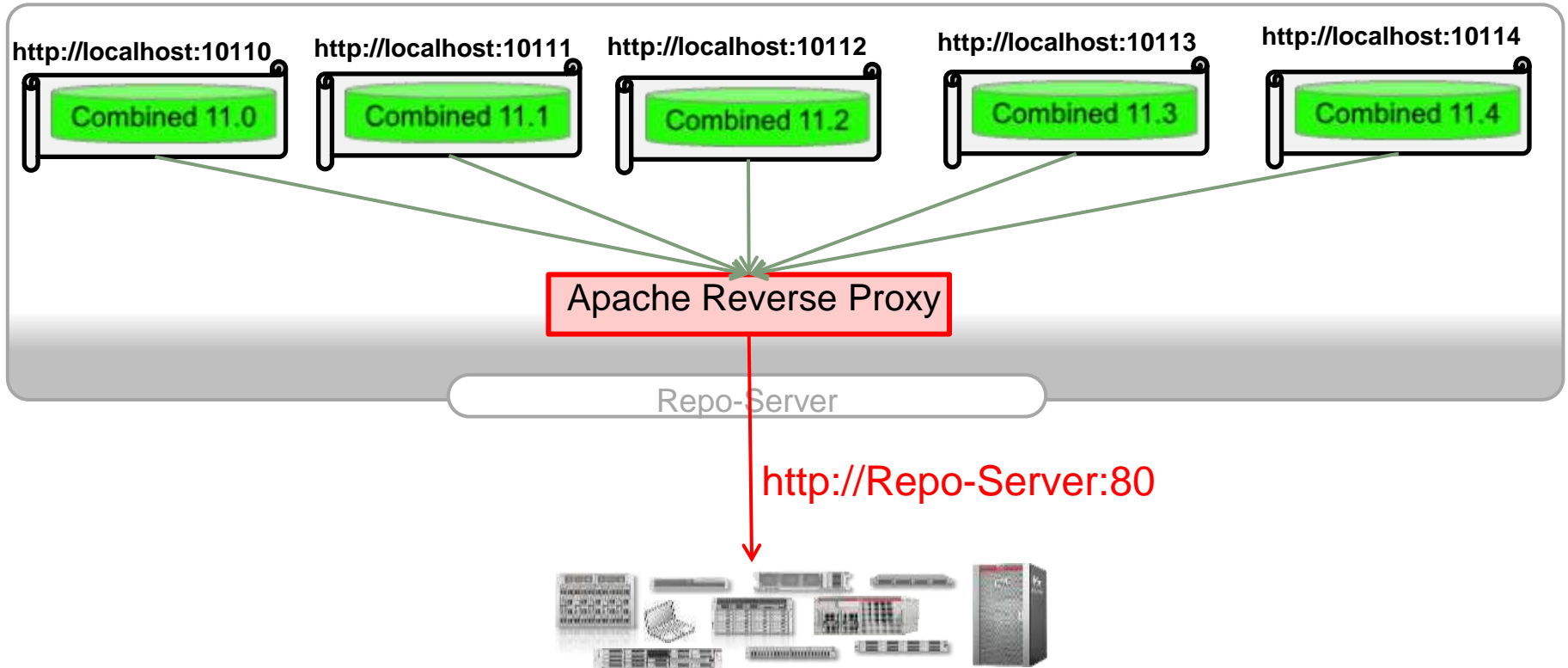
Mehrere Combined Repositories



Mehrere Combined Repositories

- Wieviele TCP-Ports werden benötigt, wenn wir z.B. bei Solaris 11.5 sind und Combined- oder SRU-Repositories verwendet haben ?
 - Verwendung in verteilten Netzen, Firewalls, Proxies, ...
- Zugriff auf alle Combined Repositories über einen gemeinsamen Reverse Proxy Apache Server und verschiedene URL

Mehrere Combined Repositories



Mehrere Combined Repositories

- <http://repo-server/>
 - /s11.0 zeigt auf <http://localhost:10110> (s11.0)
 - /s11.1 zeigt auf <http://localhost:10111> (s11.1)
 - /s11.2 zeigt auf <http://localhost:10112> (s11.2)
 - ...

Mehrere Combined Repositories

Konfiguration Depot-Server Instanz

```
# svccfg -s pkg/server add s11-0
# svccfg -s pkg/server:s11-0 addpg pkg application
# svccfg -s pkg/server:s11-0 setprop pkg/port=10110
# svccfg -s pkg/server:s11-0 setprop pkg/inst_root=/repo/s11.0
# svccfg -s pkg/server:s11-0 setprop pkg/readonly=true
# svccfg -s pkg/server:s11-0 setprop pkg/address=127.0.0.1
# svccfg -s pkg/server:s11-0 setprop pkg/proxy_base=astring: http://repo-server/s11.0
# svccfg -s pkg/server:s11-0 addpg general framework
# svccfg -s pkg/server:s11-0 addpropvalue general/complete astring: s11-0
# svcadm refresh pkg/server:s11-0
# svcadm enable pkg/server:s11-0
```

Mehrere Combined Repositories

Konfiguration Apache Server

```
# vi /etc/apache2/2.2/httpd.conf
    ServerName      repo-server
    Redirect        /s11.0 http://127.0.0.1/s11.0
    ProxyPass       /s11.0 http://127.0.0.1:10110 nocanon max=100
    AllowEncodedSlashes NoDecode
    MaxKeepAliveRequests 10000
    ProxyTimeout    30
    ProxyRequests   Off
    AddOutputFilterByType DEFLATE text/html \
                          application/javascript text/css text/plain

# svcadm enable apache22
```


Logging des Zugriffs auf Repositories

- Auswertung des Access Log-Files des Reverse Proxy Servers
- Erstellung und Auswertung des Access Log-Files des Depotd-Servers
- Auswertung z.B. mit <http://www.webalizer.org/>

Zusammenfassung

- Combined Repositories haben sich neben Single Repositories als beste Anwendungsform ergeben
- Mehrere Repositories können zur Vereinfachung mit verschiedenen URL hinter einem gemeinsamen Port betrieben werden
- HTTP-Repositories bieten die besten Eigenschaften für verteilte Anwendungen

Weitere Informationen

- „Evaluating Oracle Solaris 11“, Howto, Cheat Sheets, ...
 - <http://www.oracle.com/technetwork/server-storage/solaris11/overview/evaluate-1530234.html>
- **How to Update Oracle Solaris 11 Systems From Oracle Support Repositories**
 - <http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-018-howto-update-s11-1572261.html>
- **Solaris 11 SRU / Update relationship explained, and blackout period on delivery of new bug fixes eliminated**
 - https://blogs.oracle.com/Solaris11Life/entry/solaris_11_process_enhancement_no
- **More Tips for Updating Your Oracle Solaris 11 System from the Oracle Support Repository**
 - <http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-maintain-s11-sru-1627108.html>

Q&A

Detlef.Drewanz@oracle.com

ORACLE®