Sun Cluster and Solaris 10 Containers

Thorsten Früauf / Detlef Ulherr

Customer Engineering Conference (CEC) October 2006

Session Id: S280251

Sun Microsystems

Agenda

- Overview
- Explanation of HA Container Agent
- Explanation of 3.2 Zone Nodes
- Zone Nodes compared with Failover Zones
- Use-cases (what to use when)
- Converting to Zone Nodes
- URL References & Q/A

Overview (1)

- Already available since Sun Cluster 3.1 8/05
  - HA Container Agent
- To come in Sun Cluster 3.2
  - Zone Nodes
  - Coexistence and combination with HA Container Agent

Overview (2)
Sun Cluster HA Container Agent

- Agent is SUNW.gds based
- Treats Solaris Containers as resources
- Failover and multiple masters configurations possible
- Offers script and SMF component to integrate applications
- Some of the standard agents run on top of the HA Container Agent
- Available since Sun Cluster 3.1 8/05 for sparc and x64

Overview (3)
Sun Cluster Zone Nodes

- Treats non-global zones as virtual nodes (RG receiver)
- Integrated in RGM
- Some components of Sun Cluster run in non-global zones
- Failover, multiple masters and scalable RG configurations possible
- Most of the standard agents run in Zone Nodes
- Available starting with 3.2 GA

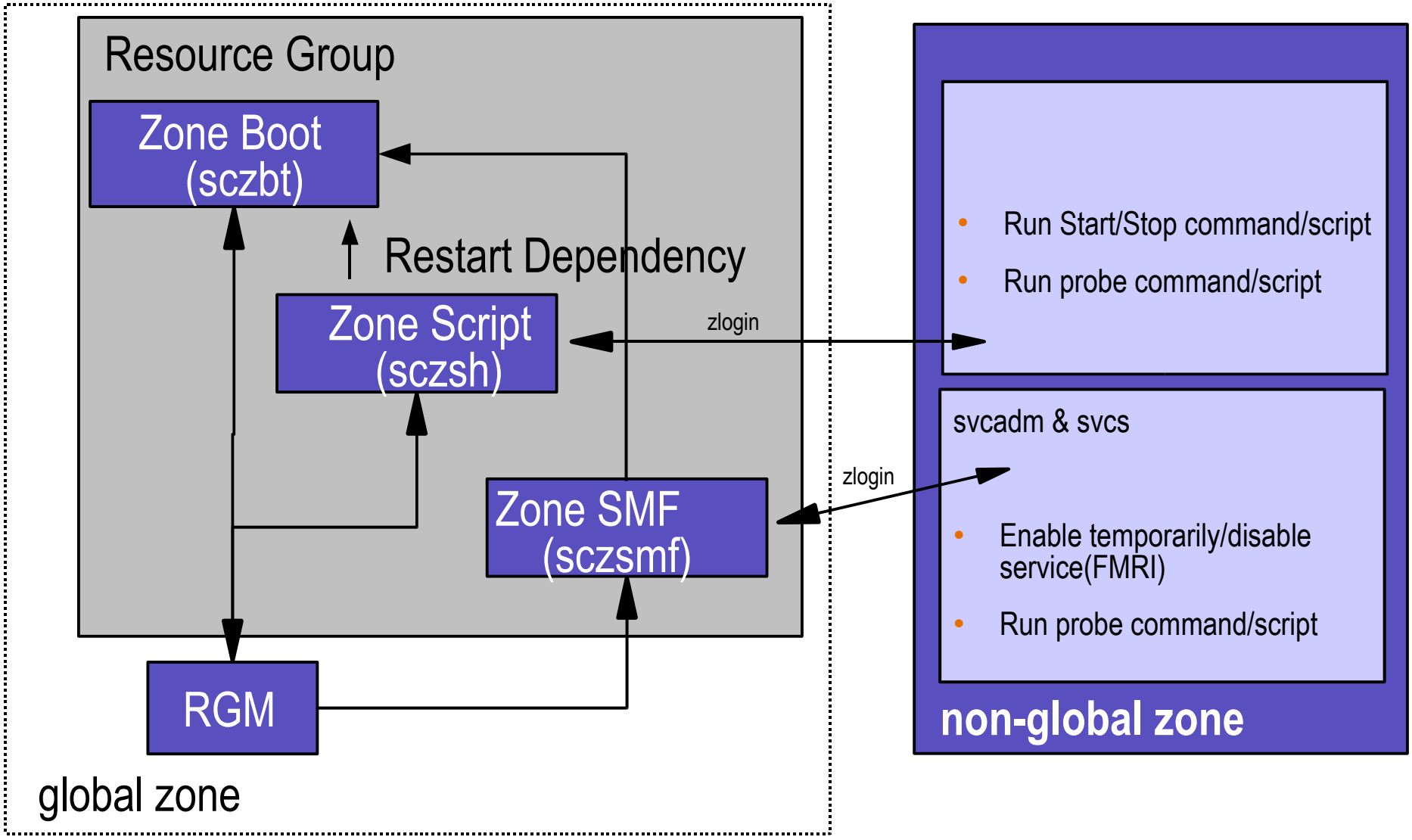
Explanation of HA Container Agent (1)

Sun Cluster 3.1 08/05 and non-global Zones

- Sun Cluster Framework runs in the global zone only
- Sun Cluster allows non-global zones coexistence
- Non-global zones can be controlled by the sczbt component of the HA Container Agent
- Applications running in non-global zones can be controlled by the sczsh/sczsmf components of the HA Container Agent

Explanation of HA Container Agent (2)

HA Container Agent Architecture



Explanation of HA Container Agent (3)

Failover zone Configuration

Node 1

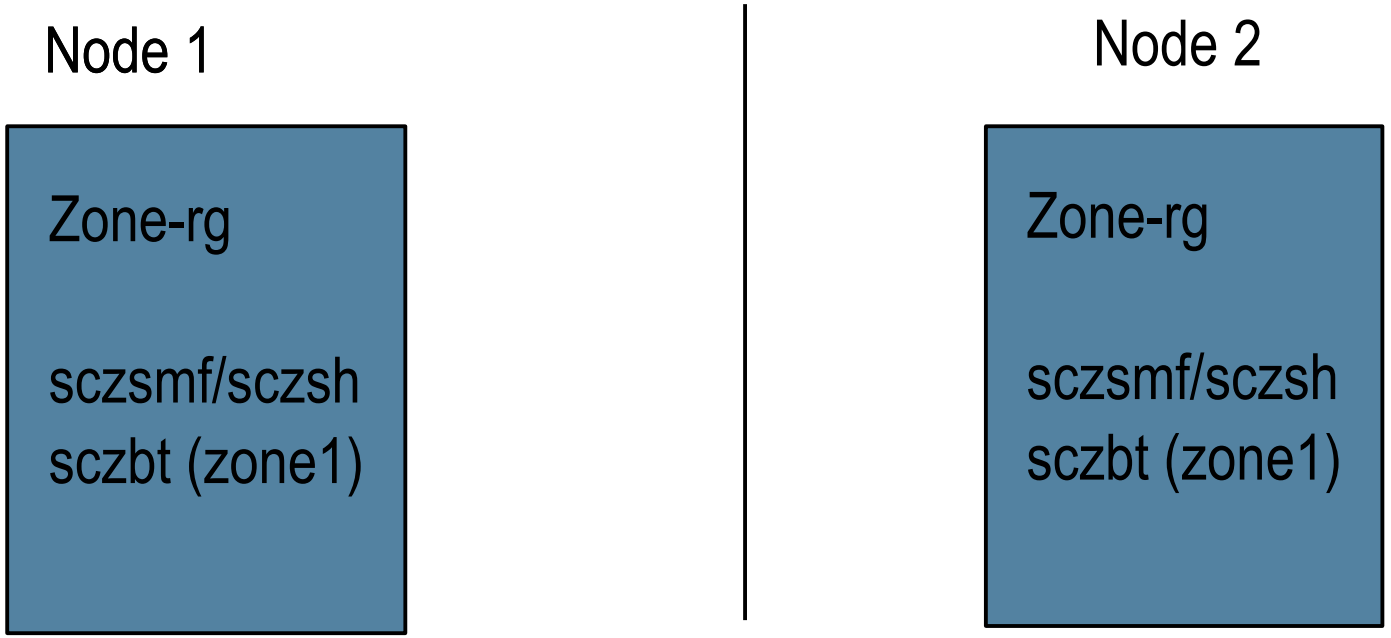
Zone-rg

sczsmf/sczsh
sczbt (zone1)
SUNW.HAStoragePlus
SUNW.LogicalHostname

Node 2

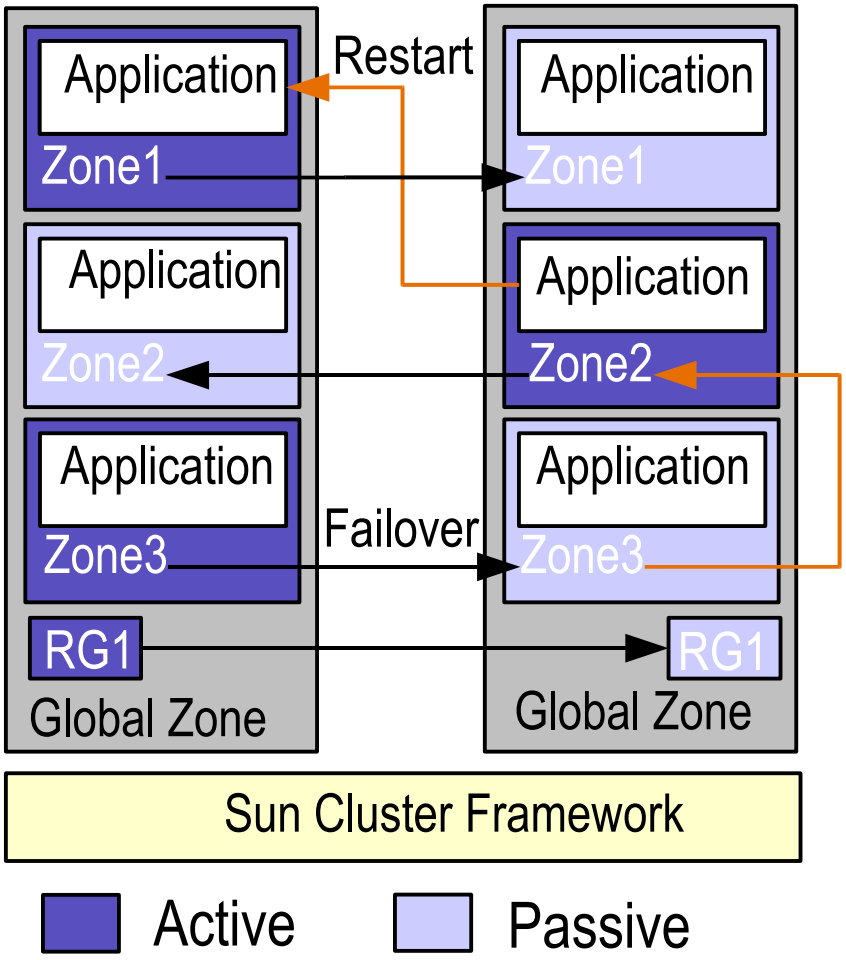
Explanation of HA Container Agent (4)

Multiple masters zone Configuration



Explanation of HA Container Agent (5)

Sun Cluster and Solaris Containers
Multiple highly available Solaris Containers



- Failover zones
- Dependencies between Applications across zones across nodes
- ++/-- Affinities or Offload zones across nodes
- Integration with Service Management Facility

Explanation of HA Container Agent (6)

HA Container Agent components explained

- Three components (within SUNWsczone package)
  - Zone Boot Agent (sczbt)
    - Boots/halts/monitors a non-global Zone
  - Zone Script Agent (sczsh)
    - Allows start/stop/probe commands/scripts to be run within the non-global zone
  - Zone SMF Agent (sczsmf)
    - Enables/disables & probes an SMF Service within the non-global zone
- Dependencies
  - Zone Script/SMF Agents are restart dependent on Zone Boot Agent

Explanation of HA Container Agent (7)

HA Container Agent Design (1)

general

- Each Zone Agent component (Zone, Script & SMF)
  - runs in the global zone (Zone 0)
  - callback via RGM / GDS using GDS like returncodes
  - all SC administration is done within the global zone
  - ability to manage the non-global zone's IP address with SUNW.LogicalHostname
  - ability to manage lofs file systems for the non-global zone
  - Non-global zone(s) configured and installed on each SC node
  - No auto-boot of the non-global zone from the global zone
  - Failover & Multiple Master Zones
    - Multiple Master: Zone name must be the same on all nodes

Explanation of HA Container Agent (8)

HA Container Agent Design (2)

Storage (1)

- Failover zones
  - zone rootpath = FFS (SUNW.HAStoragePlus) on UFS
  - Devices can be added (svm, did, ...)
  - currently no VxVM devices
  - Loopback mounts to failover and global file systems
  - Devices must have the same major and minor numbers on the possible master nodes
- Multiple master zones
  - zone rootpath = local filesystem on each node
  - Global devices can be added
  - Loopback mounts to global file systems

Explanation of HA Container Agent (9)

HA Container Agent Design (3)

Storage (2)

- Loopback mounts are the preferred solution
- Add loopback mounts either with zonecfg or in the sczbt components configuration
- Manage application file systems via SUNW.HAStoragePlus

Explanation of HA Container Agent (10)

HA Container Agent Design (4)

Network (1)

- Failover zones
  - IP addresses for failover zones can be configured in the zone (zonecfg) or within SUNW.LogicalHostname
  - Zones and logical hosts rely on IP Multipathing
  - Zones can be configured without IP addresses
  - SUNW.LogicalHostname protects against total LAN failure of one node
- Multiple master zones
  - Configure the IP addresses within the zone (zonecfg) only

Explanation of HA Container Agent (11)

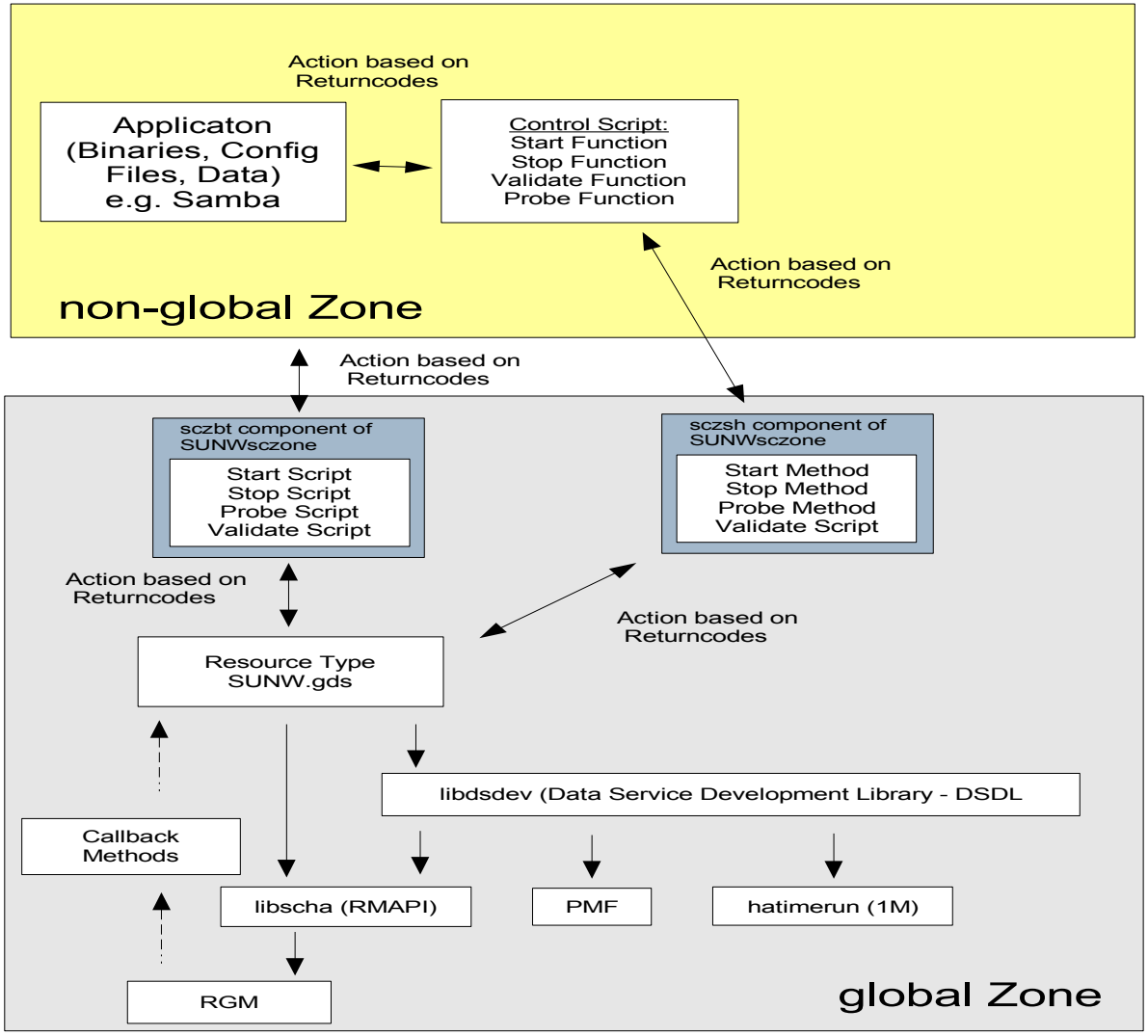
HA Container Agent Design (5)

Network (2)

- SUNW.LogicalHostname interfaces combined with network aware sczbt resource are moved in to the zone
- SUNW.LogicalHostname interfaces combined with network unaware sczbt resource remain in the global zone
- SUNW.LogicalHostname interfaces are flagged deprecated
- Interfaces configured with zonecfg are not flagged deprecated

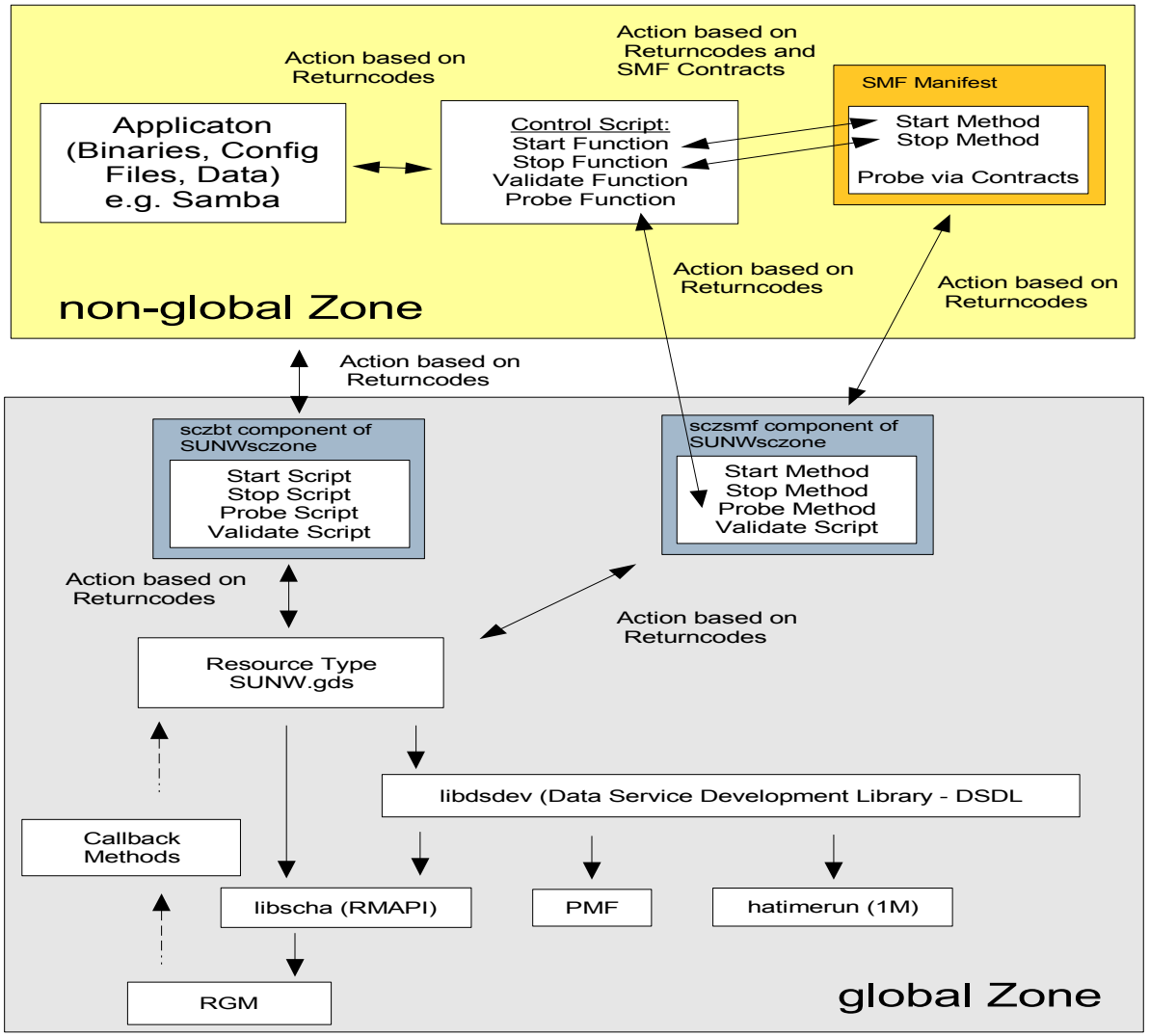
Explanation of HA Container Agent (12)

RGM with non-global zone and sczbt/sczsh components



Explanation of HA Container Agent (13)

RGM with non-global zone and sczbt/sczsmf components



Explanation of HA Container Agent (14)
Comparison between sczsh and scsmf components

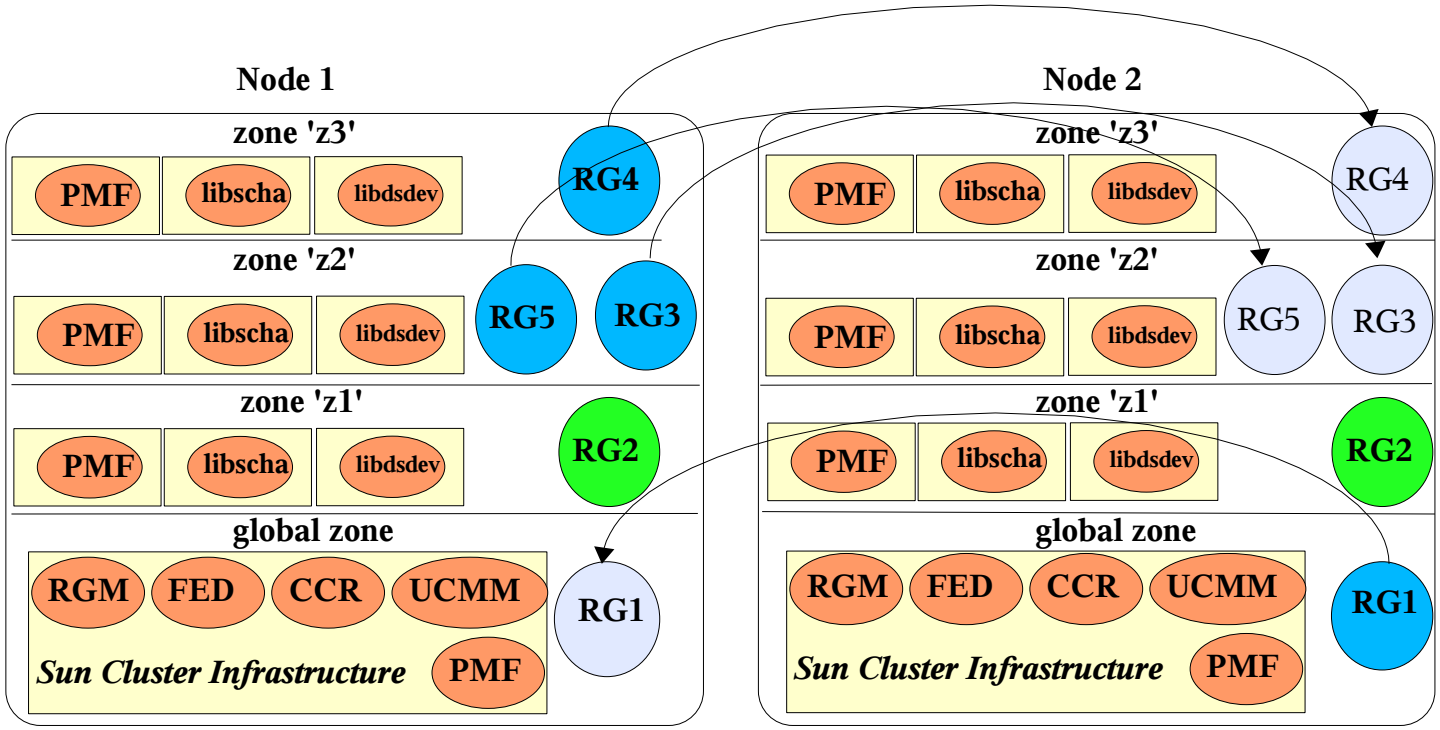
- SMF component (sczsmf)
  - Granular process monitoring by SMF
  - Optional “intelligent” probe with the component
  - Fastest error detection
  - SMF method scripts can be start and stop commands normally registered with GDS
- ZSH component (sczsh)
  - Keep it simple approach
  - PMF unaware
  - Probe is mandatory (but can be /bin/true)
  - Dealing with Solaris resource management is complex

Explanation of HA Container Agent (15)

Advantages of SMF usage in non-global zones

- since non-global zones in 3.1 8/05 have no Sun Cluster Framework inside, we need a way to
  - store configuration properties easy changeable
    - SMF has svcprop / svccfg
  - find a substitution for PMF
    - SMF uses contracts
  - handle user credentials and SRM projects properly
    - SMF offers this by design also changeable via svccfg
- offers reaction to FMRI events and a more granular way to react to signals send to our processes
- SMF is not optional anyway - customers will have to learn and use it

Explanation 3.2 Zone Nodes (1)
Sun Cluster components on a Solaris 10 Cluster



Explanation 3.2 Zone Nodes (2)
Overview (1)

- Multiple resource groups can run in the same zone and fail over independently
- Allows zones to be dynamically created/destroyed
  - Using the usual Solaris tools
  - Automatic discovery by RGM
  - Just create zone and mention zone name in the RG Nodelist
- Can coexist with failover zones on same cluster

Explanation 3.2 Zone Nodes (3)
Overview (2)

- Supports unbounded (large) number of zones
- Resource group(s) can run in any combination of zones, on any node(s)
  - Allows a resource group to fail over between zones on the same node
    - Does not really provide high availability
    - Enables prototyping and development of failover data services on single-node clusters

Explanation 3.2 Zone Nodes (4)

Zone Node Isolation

- Zone isolation is incomplete
  - new zones see cluster namespace
- User running in a non-global zone can “see” resource groups configured in other zones
- User running in a non-global zone cannot modify or affect behavior of resource groups in other zones unless those resource groups list the non-global zone in their Nodelist property
- Cluster administration is done from the global zone
- Security to be enhanced by the “Clusterized Zones” project in a future release

Explanation 3.2 Zone Nodes (5)

Representation of a Zone in the Nodelist property

- "Logical Nodename" **`nodename:zonename`**
  or **`nodename`**
  - (old) Nodelist=node1,node2,node3
  - (new) Nodelist=node1:zoneA,node2:zoneA,node3:zoneA
- Also permitted:
  - RG runs in different zone name per node:
    - Nodelist=node1:zoneA,node2:zoneB,node3:zoneC
  - RG runs in multiple zones on single physical node:
    - Nodelist=node1:zoneA,node1:zoneB,node1:zoneC

Zone Nodes vs Failover Zones (1)

- Resource/Resource Group
  - Zones are RG receivers (virtual nodes)
- Scalable/Multiple Masters
  - Supports multiple masters and scalable resources
- Zone creation
  - Zones need to be created on each node

- Resource/Resource Group
  - Zones are resources
- Scalable/Multiple Masters
  - Supports multiple masters resources only
- Zone creation
  - Zones configuration needs to be available on each node

Zone Nodes vs Failover Zones (2)

- Root path location
  - Local storage
- Zones boot process
  - Zone needs to boot automatically (autoboot=true)
- Zones content administration
  - Needs to happen on each physical node

- Root path location
  - Failover file system
- Zones boot process
  - Controlled by sczbt component (autoboot=false)
- Zones content administration
  - Needs to happen on one physical node only

Zone Nodes vs Failover Zones (3)

- Affinities between RGs running in global and non-global zones
  - > No, because Nodelist will differ for the RGs
- Standard agents
  - > Most of the agents run in zone nodes
- Custom Agents
  - > Yes

- Affinities between RGs running in global and non-global zones
  - > Yes, because Nodelist can be identical
- Standard agents
  - > Some of the agents run with failover zones
- Custom Agents
  - > GDS based agents leveraging sczsh/sczsmf

Zone Nodes vs Failover Zones (4)

- SMF services
  - Yes, with proxy resource type. No application specific probe. SMF repository exists in each zone and must be manually kept in sync.
- Resource type methods
  - Most methods run directly in the non-global zone, except: HASTP, IP

- SMF services
  - Yes, with sczsmf component. Application specific probe script optional. SMF repository exists only once, no need to sync it.
- Resource type methods
  - Methods are executed from the global zone via zlogin in the non-global zone

Zone Type Use Cases (1)

Use Zone Nodes if (1)

- The application is supported by the ISV to run in zones (prerequisite)
- The necessary standard agents are supported for non-global zones (prerequisite)
- Fine grained resource topologies are acceptable (prerequisite)
  - Every application in the non-global zone is configured as a resource
- Minimum failover time is a key requirement (ZN indicator)
  - Zone Nodes are booted at node boot time (autoboot=true)

Zone Type Use Cases (2)

Use Zone Nodes if (2)

- A SUNW.SharedAddress resource is required (ZN indicator)
- Dual partition software upgrade is required (ZN indicator)
- Minimum downtime during maintenance is required (ZN indicator)
- All RG's with affinities contain the same Nodelist
  - Nodelist RG1 Nodelist=nodea:zone1,nodeb:zone2
  - Nodelist RG2 Nodelist=nodea:zone1,nodeb:zone2
  - RG1: RG_affinities=++RG2
- No indication for HA Container Agent

Use Zone Nodes as your default design approach.

Zone Type Use Cases (3)
Use HA Container Agent if (1)

- The application is supported by the ISV to run in zones (prerequisite)
- Non-global zone needs to be configured as Blackbox (CA indicator)
  - Delegated root access
  - Application not supported in a cluster
  - Solaris integration (runlevel script or SMF service) available and agent too complex to create ie. dispersed file systems dependencies to /etc /var, ...
  - Base Solaris services (sendmail, print spooler, crontab, ...) needs to be HA

Zone Type Use Cases (4)

Use HA Container Agent if (2)

- SMF Service for application available/required and intelligent application probe is necessary (CA indicator)
- Onestop administration is required (CA indicator)
  - Application/service needs to be administered in one non-global zone only
- No fine grained resource topology acceptable (CA indicator)
  - Only the key application needs to be controlled by Sun Cluster, but the rest has to failover as well

Zone Type Use Cases (5)

Use HA Container Agent if (3)

- RG affinities needed between RG's with different Nodelists (CA indicator)
  - Nodelist RG1 Nodelist=nodea,nodeb covers zone1
  - Nodelist RG2 Nodelist=nodea,nodeb covers zone2
  - RG1: RG_affinities=++RG1
  - This can not be achieved with Zone Nodes
- Failover time is less important (prerequisite)
  - Failover time increases due to zone boot and reporting online after reaching a configurable milestone
- Service downtime for container maintenance is acceptable (prerequisite)

Zone Type Use Cases (6)

Combination of HA Container Agent and Zone Nodes

- Resources with clear indication for HA Container Agent are dependent from/to resources in Zone Nodes or the global zone
  - Blackbox zone is restart dependent to HA-Oracle in a Zone Node
  - Container or non-Container aware standard agent resource is dependent from HA-NFS
- A zone managed by the HAContainer Agent is treated as a Zone Node in an other resource group (Nodelist=nodea:ffzone,nodeb:ffzone)
  - Restriction with give-over, details see CR 6443496

Converting to Zone Nodes (1)

Design for later conversion to Zone Nodes (1)

- No Blackbox zones
- Plan for fine grained RS topologies, all applications should be covered by resources
- Use the sczsmf component without a probe script
- Use the sczsh component, if you want a probe script
- Plan the applications data in separate SUNW.HAStoragePlus resources

Converting to Zone Nodes (2)

Design for later conversion to Zone Nodes (2)

- Configure the lofs mounts of the applications file systems in the sczbt configuration file
- Configure the applications logical host with SUNW.LogicalHost and mention it in the SC_NETWORK parameter of the sczbt configuration file

Converting to Zone Nodes (3)

Convert from HA Container Agent to Zone Nodes (1)

Assumption: Sun Cluster (3.2) and/or Solaris upgrade is already done

- Deactivate the resources in the Container RG and export the RG configuration
- Delete the Container RG and resources
  - Delete the SMF manifests of Container aware standard agents with the supplied script
- Move the zone root path to local storage, and copy the zone configuration and root path to the other nodes

Converting to Zone Nodes (4)
Convert from HA Container Agent to Zone Nodes (2)

- Ensure, that your IP and application specific local data (/etc/hosts, users, ...) is the same on the zone on all nodes
- Modify autoboot=false to autoboot=true and boot the zone
- Create the RG, SUNW.HAStoragePlus and SUNW.LogicalHost resources
- Create/recreate the standard agent resources
  - Use standard agents whenever available
- Recreate the former SMF components with SMF proxy resources

Converting to Zone Nodes (5)
Convert from HA Container Agent to Zone Nodes (3)

- Convert sczsh resources to GDS agents
  - The start, stop and probe scripts are compatible if they leave processes running to satisfy PMF
- Activate and test your resource group

URL References

- http://docs.sun.com/app/docs/doc/819-2664
  - Sun Cluster 3.1 08/05 Data Service for Solaris Containers
- http://docs.sun.com/app/docs/prod/sun.cluster32#hic
  - Sun Cluster 3.2 Dokumentation
- http://www.opensolaris.org/os/community/smf/manifests/
  - Open Solaris Community for SMF, Converted services: manifests and methods
- http://www.sun.com/bigadmin/content/selfheal/smf-quickstart.html
  - Solaris Service Management Facility - Quickstart Guide
- http://www.sun.com/bigadmin/content/selfheal/sdev_intro.html
  - Solaris Service Management Facility - Service Developer Introduction

Thank You!

Thorsten Früauf / Detlef Ulherr

thorsten.frueauf@sun.com / detlef.ulherr@sun.com