ICP Architectural Considerations
HA & DR aspects

—

Eduardo Patrocinio
DE, Cloud Solutions

Designing for HA&DR

To make the correct HA & DR design decisions we need to understand:

Application HA & DR options

The different layers: application, kubernetes platform and infrastructure

Performance and recovery aspects on the different layers

The different integrations for security and service management components

What latencies do we have between the different locations?

Etc.

Specifically for DR:

What RPO and RTO?

How is data (cluster and apps) handled and recovered?

Etc.

Infrastructure matters

Why does it matter for HA&DR? Isn’t Kubernetes doing everything?

Infrastructure blocks

The infrastructure layer will dictate which style and method of HA & DR you can apply.

Compute

Networking

Storage

(Hyper)convergence & public cloud

Infrastructure blocks

The infrastructure layer will dictate which style and method of HA & DR you can apply.

Compute	Kubernetes “virtualizes” applications not infrastructure
Networking	Optimize performance vs. Infra resources (baremetal vs virtualization)
Storage	Recovery aspects will influence
(Hyper)convergence & public cloud	

Infrastructure blocks

The infrastructure layer will dictate which style and method of HA & DR you can apply.

Compute

Networking

Storage

(Hyper)convergence & public cloud

You need Stretched VLANs (even stretched IaaS clusters) if we want to consider stretched clusters.

How do we achieve DC interlinks?

Infrastructure blocks

The infrastructure layer will dictate which style and method of HA & DR you can apply.

Compute	With cloud native applications the data availability/replication is done at the app layer.
Networking	Middleware usually counts on infrastructure components.
Storage	You will need active-active (cross DC) capabilities for your cluster as well as for your persistent volumes.
(Hyper)convergence & public cloud	

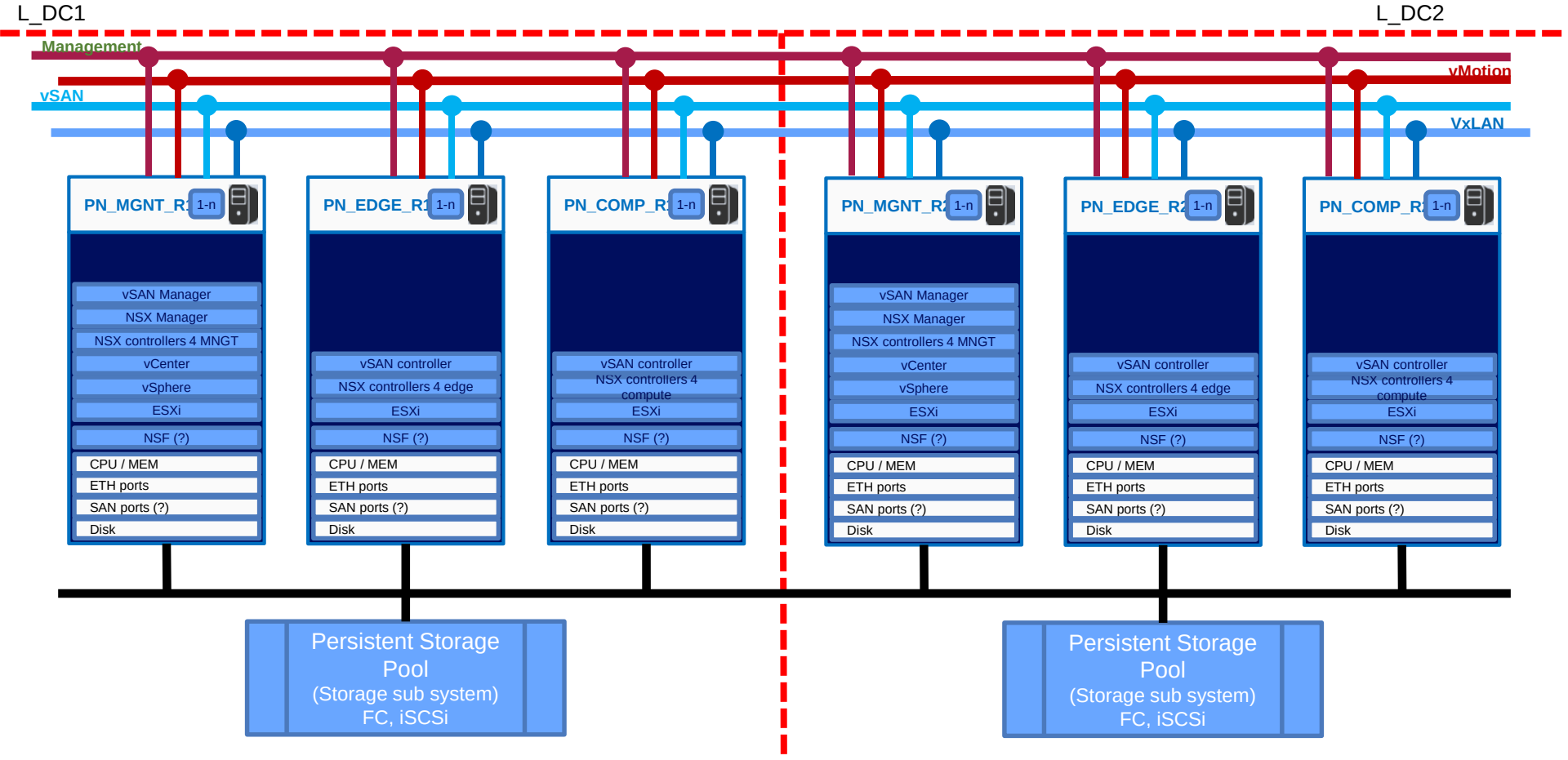
Infrastructure blocks

The infrastructure layer will dictate which style and method of HA & DR you can apply.

Compute	
Networking	Appliances are in many cases designed for HA, if you do the proper configuration.
Storage	For DR still proper design needs to be done, you need to understand the DR capabilities that are offered from the appliance not from the individual components.
(Hyper)convergence & public cloud	

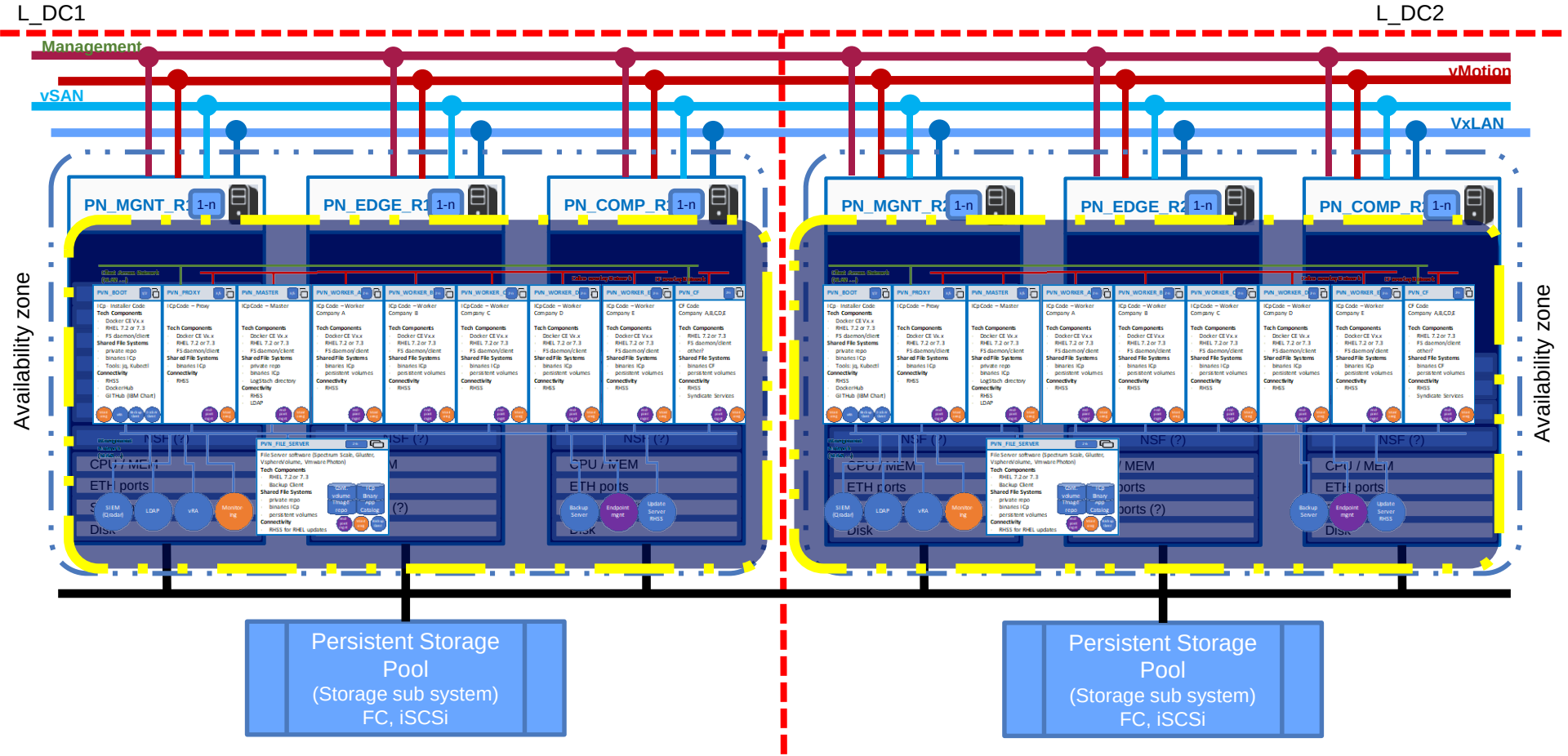
VMware SDDC Example

SDDC can have different regions which can be joined together
– giving separate physical implementation per DC even per DC per security zone with split Availability zones across the DCs



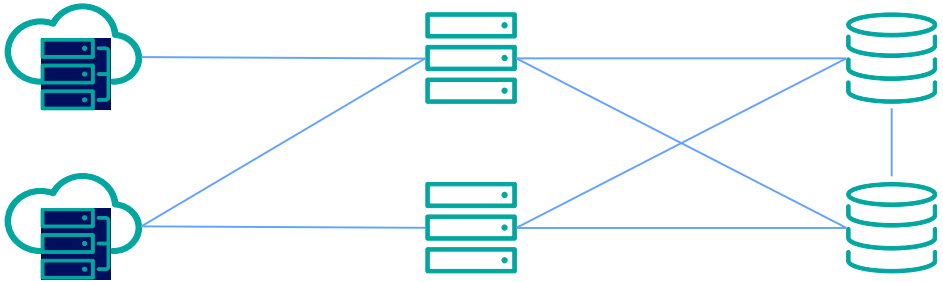
Physical view (VMware SDDC)

SDDC can have different regions which can be joined together
– giving separate physical implementation per DC even per DC per security zone with split Availability zones across the DCs

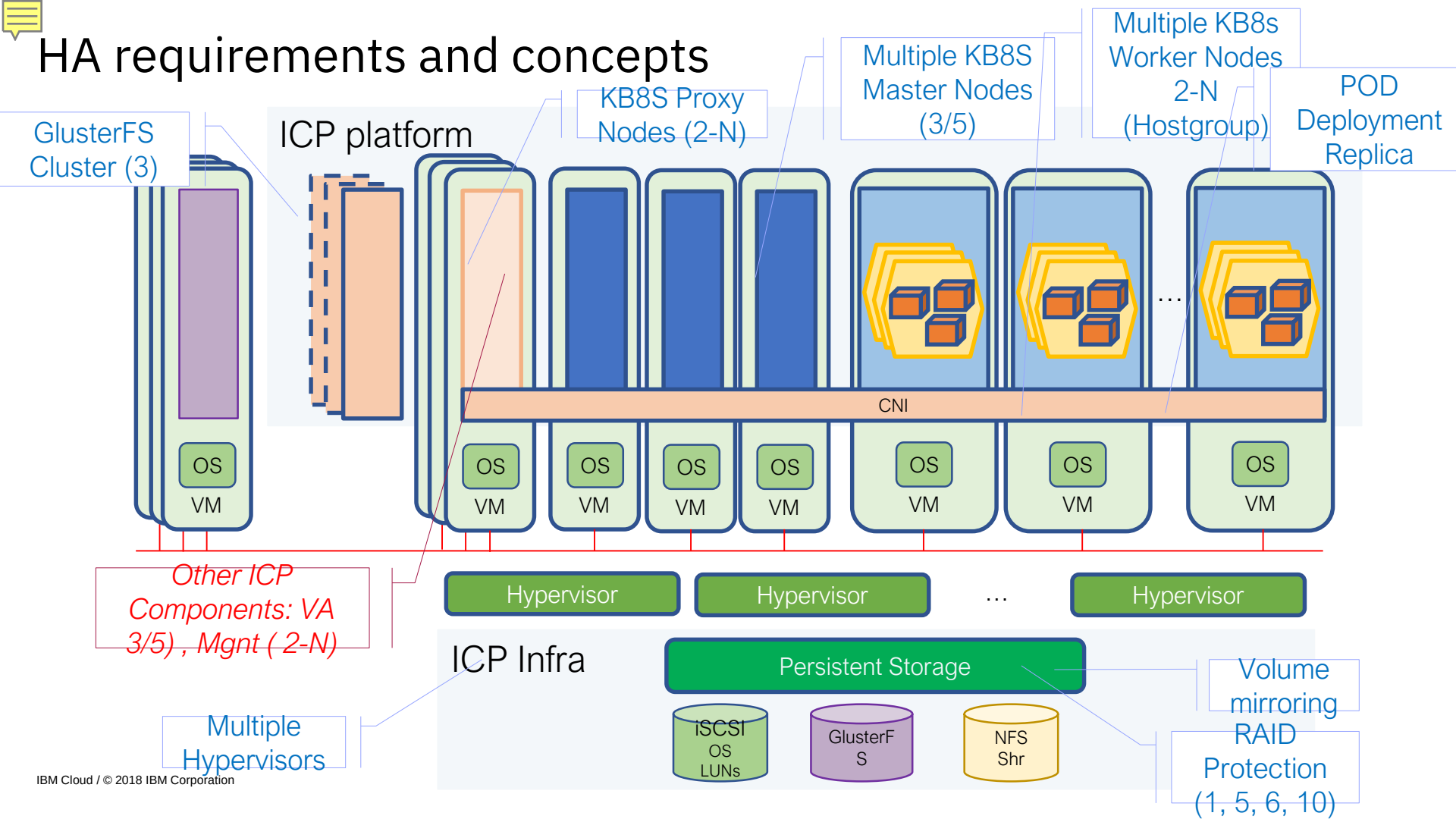


Platform HA considerations

HA definition?

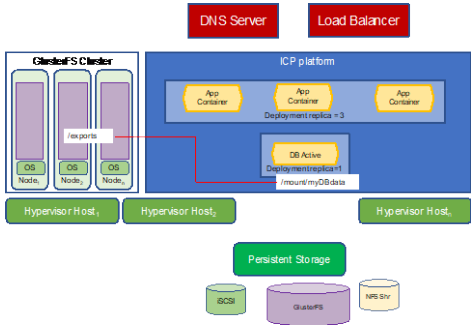


HA requirements and concepts

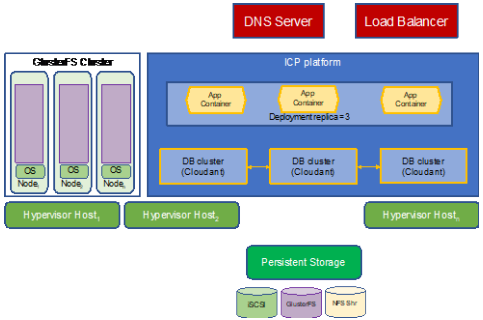


HA models

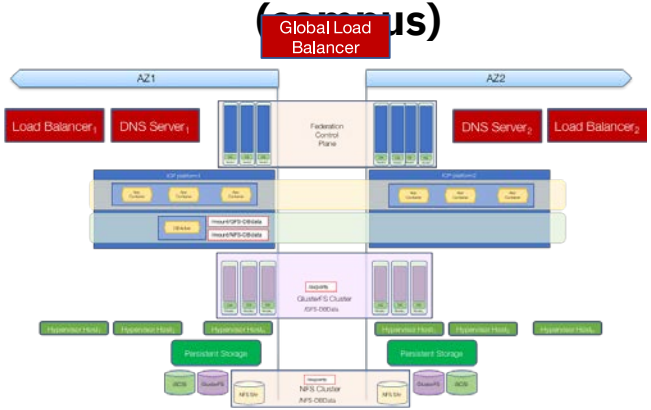
1. intra ICP/Kubernetes Application POD cluster



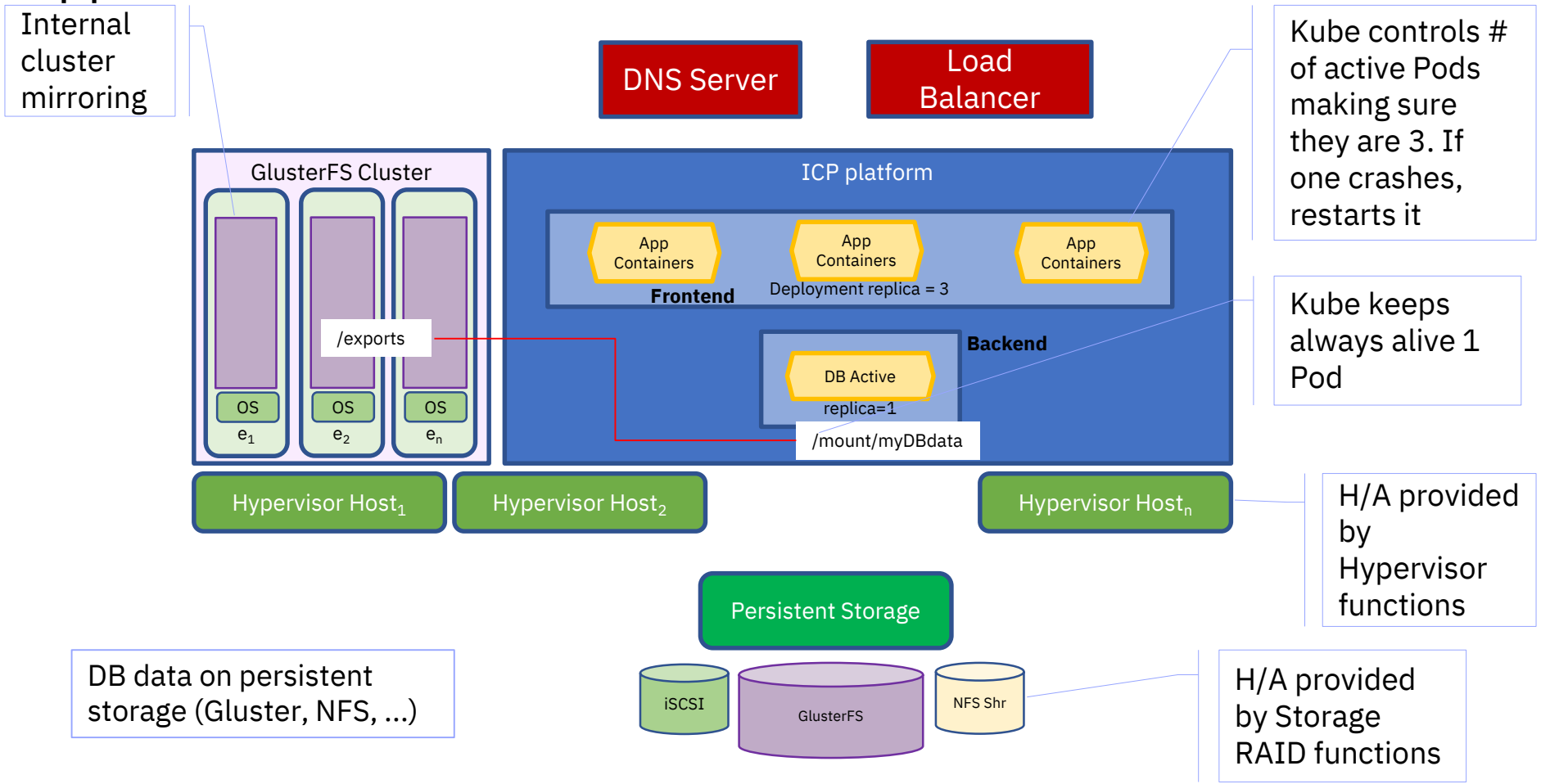
2. intra ICP/Kubernetes Application POD cluster and Database cluster



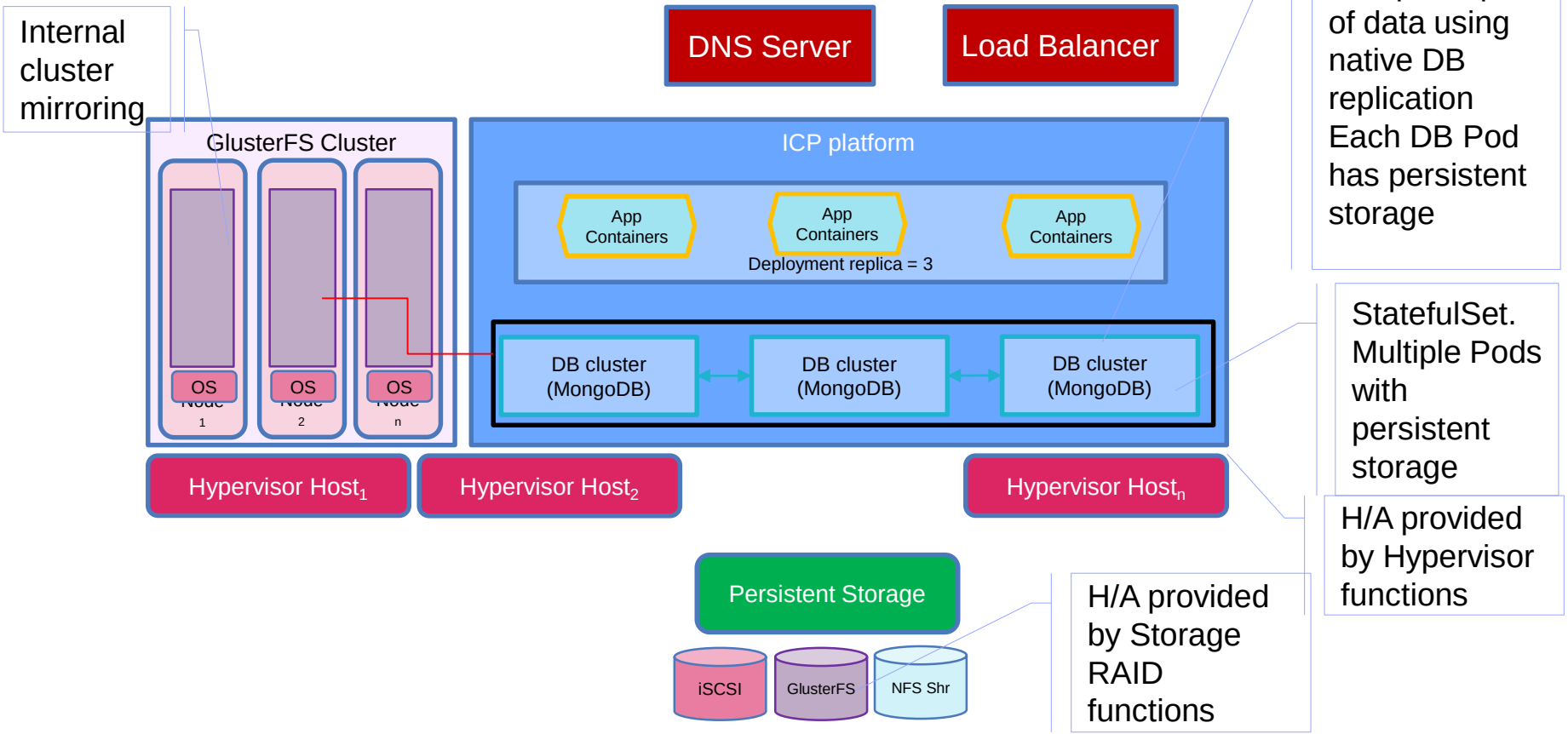
3. Inter ICP/Kubernetes cluster on different Availability Zones (campus)



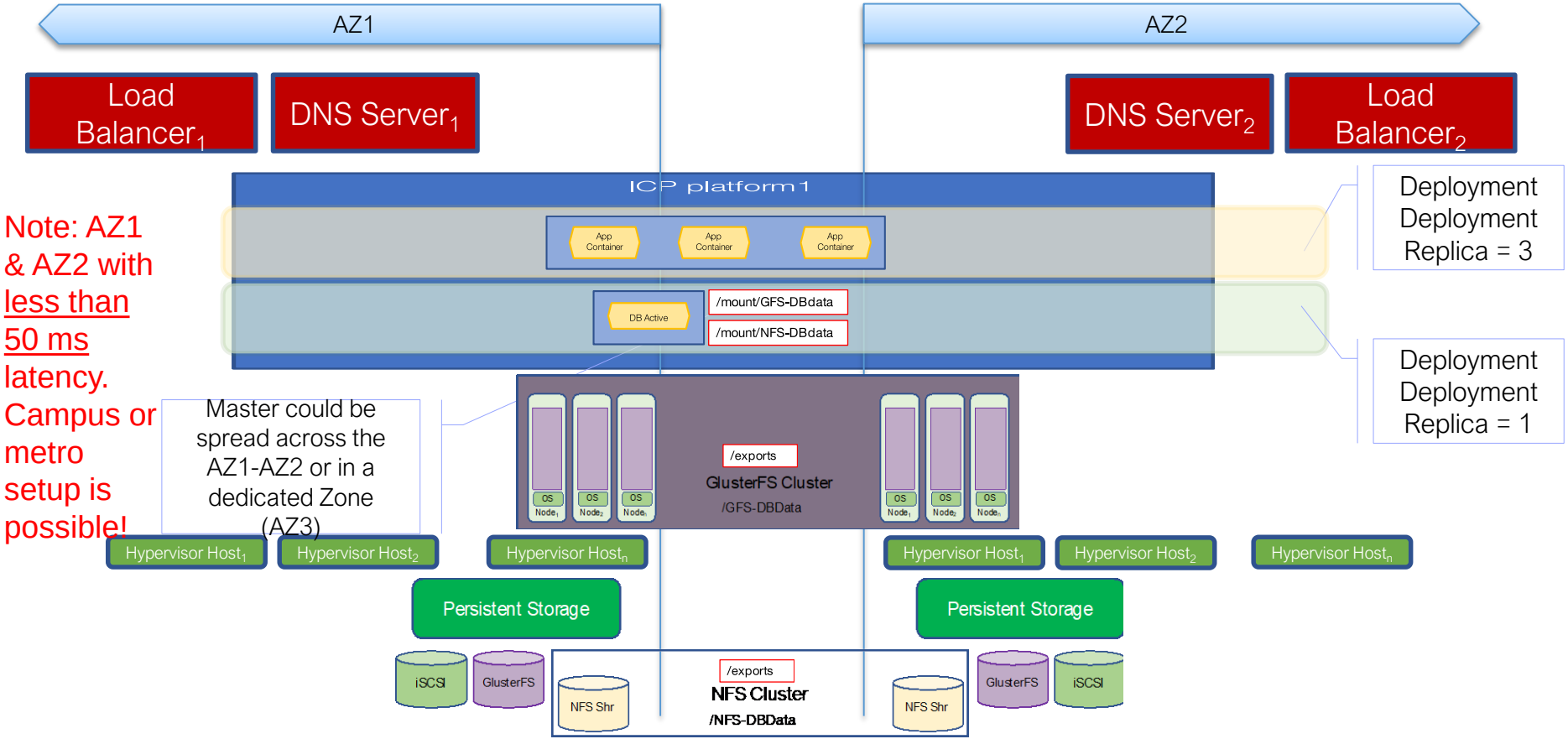
Application HA Models – intra Kubernetes cluster



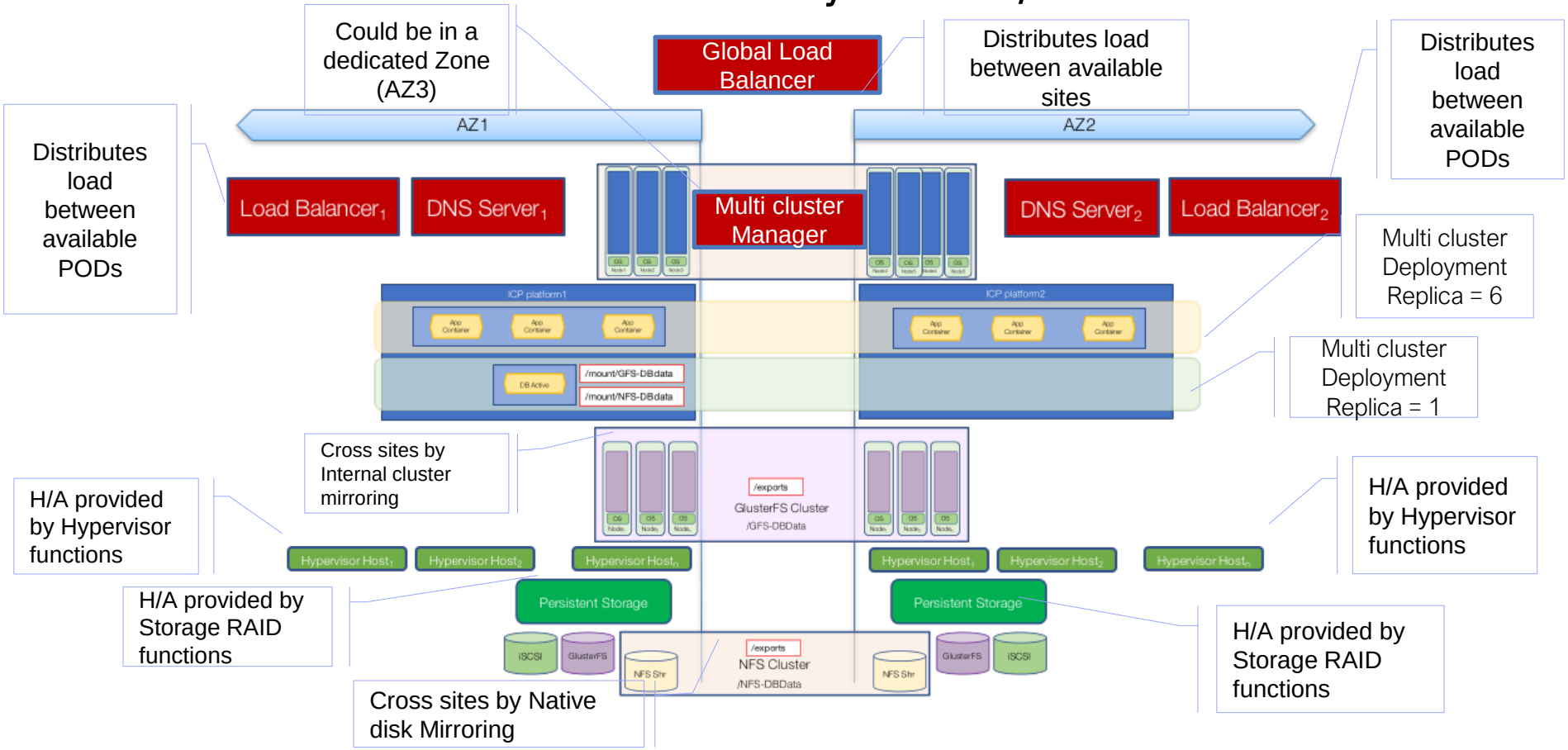
Application HA Models – intra Kubernetes cluster



Application HA Models – intra Kubernetes cluster – on different availability zones / locations

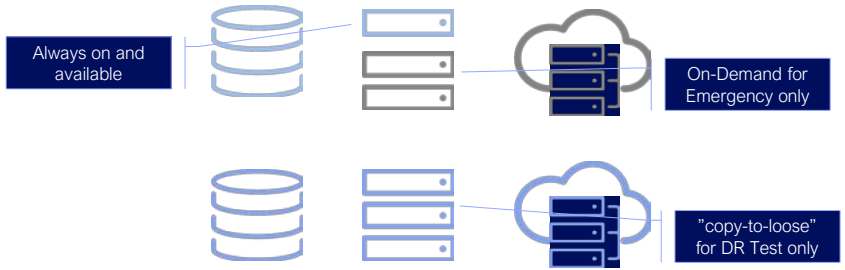


Application HA Models – intra-site Kubernetes cluster – Federation on different Availability Zones / locations



Platform DR considerations

DR definition?



DR Best Practices

1. Plan and Design for the “worst conditions”

Things will go bad when in disaster
Don’t throw your DEV/TEST away

Several other factors (non-exhaustive list):

- Key personnel availability
- External networks services
- Dependencies on Critical Providers
- Road conditions (when planning to physical transfers of personell, backup, ...)
- DR resources availability when required

DR Best Practices

2. DR Test frequency

IF you don't test, don't spend the money

Test entire groups of applications, not just a single core system

3. How you Test

There are two type of DR testing you can execute to verify your resiliency capabilities:

- DR simulation
- Switch-over

Backup Aspects

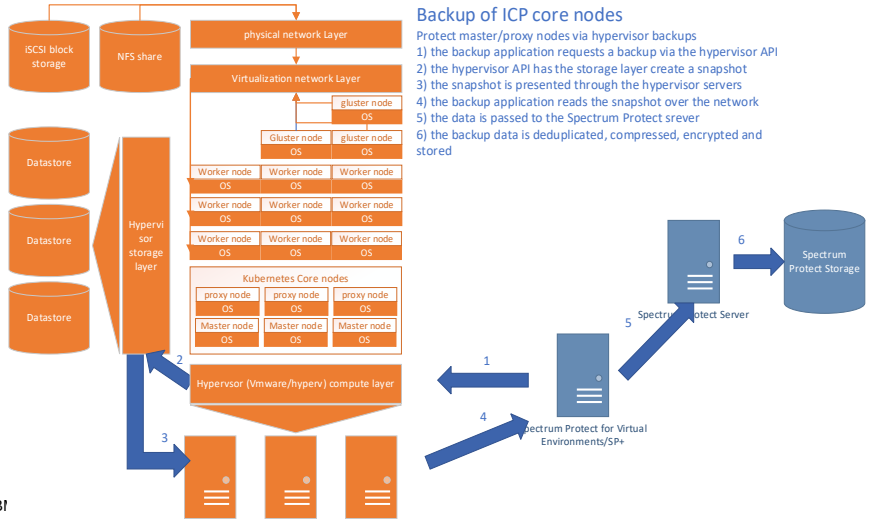
Backup use cases

UC 1: Recovery of ICP platform components

UC 2: Recovery of persistent storage volumes

UC 3: Granular recovery of storage volume contents

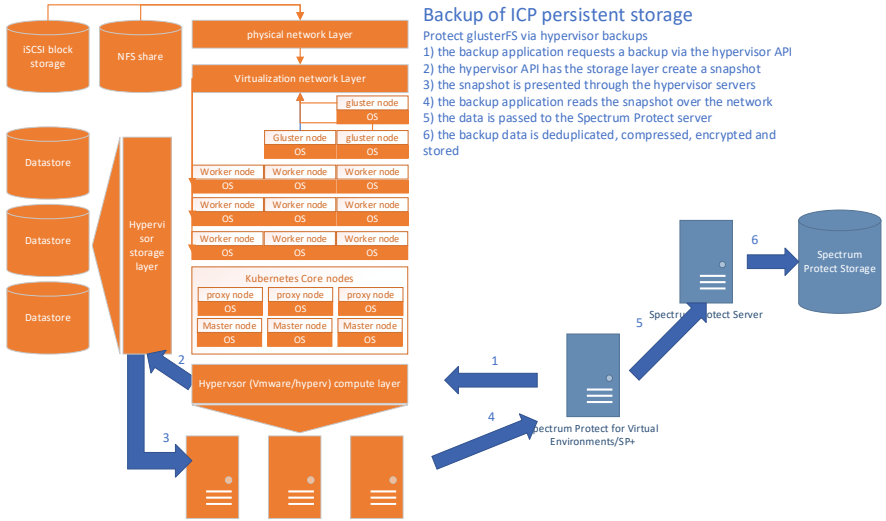
UC 4: Recovery of container images (Registry)



Backup of ICP core nodes

Protect master/proxy nodes via hypervisor backups

1) the backup application requests a backup via the hypervisor API
2) the hypervisor API has the storage layer create a snapshot
3) the snapshot is presented through the hypervisor servers
4) the backup application reads the snapshot over the network
5) the data is passed to the Spectrum Protect srever
6) the backup data is deduplicated, compressed, encrypted and stored

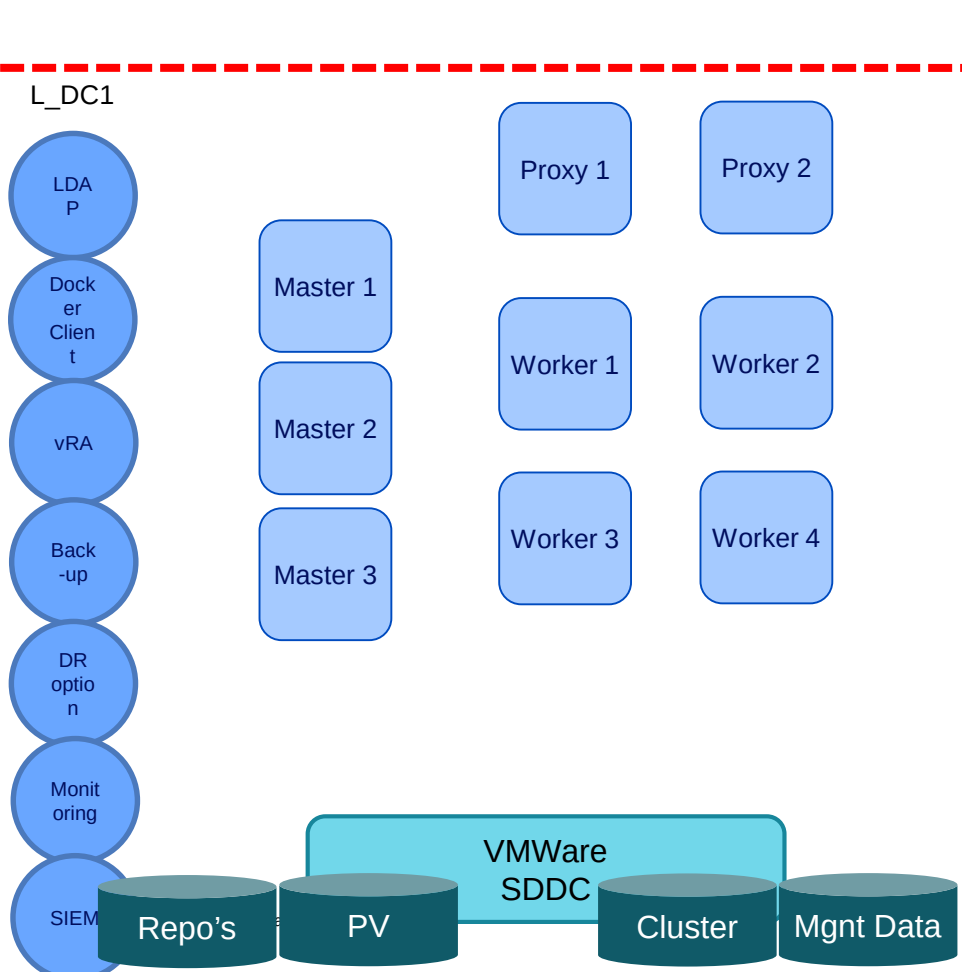


Backup of ICP persistent storage

Protect glusterFS via hypervisor backups

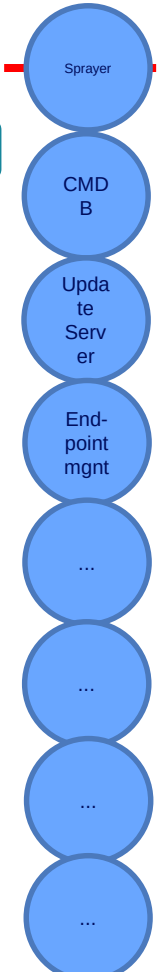
1) the backup application requests a backup via the hypervisor API
2) the hypervisor API has the storage layer create a snapshot
3) the snapshot is presented through the hypervisor servers
4) the backup application reads the snapshot over the network
5) the data is passed to the Spectrum Protect server
6) the backup data is deduplicated, compressed, encrypted and stored

Backup and Recovery of an ICP Cluster



Backup

1) VM based backup of the Master nodes
   - Recovery of individual Master nodes is per VM as we need to keep the IP addresses
2) VM based backup VA, Mgnt, Proxy and Worker Nodes
   - individual nodes can be recovered by add node but full cluster recovery will be done through the VM
3) Backup of the shared volumes for the cluster, Mgnt data and repo data – this needs to be done as much as possible in sync
4) Backup of the consumed file systems through the persistent volumes need to be done depending on the needs of the workload



ETCD backup and Recovery

Etcd (a key-value store) has built in HA capabilities when deployed across multiple nodes.

In addition to backing up the files and/or VM of the master database to ensure application consistency the kubernetes admin should schedule regular consistent dumps of the etcd which can be used for local recovery in addition to being picked up by the backup software for off host protection.

To create the consistent copy use the “etcdctl snapshot save” command

ETCD backup and Recovery

Recovery multiple masters:

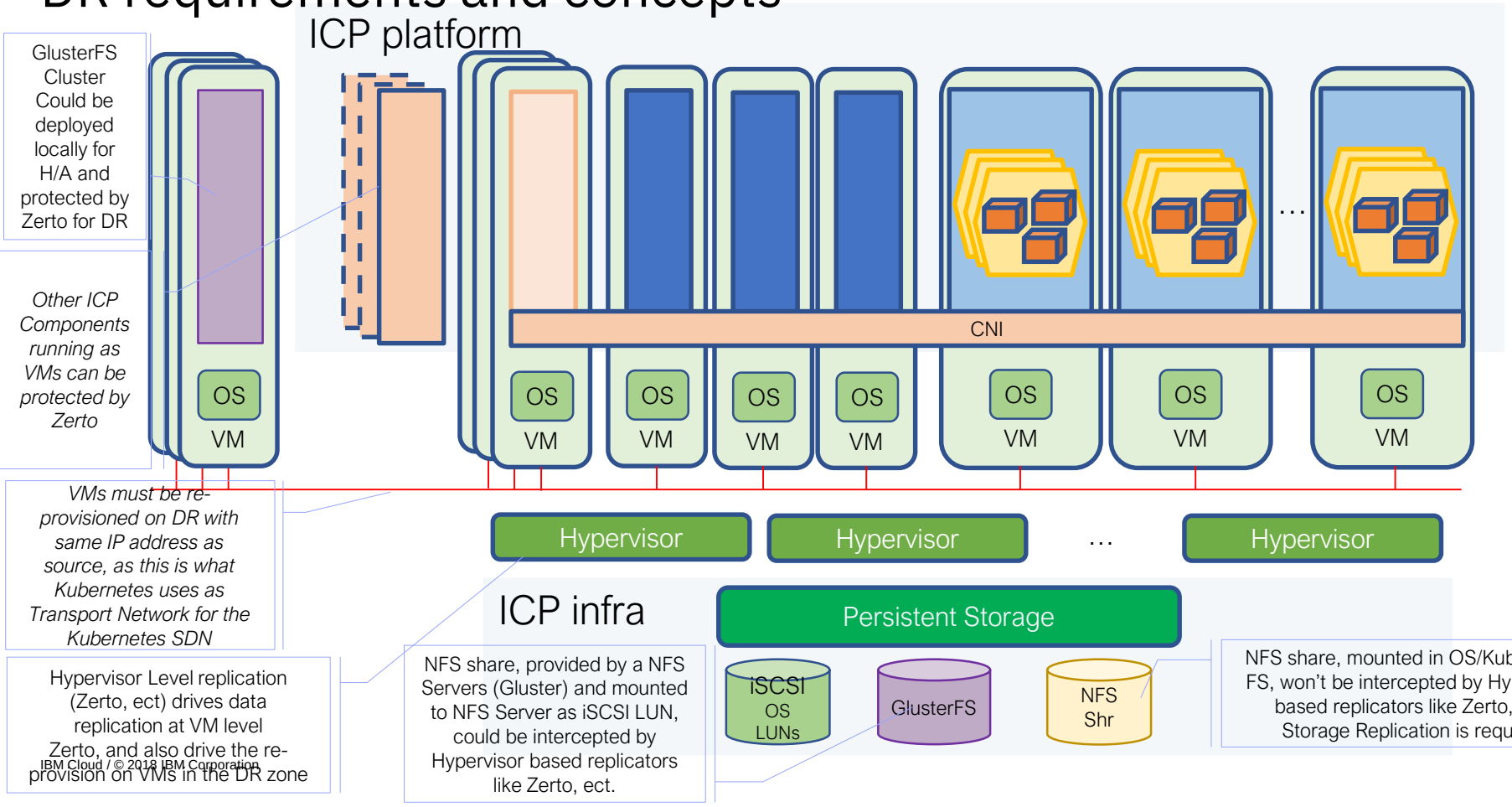
- Restore the failed Master
- Make sure etcd (and kubelet) has stopped on (all) the masters
- Take snapshot of the etcd from one of the masters that was running
- Purge etcd data of the restored master
- Copy etcd snapshot to restored master node
- Restore the snapshot on restored master nodes
- Reconnect etcd (and kubelet)
- Validate etcd cluster health
- Start the rest of the ICP cluster pods
- Validate the configuration

Reference: https://github.com/ibm-cloud-architecture/icp-backup/blob/master/docs/etcd.md#etcd-restore-on-multi-master-icp-configuration

Reference: https://github.com/coreos/etcd/blob/master/Documentation/op-guide/recovery.md

DR aspects

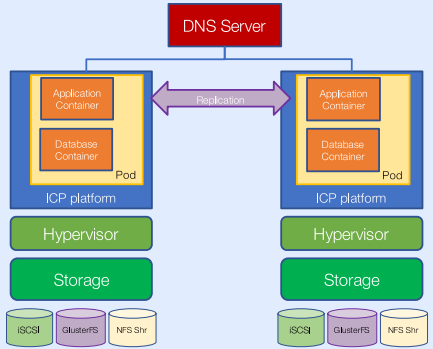
DR requirements and concepts



DR Models

Transaction Consistency

(what-ever has been processed as a transaction is safe)

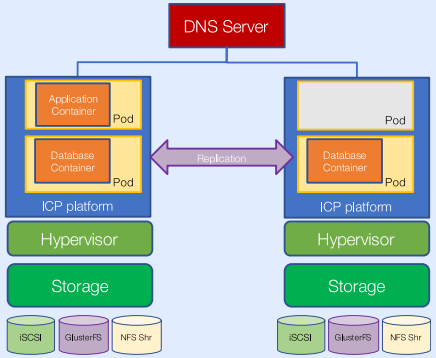


1. Application Level

- 2 ICP platform
- Application Containers active and deployed on both ICPs
- Application takes care of DR

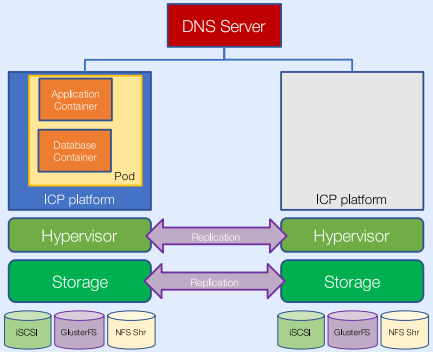
2. Database Level

- 2 ICP platform
- Application Container active only on one ICP
- Pod config deployed on both ICPs
- DB takes care of DR



Technology Consistency

(what-ever has been secured on disk is safe)

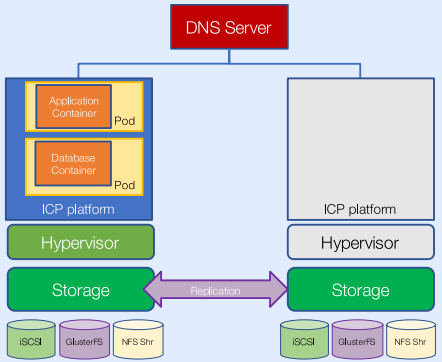


3. Hyper-Visor / VM level

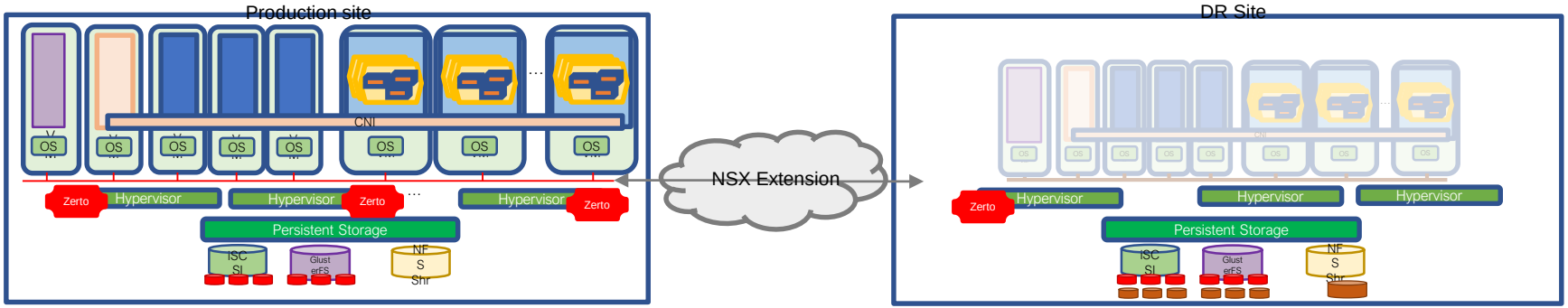
- 1 ICP platform
- Duplicated infrastructure
- all application and config data moved with VMs
- Replication/Backup
  - iSCSI: of all VMs at hyper-visor level (ie: Zerto, Veeam)
  - NFS: requires a Storage replication

4. Storage Level

- 1 ICP platform
- Storage with Replication Function
- all application and config data moved with Data replicated
- DR Server on-Demand
- Boot Hyper-Visor from replicated data



DR Scenarios – Option 3 – DR setup with ZeRTO



Zerto — VM OS Disk updates are intercepted in real time and asynchronously replicated to the DR site

GlusterFS iSCSI LUNs updates are intercepted in real time and asynchronously replicated to the DR site

Application data, stored on Persistent storage, are replicated sync/async on the DR Site via Storage Replication functions:
- Endurance / Performance replication – Asynch, snapshot replication
- vSAN – sync/async, continuous replication
- NFS Server/Storage replication function

Zerto — Only the Zerto replication is active on a minimal infrastructure required to sustain the replication and to drive day by day operations.
Source VM OS Disk updates are stored in the iSCSI disks in the DR site

Source GlusterFS iSCSI LUNs updates are stored in the iSCSI disks in the DR site

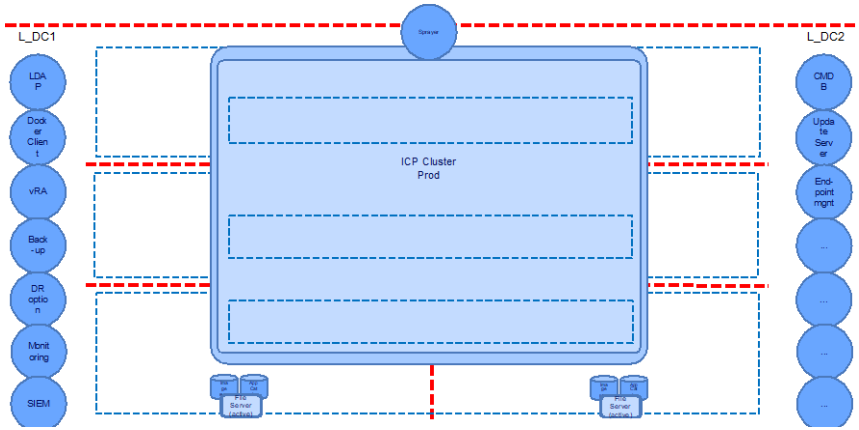
In the DR site, a copy of the replicated data is stored in the corresponding repository

NSX Extension is responsible to allow communications amongst the sites.
NSX in each site is responsible for maintaining the network virtualization layer in each site
NSX in DR site must allow to access the DR Test environment from users, pointing to the DR Test (VPN-NAT, …)

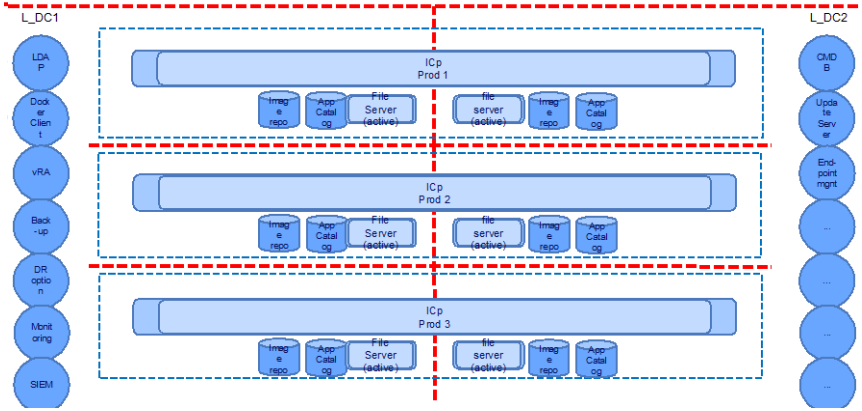
Cluster topologies

Cluster segregation (HA/DR)

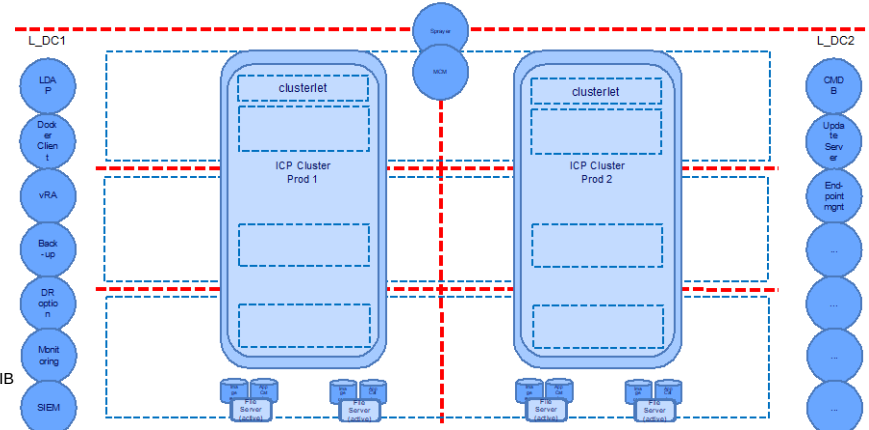
Dual DC – Stretched with security zone isolation



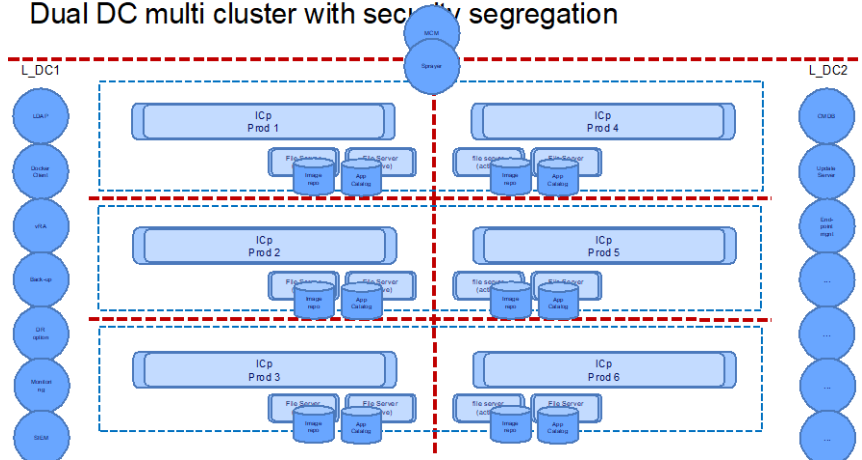
Dual DC – Stretched but segregated by security zone



Dual DC – Multi cluster with security zone isolation

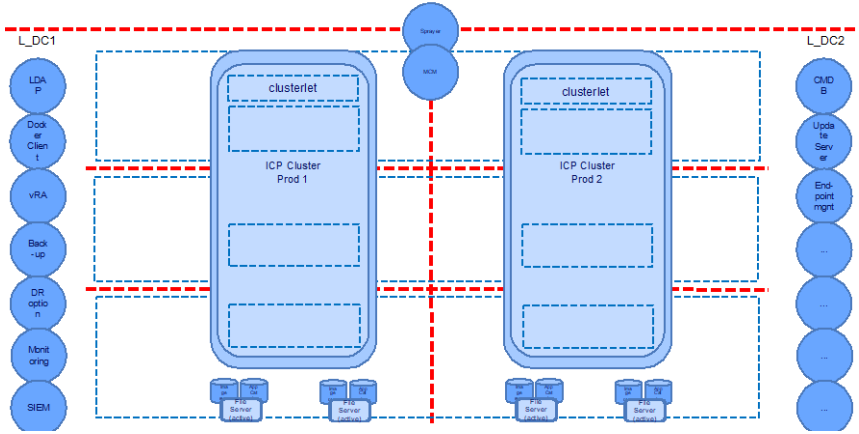


Dual DC multi cluster with sec[illegible]y segregation

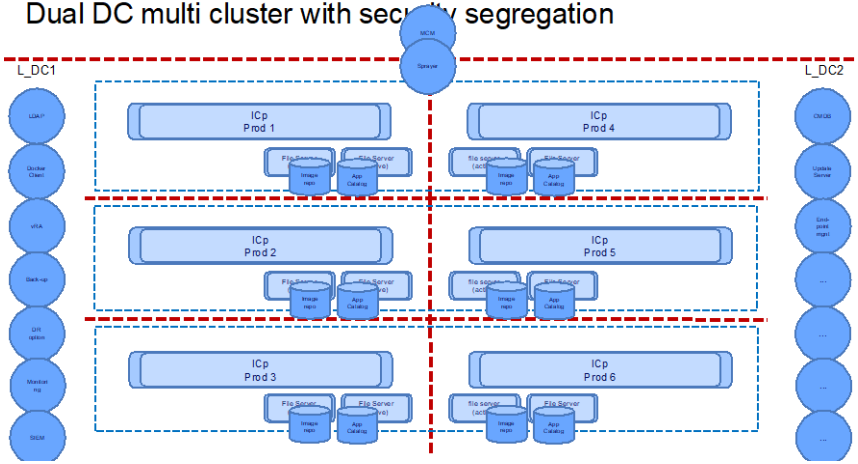


Isolate intra or inter cluster

Dual DC – Multi cluster with security zone isolation

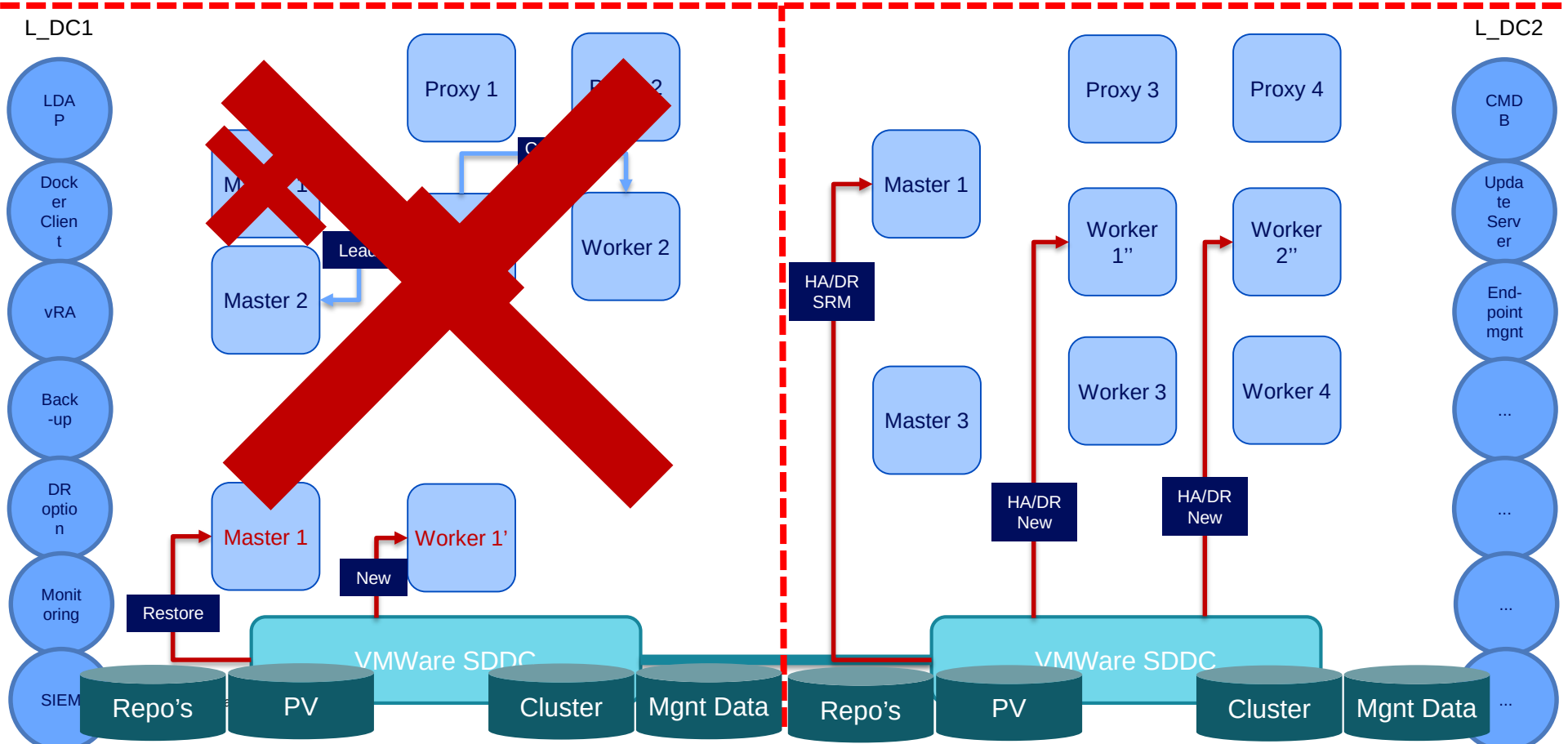


Dual DC multi cluster with security segregation

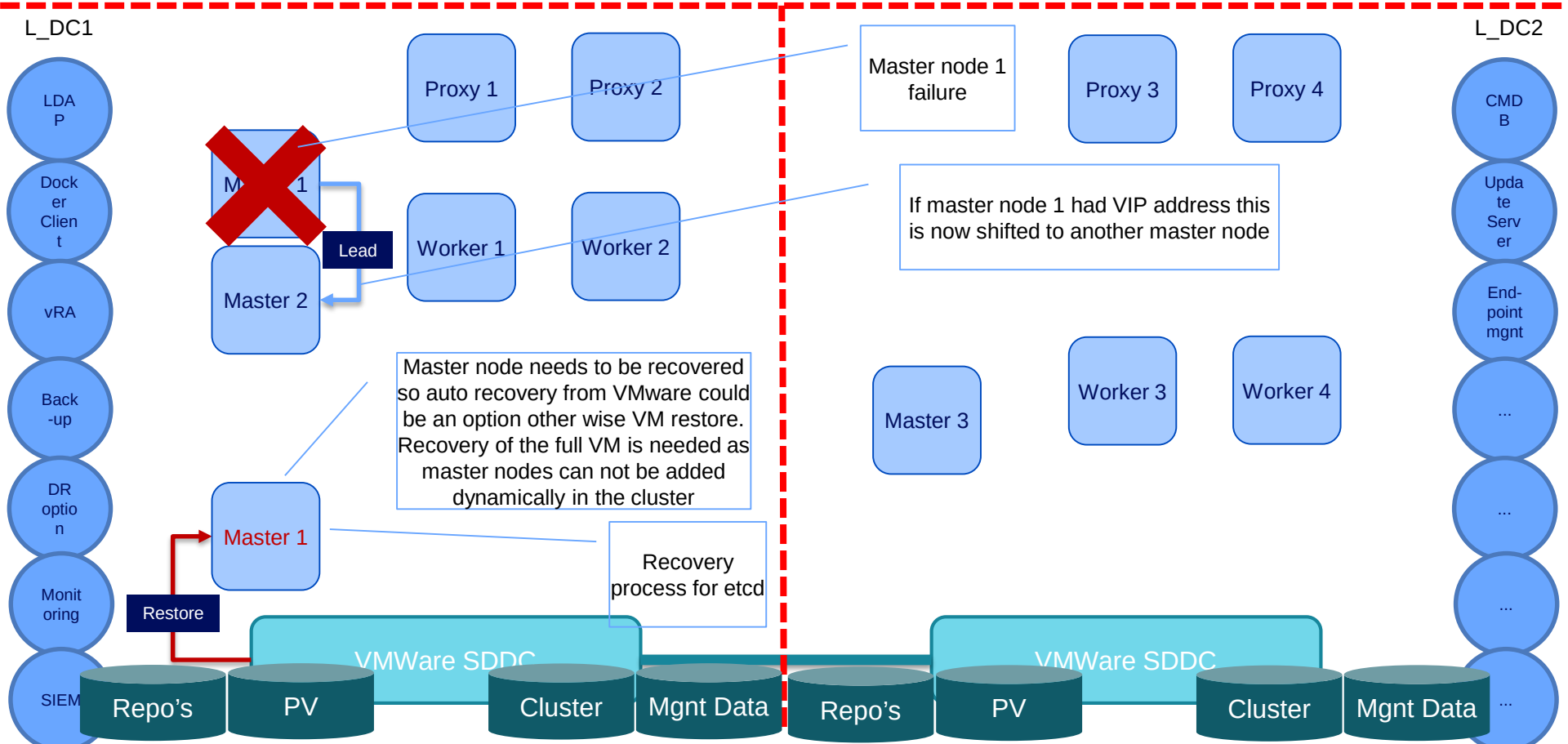


What are the policies? Exceptions? Zoning and networking? Service management? integration?

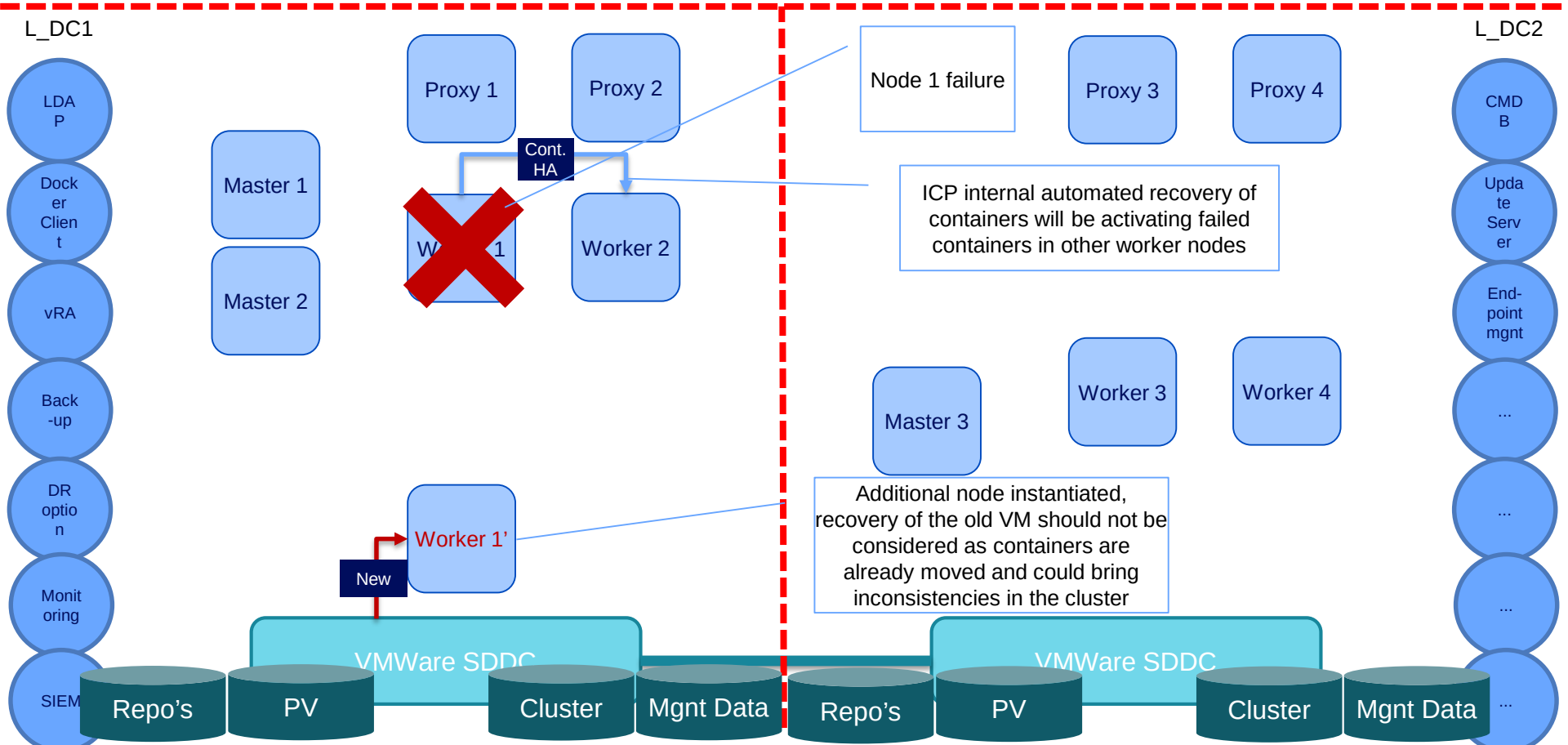
Dual DC with stretched cluster approach



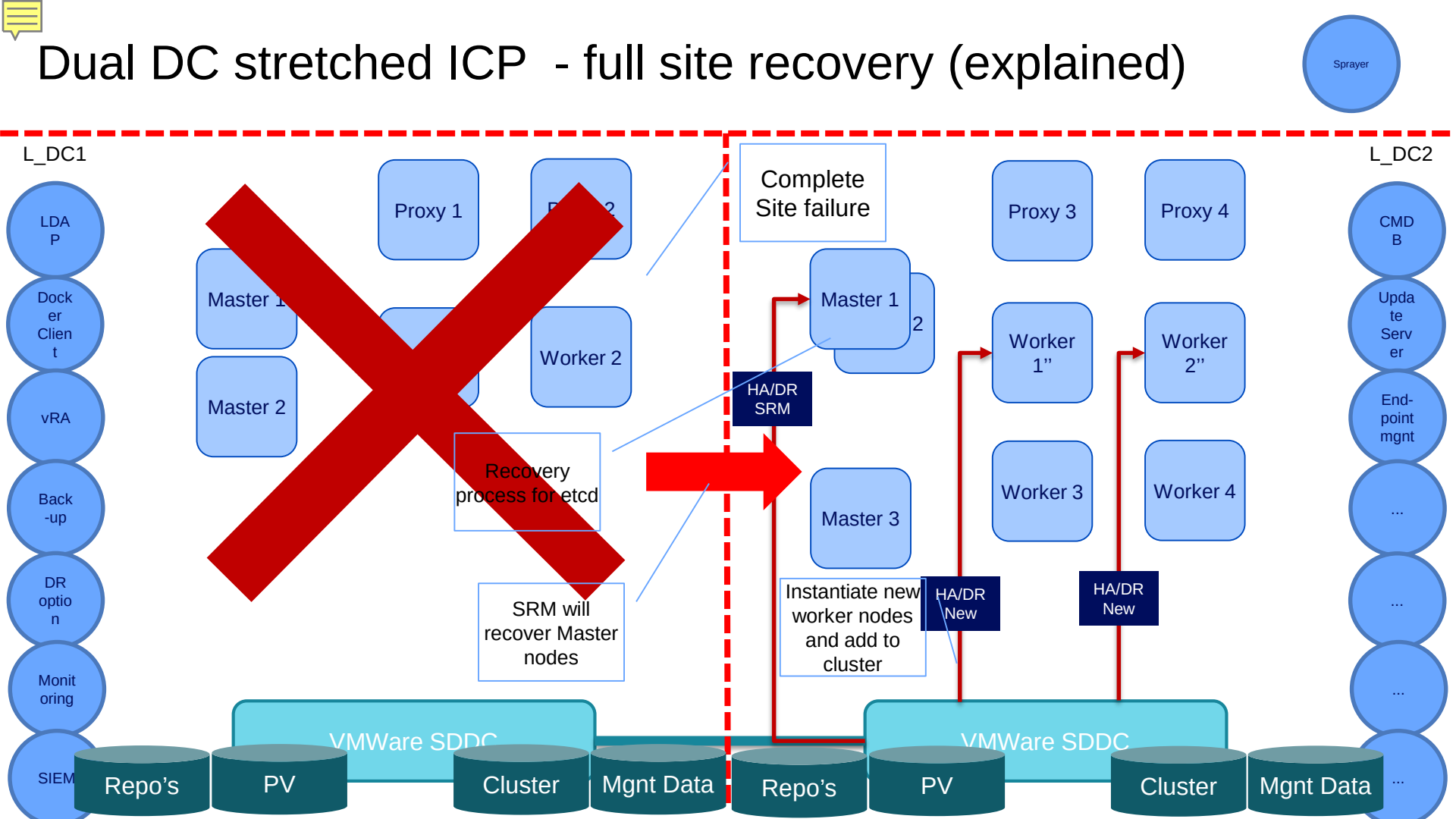
Dual DC stretched ICP – master node failure local recovery (explained)



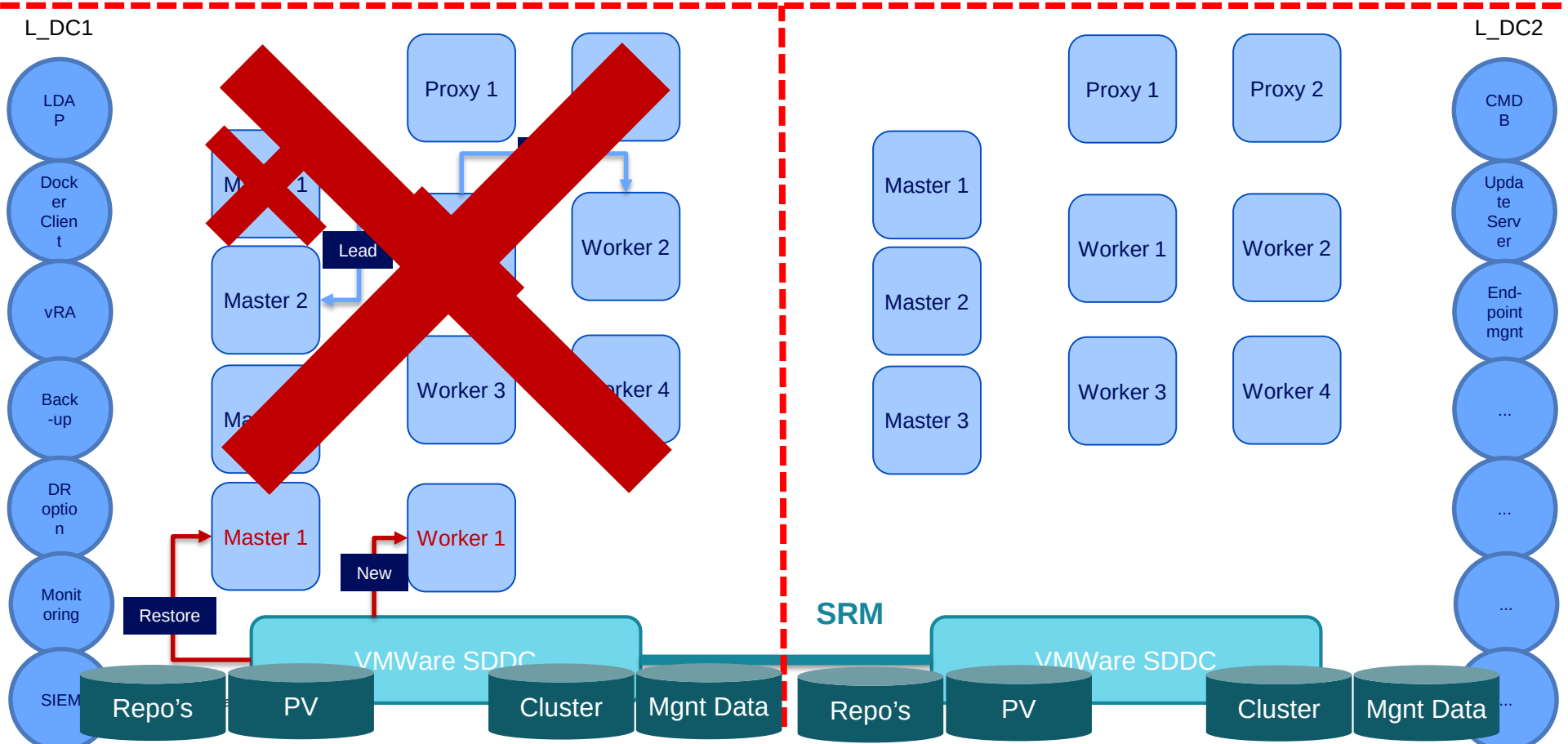
Dual DC stretched ICP – node failure local recovery (expla



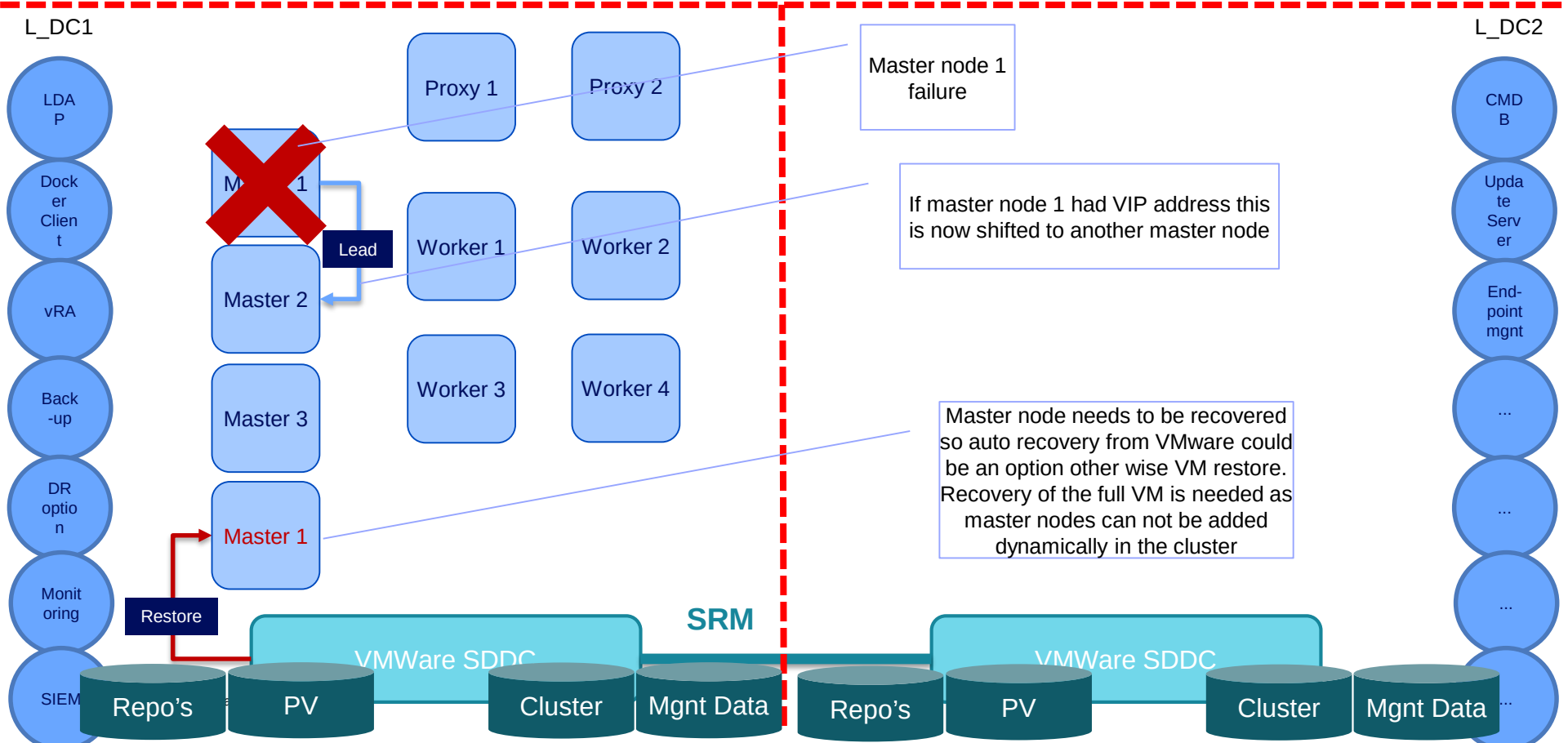
Dual DC stretched ICP - full site recovery (explained)



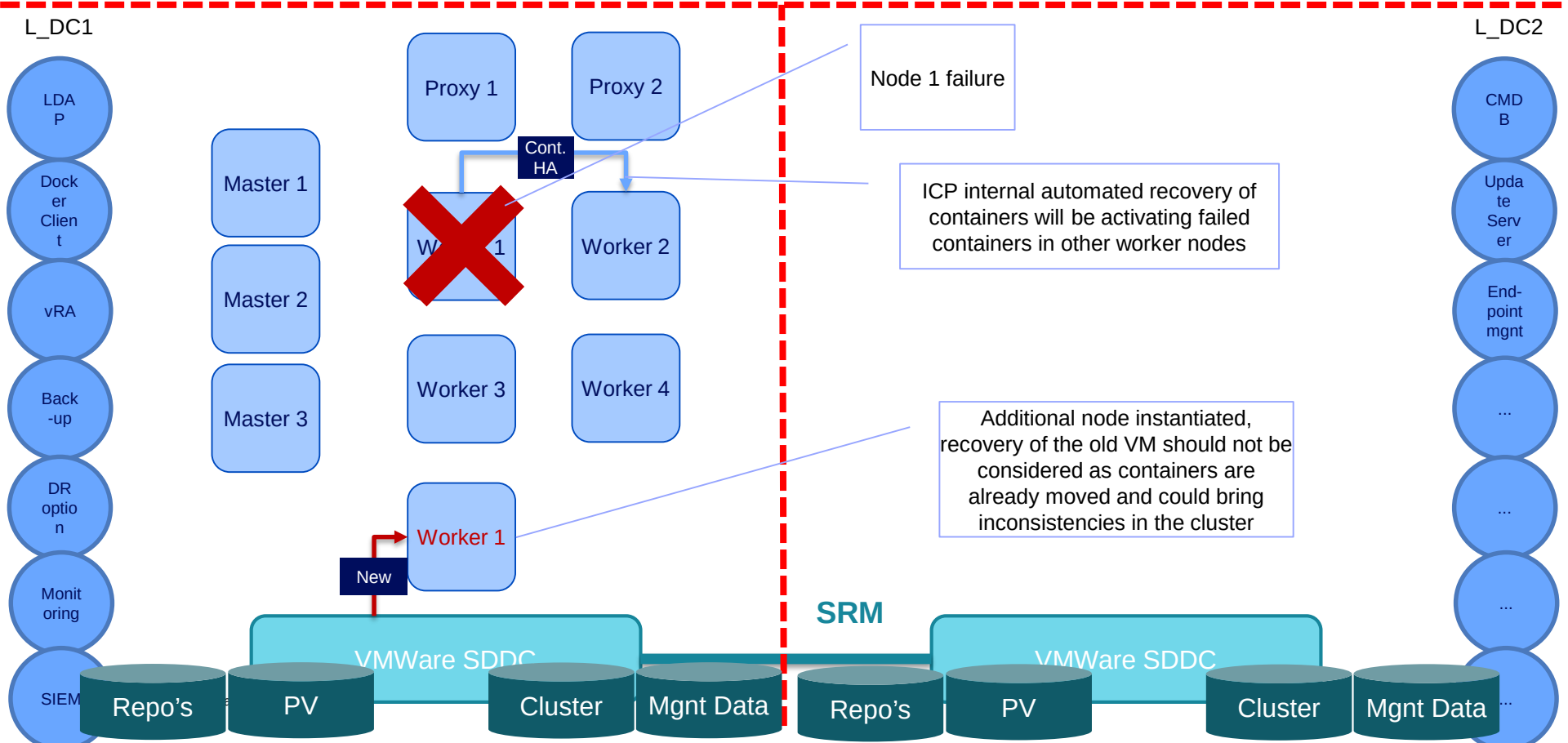
Dual DC DR with SRM to secondary DC



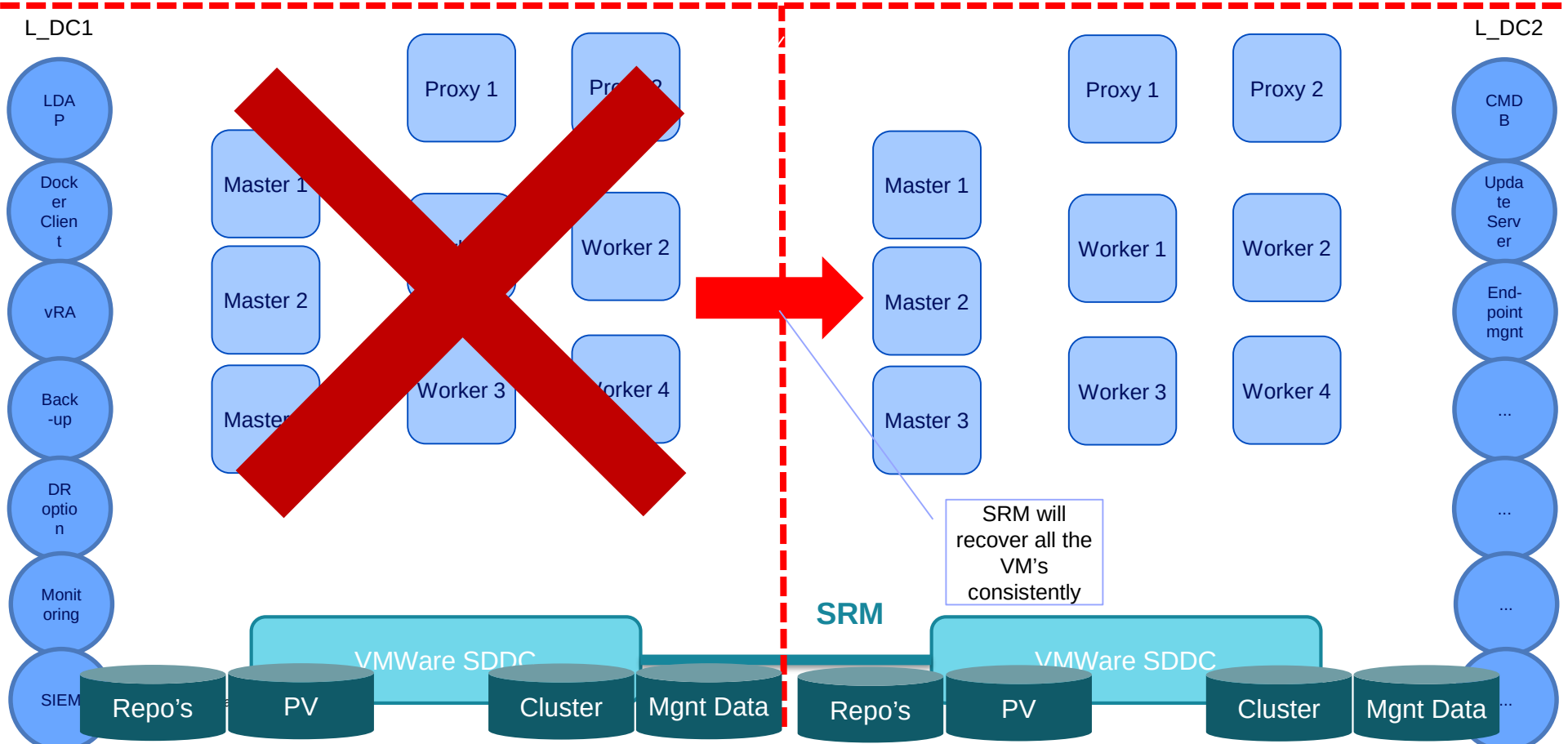
Dual DC SRM DR – master node failure (Explained)



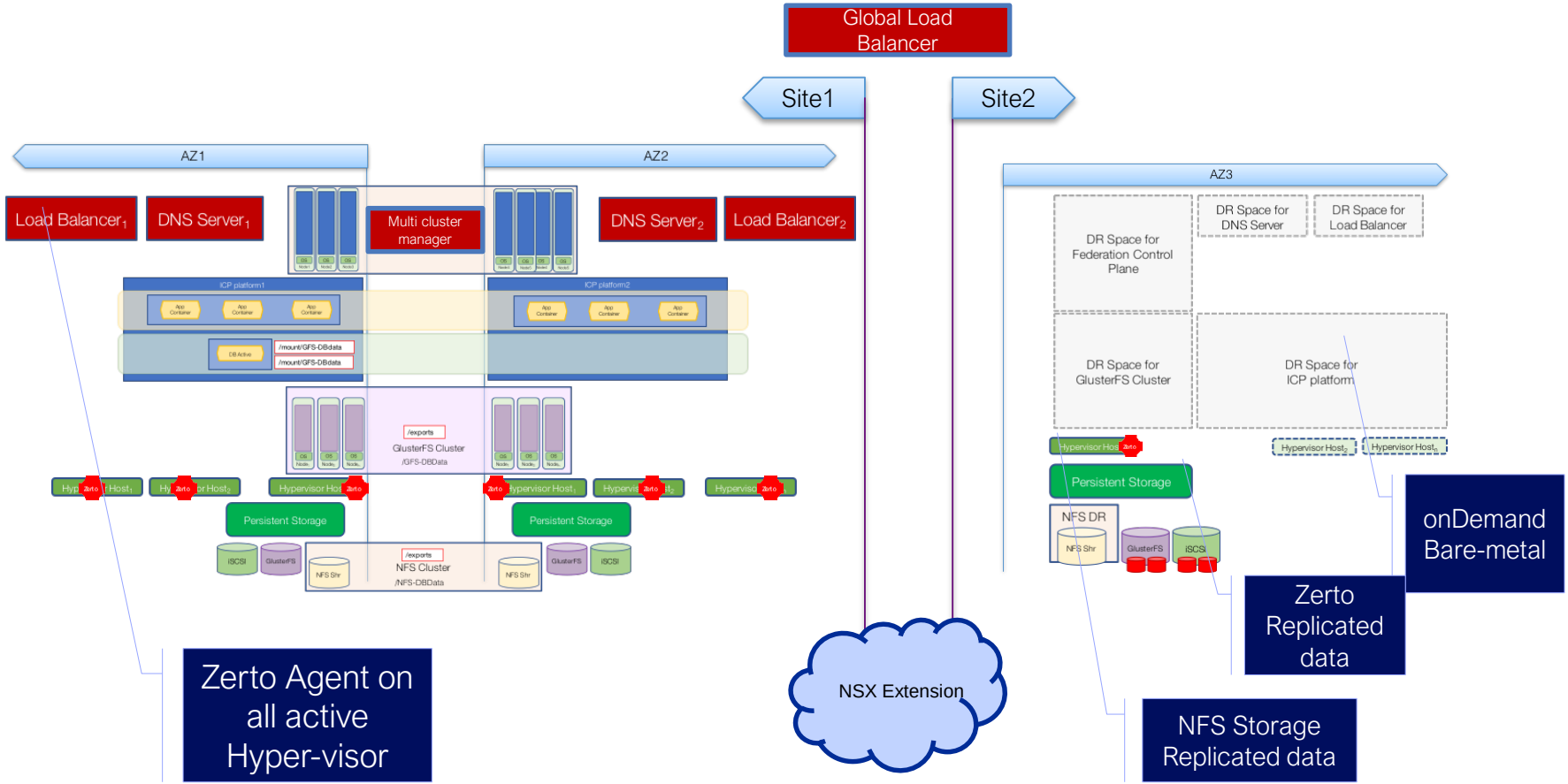
Dual DC SRM DR – master node failure (Explained)



Dual DC SRM DR – master node failure (Explained)



Combining HA & DR



Important aspects for HA&DR

Workload and Application considerations

What availability is needed for the application?

RPO and RTO for the application

Container considerations

How is the traffic sprayed over the HA components?

Do we allow DR for individual applications or only for the cluster?

Platform Considerations

Are we understanding the capacity needs? Do we need N+1 or N+2?

What tooling is foreseen to replicate the data between prod and DR?

Important aspects for HA&DR

Infrastructure considerations

How is the HA foreseen for the infrastructure?

Do we have a dark DR site or an active one?

Organization considerations

Who is managing and monitoring the platform for availability?

Do we apply regular switches between prod and DR?