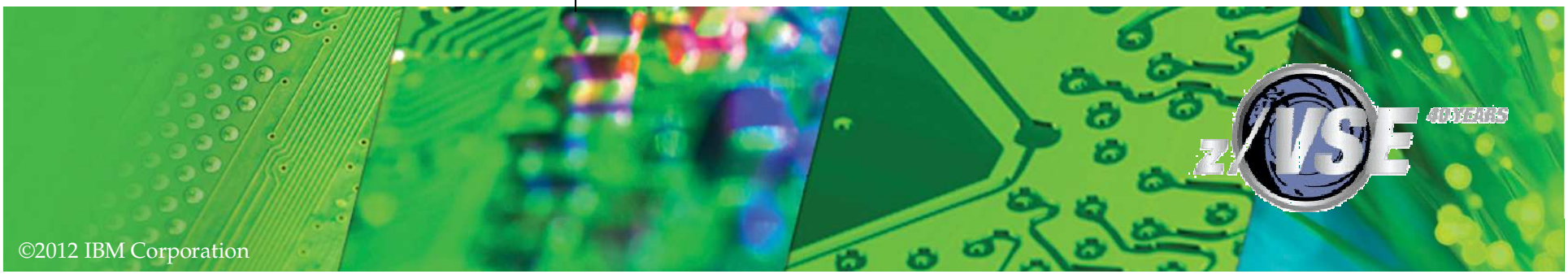
2012
IBM System z Technical University

Enabling the infrastructure for smarter computing

High Availability with Linux on System z and z/OS

zLG23

Wilhelm Mild



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.

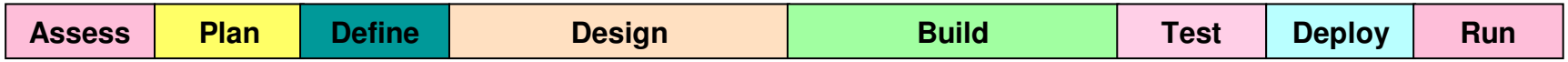
- Continuous Availability (CA) -- Deliver non-disruptive service to the end user 7 days a week, 24 hours a day (there are no planned or unplanned outages).

y.

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

- The effort of an IT organization to support 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

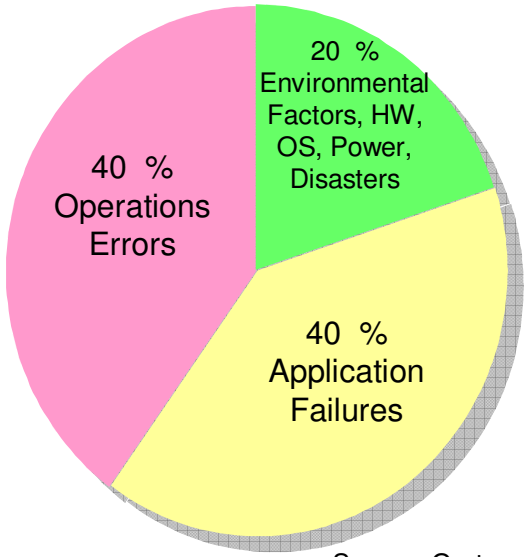
- High Availability typically requires an one time investment, with ongoing improvements
  - Initial Assessment, business and process requirements
    - identify gaps, and build a strategic road map.
    - Priority and Dependencies
    - How to approach and staff
  - Identify and prioritize key initiatives 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.



Business Continuity Issues

What are the reasons for system outages?

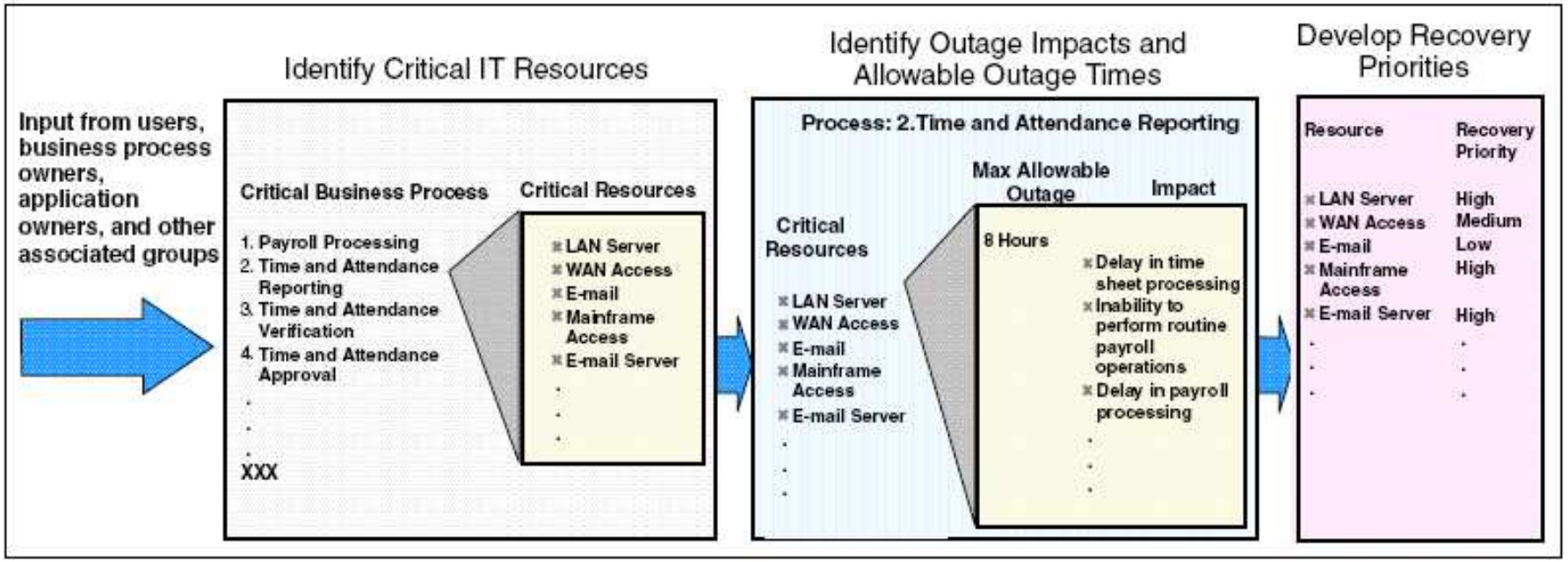
- **Planned** outages
  - Maintenance
  - Tests
- **Unplanned** outages
  - Operator errors
    - Lack of application skills
    - Lack of OS skills in heterogeneous environment
  - Application failures
    - SW exceptions
    - Environment / Configuration problems
  - Environmental failures
    - OS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



Source: Gartner Group

The Business Impact Analysis (BIA)

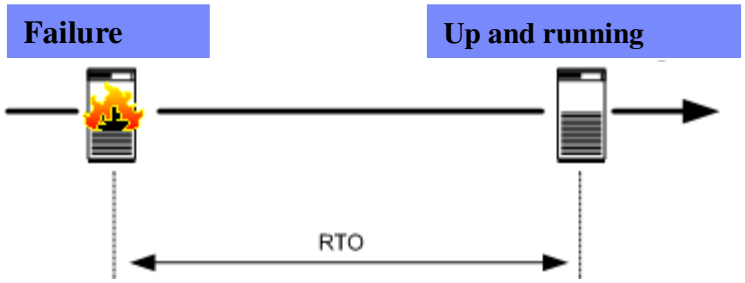
- IT Resource relation and priorities for DR
- Consider all environments
- Prioritize based on business importance



Example of the Business Impact Analysis process

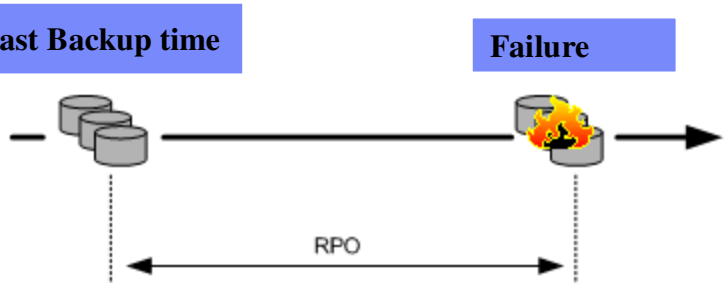
Identify RTO, RPO und NRO

Business Resiliency Plan



Recovery Time Objective (RTO)

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



Recovery Point Objective (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 Recovery Objective (NRO)

Time requirements for network availability.

Differences between HA and DR

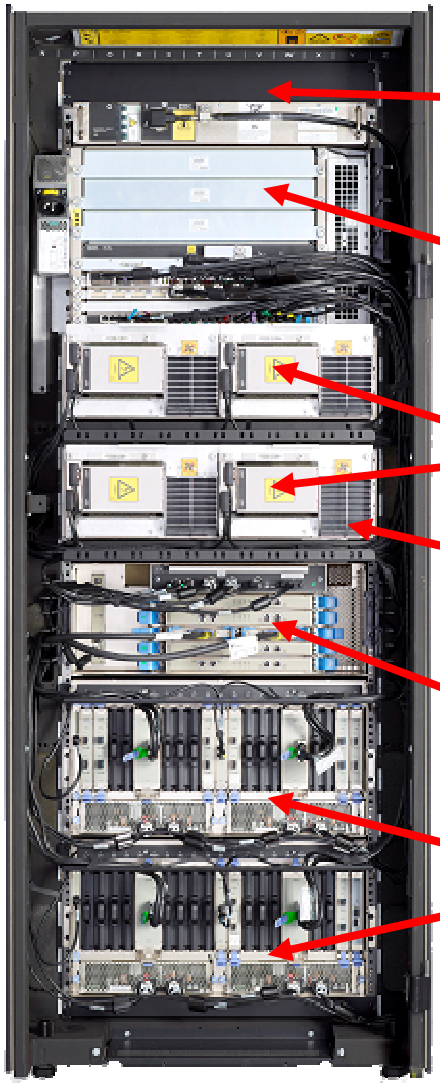
- **High Availability - HA:**
  - Failover is typically realized via duplication and clustering
  - Failover times measured in seconds and minutes
  - Reliable inter-node communication

- **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

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)

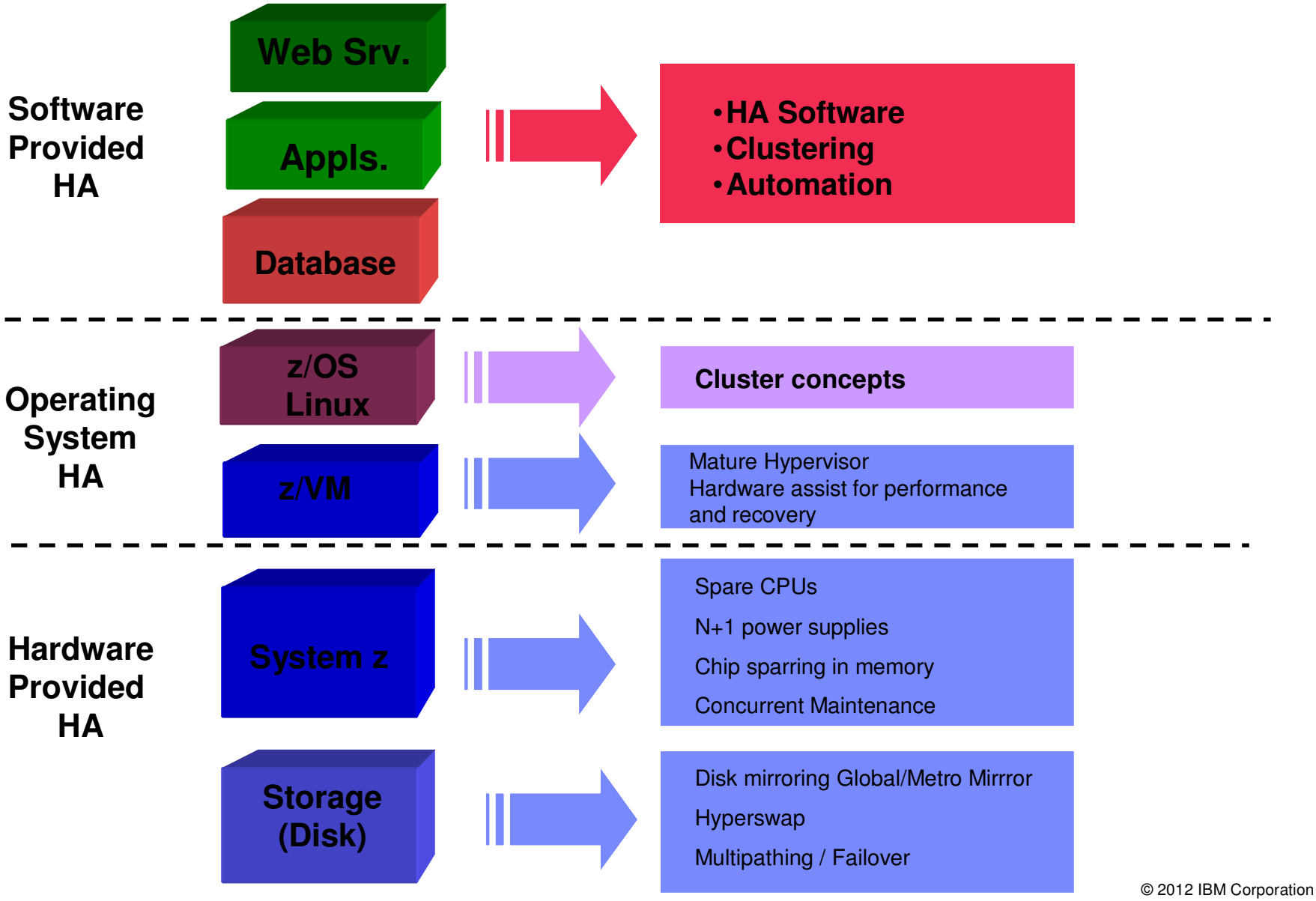
PCIe I/O drawers (redundant)

2 x Support Elements

Rear View

Front View

Components of HA with z/OS and Linux on System z



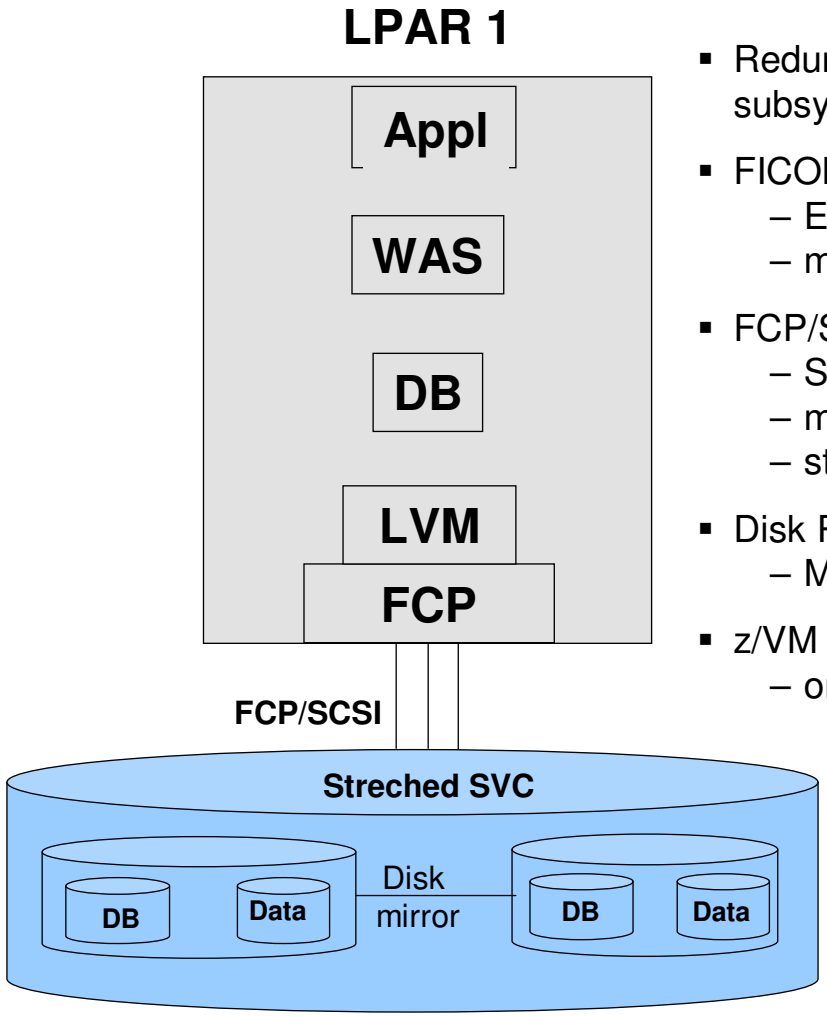
High Availability Considerations for System z

Single Point of Failure	Probability of Failure	Cost to fix
System z hardware	Very Low	High
LPAR	Very Low	Low
z/VM	Low	Low
Linux	Low	Very Low
Disk Subsystem microcode	Low	Medium
Application	High	Very Low

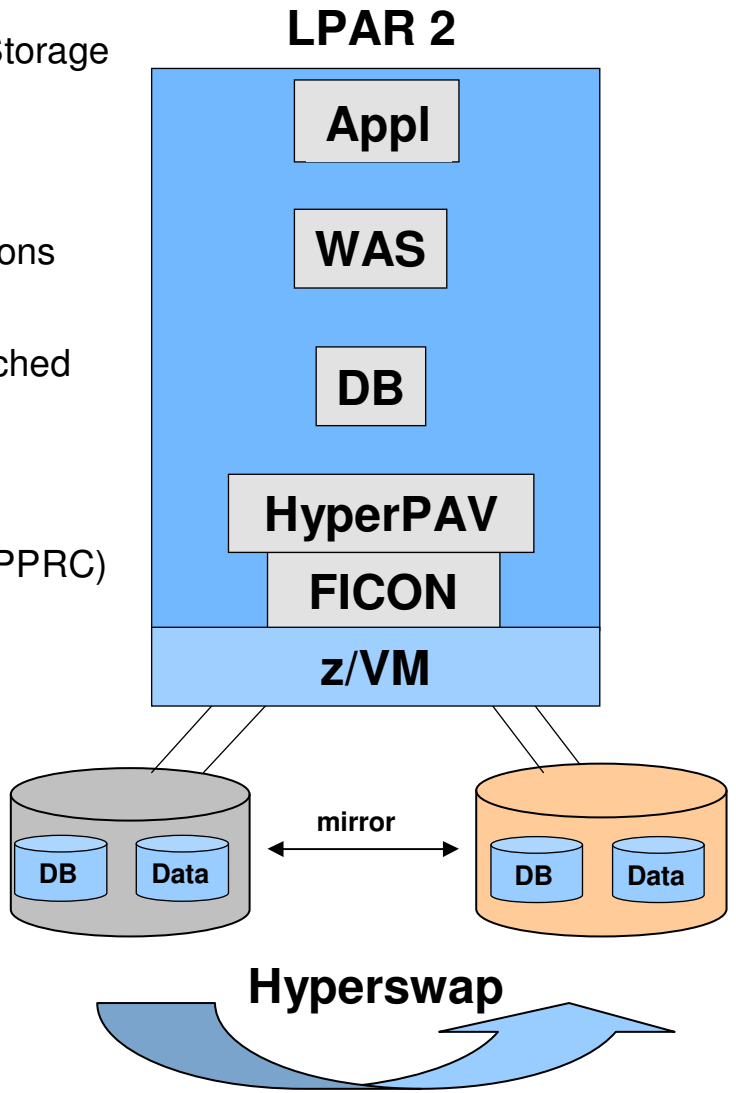
Besides hardware and software failures, what else can cause production down time?

- System z hardware upgrades requiring Restart
- LPAR configuration changes requiring reboot of the LPAR
- z/VM maintenance (if not SSI clustered)
- Linux kernel maintenance that requires reboot
- Application maintenance

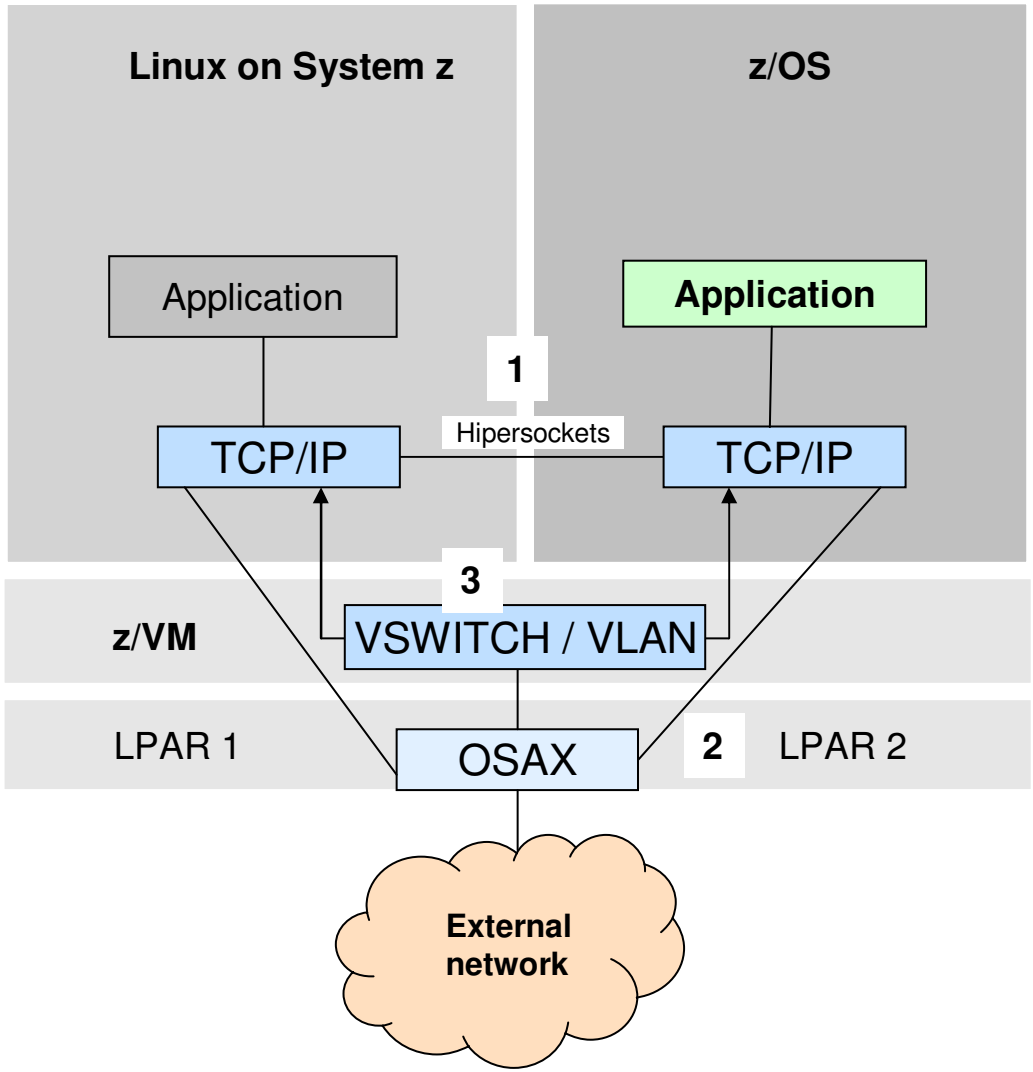
Storage HA Options



- 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)
- z/VM Hyperswap
  - online disk swap

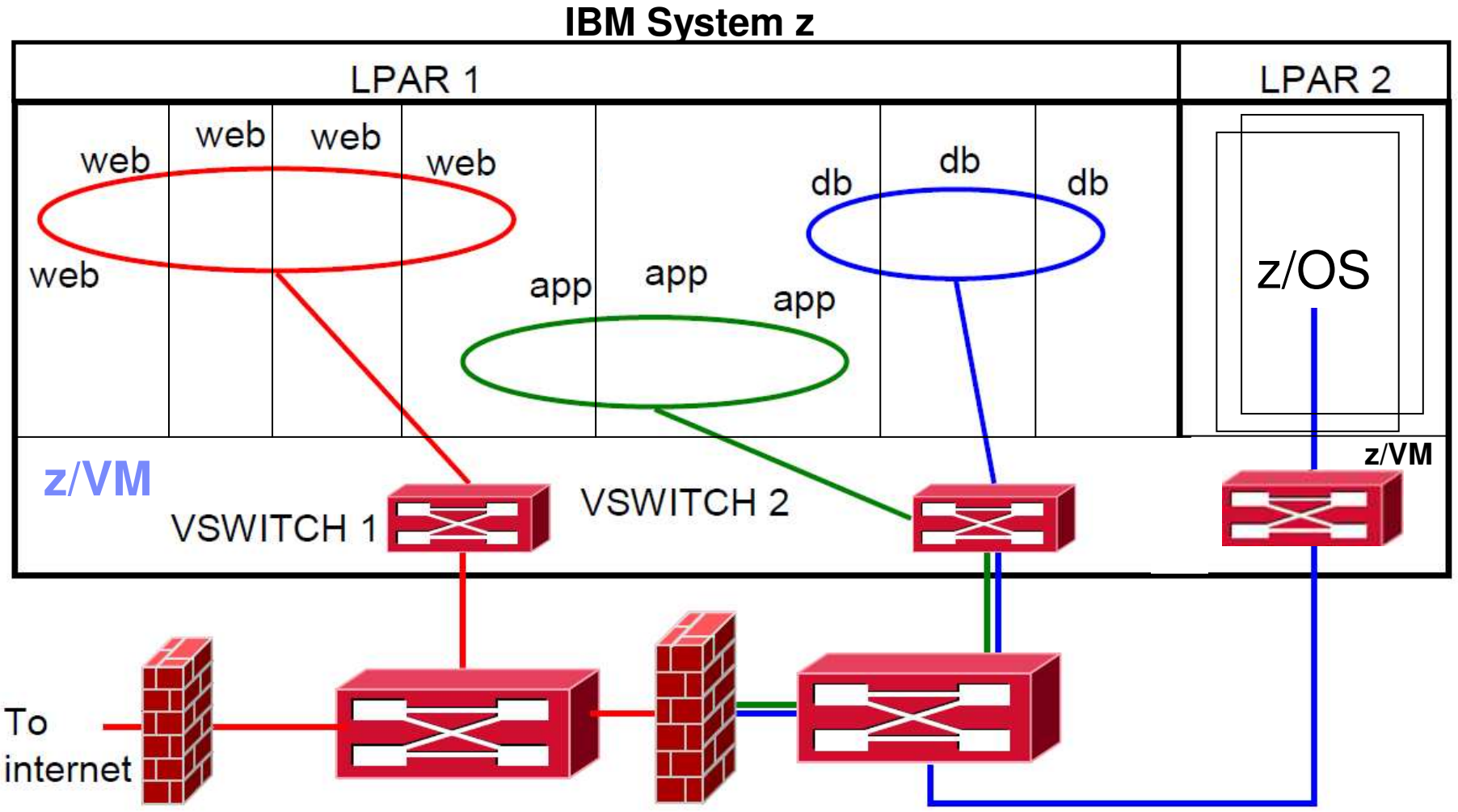


Linux and network alternatives in System z



System z network HA options

Multi-zone Network VSWITCH (red zone physical isolation)



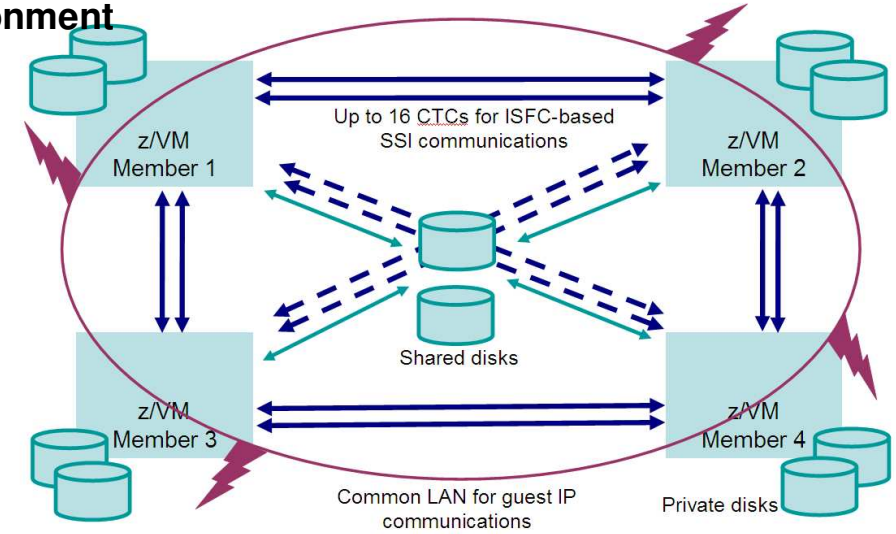
With 2 VSWITCHes, 3 VLANs, and a multi-domain firewall

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



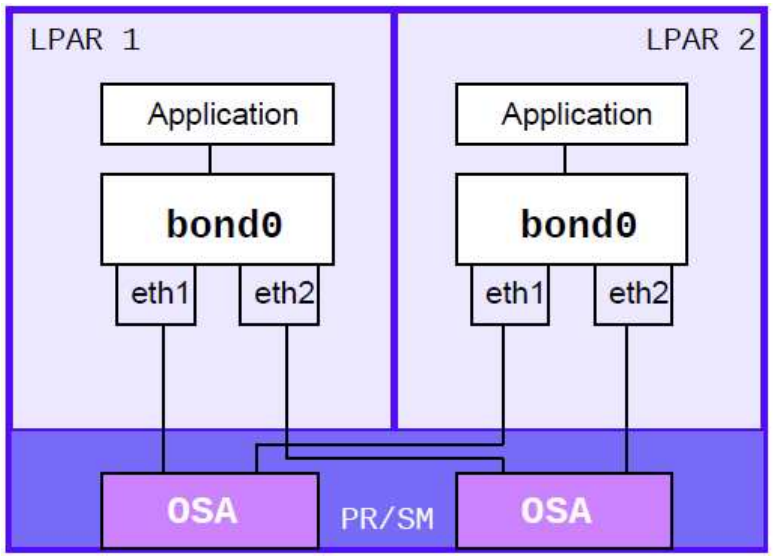
Live Guest Relocation (LGR)

- **– 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

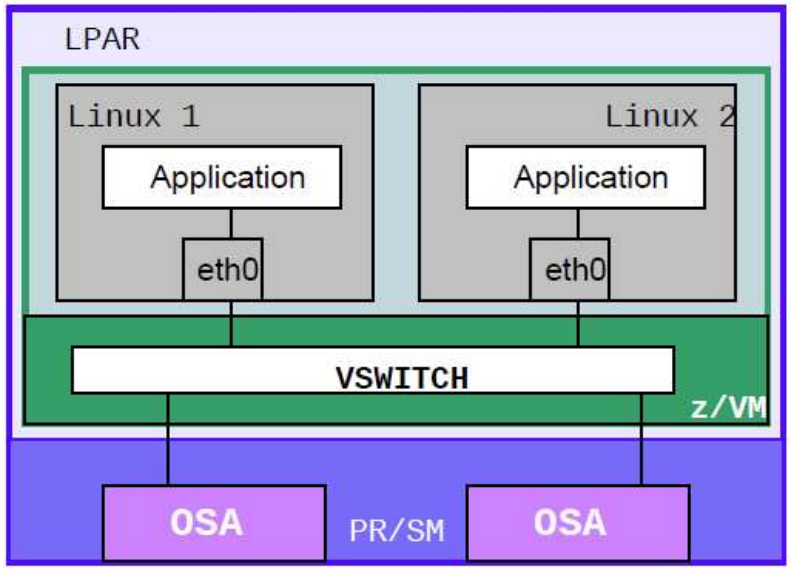
Network Interface Redundancy and Automated Failover

OSA Channel Bonding



- 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**

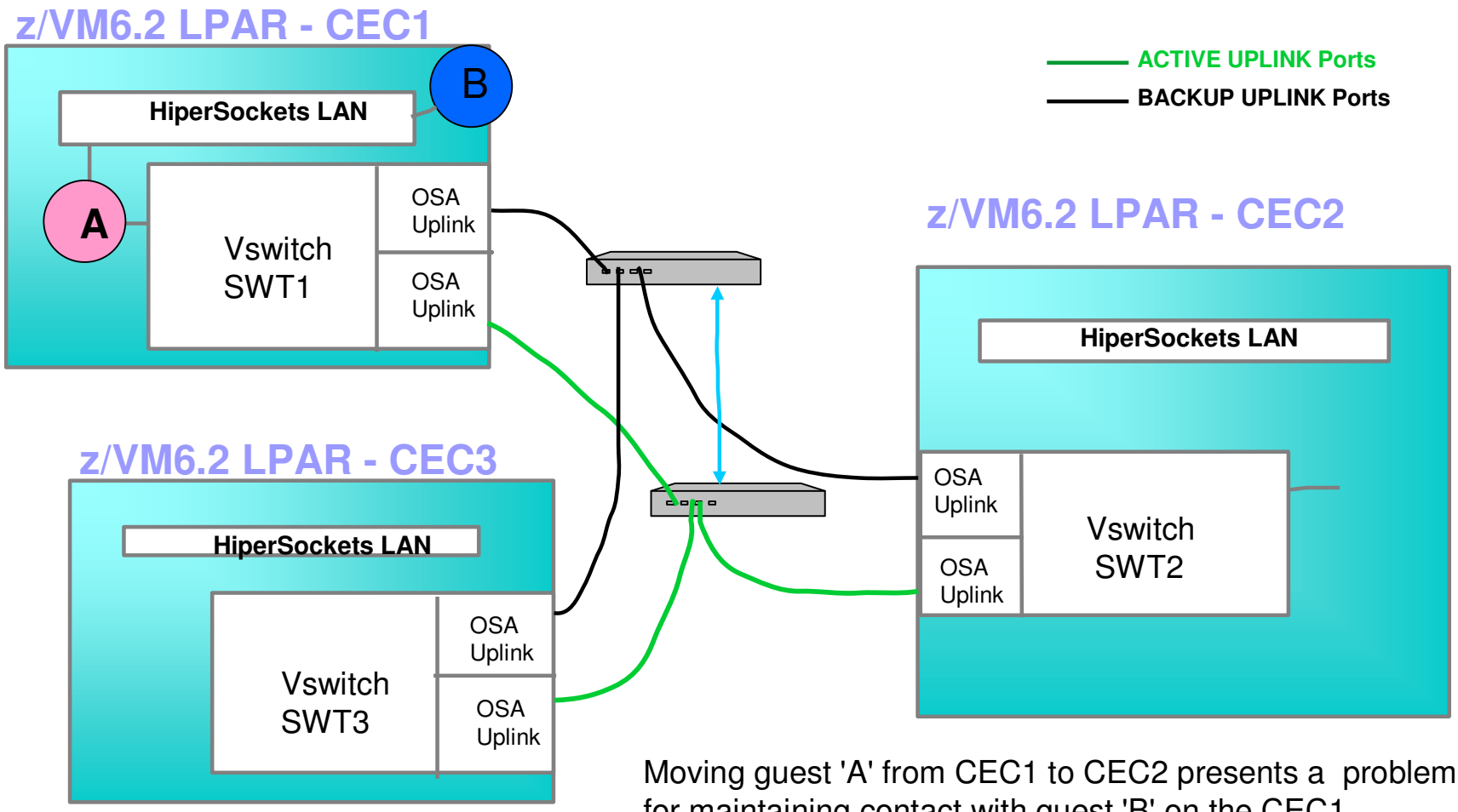
z/VM VSWITCH



- 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**

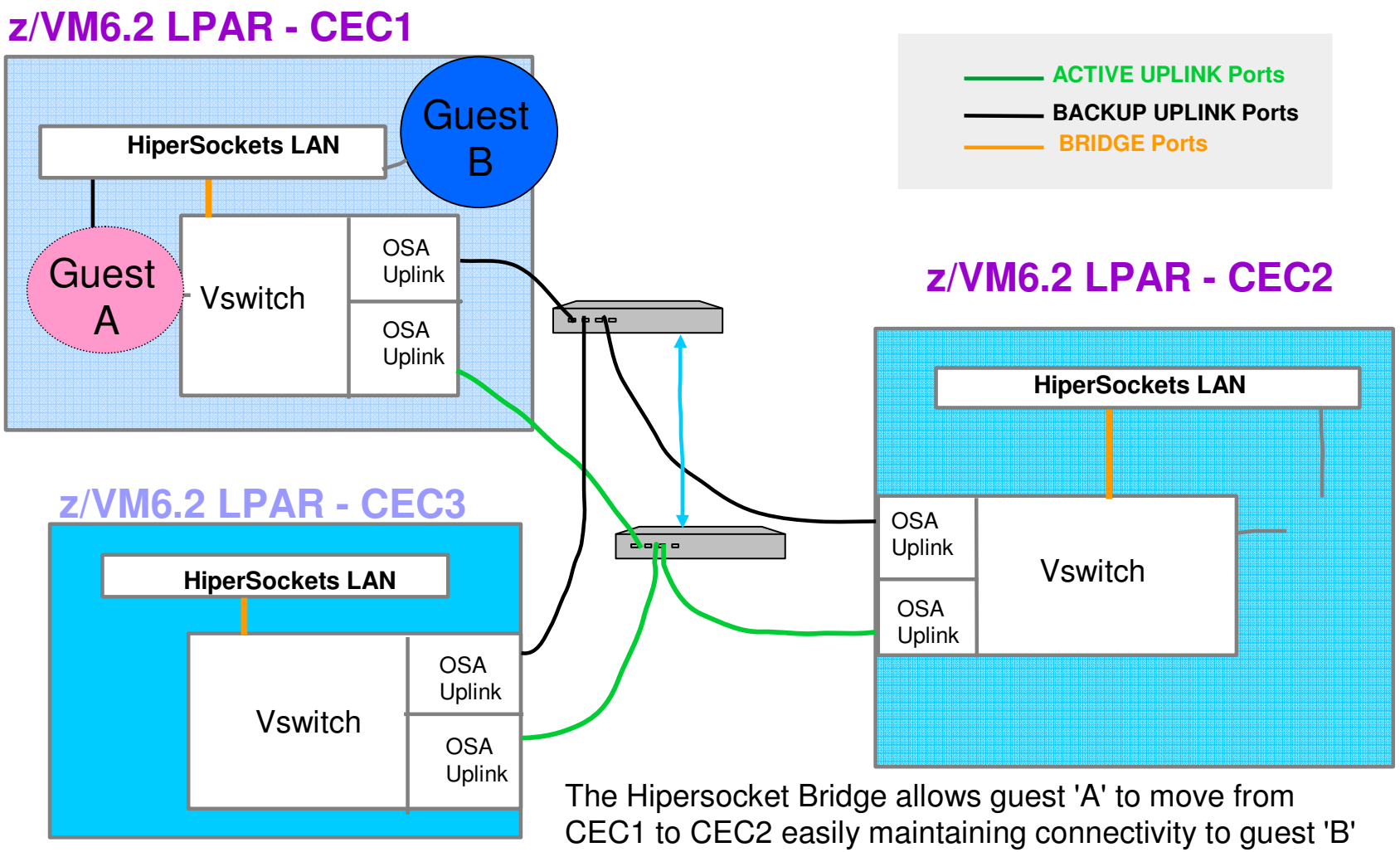
VSWITCH Topology

A typical Vswitch topology for multiple CECs. Active and Backup Uplink Ports to redundant Ethernet switches.



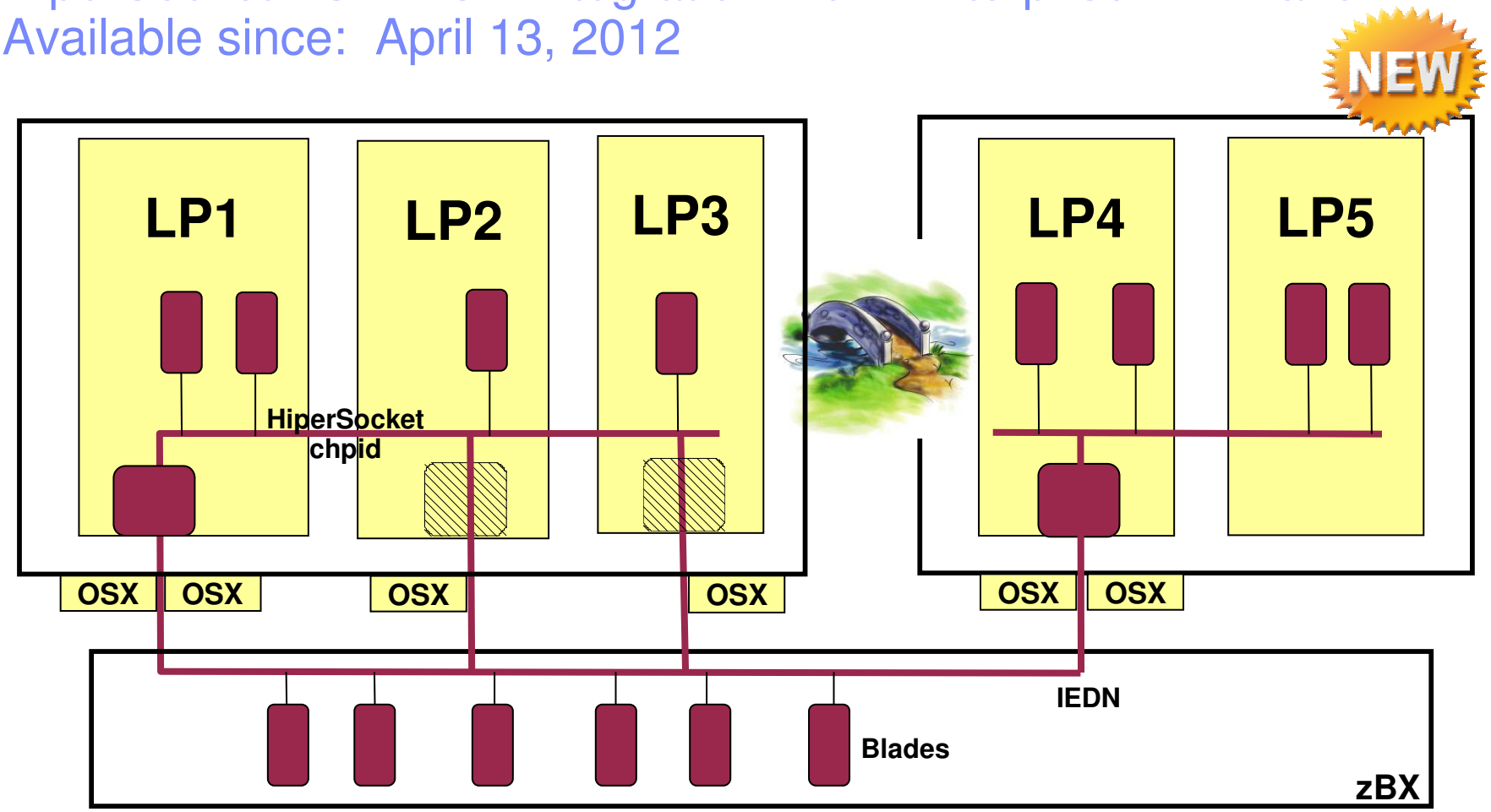
Moving guest 'A' from CEC1 to CEC2 presents a problem for maintaining contact with guest 'B' on the CEC1 hipersockets LAN segment.

VSwitch – With Hipersocket Bridge



The Hipersocket Bridge allows guest 'A' to move from CEC1 to CEC2 easily maintaining connectivity to guest 'B'

HiperSocket VSWITCH Integration with zEnterprise IEDN and zBX
Available since: April 13, 2012



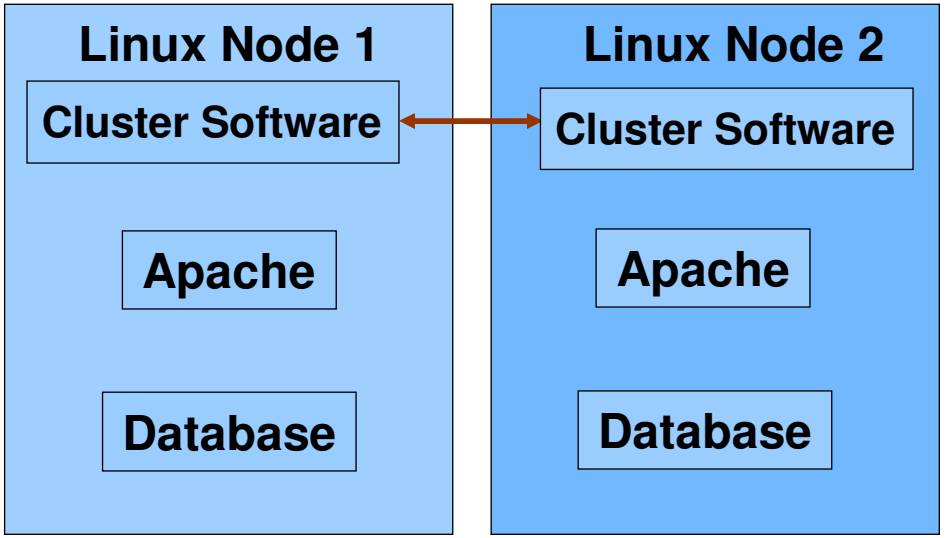
- Built-in failover and failback
- Bridge new IQDX chpid to OSX chpid
- Also works for IQD to OSD

- Same or different LPAR
- One active bridge per CEC
- PMTU simulation

High Availability for Operating Systems: z/OS and Linux

- z/OS Parallel Sysplex HA per excellence

- 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

- 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
- If this happens, then bad things will happen, 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 this
- Most common kind of quorum is voting – and only one partition can run the cluster

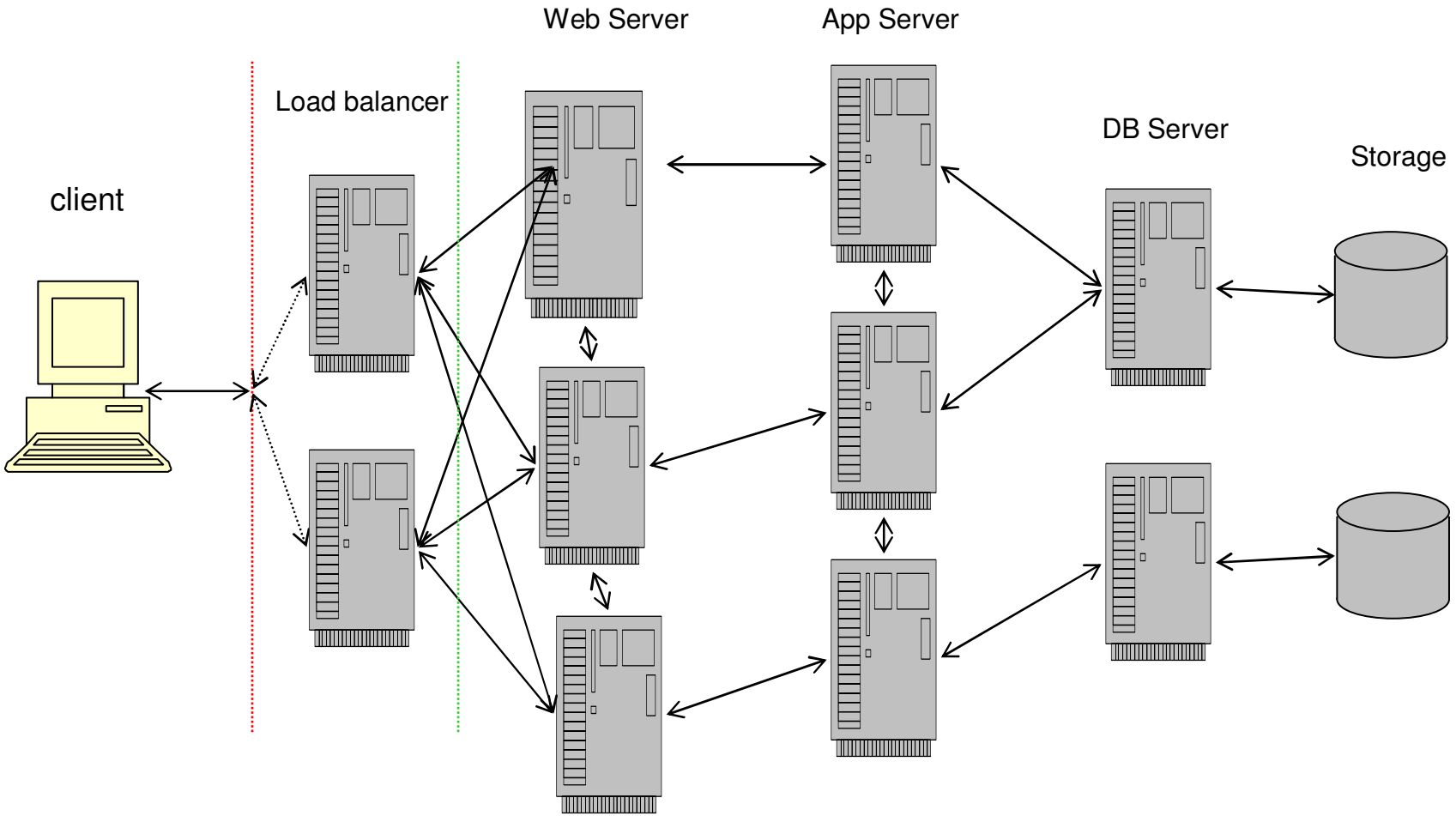
http://www.linux-ha.org/quorum

Fencing

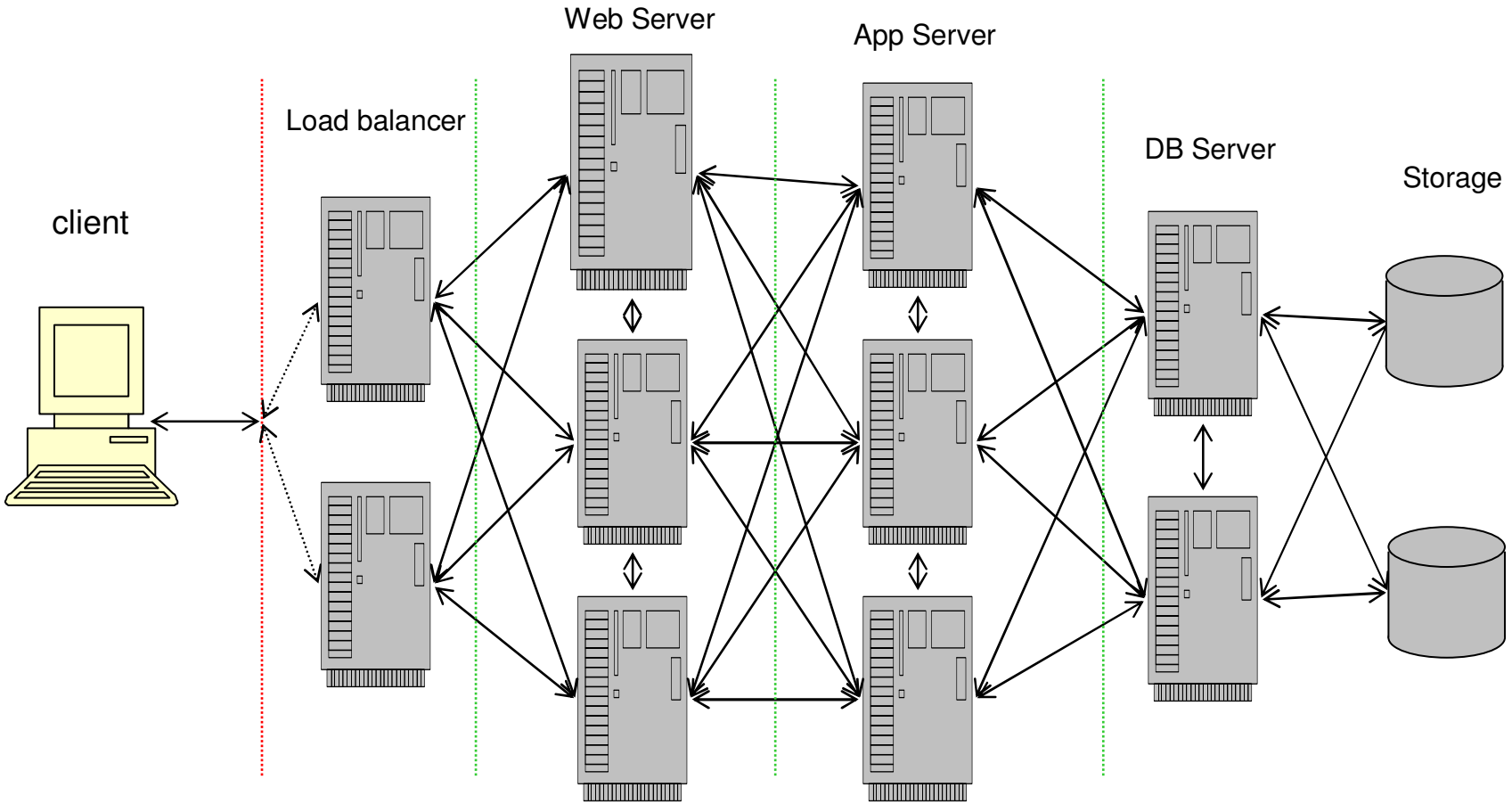
- Fencing tries to put a fence around an errant node or inhibit nodes from accessing cluster resources
- This way one doesn't have to rely on correct behavior or timing of the errant node
- This is often implemented via STONITH
  - STONITH: Shoot The Other Node In The Head
- Other techniques also work
- Fiber channel switch lockout

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

HA with Independent Complete Path Execution Streams



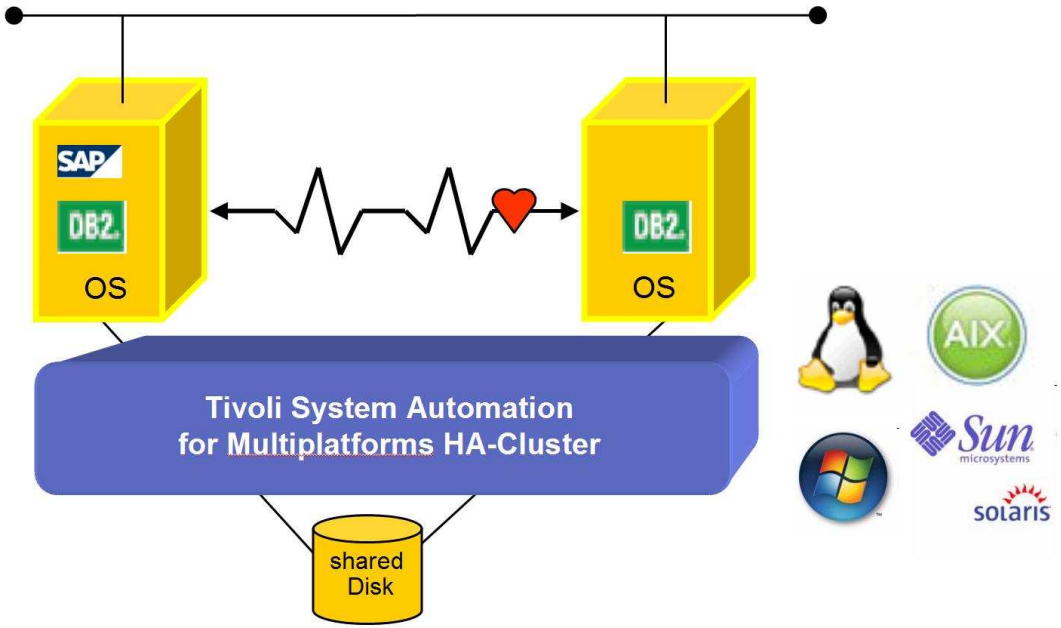
HA with Independent Tiered Execution Streams



High Availability Clustering

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

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

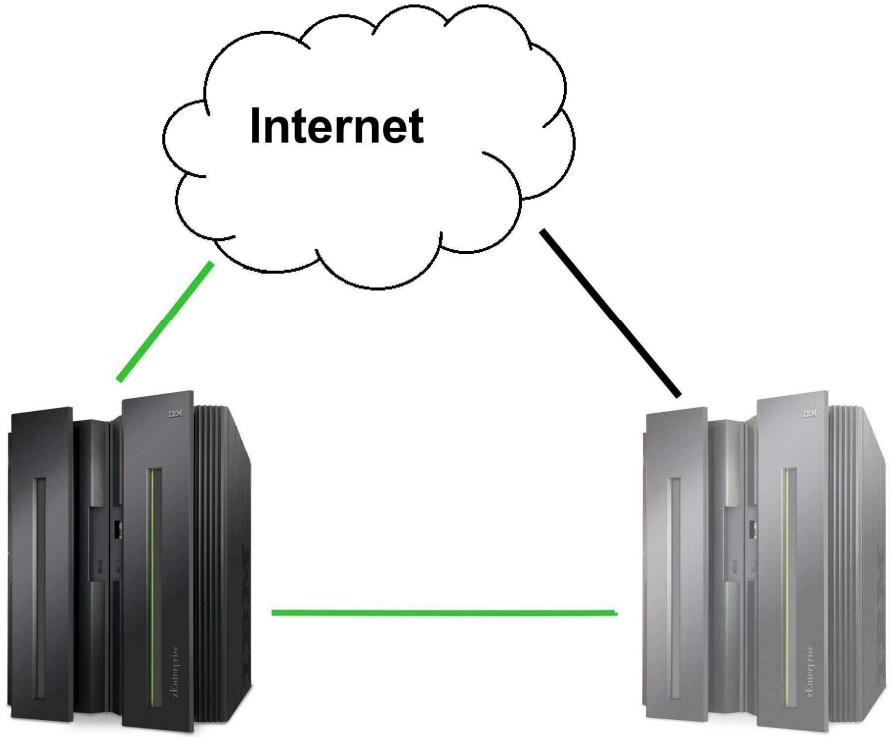


High Availability scenario as Active/Passive with System z

- **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

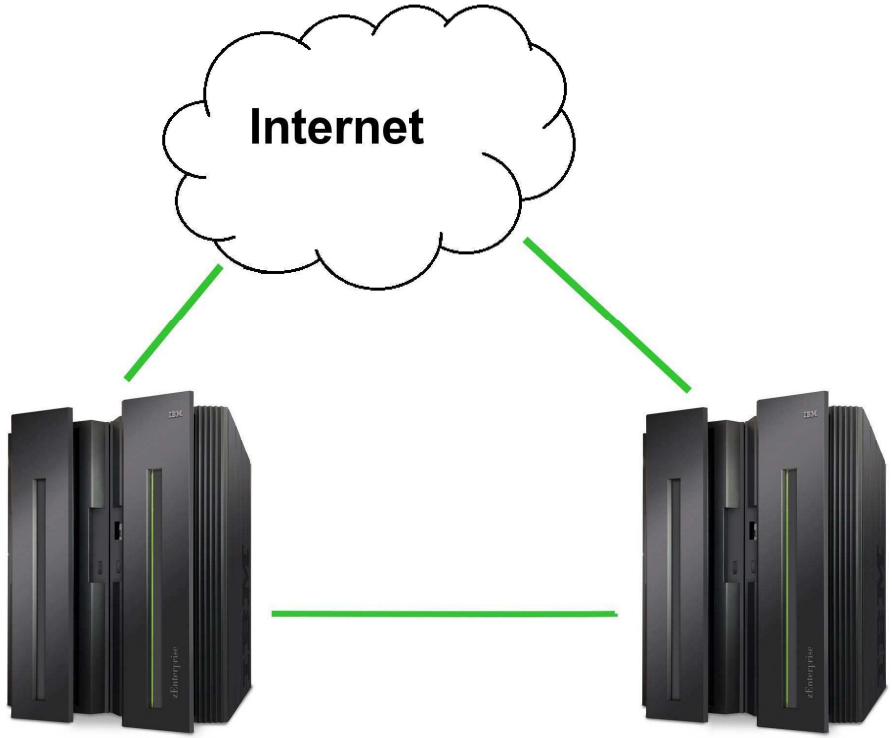
- **Hot / Warm scenario**
  - Workload is not split
  - Each site is configured to handle all operations
  - Warm environment is idling.



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 activitries
  - 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.



Linux-HA 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 RHEL Distribution for Linux on System z

- Linux clustering
  - is not implemented in the RHEL Distro for Linux on System z
  - It can be introduced by using Linux-HA packages
    - Linux-HA packages are open source and need to be adapted and compiled individually fro RHEL on Linux on System z

- 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

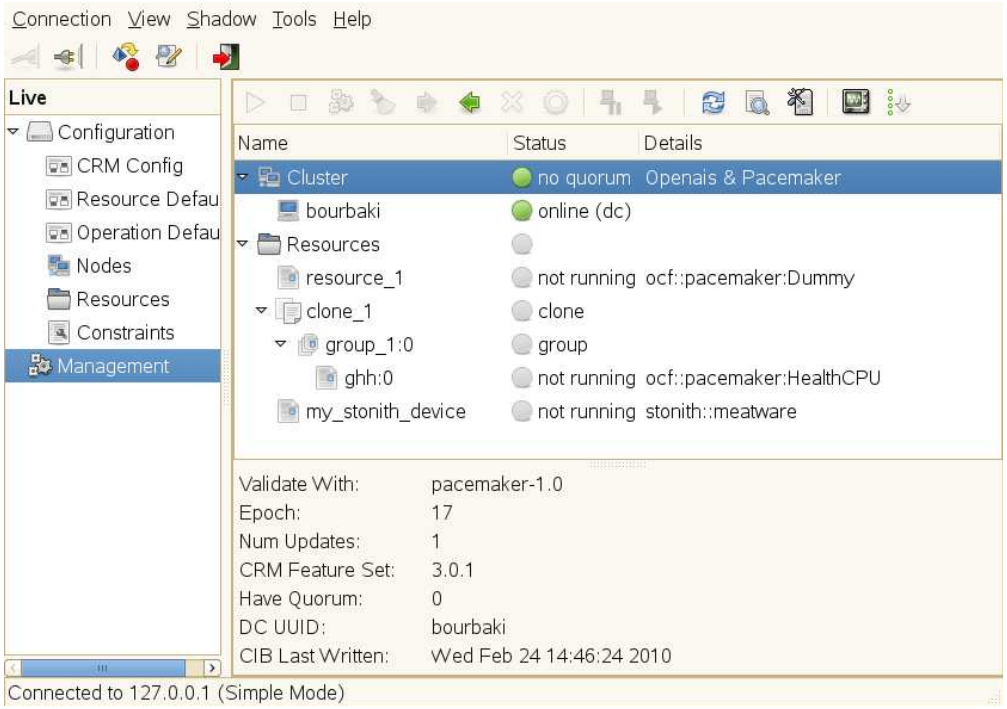
HA support in SUSE SLES Distribution for Linux on System z

- 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
- 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

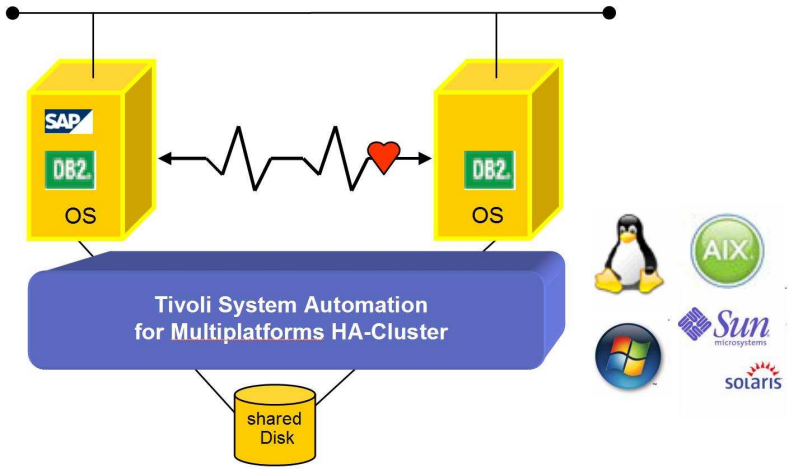
- SUSE Linux Enterprise Server 11
- for x86, x86_64,
- Itanium*,
- Power*
- System z* architectures

HA multiplatform support with IBM Tivoli System Automation

- IBM Tivoly 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.

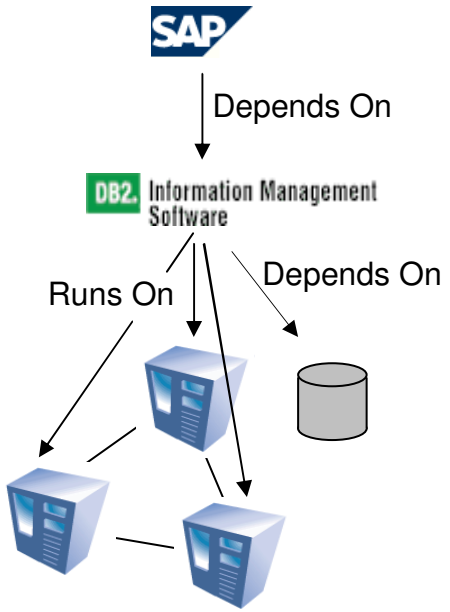


Tivoli System Automation for Multiplatforms (SA MP)

*Tivoli System Automation for Multiplatforms is a **high availability clustering solution** with advanced automation capabilities.*

It monitors and ensures availability.

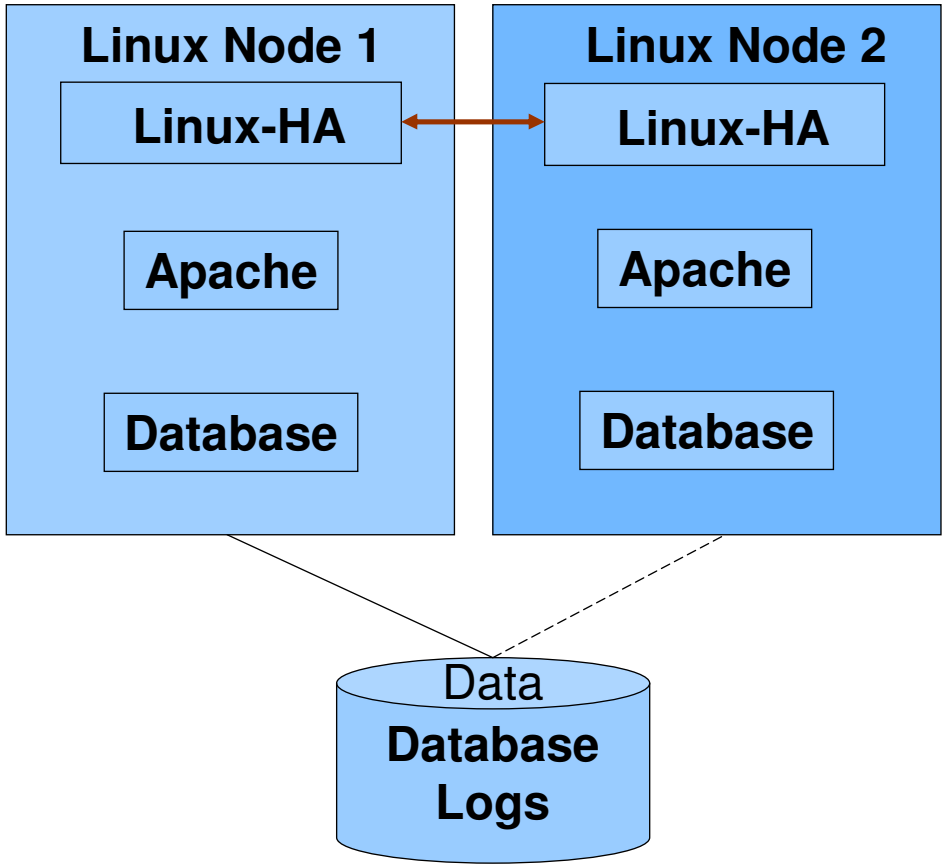
It automates the starting, stopping, and recovery of applications and application components anywhere in the cluster



Tivoli System Automation for Multiplatforms can automatically recover from …
- … hardware failures
(server down, lost storage connection)
- … software failures
(application components down)

…by the way: SA MP is often called TSA

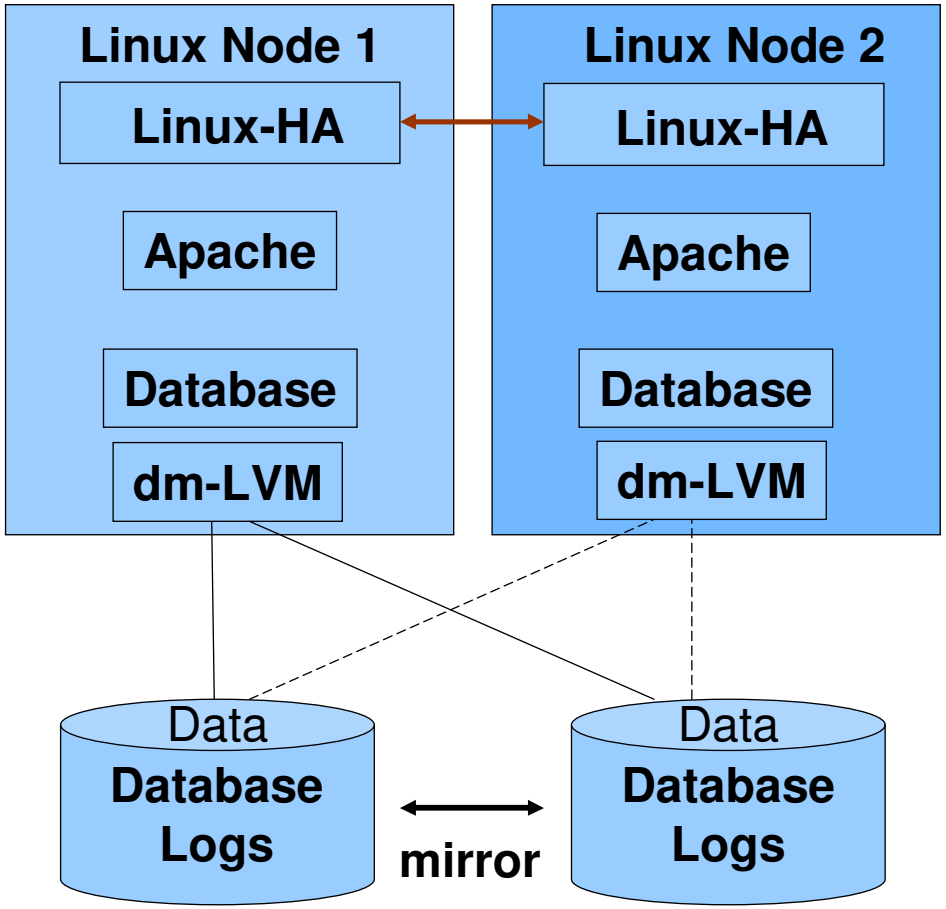
Option / Step 1: HA with Open Source and Shared Data



Linux-HA components

- **Linux-HA**
  - **Heartbeat**
- **Shared Disk**

Option 2: HA with Open Source and Linux-HA with mirrored Data



Linux-HA components

- Linux-HA
  - Heartbeat
- LVM

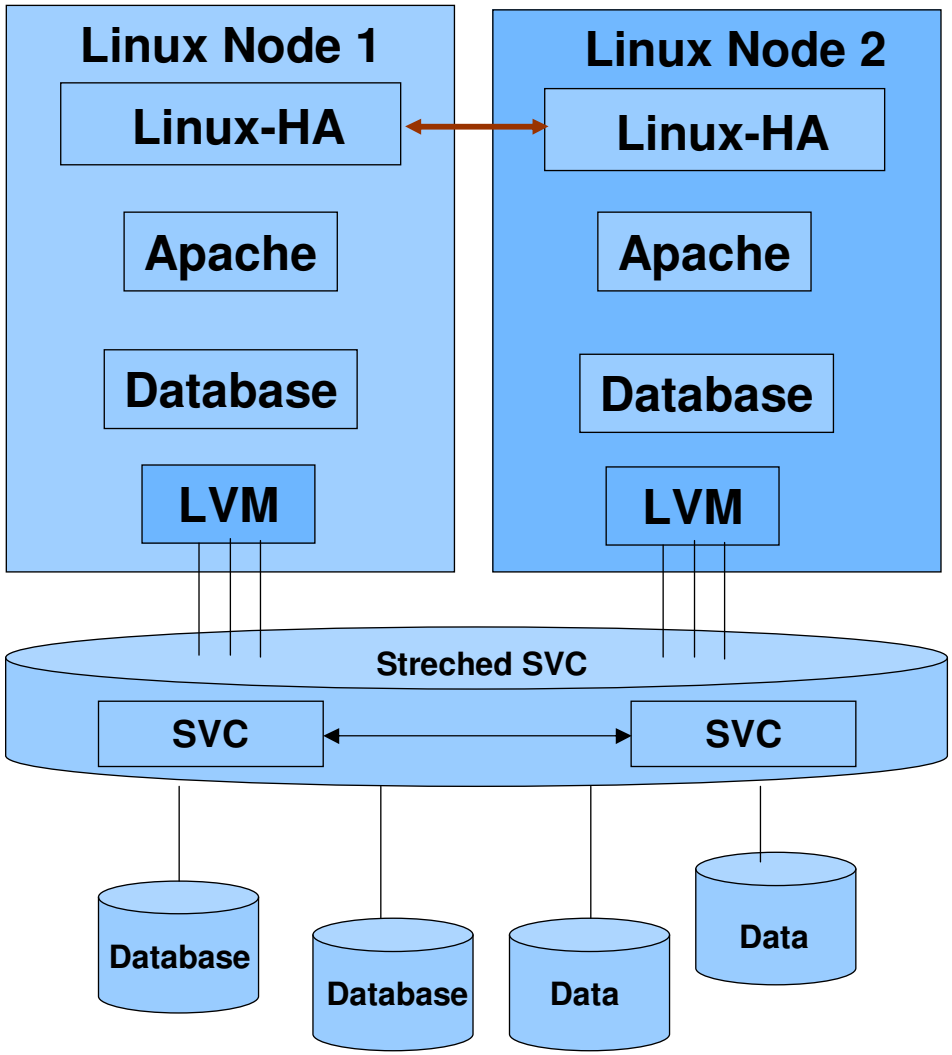
Disk HA:

- HA via Linux dm-LVM
- dm – device mapper

Linux Device Mapper (dm-LVM)

- The device-mapper driver is a kernel driver that provides a framework for volume management in Linux kernel 2.6.
- The Logical Volume Manager (LVM2) can be used as a user space interface to the device mapper.
- Volume management creates a layer of abstraction over physical storage
  - *Physical volumes* (disks) are combined into *volume groups*.
  - Volume groups are divided into logical volumes, like partitions on a disk.
  - Logical volumes are used by file systems and applications.
- To use the device-mapper driver in the kernel you need the userspace configuration tool dmsetup and the library libdevmapper.

Option 3: HA with Open Source and Linux-HA with Data HA

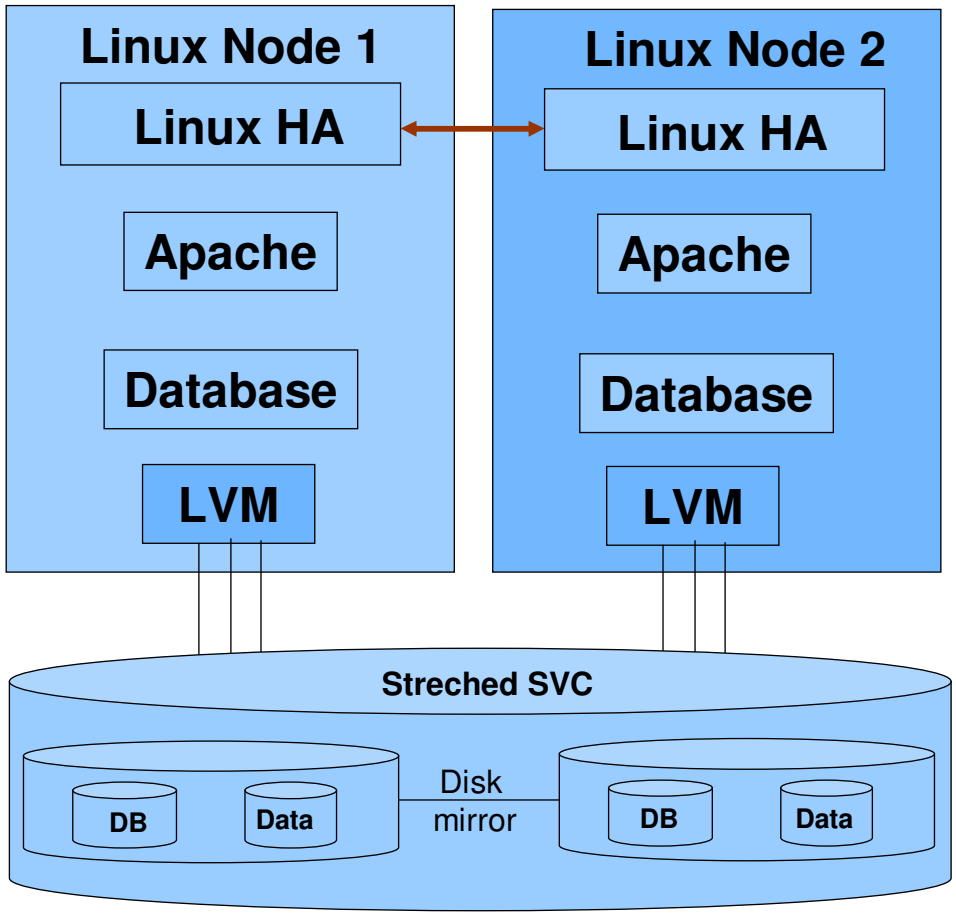


Linux-HA components

- Linux-HA
  - Heartbeat
- LVM

Disk HA:
- Concurrent update from Linux
- HA via LVM and stretched SVC (SAN Volume Controller)

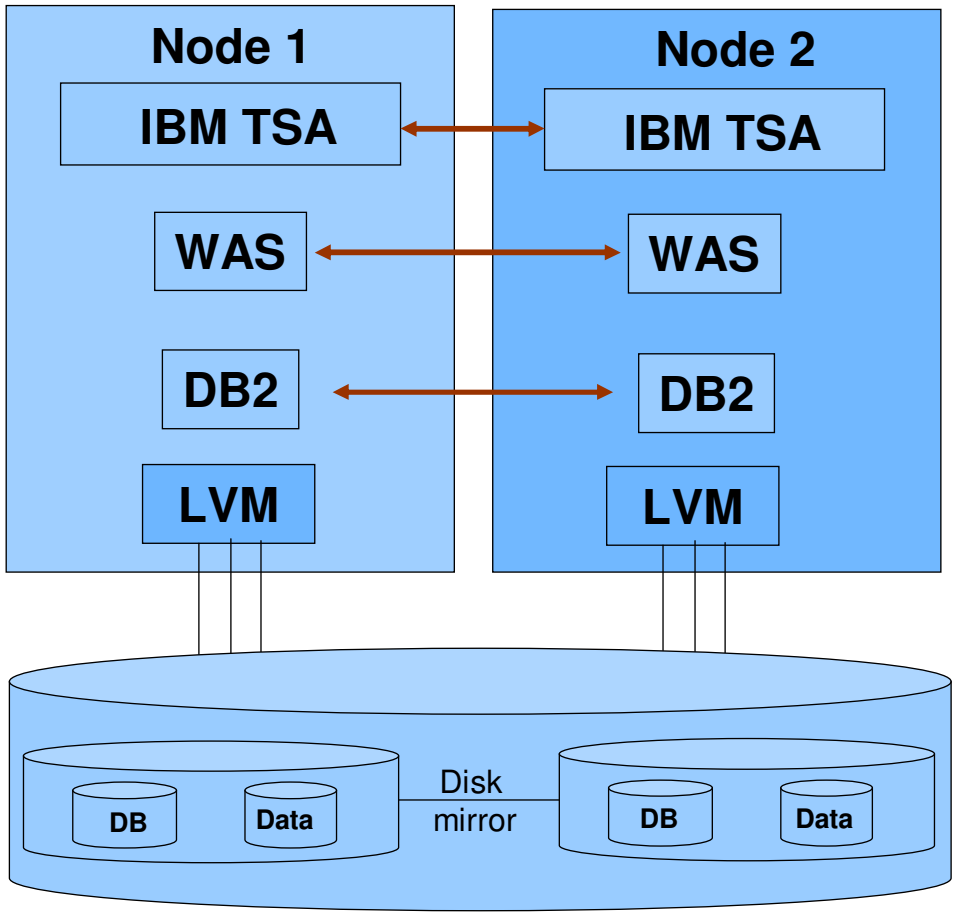
Option 4: HA with Open Source and Linux-HA with Data-mirroring



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

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

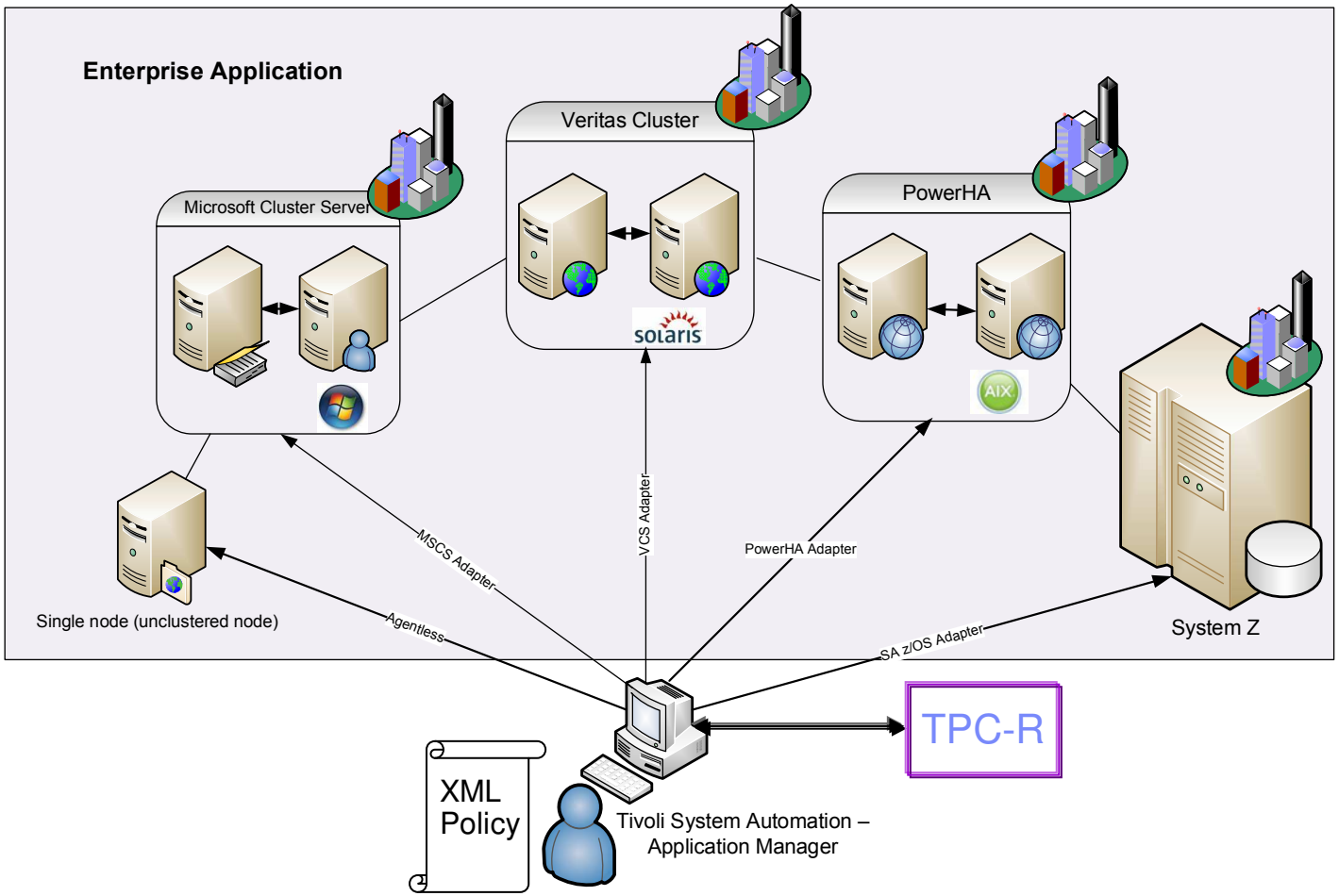
HA with IBM products clustered on each level with Data-mirroring



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

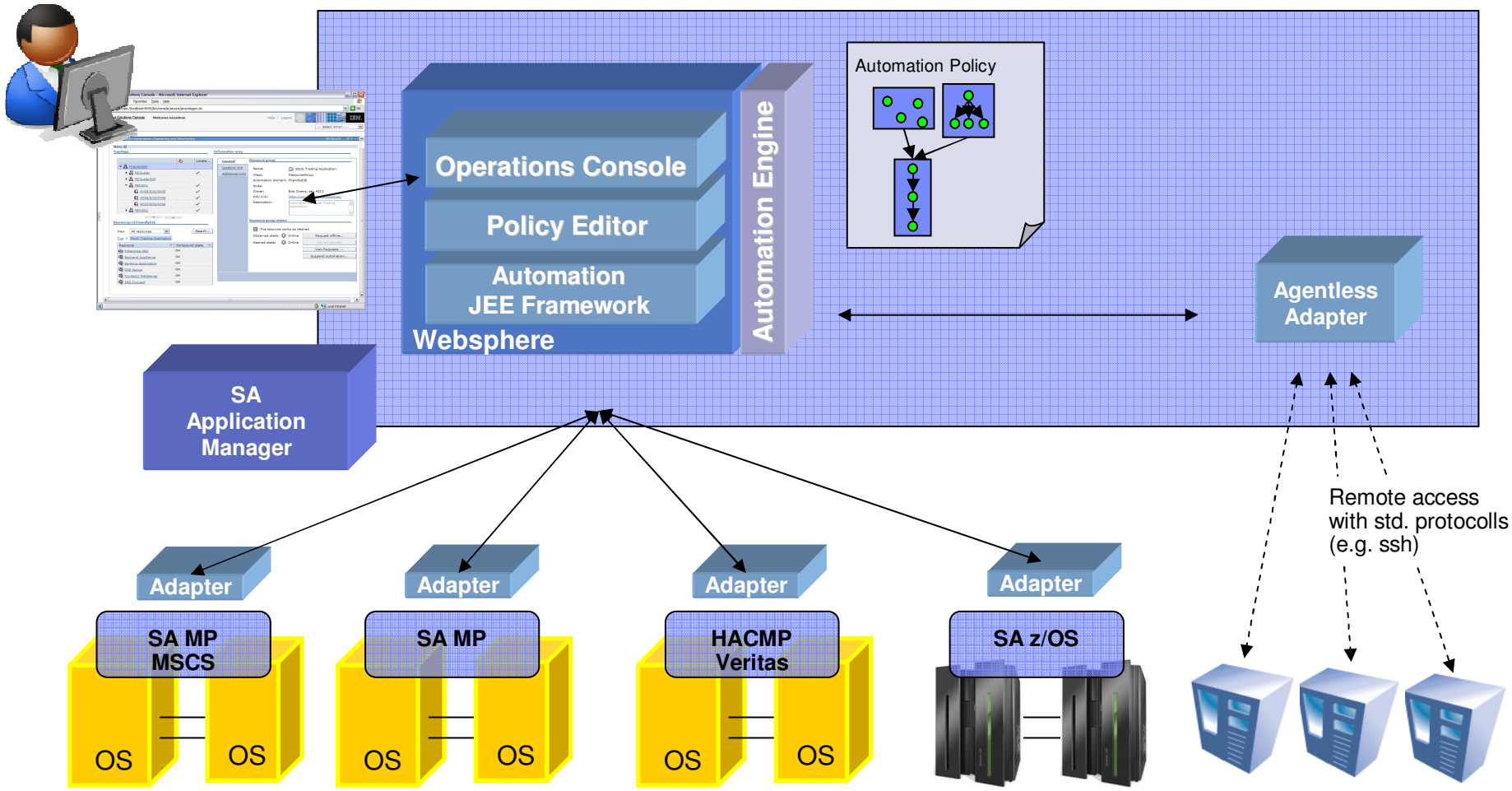
Disk HA and Virtualization:
- **HA with stretched SVC**
- **HA with IBM Storage Mirroring**

Tivoli System Automation (TSA) – Application Manager Overview



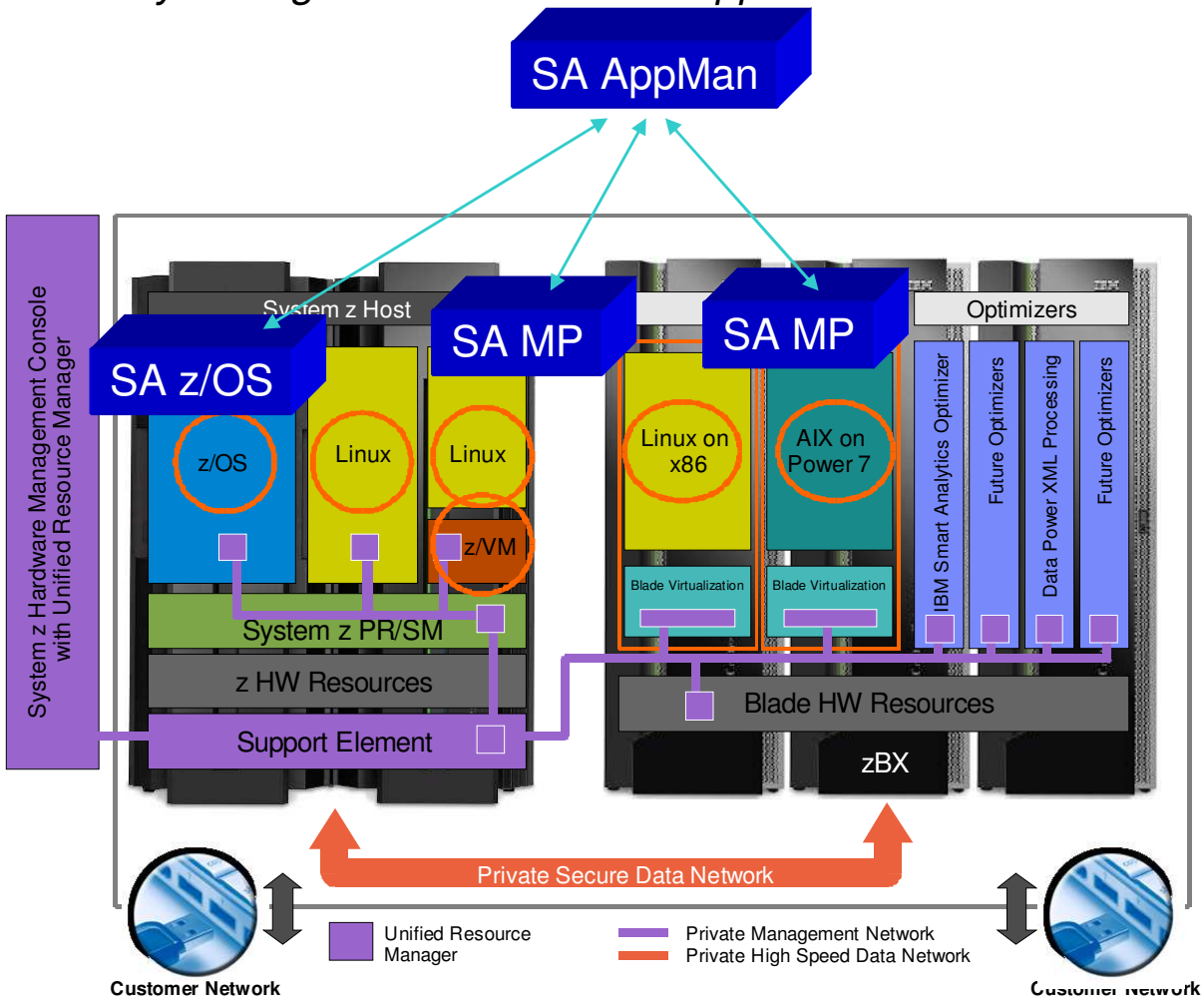
"SA MP and SA AM help us manage business applications over 200 AIX Lpar's and saves us over 3 person-years of scripting effort. We also manage application components on SA z/OS from SA AM for end to end application management" – large financial customer

SA Application Manager Adapter Infrastructure



System Automation and zEnterprise

Tivoli System Automation family provides visibility, control and automation into z/OS, Linux on System z as well as Linux and AIX running on IBM blades for automated operations and high availability of single-tier and multi-tier applications across the entire zEnterprise System

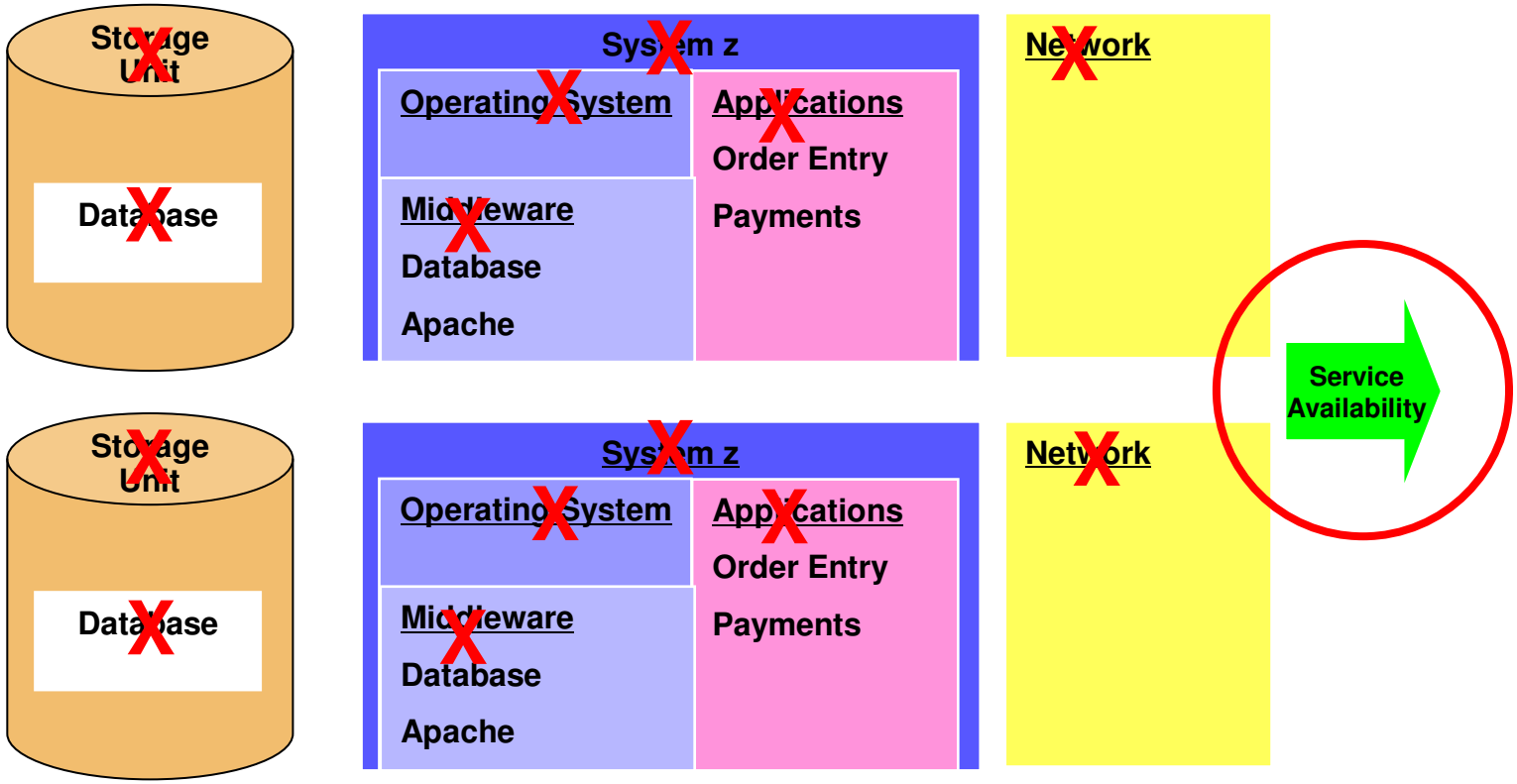


The DON'Ts for HA:

Be aware and handle with caution:

- Don’t directly re-use concepts from other cluster solutions
- Don’t use storage based mirror for high availability under cluster control
- Do not use storage based mirror with inactive paths visible
- Don’t use multipath user friendly names, but not all names pre-defined
- Avoid cluster resource, STONITH, and SBD timings shorter than SAN timings
- No stonith at all if possible
- No OCFS2 if no concurrent access is needed
- No OCFS2 on MD-RAID
- Don’t use other software like watchdog in parallel to SBD
- Don’t go live without extensive tests of planned and un-planned HA-scenarios done

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

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

- 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 IT Architect

THE Open GROUP
Master
Certified IT Architect

IBM

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

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

- Comprehensive education, training and service offerings
- Expert instructors and consultants, world-class content and skills
- Multiple delivery options for training and services
- Conferences explore emerging trends and product strategies

Special Programs:

- IBM Systems 'Guaranteed to Run' Classes -- ***Make your education plans for classes with confidence!***
- Instructor-led online (ILO) training ***The classroom comes to you.***
- Customized, private training
- Lab-based services assisting in high tech solutions

www.ibm.com/training