High Availability solutions with Linux on System z

Wilhelm Mild
IT Architect
IBM Germany

Trademarks

- This presentation contains trade-marked IBM products and technologies. Refer to the following Web site:

  http://www.ibm.com/legal/copytrade.shtml

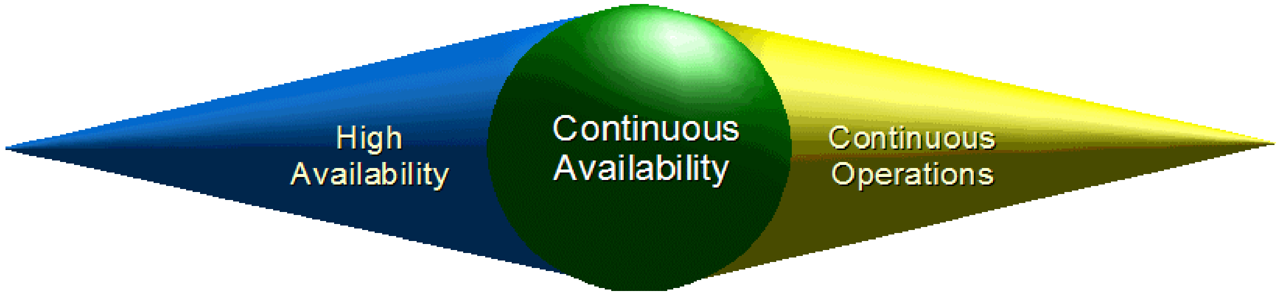
Definitions

- High Availability (HA) – Provide service during defined periods, at acceptable or agreed upon levels, and masks *unplanned* outages from end-users. It employs Fault Tolerance; Automated Failure Detection, Recovery, Bypass Reconfiguration, Testing, Problem and Change Management

- Continuous Operations (CO) -- Continuously operate and mask *planned* outages from end-users. It employs Non-disruptive hardware and software changes, non-disruptive configuration, software coexistence.

- Continuous Availability (CA) -- Deliver non-disruptive service to the end user 7 days a week, 24 hours a day (there are no planned or unplanned outages).

y.



Achieving Continuous and High Availability of IT services
- requires a comprehensive long term commitment
- requires continuous improvement.

- **The effort of an IT organization for high availability varies depending on maturity**.
  - Failures across people, process, and technology can inhibit high availability
  - Therefore all potential gaps and inhibitors must be addressed
  - High availability results from doing many things controlled

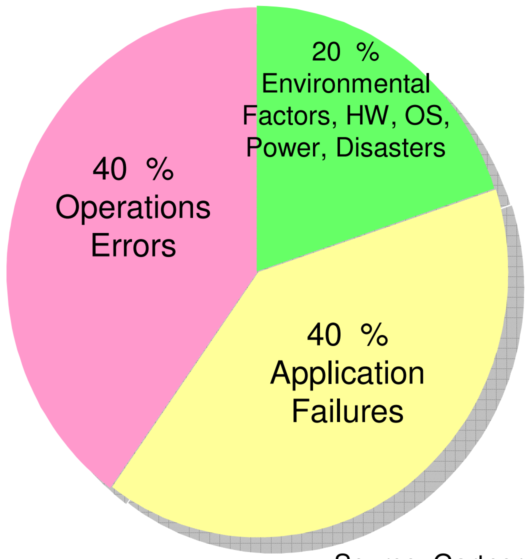
- **High Availability typically requires an one time investment, with ongoing improvements**
  - Initial Assessment, business and process requirements
    - identify gaps, and build a strategic road map.
    - Priority and Dependencies
  - Identify and prioritize key initiatives to improve availability
    - System, application, and data architecture
    - Required support structure and skill development
    - Processes, procedures, and methods supporting High Availability
  - Design, develop, implement, and enhance processes, procedures, methods, technology, tools, IT applications, and skills.

Assess	**Plan**	**Define**	**Design**	**Build**	**Test**	**Deploy**	**Run**

Business Continuity Issues

What are the reasons for system outages?

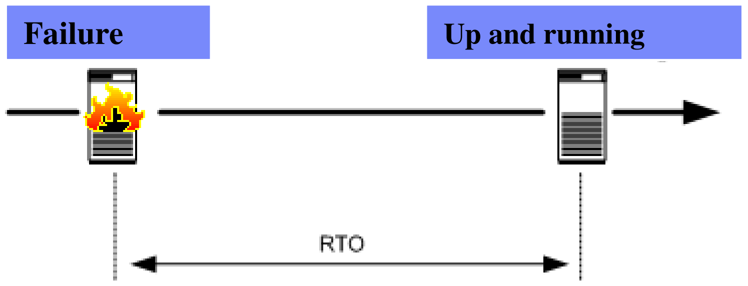
- **Planned** outages
  - Maintenance
  - Tests
- **Unplanned** outages
  - Operator errors
    - Lack of application skills
    - Lack of OS skills in heterogeneous environment
  - Application failures
    - SW exceptions
    - Environment / Configuration problems
  - Environmental failures
    - OS failures
    - HW failures
    - …
    - Disasters

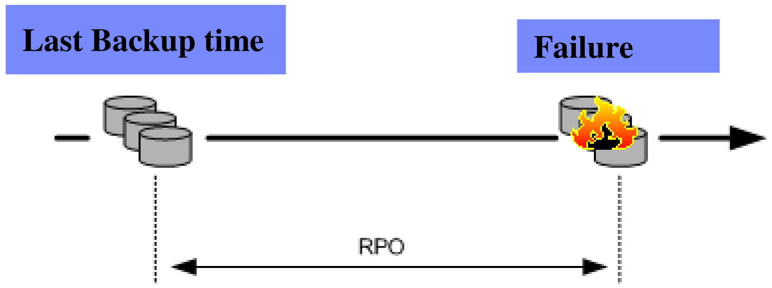


Source: Gartner Group

Identify RTO, RPO und NRO

Business Resiliency Plan





Recovery **T**ime **O**bjective (**RTO**)

What time difference can be between Failure and a total productional run level ?

Recovery **P**oint **O**bjective (**RPO**)

What is the toleration for data loss?

RPO = “0” means, NULL data loss acceptable
RPO = “5” means, data loss in last 5 min acceptable

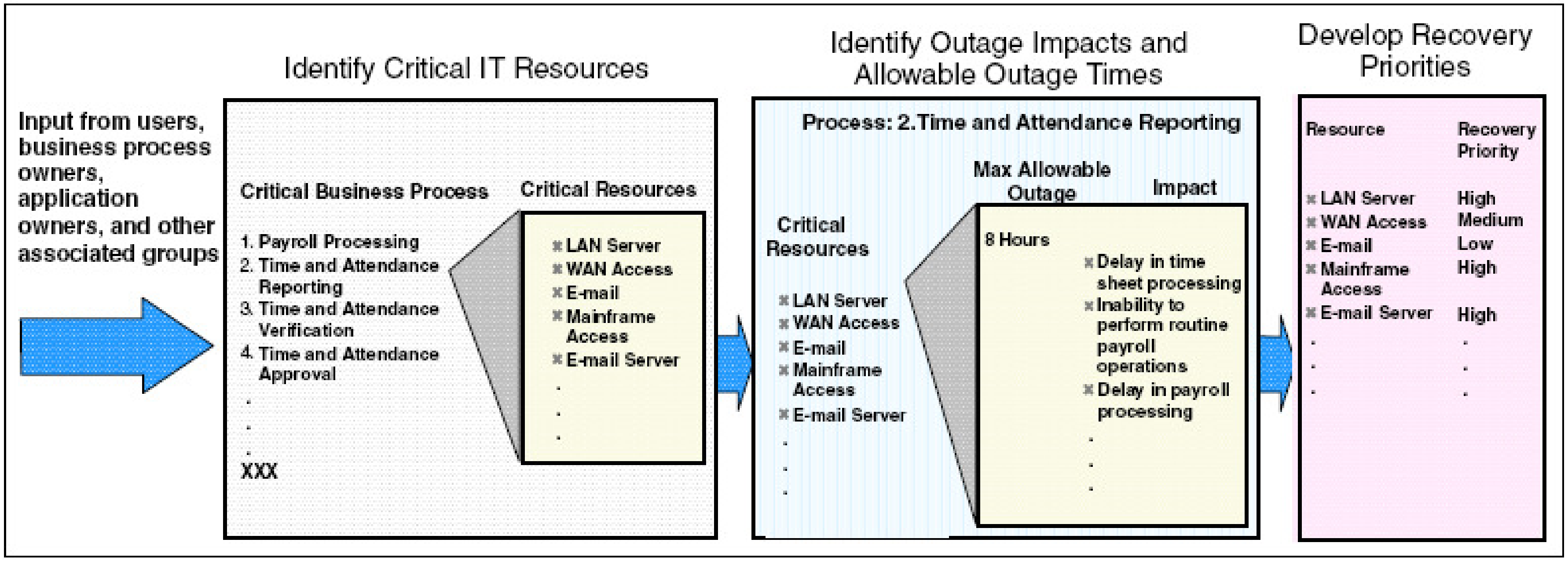
TREND: RPO = 0

Network **R**ecovery **O**bjective (**NRO**)

Time requirements for network availability.

The Business Impact Analysis (BIA)

- IT Resource relation and priorities for DR
- Consider all environments
- Prioritize based on business importance


Identify Critical IT Resources
Identify Outage Impacts and Allowable Outage Times
Develop Recovery Priorities
Input from users, business process owners, application owners, and other associated groups
Critical Business Process
1. Payroll Processing
2. Time and Attendance Reporting
3. Time and Attendance Verification
4. Time and Attendance Approval
XXX
Critical Resources
LAN Server
WAN Access
E-mail
Mainframe Access
E-mail Server
Process: 2.Time and Attendance Reporting
Critical Resources
Max Allowable Outage
Impact
8 Hours
Delay in time sheet processing
Inability to perform routine payroll operations
Delay in payroll processing
Resource
Recovery Priority
LAN Server High
WAN Access Medium
E-mail Low
Mainframe Access High
E-mail Server High

Example of the Business Impact Analysis process

The Business Impact Analysis (BIA)

- The **BIA/RPO/RTO** analysis stage focuses on organizing business requirements in such a way that systems and data can be classified by relative importance.
- The systems classification is translated into technical classifications and policies. While these elements are often included in the risk analysis phase, we separate them to stress the importance of strategic planning and classification of systems.
- In addition to the **BIA data**, the following elements must be fully understood for each system:
  - **Recovery Point Objective (RPO)** defines the amount of data that you can afford to recreate during a recovery, by determining the most recent point in time for data recovery.
  - **Recovery Time Objective (RTO)** is the time needed to recover from a disaster, or how long the business can survive without the systems.
  - **Network Recovery Objective (NRO)** details the time to recover or failover network operations. Keep in mind that the systems level recovery is not fully complete if customers cannot access the application services via network connections. For instance, in a WAN environment where data processing services are transitioned from site A to the recovery site B, numerous problems can exist outside of the local network configuration, including DNS and routing

Differences between HA and DR

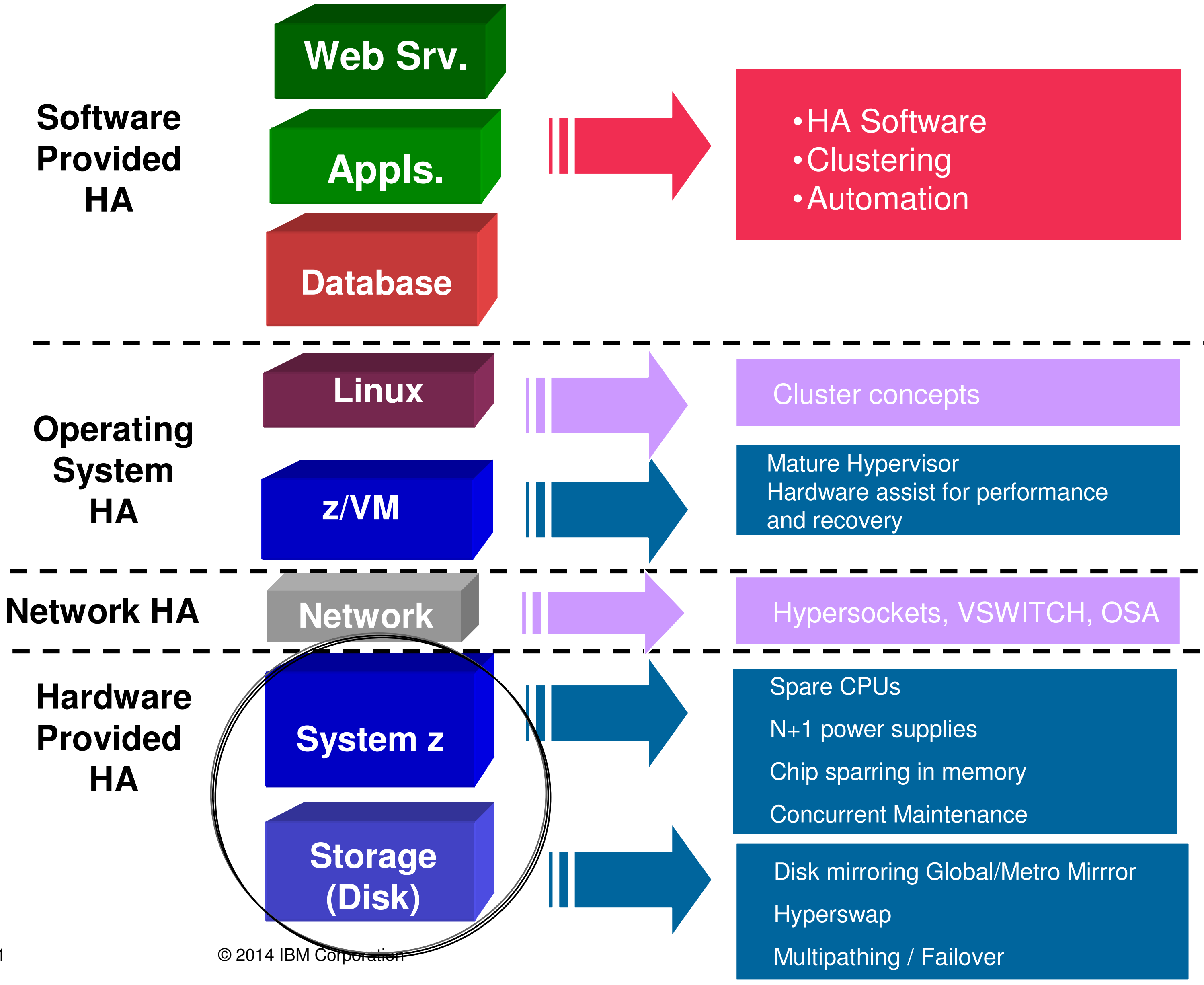
- **High Availability - HA:**
  - Failover is typically realized via duplication and clustering
  - Failover times measured in seconds and minutes
  - Reliable synchronous inter-node communication

- **Disaster Recovery - DR:**
  - Failover is typically realized with 2 or more sites in case of disasters
  - Failover times often measured in minutes and hours
  - Unreliable inter-node communication assumed

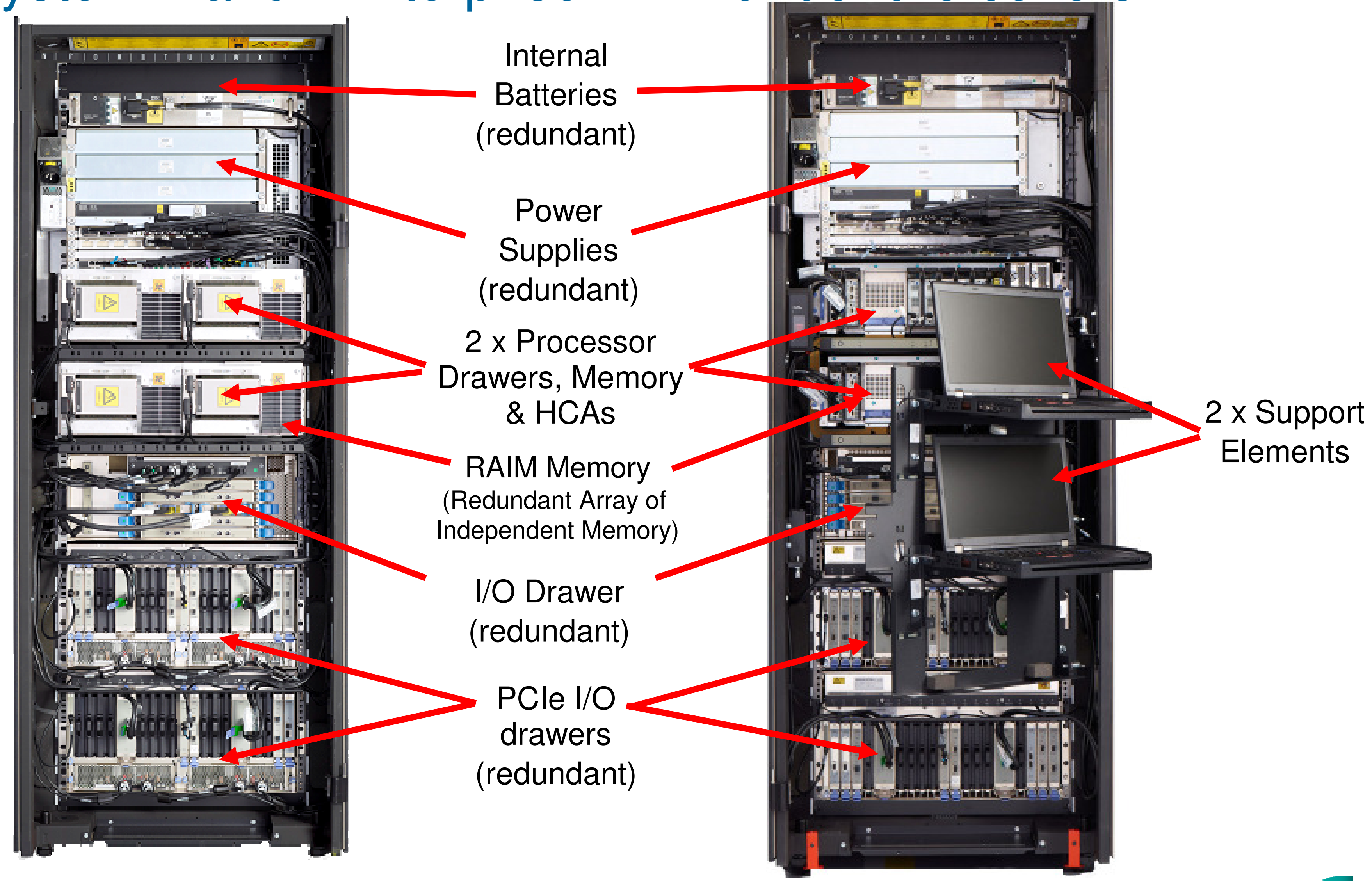
Fundamentals of High Availability

- **Redundancy**, Redundancy, Redundancy
  - Duplicate to eliminate single points of failure.
- **Early detection**
  - To keep offline time as short as possible
  - Reduce risk of wrong interpretation and unnecessary failover
  - Keep offline time as short as possible (mean-time-to-repair MTTR)
- **Protect Data Consistency** – Provide ability for data and file systems to return to a point of consistency after a crash.
  - Journaling databases
  - Journaling file systems
  - Mirroring
  - Routine database backups
- **Automate Detection and Failover -** Let the system do the work in order to minimize outage windows.
  - Multipath
  - VIPA –Virtual IP Addresses
  - Monitoring and heart-beating
  - Clustered middleware
  - Clustered operating systems

Components of HA with Linux on System z

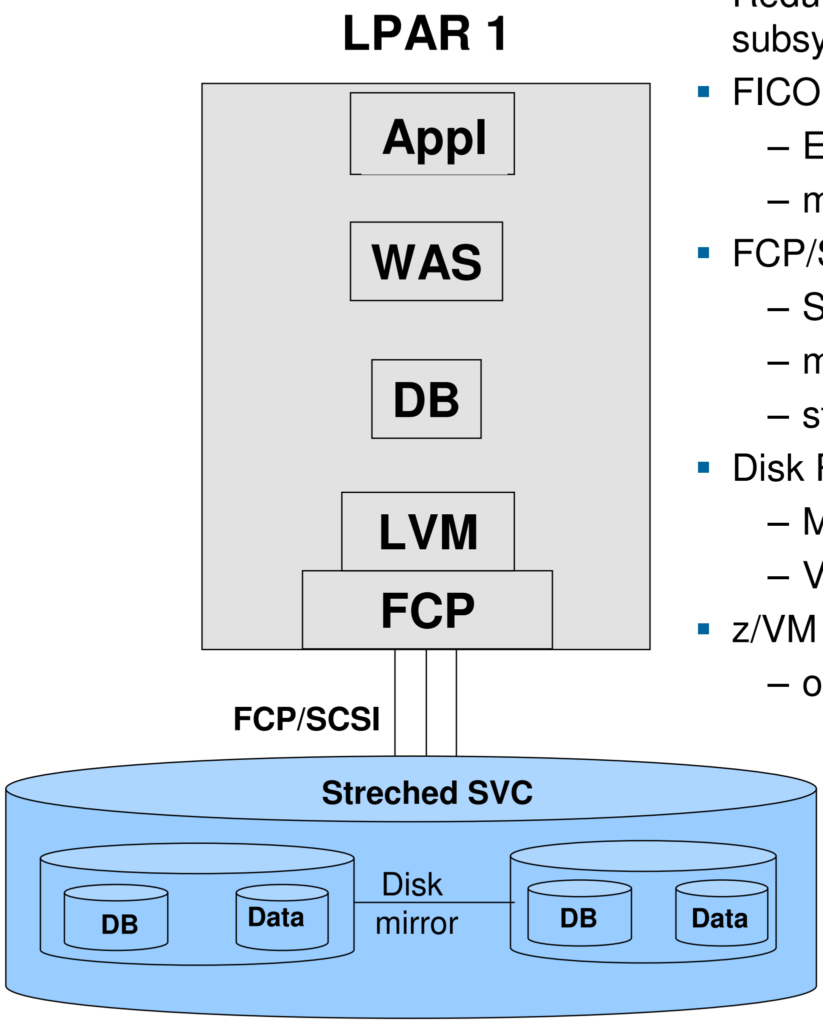


System z and zEnterprise – HA under the covers

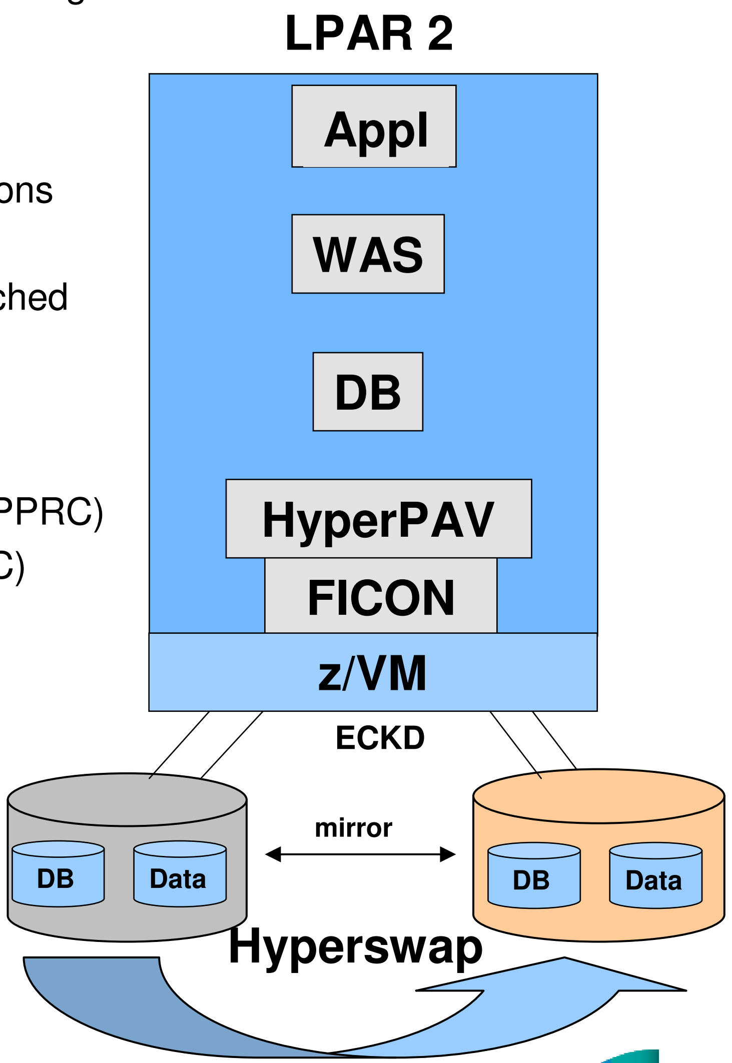

Internal Batteries (redundant)
Power Supplies (redundant)
2 x Processor Drawers, Memory & HCAs
RAIM Memory (Redundant Array of Independent Memory)
I/O Drawer (redundant)
PCIe I/O drawers (redundant)
2 x Support Elements

Rear View **Front View**

Storage HA Options



- Redundant access to the Storage subsystems
- FICON attached Disks
  - ECKD disks
  - multi channel connections
- FCP/SCSI attached disks
  - SCSI disks - LUN attached
  - multi pathing /LVM
  - streched SVC for HA
- Disk Replication -
  - Metro / Global Mirror (PPRC)
  - Virtual disk mirror (SVC)
- z/VM Hyperswap
  - online disk swap

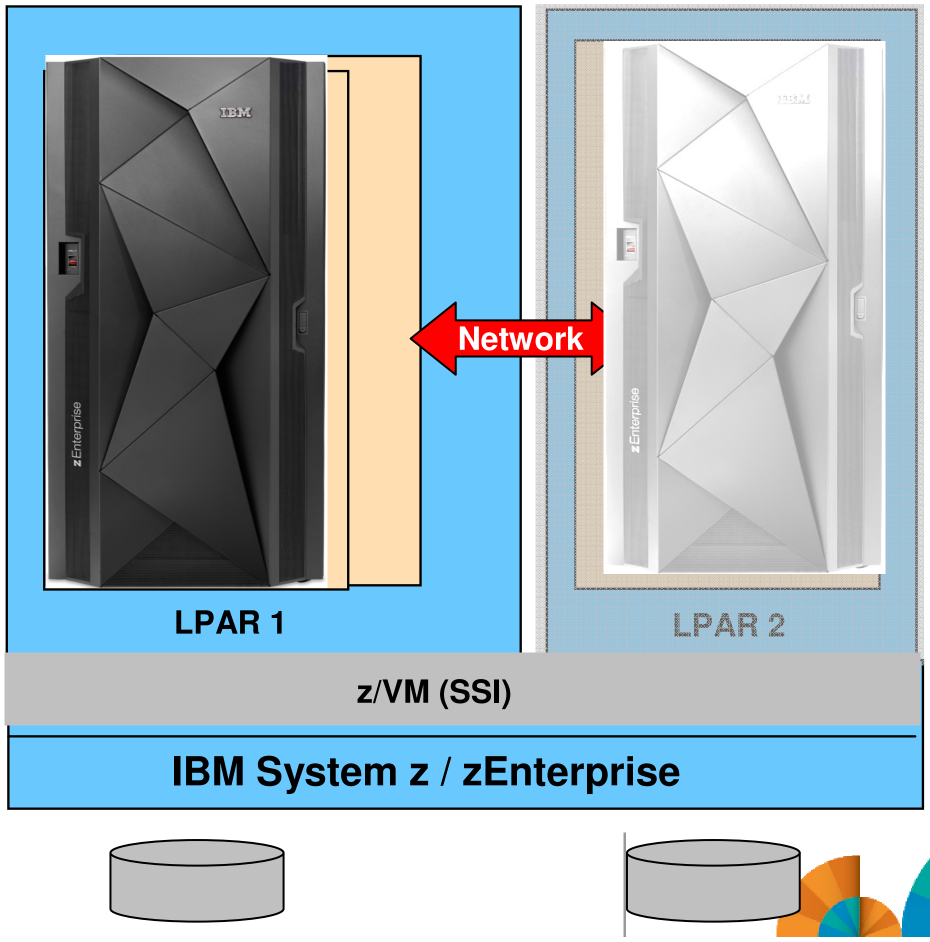


High Availability scenario as Active/Passive with System z

- **Active / Passive Deployment**.
  - Workload normally contained at Site 1, standby server capability at Site 2
  - Primary and secondary disk configurations active at both sites.
  - During fail over, Capacity Upgrade on Demand (CUoD) adds resources to operational site, and standby servers are started. Helps save hardware and software costs, but requires higher recovery time.

- **Hot / Cold scenario**
  - Workload is not split.
  - Each site is configured to handle all operations
  - Cold environment needs longer to get active – often used in DR

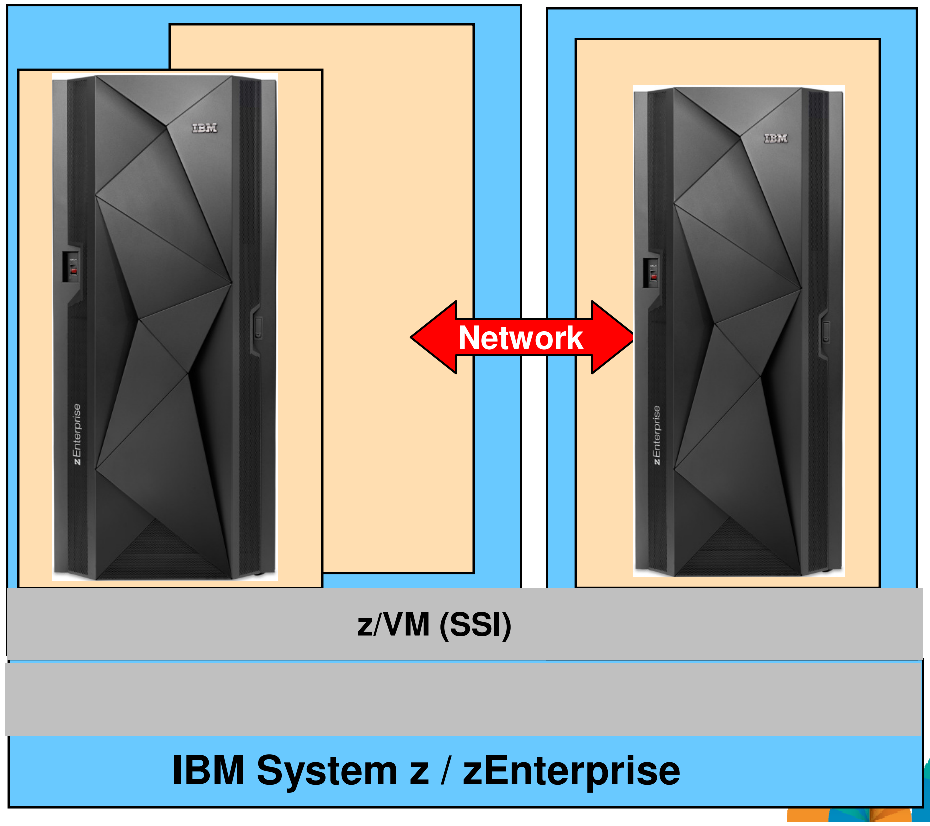
- **Hot / Warm scenario**
  - Workload is not split
  - Each site is configured to handle all operations
  - Warm environment is idling.



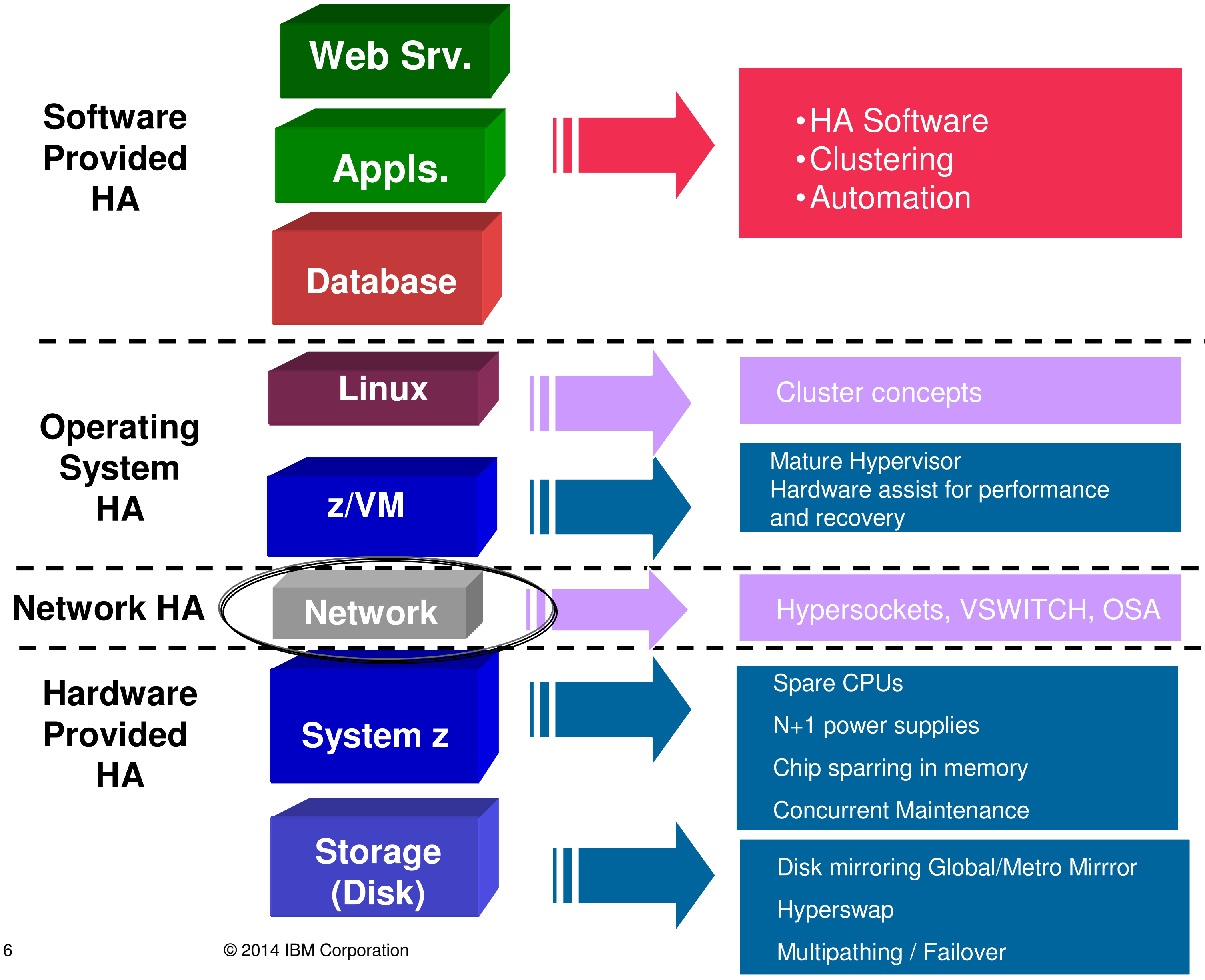
High Availability with an active/active environment on System z

- **Active / Active Deployment -Expendable work.**
  - Workload is normally split between 2 or more sites
  - Each site is (over) configured to be able to instantly cover the workload if needed.
  - During normal operation, excess capacity at each site is consumed by lower priority, work like development or test activities
  - In a failover situation, low priority work is stopped to free up resources for the production site's incoming work.

- **Capacity Upgrade on Demand (Active / Active)**
  - Workload is normally split between sites.
  - Each site is configured with capacity to handle normal operations
  - Special setup with Capacity Upgrade on Demand (CUoD).
  - In a failover situation, additional CPUs are enabled at the operational site.



Components of HA with Linux on System z



- HA Software
- Clustering
- Automation

Cluster concepts

Mature Hypervisor
Hardware assist for performance and recovery

Hypersockets, VSWITCH, OSA

Spare CPUs

N+1 power supplies

Chip sparring in memory

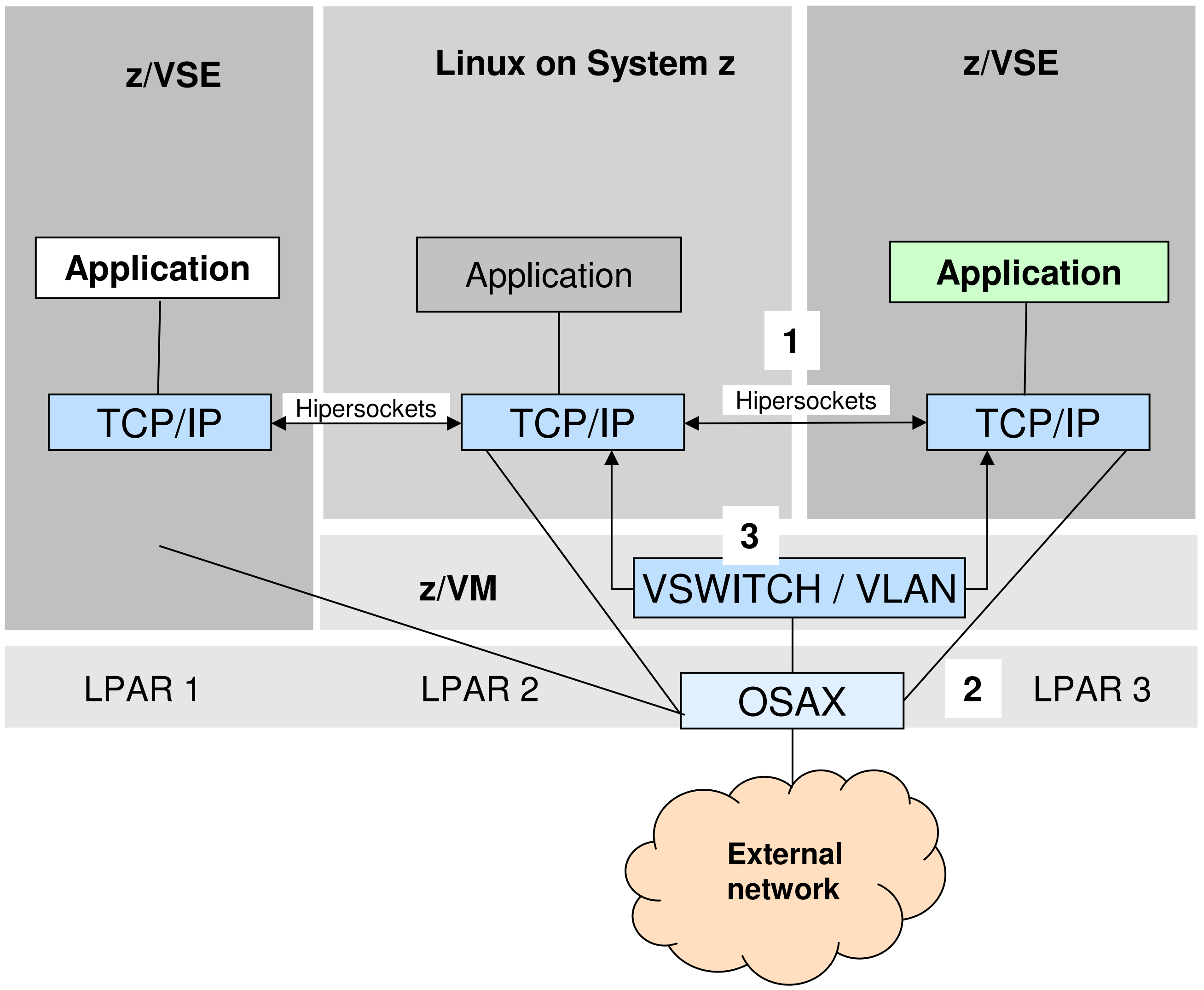
Concurrent Maintenance

Disk mirroring Global/Metro Mirrror

Hyperswap

Multipathing / Failover

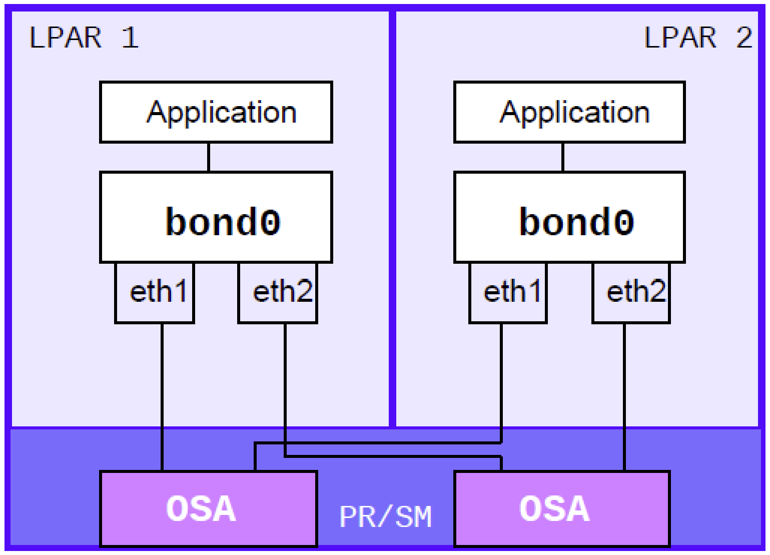
Linux and network alternatives in System z



Network Interface HA and Automated Failover

OSA-card HA

Channel Bonding for enhanced bandwidth

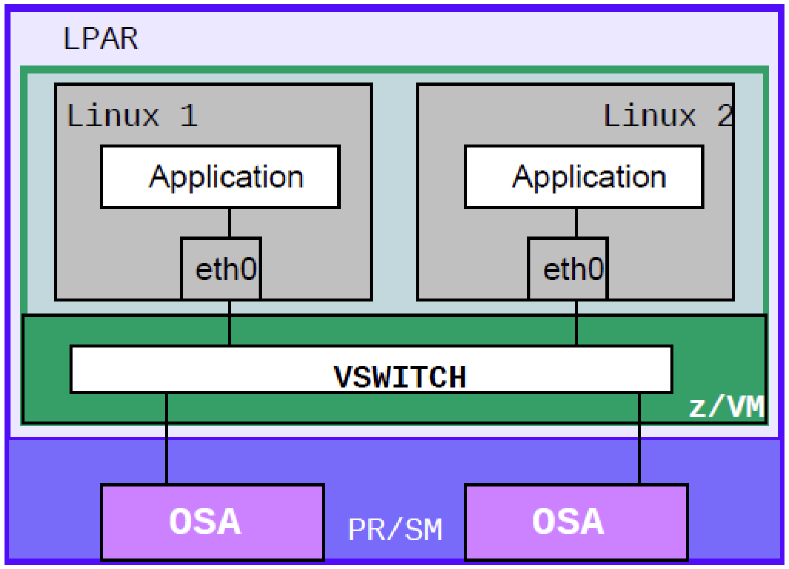


- Linux *bonding* driver enslaves multiple OSA connections to create a single logical network interface card (NIC)
- Detects loss of NIC connectivity and automatically fails over to surviving NIC
- Active/backup & aggregation modes
- **Separately configured for each Linux**

Network HA with

z/VM VSWITCH

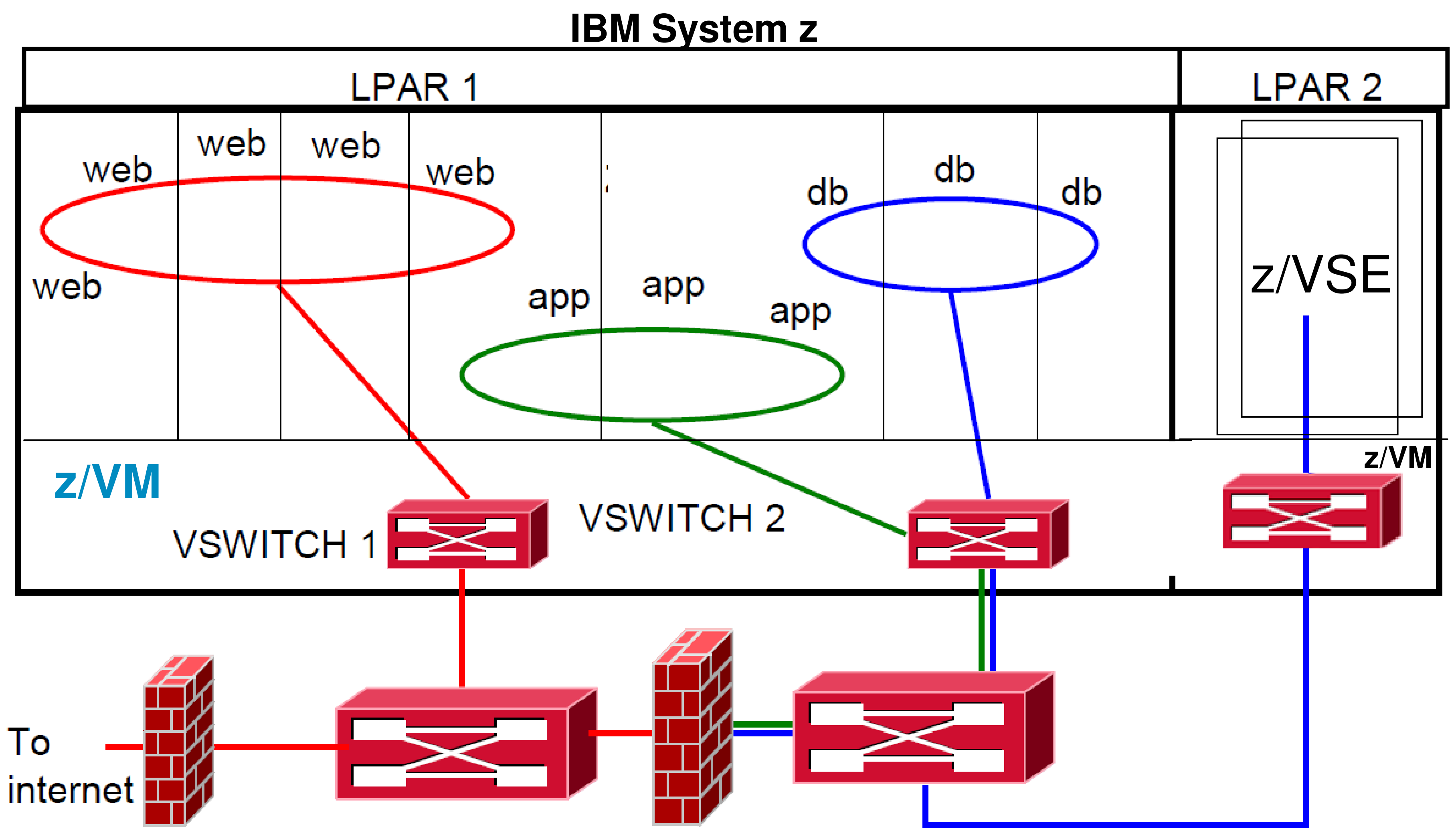
Port aggregation for enhanced bandwidth



- z/VM *VSWITCH* enslaves multiple OSA connections. Creates virtual NICs for each Linux guest
- Detects loss of physical NIC connectivity and automatically fails over to surviving NIC
- Active/backup & aggregation modes
- **Centralized configuration benefits all guests**

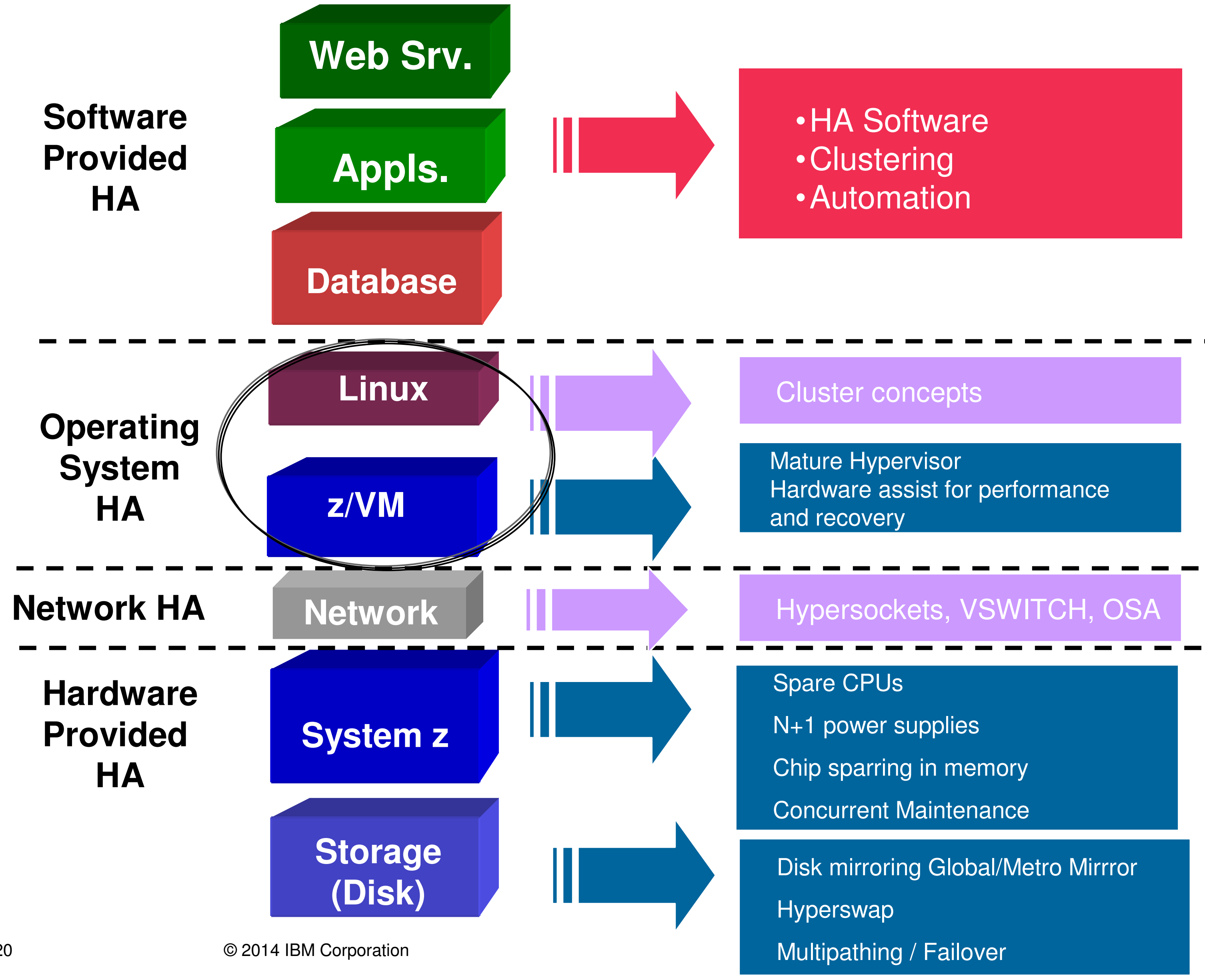
System z network HA options

Multi-zone Network VSWITCH (red zone physical isolation)



With 2 VSWITCHes, 3 VLANs, and a multi-domain firewall

Components of HA with Linux on System z

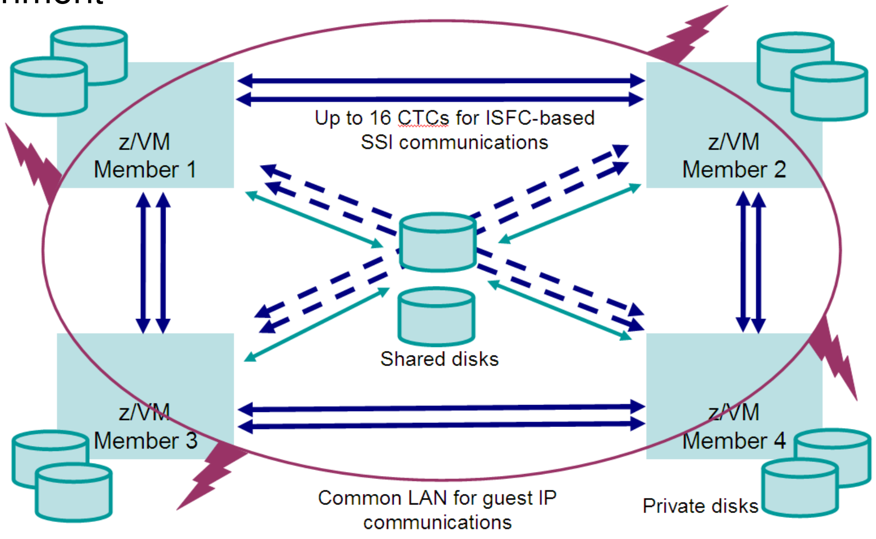


z/VM V6.2 - Increase Availability for Linux guests

Single System Image, Clustered Hypervisor, Live Guest Relocation

Single System Image (SSI)

- connect up to four z/VM systems as members of a cluster
- Provides a set of shared resources for member systems and their hosted virtual machines
  - Directory, minidisks, spool files, virtual switch MAC addresses
- Cluster members can be run on the same or different z10, z196, or z114 servers
- **Simplifies systems management** of a multi-z/VM environment
  - Single user directory
  - Cluster management from any member
    - Apply maintenance to all members in the cluster from one location
    - Issue commands from one member to operate on another
  - Built-in cross-member capabilities
  - Resource coordination and protection of network and disks


Up to 16 CTCs for ISFC-based SSI communications
z/VM Member 1
z/VM Member 2
z/VM Member 3
z/VM Member 4
Shared disks
Common LAN for guest IP communications
Private disks

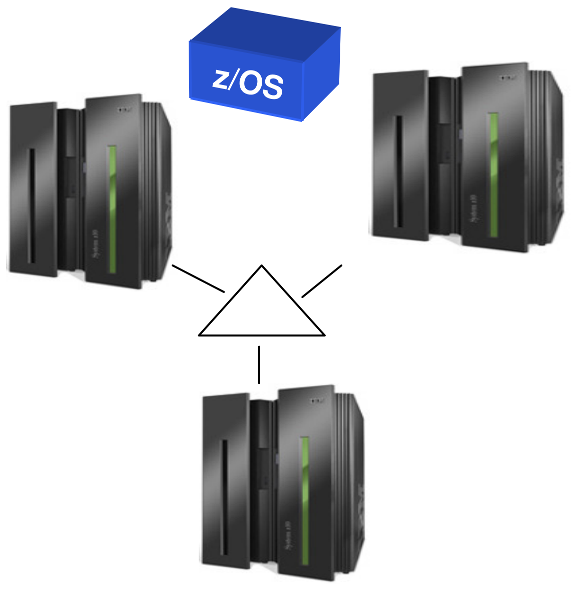
Live Guest Relocation (LGR)

- – Dynamically move Linux guests from one z/VM member to another

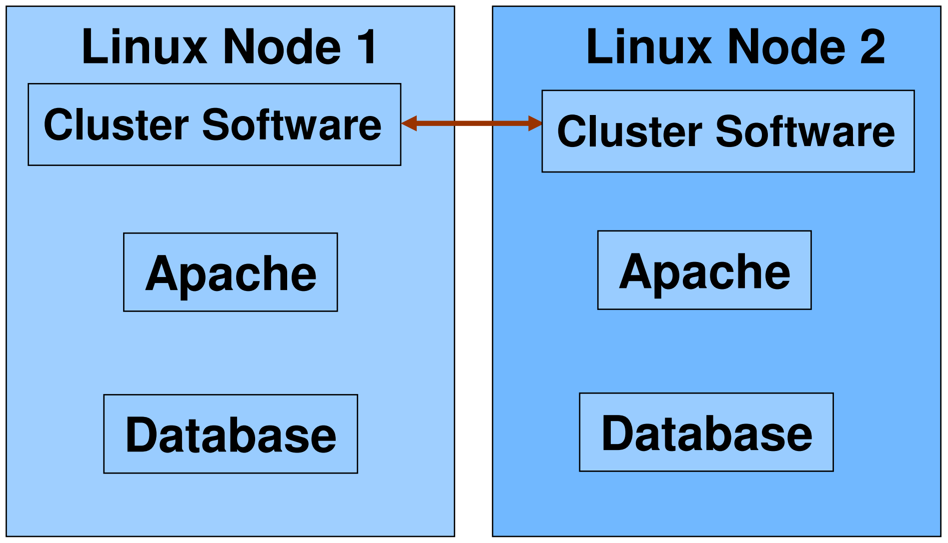
  Reduce planned outages; enhance workload management
  - Non-disruptively move work to available system resources **and** non-disruptively move system resources to work
  - When combined with Capacity Upgrade on Demand, Capacity Backup on Demand, and Dynamic Memory Upgrade, you will get the best of both worlds

High Availability for Operating Systems: z/OS and Linux

- z/OS Parallel Sysplex HA incl. memory sharing



- Linux on z HA using cluster software



Clustering Concepts

Computer Cluster

- A computer cluster consists of a set of loosely connected computers that work together so that in many respects they can be viewed as a single system. (Wikipedia definition: Computer Cluster)

High Availability Cluster

- A computer cluster where each cluster operates as workload node. When one node fails another node takes over the entire workload: IP address, data access, services, etc.
- The key of High Availability is avoiding single points of failure
- High Availability adds costs because of added complexity due to redundant resources in the environment

High Availability Cluster concepts

- Split-Brain
- Quorum
- Fencing
- Data Sharing

Split Brain

- If the heartbeat between nodes fails, all nodes can still be active and:
  - detect the other as failing
  - the status of an unreachable node is unknown
- Communication/heartbeat failures between cluster nodes can lead to isolated actions in separated partitions of the cluster
- If those partitions each try and take control of the cluster, then it's called a split-brain condition
- This can lead to data corruption or inconsistency, therefore split brain has to be inhibited

http://www.linux-ha.org/SplitBrain

Quorum

- Quorum is an attempt to avoid split brain for most kinds of failures

- Typically the Cluster Management Software tries to make sure only one partition can be active

- Quorum is the term for methods for enforcing which part of the cluster is active:
  - A quorum server – as additional node-can decide more reliably
  - Quorum server is in a quorum daemon

- Most common kind of quorum is voting – and only one partition can run the cluster

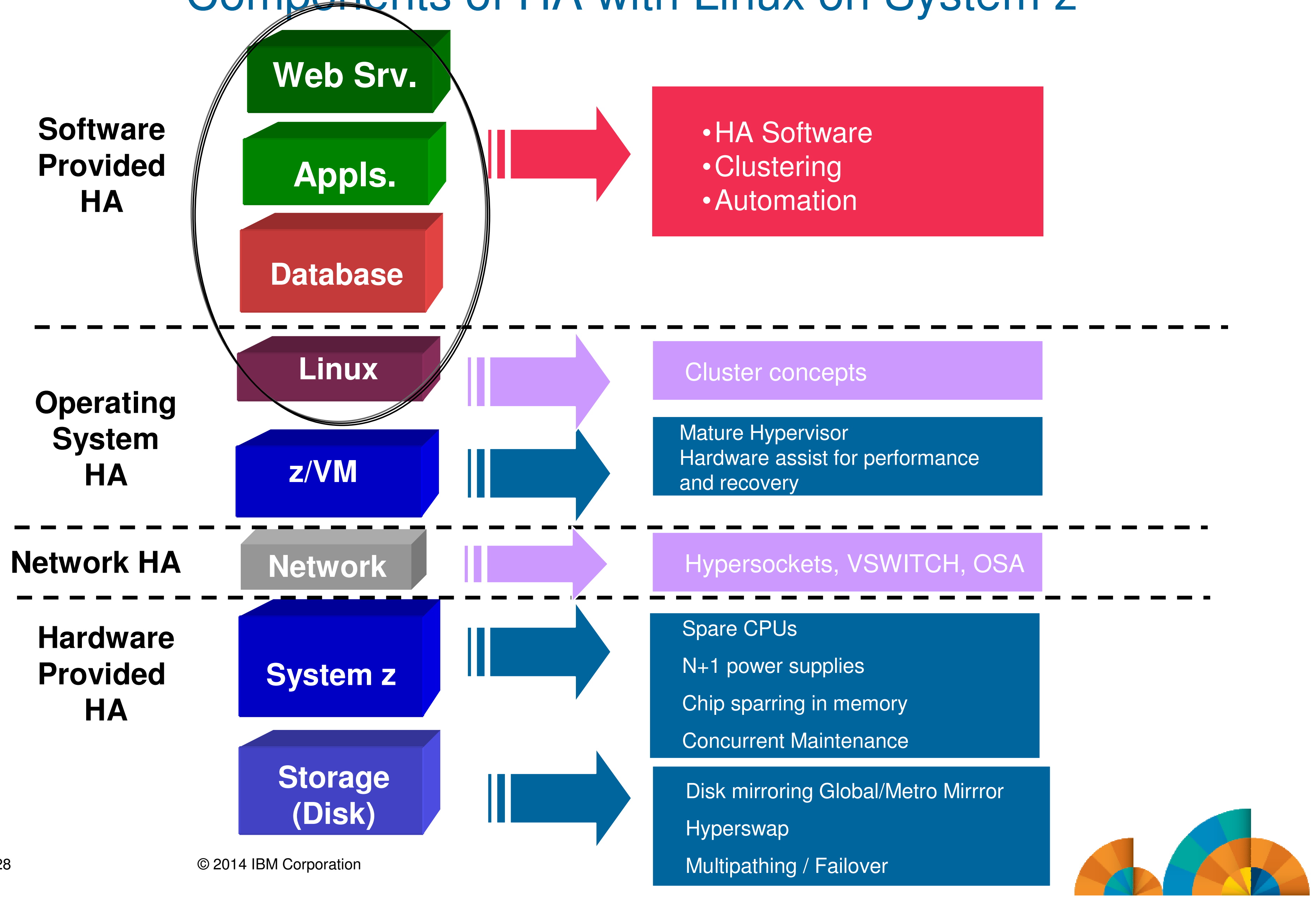
http://linux-ha.org/wiki/Cluster_Concepts

Fencing

- Fencing puts a fence around an errant node or inhibit a failed node from accessing cluster resources
- This way one doesn't have to rely on correct behavior or timing of the errant node
- This is often implemented via STONITH
  - STONITH: Shoot The Other Node In The Head
- Other techniques also work
  - use of hardware or software watchdog timers
  - self shutdown / restart
  - Shared device for notification instead of heartbeat

http://www.linux-ha.org/fencing

Components of HA with Linux on System z



- HA Software
- Clustering
- Automation

Cluster concepts

Mature Hypervisor
Hardware assist for performance and recovery

Hypersockets, VSWITCH, OSA

Spare CPUs

N+1 power supplies

Chip sparring in memory

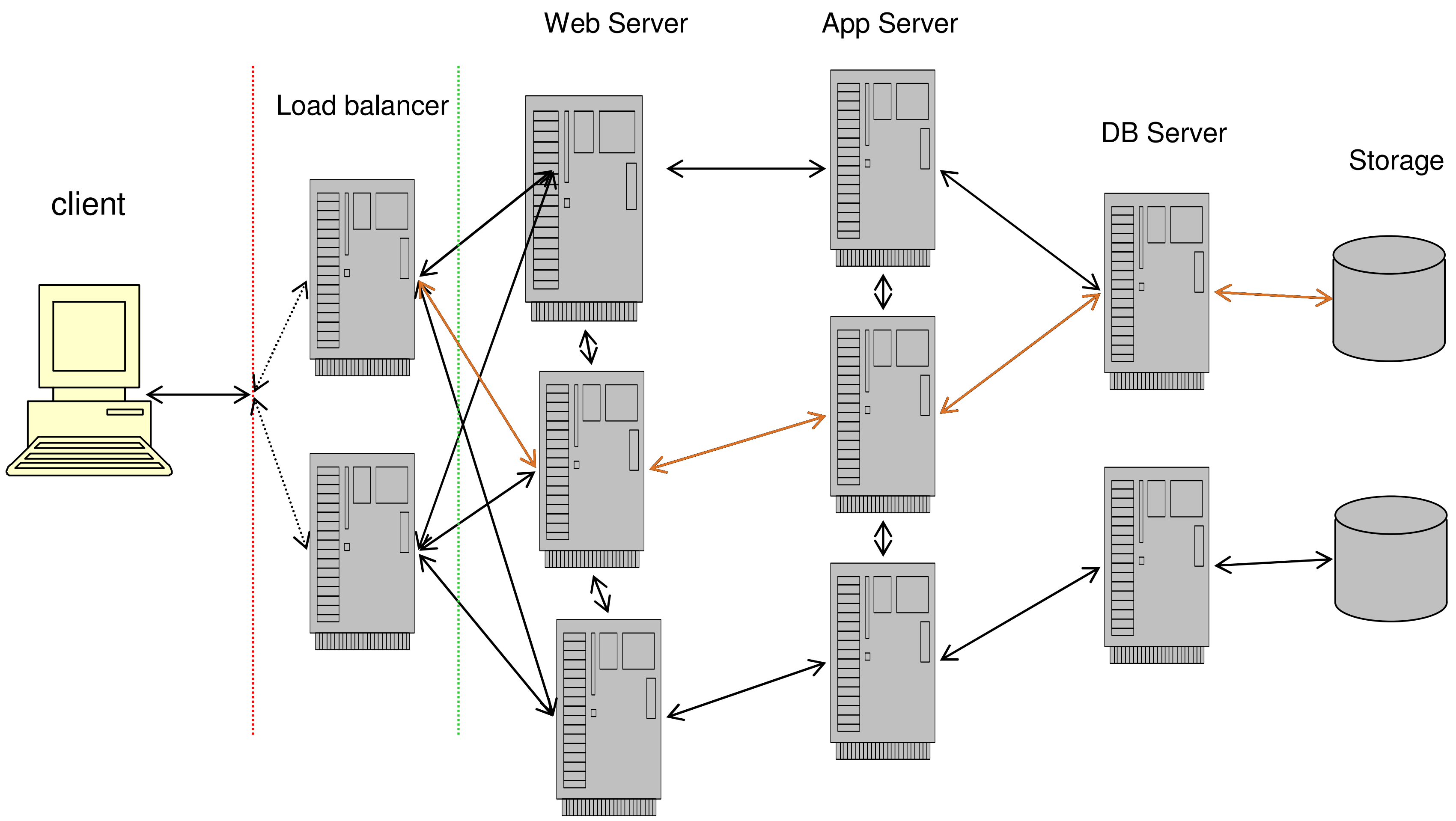
Concurrent Maintenance

Disk mirroring Global/Metro Mirrror

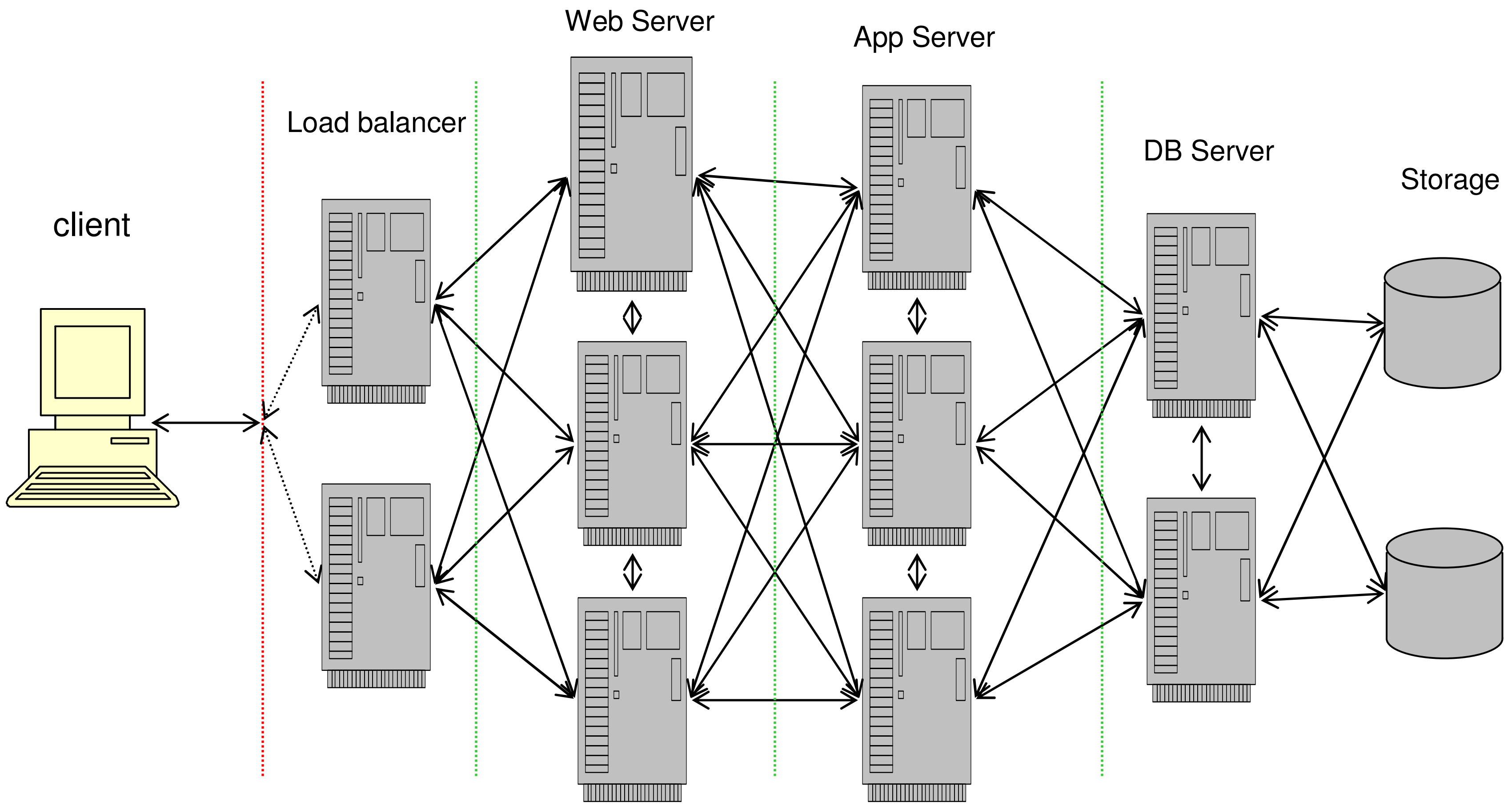
Hyperswap

Multipathing / Failover

HA with Independent Complete Path Execution Streams



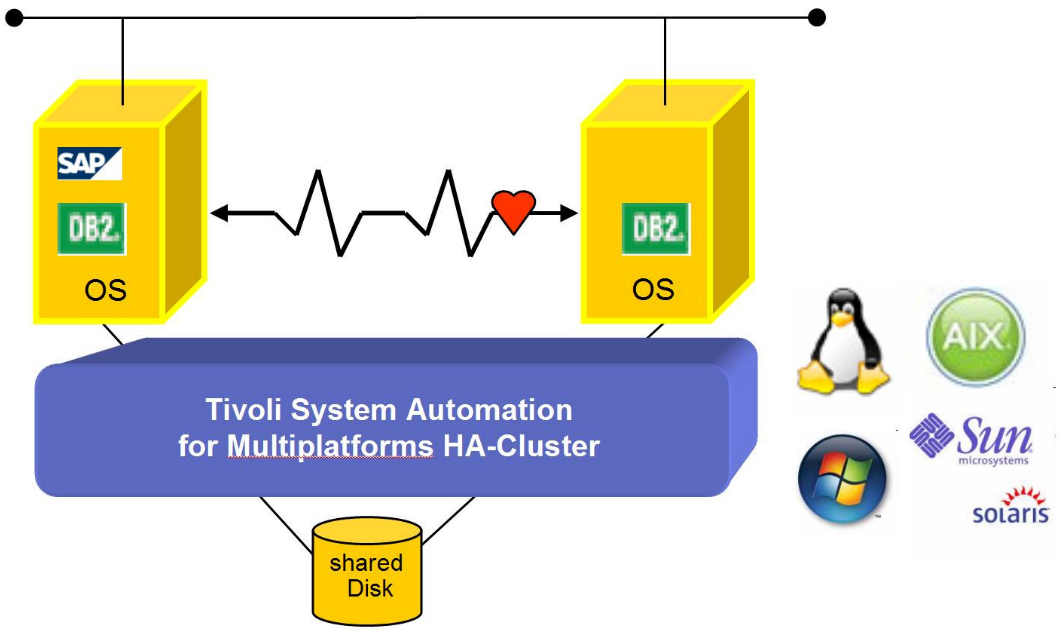
HA with Independent Tiered Execution Streams



High Availability Clustering

- **Linux-HA package implementations**
  - **for Linux environments**

- **Tivoli System Automation for Multiplatforms**
  - **for z/OS**
  - **for multiplatforms**
  - **for distributed heterogeneous environments**



Linux-HA package - High Availability components

- Heartbeat
  - Messaging between nodes to make sure they are alive and available
  - Action required if heartbeat stops after certain tries
- Cluster-glue
  - Everything that is not messaging layer and not resource manager
- Resource-agents
  - Agents running in clustered systems or remote
  - Agents are able to start, restart or stop services
- Pacemaker
  - A Cluster Resource Manager (CRM)
- OpenSAF checkpoint APIs
  - SAF -> Service Availability Forum – created the Service Availability Specifications
  - OpenHPI - The Hardware Platform Interface (HPI) abstracts the differences between hardware implementations, providing a uniform interface to hardware features.
  - OpenAIS - The Application Interface Specification (AIS) specifies an interface that applications interchange information with the service availability middleware (i.e. CRM).

HA support in RedHat Distribution for Linux on System z

- Linux clustering
  - **is not implemented/supported in the RHEL Distro for Linux on System z**
  - It can be introduced from 3rd party provider – Sine Nomine
    - Sine Nomine provides support for Linux-HA on RHEL on System z
      http://www.sinenomine.net/products/linux/systemz/hao4relz

- HA Concepts with RHEL rely on the layered duplication only:
  - Using application HA, like Oracle RAC for example
  - Mirroring the disks DASD/FCP volumes with IBM GDPS/PPRC
  - Strengthening parts of the operating system, like using multipath for disk failover and VSWITCH for network failover

- An alternative is NFS version 4, which has cluster/locking built in
  - The idea is to use virtual networking (hipersockets ideally, or VSWITCH) to connect to a virtualized NFS share.
  - Performance will be similar to I/O to disk and NFS handles multiple read/write access to the same data

HA support in SUSE Distribution for Linux on System z

- HA for Linux on z using clustering
  - is implemented in the SLES Distro for Linux on System z
  - License is part of SLES for Linux on System z
  - HA implementation bases on Linux-HA
  - Graphical tools included for cluster management and monitoring resources
- SUSE Linux Enterprise High Availability Extension delivers all the essential monitoring, messaging and cluster resource management functionality
- HA Add-on: Geo Clustering for SUSE Linux Enterprise High Availability Extension lets you deploy Linux clusters between data centers spread anywhere in the world.

- In the SLES HA Extension, Pacemaker is included, a scalable cluster resource manager with a flexible policy engine that supports n-node clusters

- OpenAIS, as one of the leading standards-based communication protocol for server and storage clustering is used for communication

- Using OpenAIS and Pacemaker, you can continuously monitor the health of your resources, manage dependencies, and automatically stop and start services based on highly configurable rules and policies

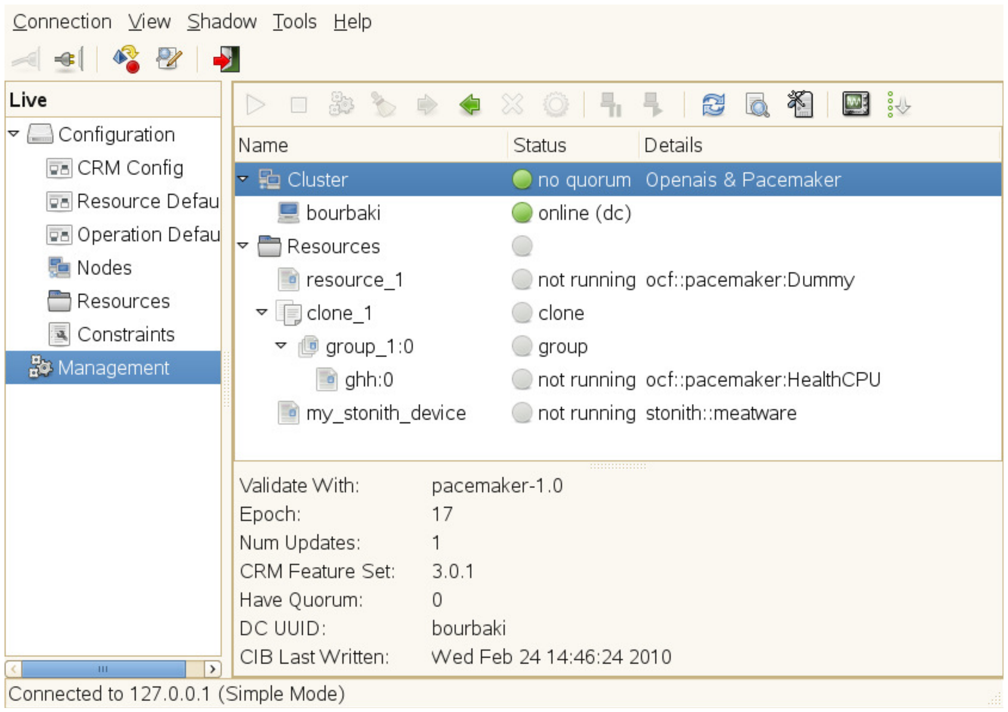
User-friendly management tools

SUSE Linux Enterprise High Availability Extension includes a powerful new unified command-line interface for experienced IT managers to quickly and easily install, configure and manage their clustered Linux servers.

- Graphical user interface that provides operators with a simple, user-friendly tool for monitoring and administering their clustered environment.
- New YaST2 modules for the configuration:
  - of DRBD,
  - openAIS
  - multipath

Supported Platforms

- SUSE Linux Enterprise Server 11
- for x86, x86_64,
- Itanium*,
- Power*
- System z* architectures

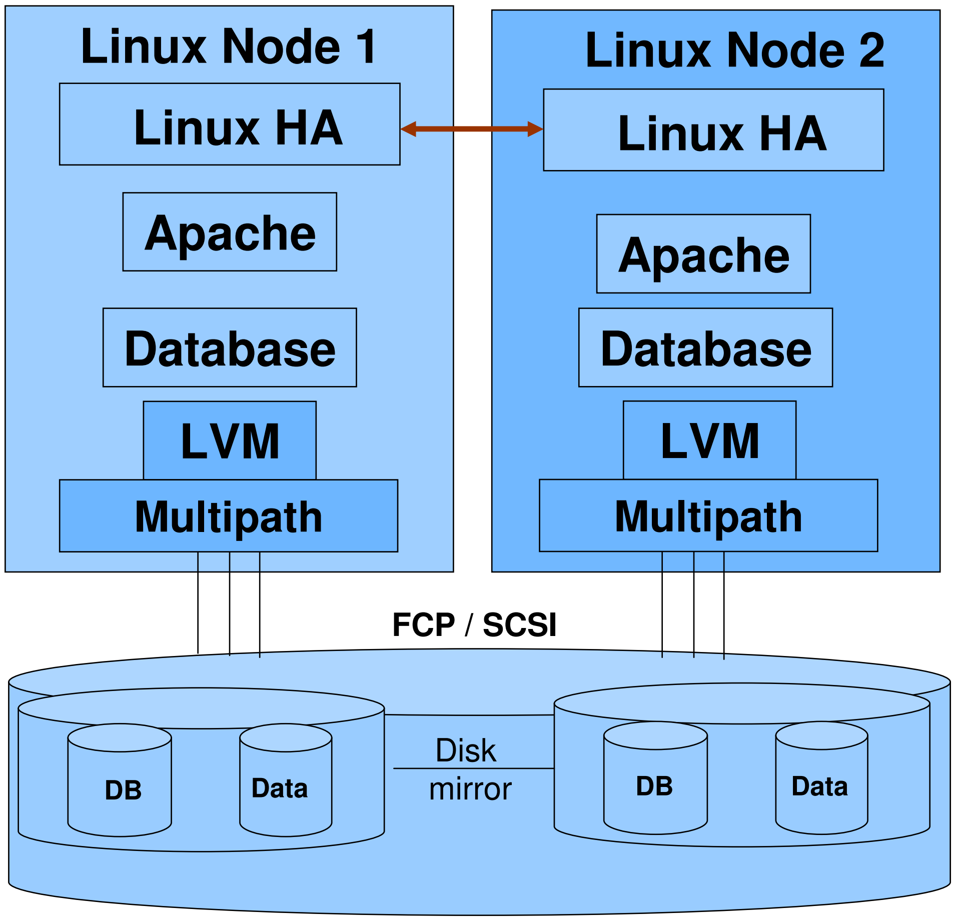


Name	Status	Details
Cluster	no quorum	Openais & Pacemaker
bourbaki	online (dc)	
Resources		
resource_1	not running	ocf::pacemaker:Dummy
clone_1	clone	
group_1:0	group	
ghh:0	not running	ocf::pacemaker:HealthCPU
my_stonith_device	not running	stonith::meatware

```
Validate With:       pacemaker-1.0
Epoch:               17
Num Updates:         1
CRM Feature Set:     3.0.1
Have Quorum:         0
DC UUID:             bourbaki
CIB Last Written:    Wed Feb 24 14:46:24 2010
```

Connected to 127.0.0.1 (Simple Mode)

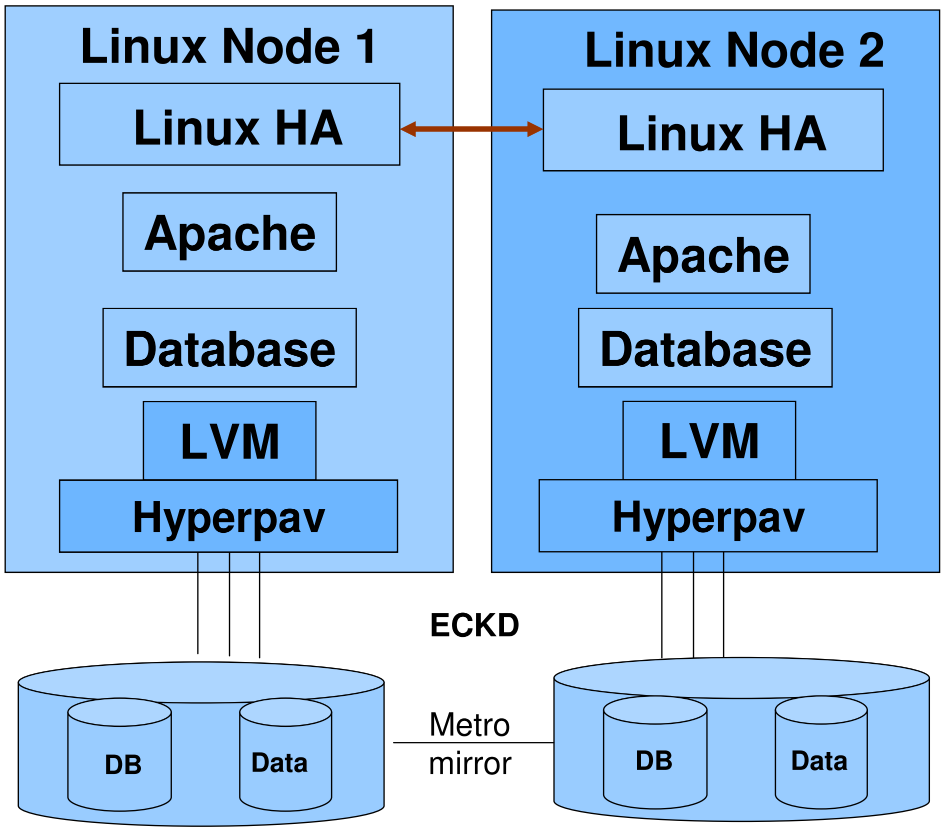
HA with Open Source SW and Linux-HA with Data-mirroring



Linux-HA
- **Heartbeat**
- **LVM**

Disk HA and Virtualization:
- **HA with stretched SVC**
- **HA with IBM Storage Mirroring**

HA with Open Source SW and Linux-HA with Data-mirroring



Linux-HA
- **Heartbeat**
- **LVM**
- **Hyperpav**

Disk HA and Virtualization:
•HA with IBM Storage Mirroring

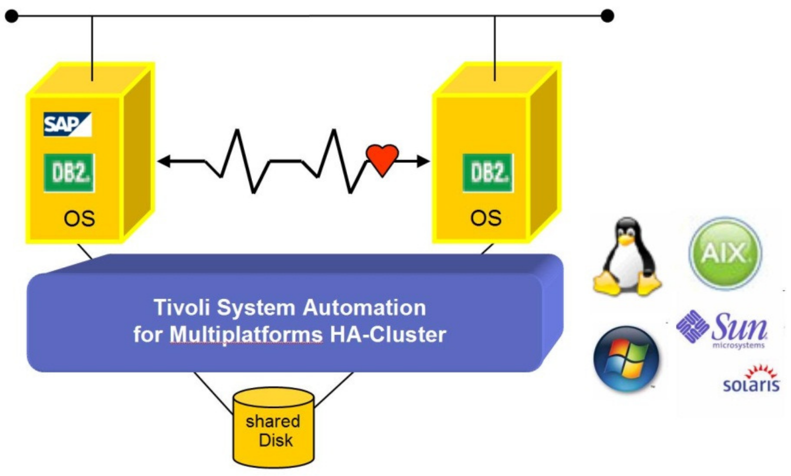
HA with IBM Tivoli System Automation multiplatform support

- IBM Tivoli System Automation for Multiplatform (SA MP) can take advantage of:
  - Linux-HA heartbeat environment
  - enable cross platform HA
  - z/OS High availability together with Linux
  - implements advanced resource group automation
  - dependencies and policy management and hierarchies
  - can be used in HA for non-clustered systems and applications
  - supports various platforms including System z
  - contains a variety of predefined HA adapters for middleware (i.e. DB2, SAP, WebSphere, Apache …)

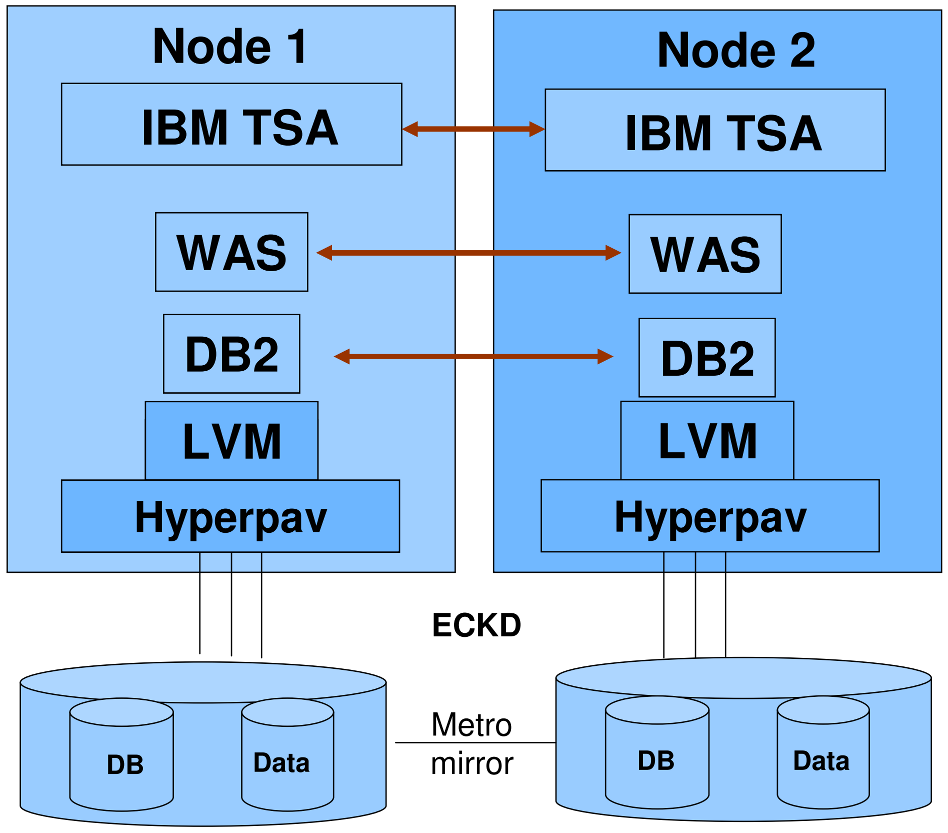
High Availability Clustering

- **Tivoli System Automation for Multiplatforms**
  - Provides a **High Availability Cluster**
  - **Automates startup and shutdown** in correct sequence of complex, statefull applications
  - **Actively monitors** all resources and **reacts on outages of SW and HW** components by automatic restart in correct context

- **Automation Policies** define the **Automation Scope**
  - Describe **resources**, **groups** and **relationships**
  - **Define the desired target availability situation**
  - No need to develop automation workflow scripts.



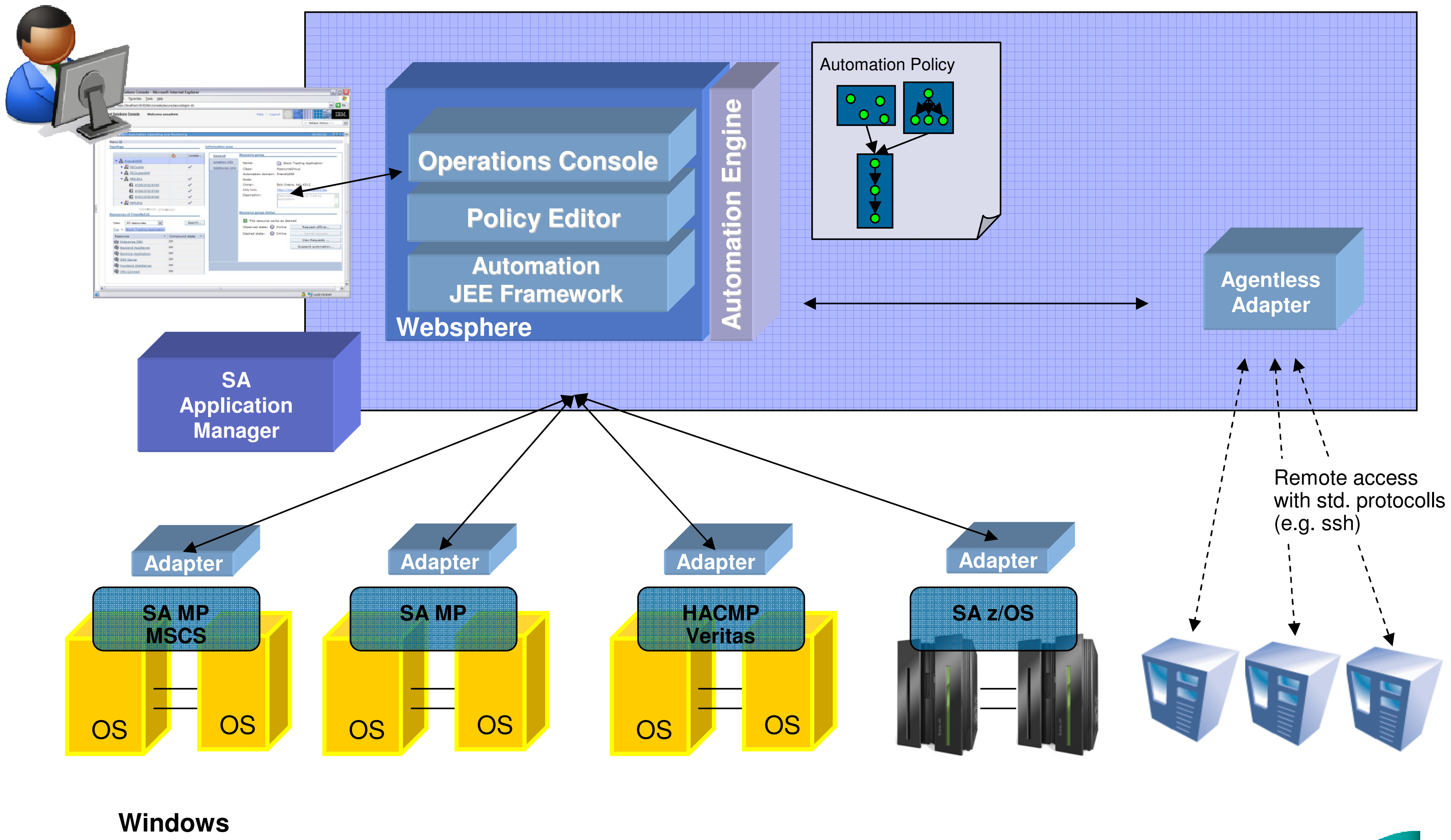
HA with IBM SW products clustered on each level including Data HA and mirroring



IBM Tivoli System Automation
- **Heartbeat**
- **WebSphere App. Server**
- **DB2**
- **LVM**

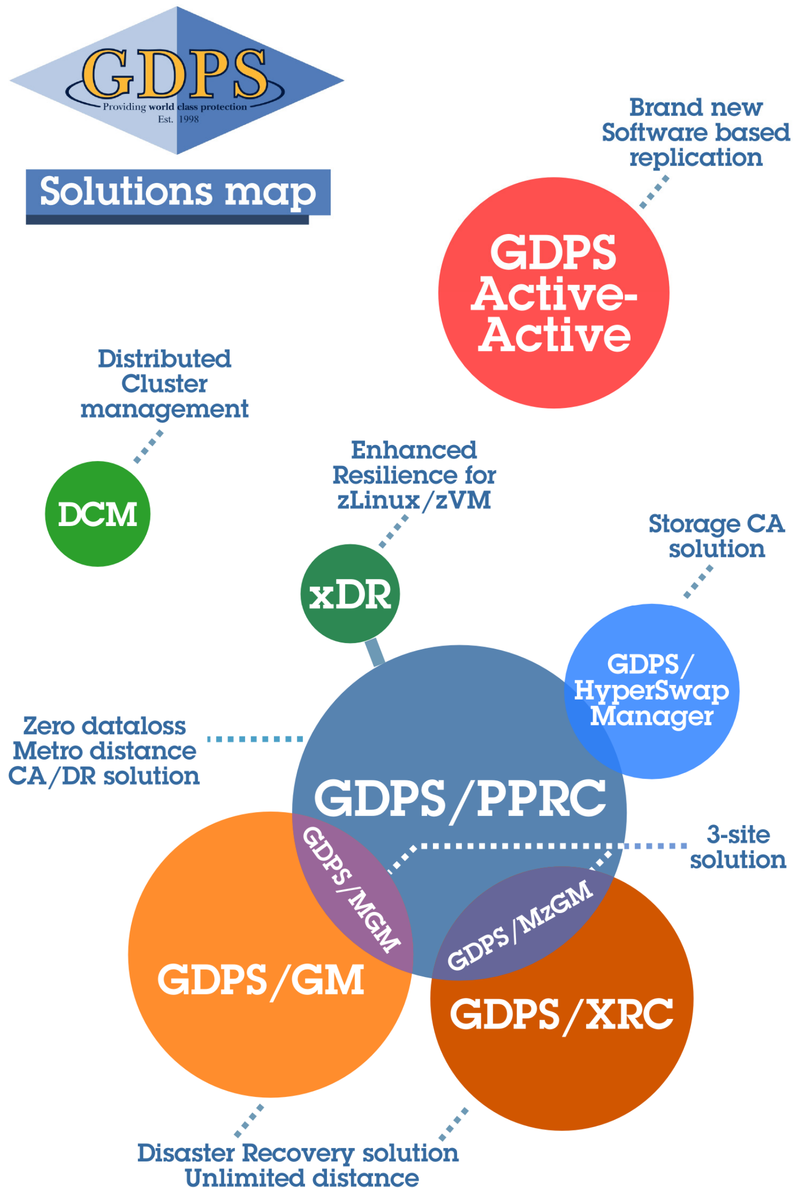
Disk HA and Virtualization:
- **HA with stretched SVC**
- **HA with IBM Storage Mirroring**

SA Application Manager Adapter Infrastructure



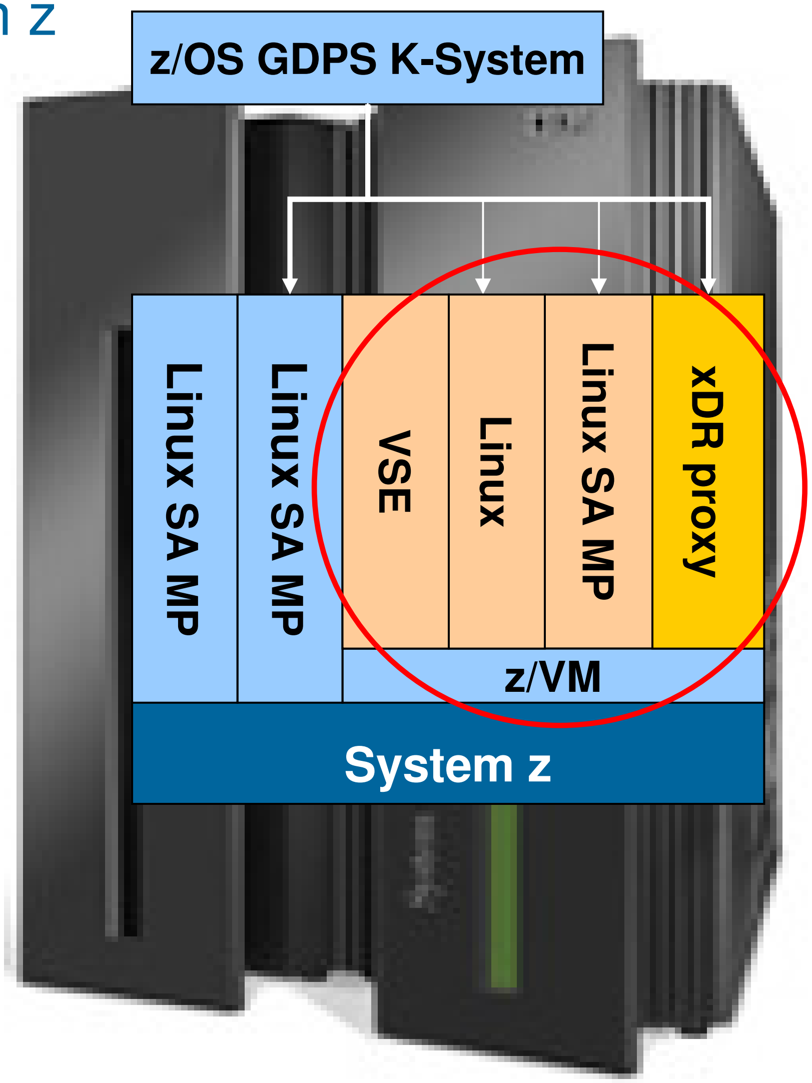
What is GDPS?

- Integrated / Automated solution
- Manages application and data availability in and across sites
  - Monitors systems, disk & tape subsystems
  - Manages planned and unplanned activities
    - System/disk maintenance/failure
    - Site maintenance/failure
- Builds on proven high availability technologies
  - Clustering
  - Remote copy (disk and tape)
  - Automation
- Easy to use interface
  - Intuitive panel interface
  - Simple scripting

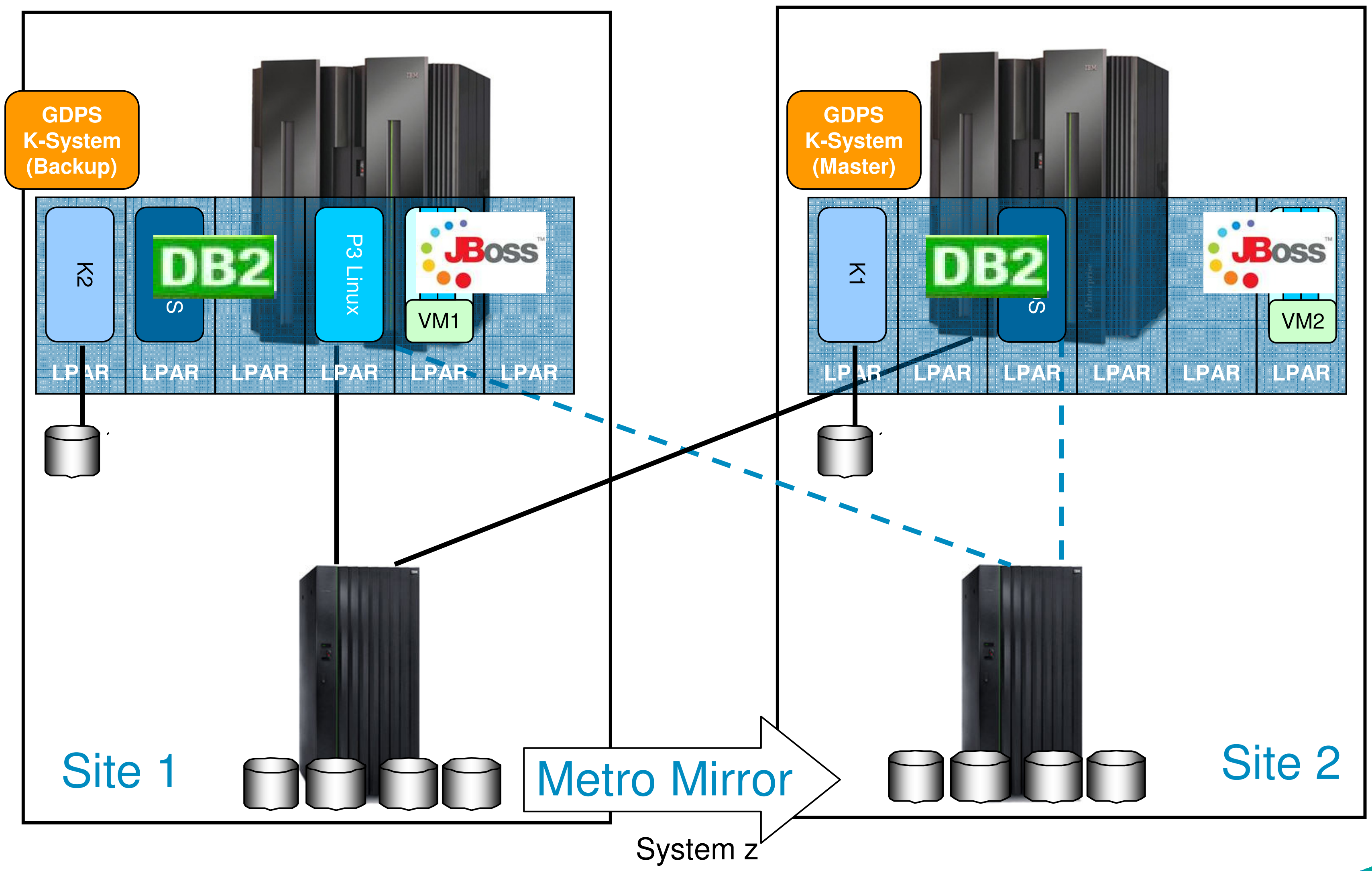

GDPS
Providing world class protection
Est. 1998
Solutions map
Brand new Software based replication
GDPS Active-Active
Distributed Cluster management
DCM
Enhanced Resilience for zLinux/zVM
xDR
Storage CA solution
GDPS/HyperSwap Manager
Zero dataloss Metro distance CA/DR solution
GDPS/PPRC
3-site solution
GDPS/MGM
GDPS/MzGM
GDPS/GM
GDPS/XRC
Disaster Recovery solution Unlimited distance

xDR – the GDPS extension for Linux on z

- xDR provides similar DR capabilities for Linux running on System z as GDPS for z/OS
- Highly automated tasks reduce risk for operating errors
- Disaster Detection on Linux e.g. disk failure, system failure
- Clustering and High availability – provides high availability in case of system, application or network failure
- Single point of control from GDPS
- A complete cross-platform disaster recovery task can be done by operator – no need for availability of all experts for e.g. storage team, hardware team, OS team, application team etc.
- Supported Platforms
  - Linux running as guest on z/VM (xDR on z/VM)
  - Linux running native in LPAR (xDR native)
    - Only reduced set of Linux Distributions are supported

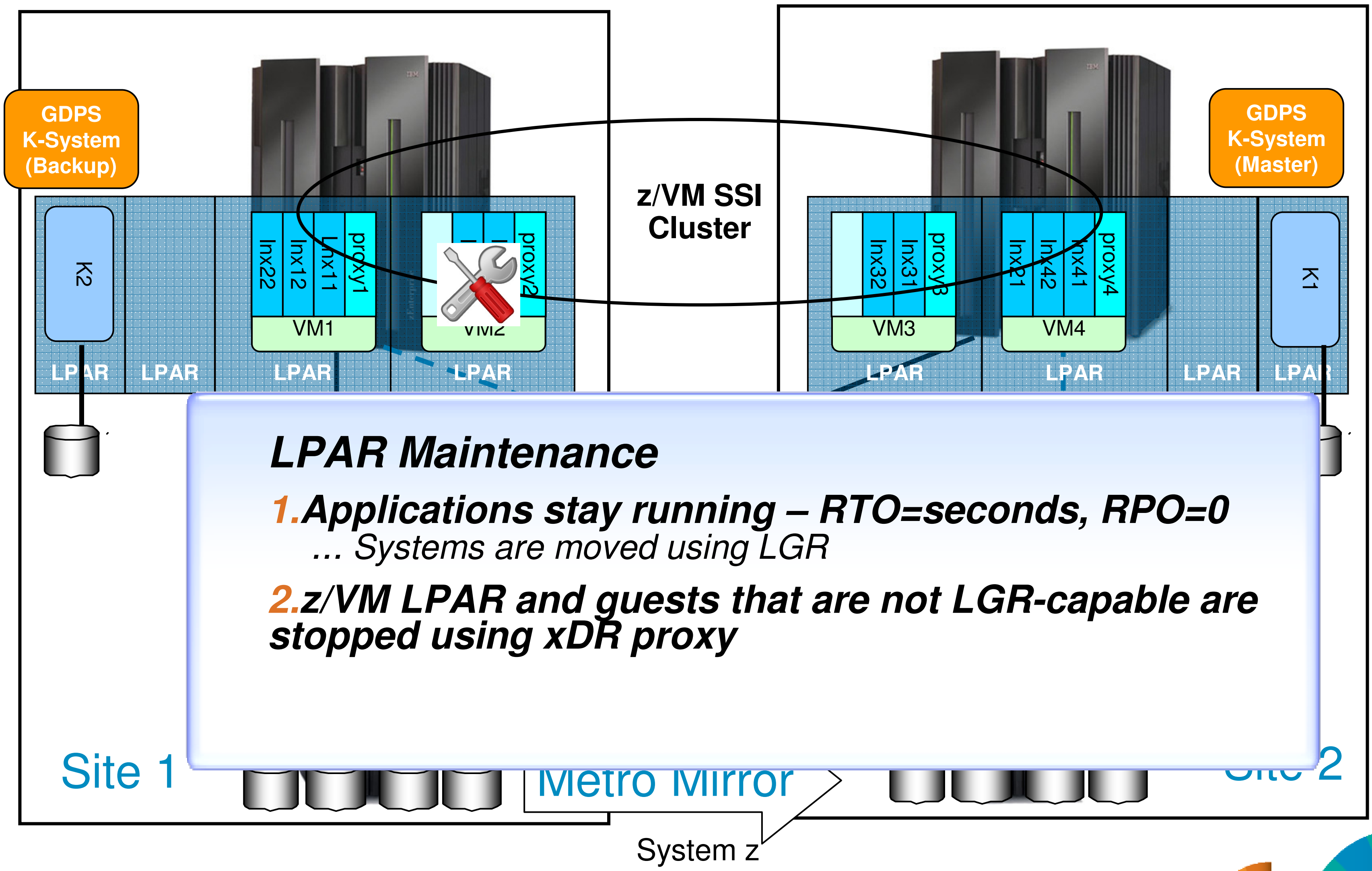


GDPS with xDR Topology



z/VM SSI and GDPS support

z/VM or LPAR Maintenance - Continuous Availability of z/VM Guests



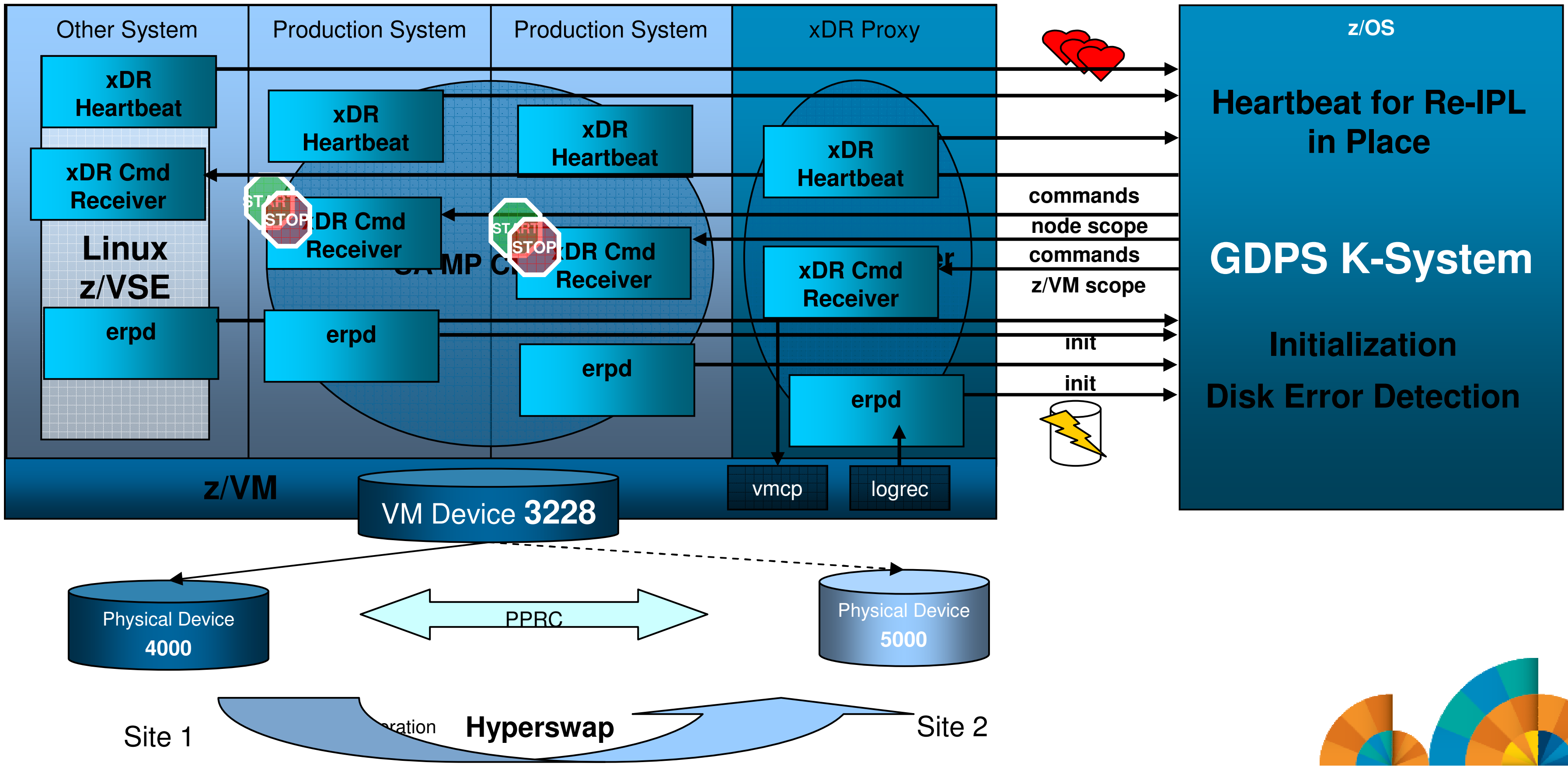
LPAR Maintenance

1.Applications stay running – RTO=seconds, RPO=0
... Systems are moved using LGR

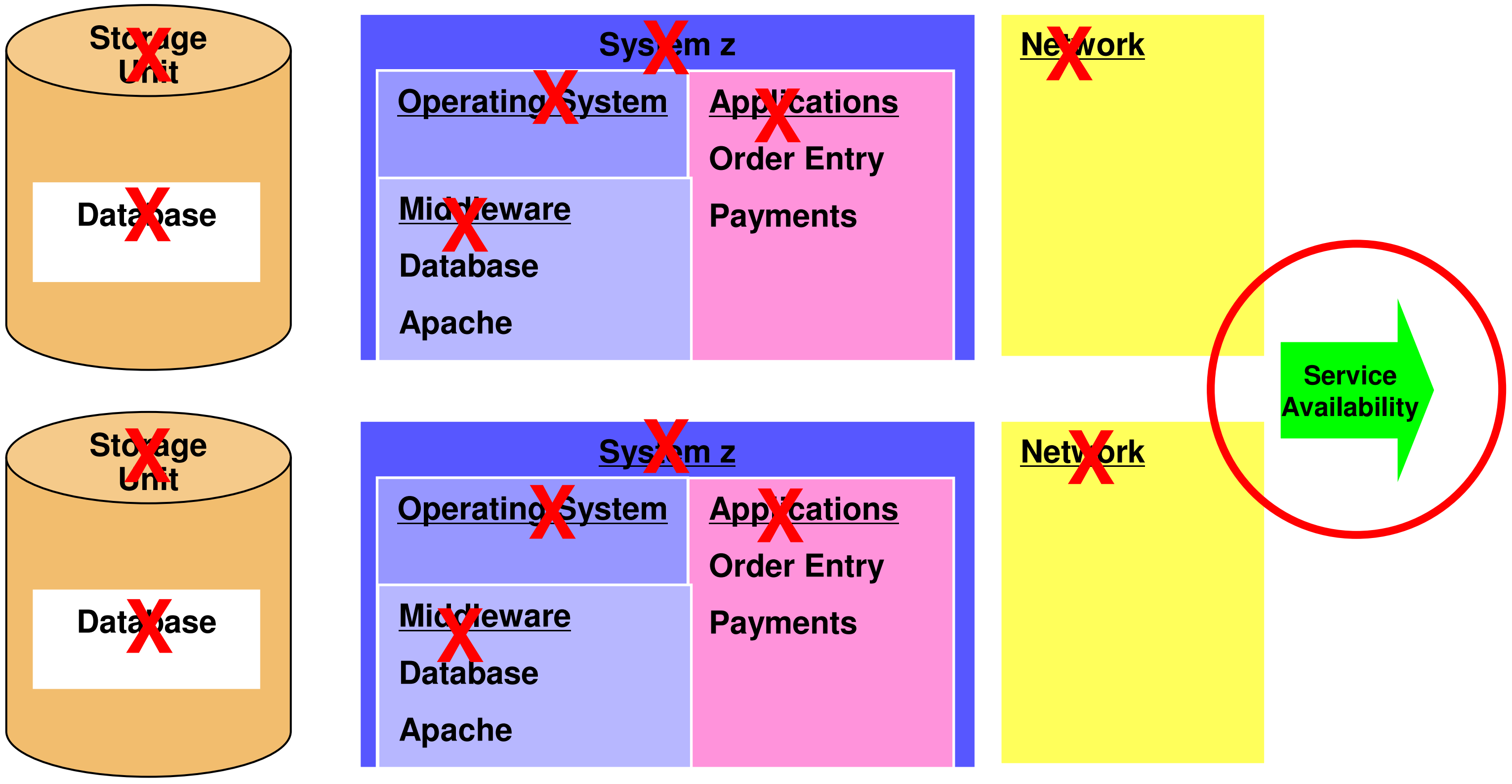
2.z/VM LPAR and guests that are not LGR-capable are stopped using xDR proxy

HA with z/OS GDPS and xDR with z/VM guests

- Proxy
  - One linux system is configured as Proxy for GDPS which has special configuration
    - (Memory locked, Access rights to VM, One-Node-Cluster)
  - Is used for tasks that have z/VM scope
    - HyperSwap, shutdown z/VM, IPL z/VM guest
- Production Nodes
  - Run Linux Workload
  - Are used for local actions (Shut down node, Maintenance Mode)
- Other Systems
  - Enabled for HyperSwap via xDR Proxy (Linux, z/VSE)
  - No re-IPL in place, no start/stop via GDPS (init, reipl, maint)



An ideal High Availability architecture allows service to continue no matter what fails.



- An HA architecture protects the service from product failures by eliminating Single Points of Failure (SPoFs) at all layers (not just internal within the box).
  - –Facilities, HW & SW components, Middleware or subsystems, Applications, Dat, etc.a
- This ideal approach is typically referred to as an active/active solution and may eliminate any service disruption for a single failure scenario.

Find Information Online

https://www.ibm.com/developerworks/servicemanagement/dca/index.html

Data Center Automation

Connect, learn, and share with the experts

Welcome to the **Data Center Automation** community, where you can connect, learn, and share with the Data experts.

Connect and Collaborate

Join the Data Center Automation group, and connect with other members who have an interest Automation. You can also collaborate with the experts by accessing the blogs, forums, and wikis

Blog

Read the perspectives of Data Center Automation experts.

Additional documentation

- Linux-HA project Open source
  http://www.linux-ha.org/wiki/Main_Page

- Suse SLES 11 SP2 High Availability Guide
  http://www.suse.com/documentation/sle_ha/pdfdoc/book_sleha/book_sleha.pdf

- Tivoli System Automation for Multiplatforms
  http://www-01.ibm.com/software/tivoli/products/sys-auto-multi/

- Redbook:
  – Achieving High Availability on Linux for System z with Linux-HA Release 2
    SG24-7711 : http://www.redbooks.ibm.com/abstracts/sg247711.html?Open

Questions?

Wilhelm Mild

IBM Executive IT Architect

THE Open GROUP
Master
Certified IT Architect

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Office: +49 (0)7031-16-3796
mildw@de.ibm.com