LINUXCARE SM

Platform Operations Concept for zSeries Linux with z/VM

Art Olbert
Linuxcare, Inc
AOlbert@linuxcare.com
415-354-4346

OPERATIONS CONCEPT

Operations Concept defines the process, roles, and approach to effectively use and support Linux on the z/VM platform

- Capabilities and key concepts of the Linux on z/VM platform
  - Building blocks of the z/VM platform?
  - Capabilities of the platform?
  - Key concepts?
- User roles
  - Who uses the platform and why?
  - Who is responsible for what?
- Operations scenarios
  - How is the platform operated?

OPERATIONS CONCEPT

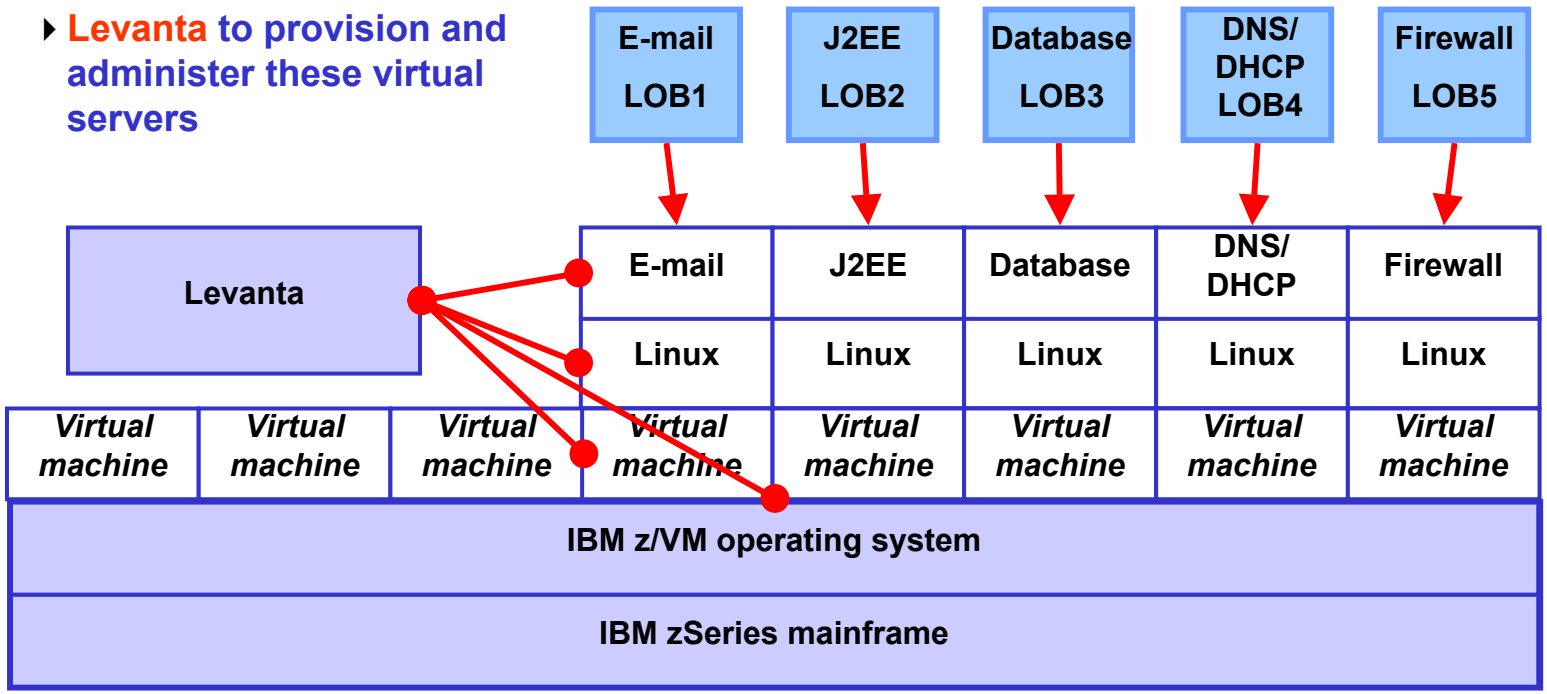
Operations Concept defines the process, roles, and approach to effectively use and support the Linux on z/VM platform

- **Capabilities and key concepts of the Linux on z/VM platform**
  - **Building blocks of the z/VM platform?**
  - **Capabilities of the platform?**
  - **Key concepts?**
- User roles
  - Who uses the platform and why?
  - Who is responsible for what?
- Operations scenarios
  - How is the platform operated?

BUILDING BLOCKS OF LINUX ON z/VM PLATFORM

The platform consists of:

- **IBM z/VM to virtualize servers running standard Linux OS and software stacks, and**
- **Levanta to provision and administer these virtual servers**



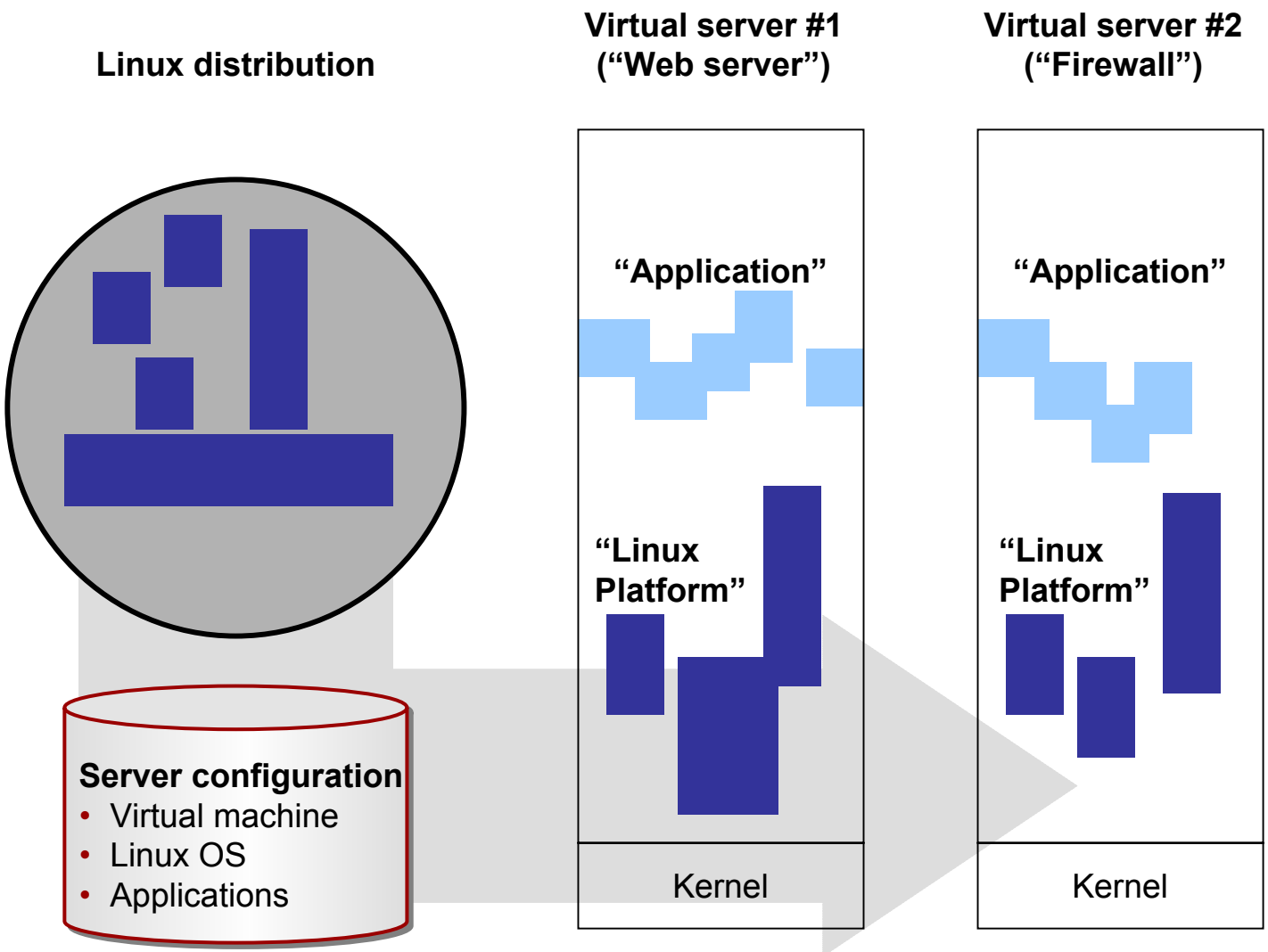
WHY USE THE PLATFORM

Platform capabilities	Example value	Typical software applications used
Provide a state-of-the-art development, test, and production environment	• Provide developers with own Linux servers on demand • Smoothly migrate software: development - test - production • Run any Linux application (without modification) • Follow mainframe standards regarding reliability and disaster recovery • Retain flexibility of decentralized Linux systems	• Linux development • Java development
Consolidate distributed servers	• Provide virtual Linux servers instead of distributed servers for use in infrastructure applications • Optimize Linux applications by leveraging mainframe capabilities • Generate a new virtual server (instance) in minutes • Allow for efficient administration of virtual servers through management software	• Domain name server (DNS) • Firewall • File/Print • Web serving • Java/J2EE • E-Mail
Web-enhance mainframe business processes	• Bring mainframe business logic and data to the Internet by using Linux z/VM and Hipersocket fast networking	• Apache web server • Java/J2EE • Database connectors

NECESSITIES FOR EFFECTIVE DEPLOYMENT OF LINUX ON z/VM PLATFORM

Platform component	High-level description	Benefit
Best practice server configs	• Description of components that define a virtual server (e.g., hdw, op sys, appls)	• Easy creation and modification of server configs • Re-use server configs in multiple virtual servers • Create (harvest) of new server configs from running virtual servers
Virtual server	• Instantiation of virtual server config, in a virtual machine	• Efficiently creat new virtual servers (in minutes) • Group of virtual servers for easier management and administration
Backup and Restore	• Sufficient data to reconstruct a virtual server, and a method to do so	• Capture of installation / config changes • Ability to restore a virtual server to consistent state • Propagate specific change(s) to group of virtual servers
Shared storage	• Location to store common, best practice server configs, binaries, and user content	• Access / share mainframe resources among virtual servers (e.g., DASD storage, processors) • Storage of arbitrary data (e.g., server configs, software components, any user file) • Simple/efficient backup & disaster recovery (DR)
Virtual machine	• Isolated / encapsulated portion of the mainframe, to run virtual servers	• Easily partition mainframe into virtual machines • Flexible & efficient mgm't of virtual machines • Strong isolation between virtual machines
Mainframe	• Physical hardware platform, and core hypervisor to operate the hardware	• Reliable / efficient operation of mainframe hardware and hypervisor (z/VM) • Hardware and software support of efficient "virtualization"

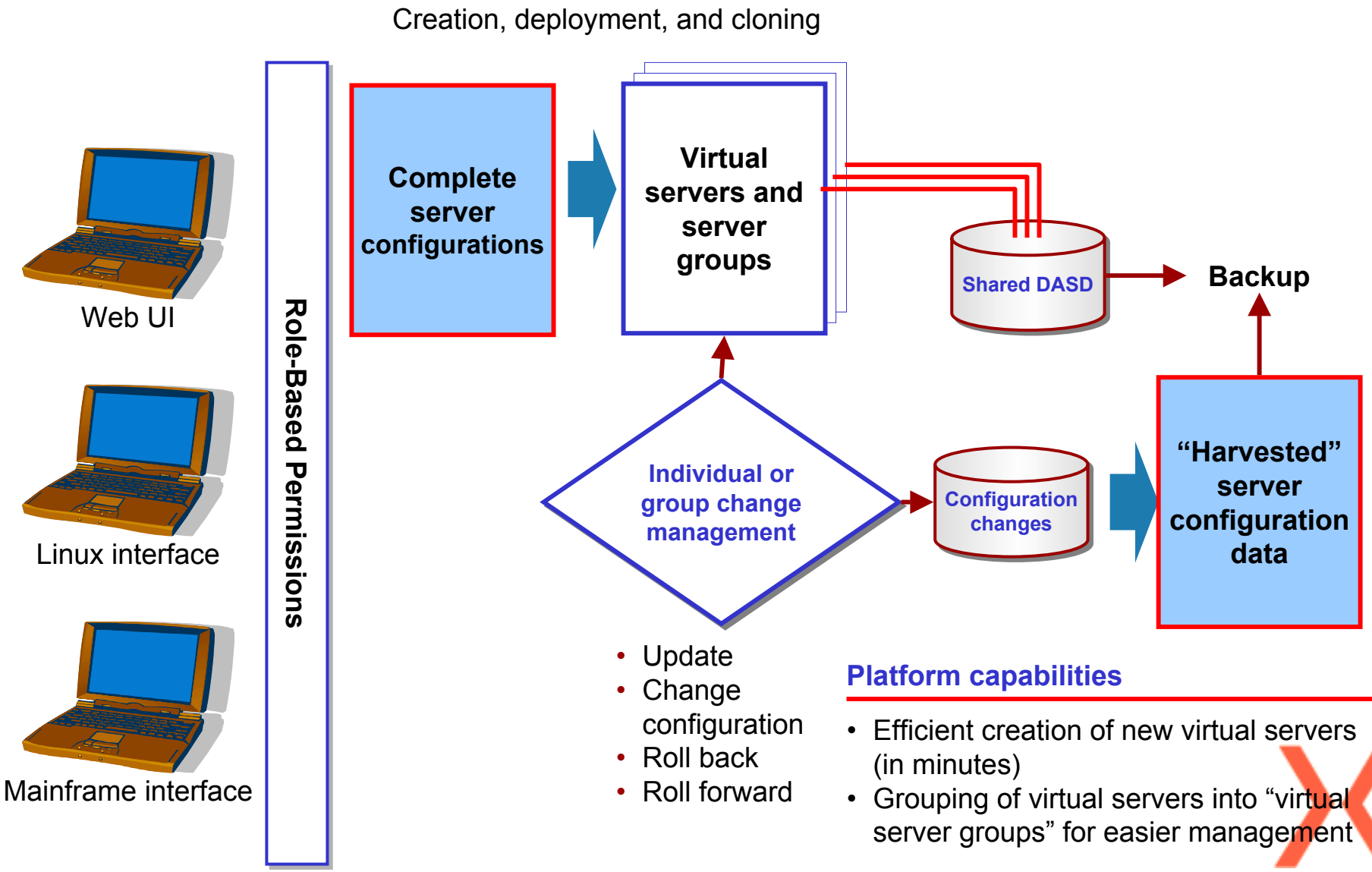
“BEST PRACTICE SERVER CONFIGURATIONS” DESCRIBE THE COMPONENTS OF VIRTUAL SERVERS



Platform capabilities

- Easy creation and modification of server configurations
- Re-use of server configurations
- Creation of new server configurations from running virtual servers

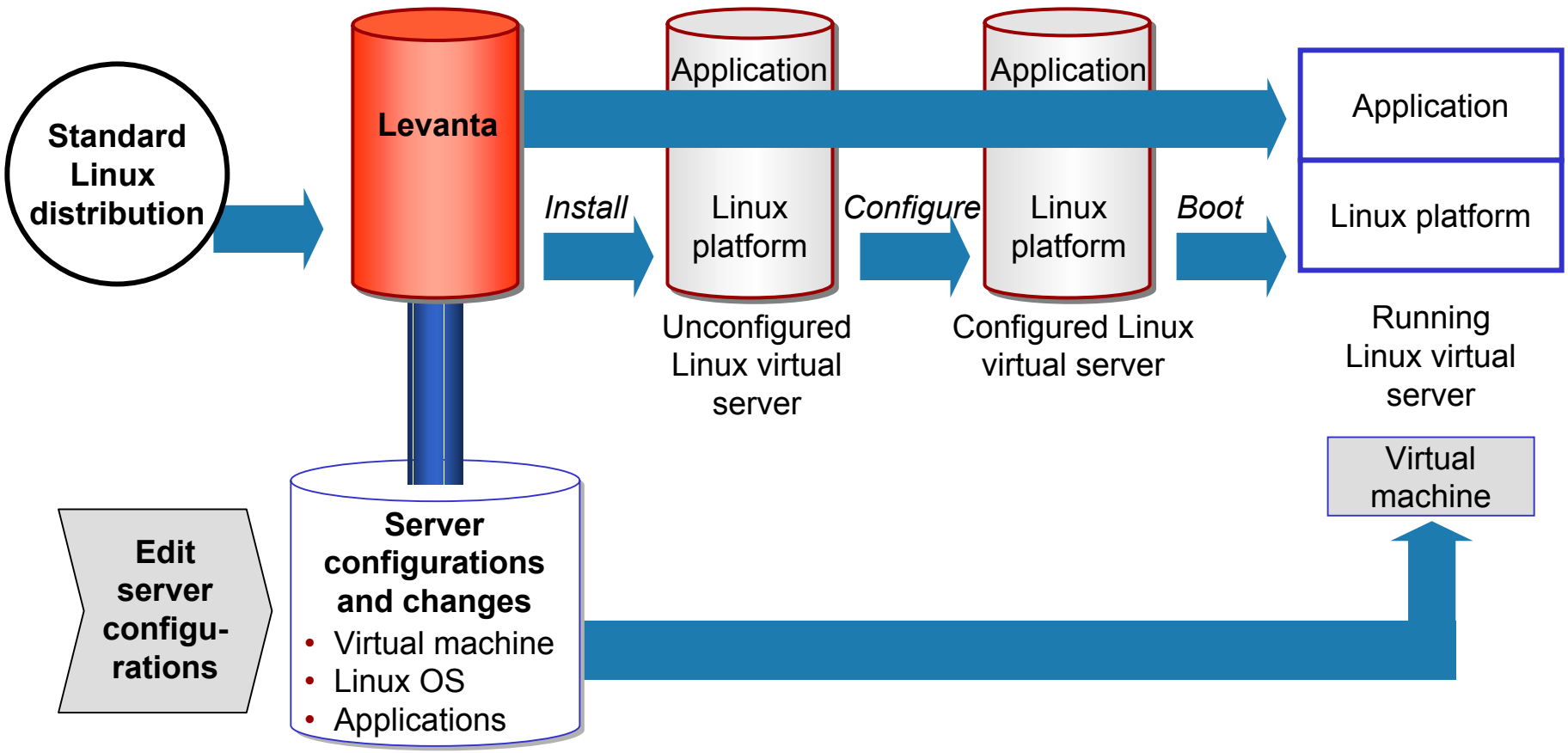
“VIRTUAL SERVERS” CAN BE CREATED FROM SERVER CONFIGURATIONS AND MANAGED IN LARGE NUMBERS



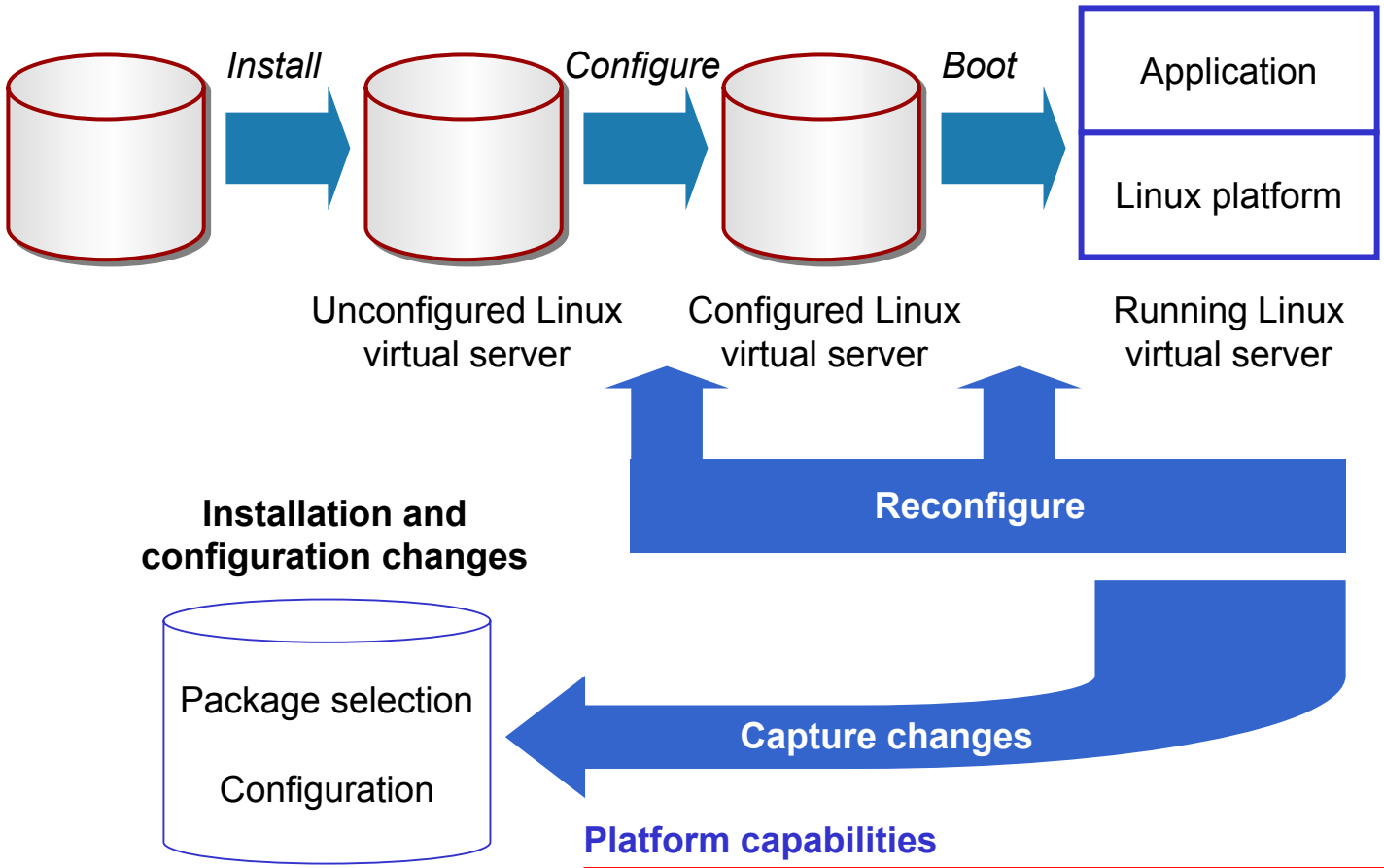
Platform capabilities

- Efficient creation of new virtual servers (in minutes)
- Grouping of virtual servers into “virtual server groups” for easier management

TAILORED LINUX “VIRTUAL SERVERS” ARE CREATED WITH LEVANTA PROVISIONING / ADMIN TOOL



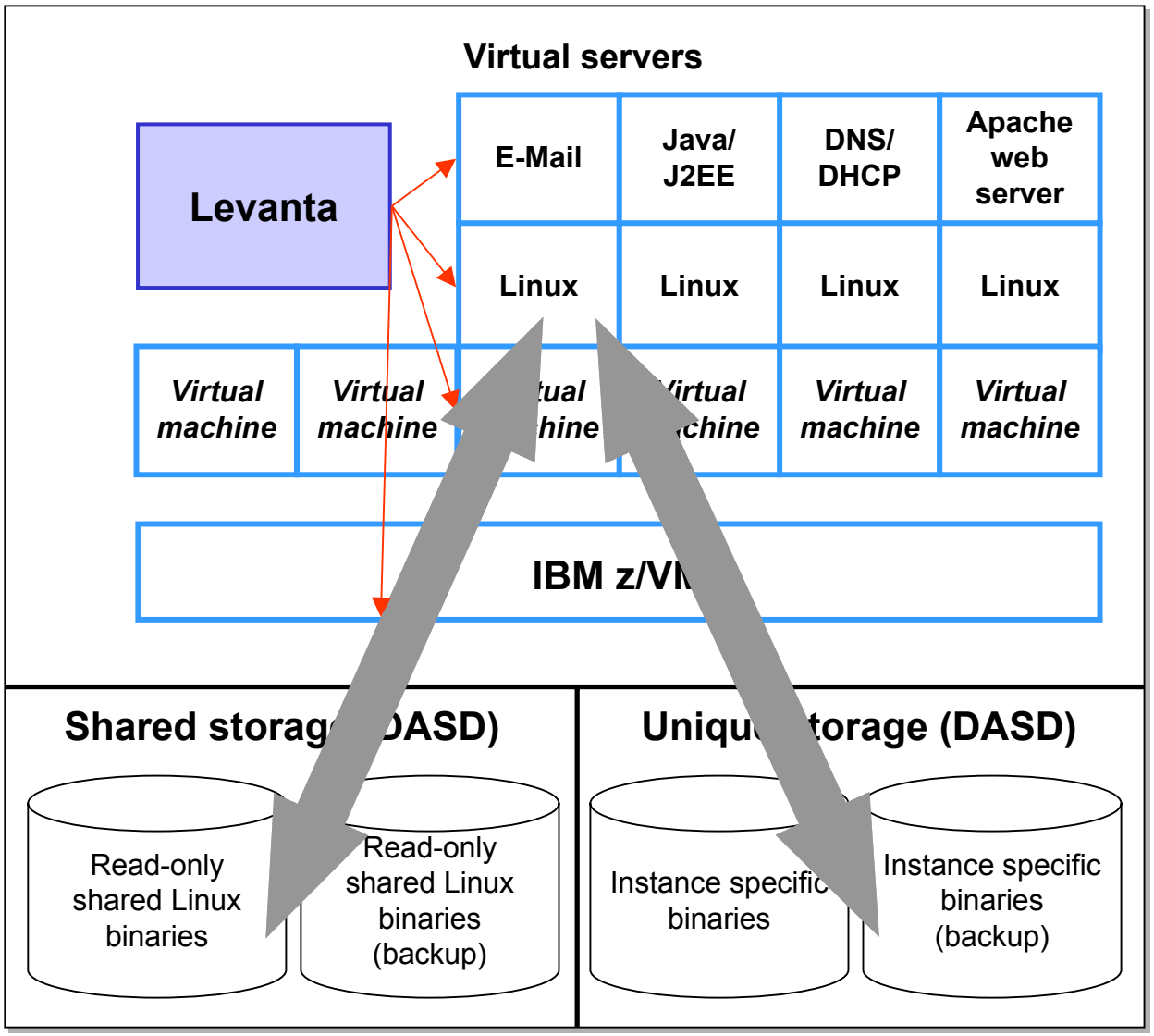
INSTALLATION AND CONFIGURATION CHANGES ARE RECORDED AND "ROLL-BACK" IS ENABLED



Platform capabilities

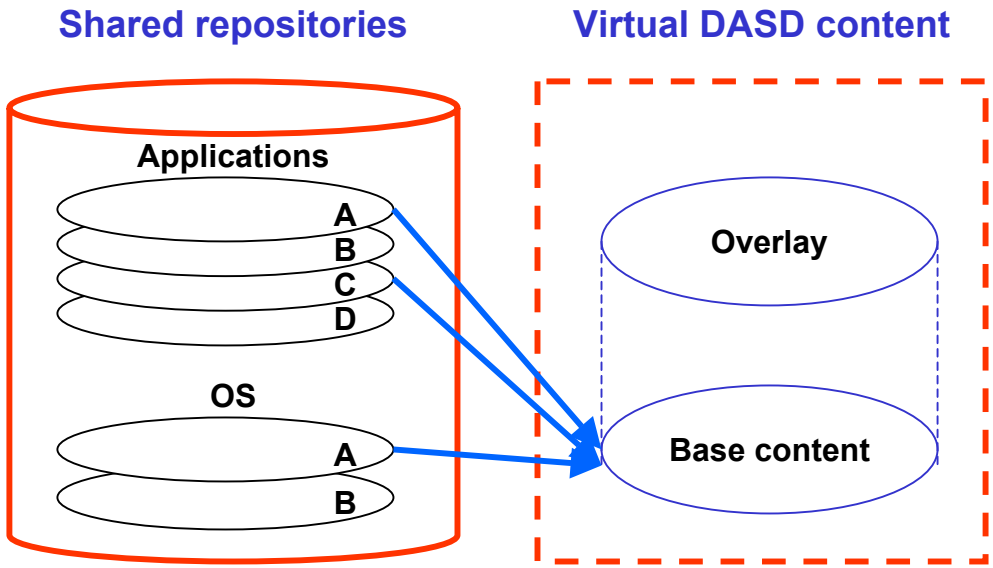
- Capture of installation and configuration changes
- Ability to always restore a virtual server to a consistent state
- Propagation of a specific change to any group of virtual servers

COMMON BINARIES ARE AUTOMATICALLY PLACED IN "SHARED STORAGE" TO SAVE SPACE AND IMPROVE MGMT



- Reduces virtual server creation time
- Simplifies backup and disaster recovery
- Simplifies considerations for mirroring and clustering
- Saves disk space
- Leverages mainframe resources

Overlays and automatic shared DASD



- When a virtual server is created by Levanta, all its files and directories are obtained from repositories
- All disks containing Levanta repositories are shared read-only by all virtual servers
- Files that are modified on a virtual server are copied to a special partition called an overlay; added and deleted files are also handled using the overlay

Benefits of overlays

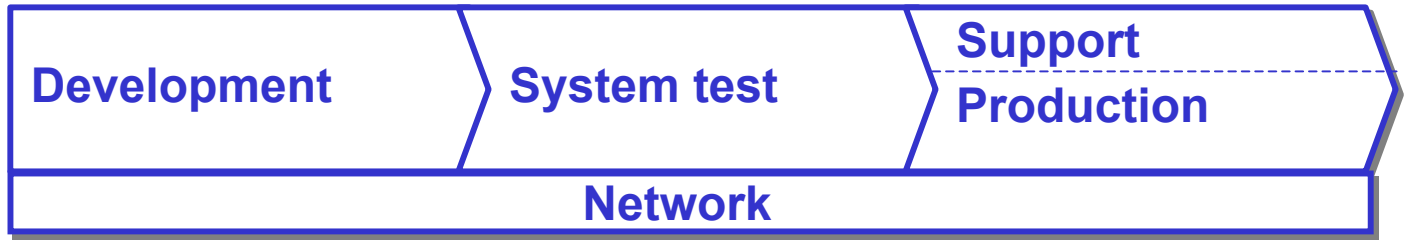
- All disks containing Levanta repositories are shared read-only by all virtual servers
- Files that are modified on a virtual server are copied to a special partition called an overlay; added and deleted files are also handled using the overlay
- Only changes are stored on DASD uniquely

OPERATIONS CONCEPT

Operations Concept defines the process, roles, and approach to effectively use and support the Linux on z/VM platform

- Capabilities and key concepts of the Linux on z/VM platform
  - Building blocks of the z/VM platform?
  - Capabilities of the platform?
  - Key concepts?
- **User roles**
  - **Who uses the platform and why?**
  - **Who is responsible for what?**
- Operations scenarios
  - How is the platform operated?

USERS OF PLATFORM BY ORGANIZATIONAL UNIT



Sample requirements

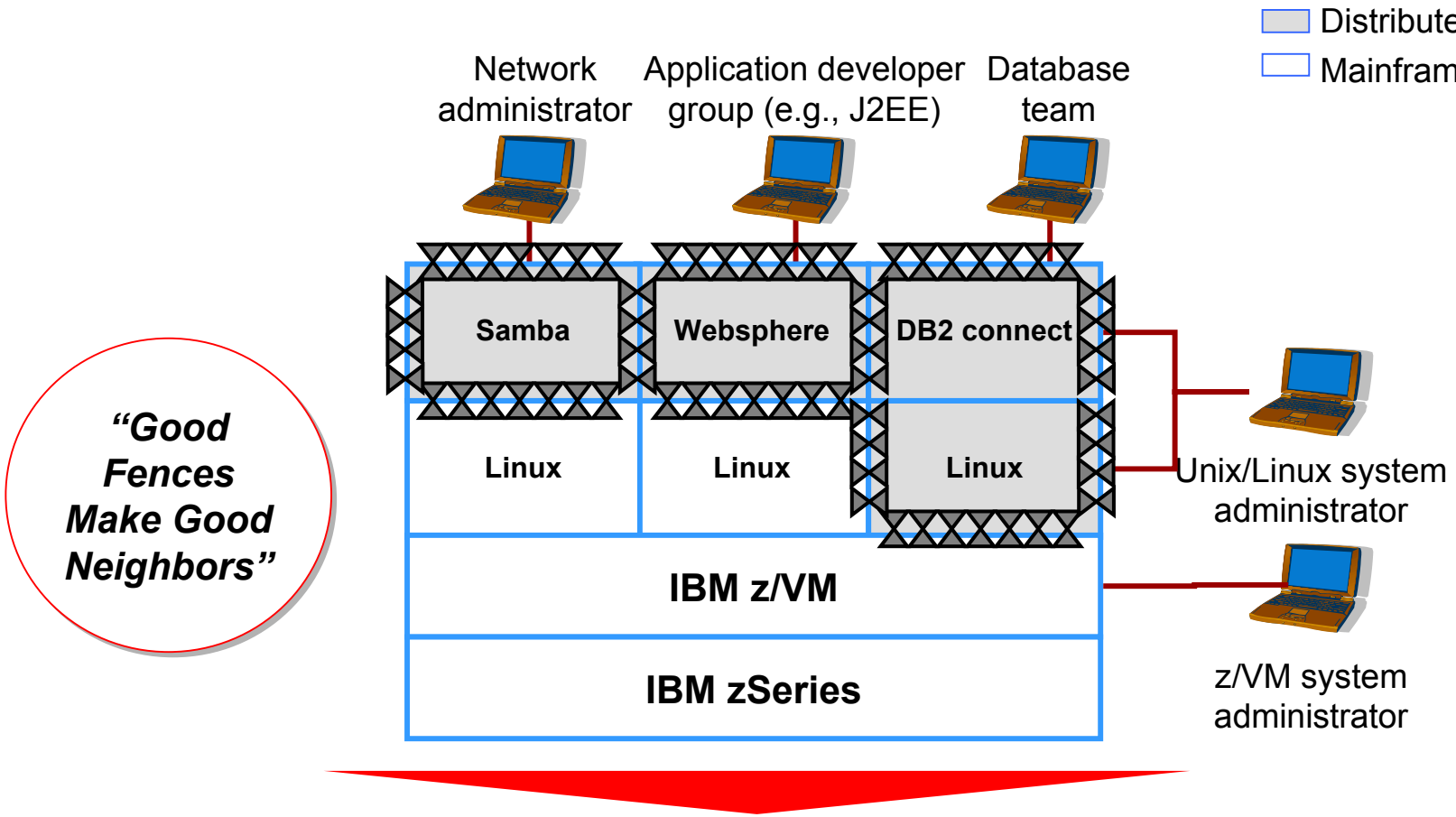
- Provision / reconfigure *dozens* of experimental servers
- Develop specialized and optimized software versions for various virtual servers configs
- Requires *perfect* copies of production environment for testing
- Create fine-tuned *variations* of production servers, with differing workload & user profile
- Quick migration and deployment of new servers from test to production
- Capacity planning asks for more servers in response to growing system load

Typical tasks performed

- Create virtual server
  - Provision virtual machine (e.g., memory, DASD storage, CPU)
  - Install OS and application
  - Change server configs
  - Cycle/upgrade the OS
  - Re-configure virtual machines
- Develop and test applications
  - Modify configs to debug
  - Track changes to configurations
  - Tune configurations
  - Configure network
- Deploy virtual servers
  - Install multiple images
  - Deploy image updates
  - Capture changes
- Manage virtual servers
  - Roll back / forward changes
  - Backup and restore servers
  - Grow / shrink server numbers
  - Define standards, security guidelines and service level agreements (SLAs)
- Support users
  - Capture/resolve problems
  - Train users

DIFFERENT TEAMS COLLABORATE ON THE PLATFORM

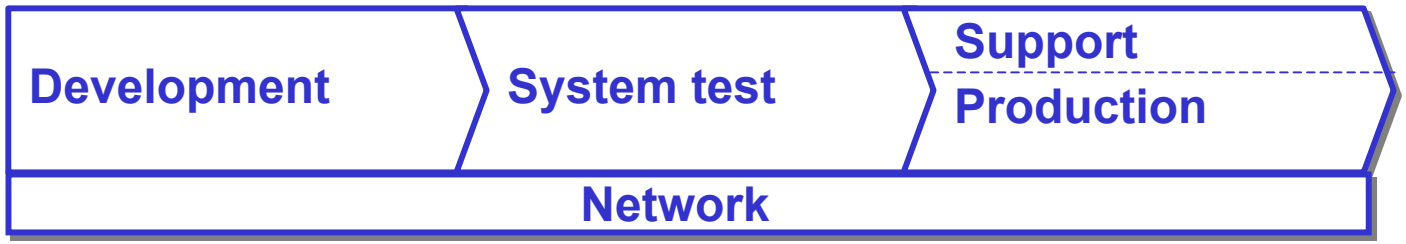
Mainframe teams work with distributed teams



Linux zSeries with Levanta fosters collaboration among different teams

- Effect self-service model through roles, enforced by access permissions
  - Enable defined interaction with system and provide protection
  - Logs capture changes and allow for rollback
- Reduce cultural barriers by providing multiple, familiar interfaces

ROLES ON PLATFORM (1/2)

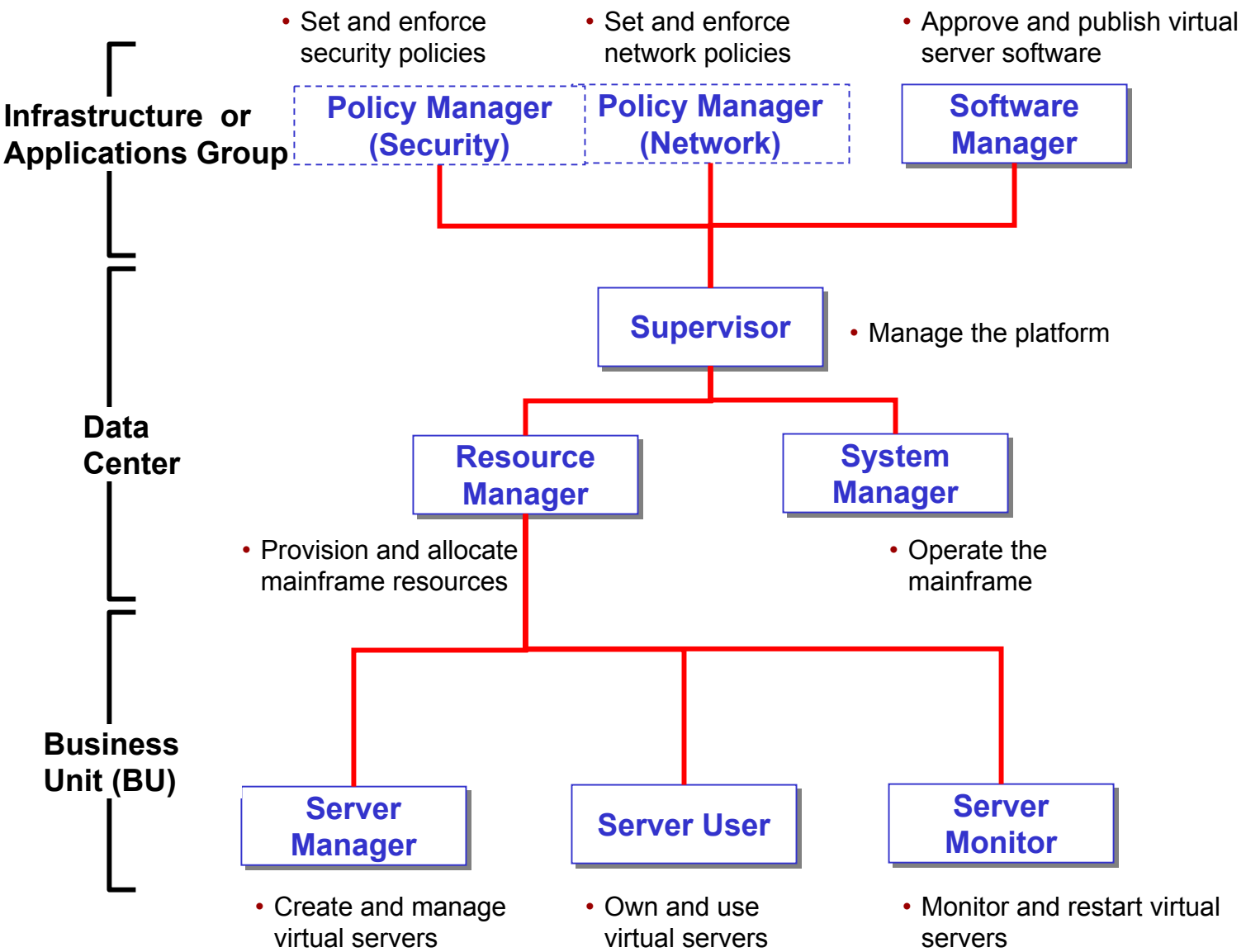


Platform Roles	Played by Whom		
Server Monitor *("monitor / restart virtual servers")*			• Server operator • Basic support of users
Server User *("own / use virtual servers")*	• Application developer – Web applications – Linux applications	• QA test engineer, e.g. unit tests, dependency checks	• Server admin - basic ops • Data base admin
Server Manager *("create / manage virtual servers")*	• Linux server mgr/admin for a development team • Sys prog with ability to create modify servers	• Linux server mgr/admin for unit test team •Sys level QA/test engineer, able to create / modify servers	• Server mgr - advanced operations e.g. restore virtual server (DR) • Database mgr (advanced) • Advance/expert user support
Software Mgr *("approve / publish virtual server")*	• Appl development / certification mgr	• Sys test / final release manager	• Mainframe res mgr (e.g allocate DASD, plan consolidation
Resource Mgr *("provision / allocate mainframe resources")*			• Mainframe operator • Add / config hdw
System Mgr *("operate mainframe")*			• Head of platform • Oversee operations
Supervisor *("manage platform")*			

ROLES ON PLATFORM (2/2)

Platform role…		…Key activities and capabilities…	…Played by Whom
Server Monitor	*"Monitor and restart virtual servers"*	• Monitor and operate virtual servers • Restart virtual servers if needed (e.g., upon failure)	• Virtual server operator (monitoring) • First-level support (FLS), e.g., – Basic support of a user group
Server User	*"Own and use virtual servers"*	• Start and stop virtual servers • Perform routine operations (e.g., daily backup of virtual servers) • Use virtual servers ("produce")	• End user (e.g., application developer, QA/test engineer, production operator) • Virtual server administrator (day-to-day) • Database administrator (day-to-day)
Server Manager	*"Create and manage virtual servers"*	• Procure, create and administer virtual servers (through self-service) • Perform advanced virtual server operations (e.g., disaster recovery) • Provide advanced/expert support	• Virtual Linux server manager/admin • System-level programmer • System-level QA/test engineer • Database manager (advanced) • Second-level support (SLS)
Software Manager	*"Approve and publish virtual server software"*	• Develop new applications and define valid server configurations • Approve software configurations for system testing and production	• Application development, configuration and certification manager • System test manager (for application release to production)
Resource Manager	*"Provision and allocate mainframe resources"*	• Manage virtual mainframe resources, e.g., – Provision and allocate resources (for later self-service by server managers) – Manage mainframe resource pool	• Mainframe resource manager • Project manager for server consolidation
System Manager	*"Operate the mainframe"*	• Perform basic mainframe operations (non-virtual machine view), e.g., – Install mainframe/VM environment – Add and configure new hardware	• Mainframe operator (day-to-day) – Mainframe and VM installation – New hardware addition and configuration
Supervisor	*"Manage the platform"*	• Oversee Linux on z/VM operations • Provide overall direction and guidance	• Head of data center • Manager of technical services

LINKAGE BETWEEN ROLES AND IS / IT DEPARTMENTS



Note: Several roles may be held by the same individual

OPERATIONS CONCEPT

Operations Concept defines the process, roles, and approach to effectively use and support the Linux on z/VM platform

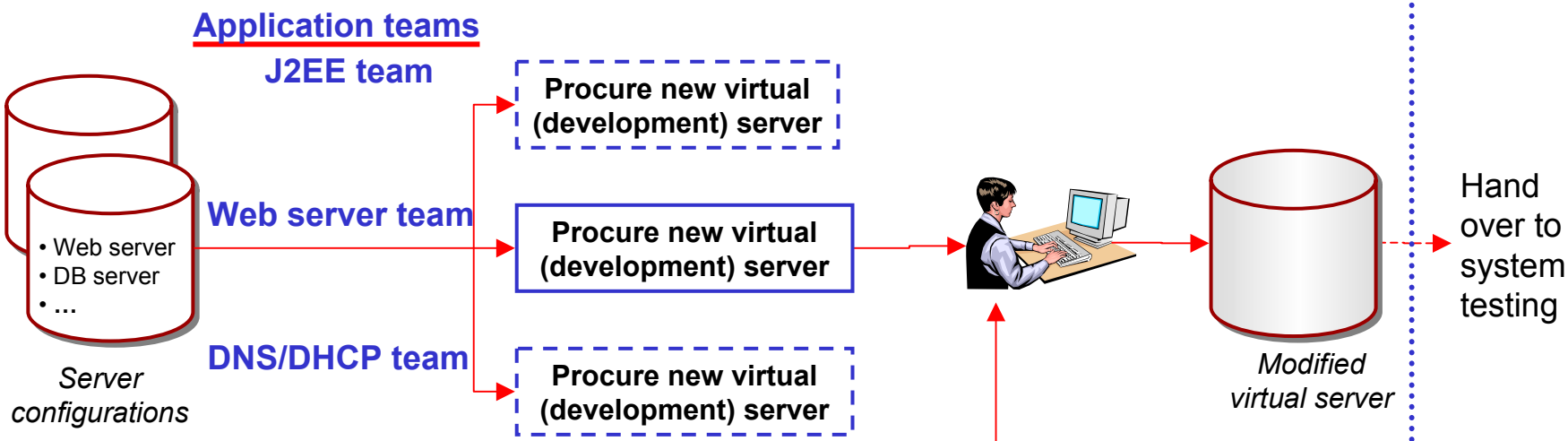
- Capabilities and key concepts of the Linux on z/VM platform
  - Building blocks of the z/VM platform?
  - Capabilities of the platform?
  - Key concepts?
- User roles
  - Who uses the platform and why?
  - Who is responsible for what?
- **Operations scenarios**
  - **How is the platform operated?**

OPERATIONS SCENARIOS

Scenario	Key activities	Involved parties	Platform roles
Software Development	• Develop software applications • Define and approve valid software configurations	• Application developer • Application certifier	• Server User • Software Manager
System Testing	• Test software, using defined test scenarios and procedures • Define/publish test guidelines	• Quality assurance (QA)/ test engineer	• Server User • Server Manager (if needed)
Software Deployment	• Ensure rapid rollout/upgrade/staging of OS and applications (on a large number of virtual servers)	• QA/test engineer • Production team	• Server User • Server Manager (if needed)
Software Change Management	• Ensure smooth transition from development to test to production	• Application developer • QA/test engineer • Production team	• Server User • Server Manager (if needed)
Backup/Restore and Disaster Recovery	• Perform regular backup of virtual servers for recovery purposes • Restore virtual server if needed	• Virtual server admin • Virtual server manager • Database manager	• Server User • Server Manager
Resource and Capacity Management	• Provision new mainframe resources, in response to growing system load • Re-distribute load to newly created virtual servers	• Mainframe resource manager • Virtual Linux server managers/administrators	• Platform Manager • Resource Manager • Server Manager
Problem Management	• Report and escalate user problems • Analyze and resolve problems	• End user • Support and help desk • Development (if needed)	• Server User • Server Monitor • Server Manager
Server Consolidation	• Re-create a group of physical servers on the mainframe	• Head of data center • Resource manager • Virtual server manager	• Resource Manager • System Manager • Server Manager

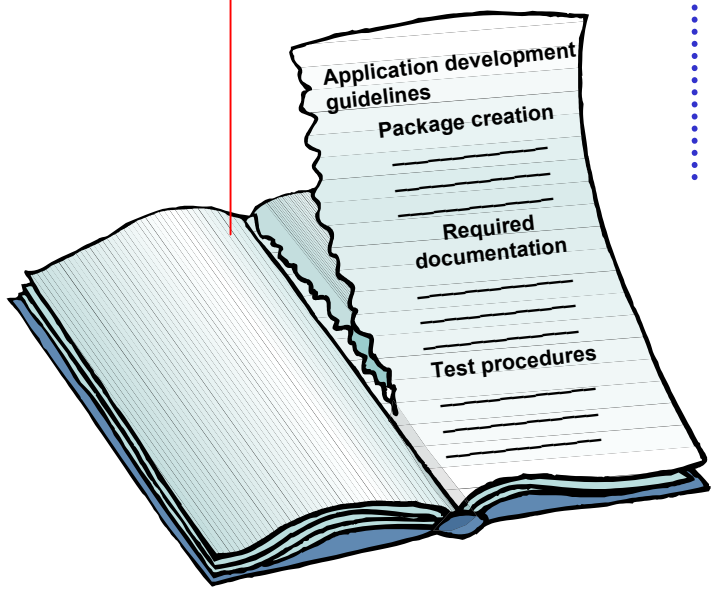
SOFTWARE DEVELOPMENT

Scenario: Rapid provisioning of virtual servers for development (e.g., 10 different configs)



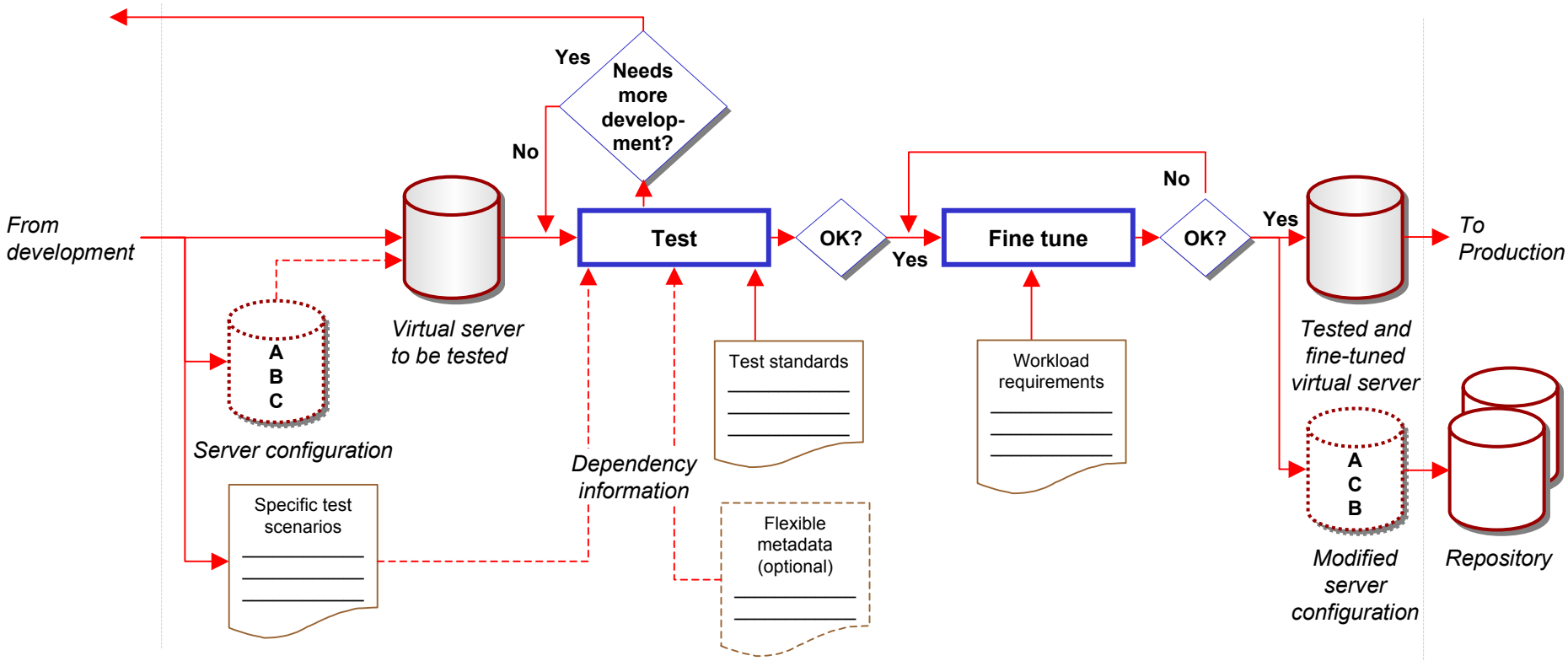
Best practice approach

- Application developer teams request a new virtual test server for development purposes
- Software development is guided by standard application development guidelines, e.g.,
  - Rules for creation of RPM packages (name conventions, installation/configuration scripts)
  - Required documentation (e.g., manual pages, user guide, operations manual)
  - Test procedures and scenarios
- Well-defined interface to system testing is in place (e.g., documented test procedures)



SYSTEM TESTING

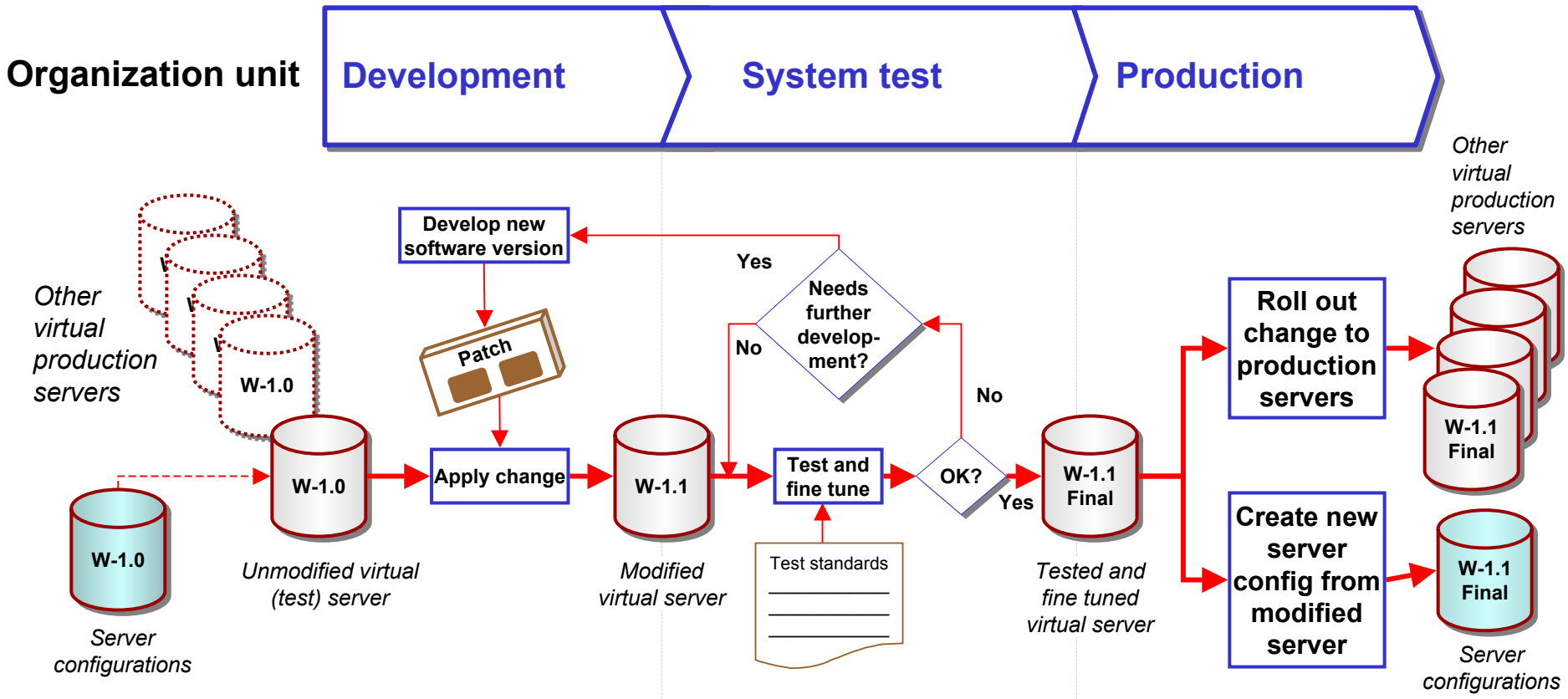
Scenario: Testing and certification of a procured (or self-developed) software application



Best practice approach	• Obtain virtual server to be tested from application development group • Ensure documentation is complete (e.g., manual, test scenarios)	• Test the virtual server using standard test guidelines • Run virtual server through the specific test scenarios provided • (Optionally) augment dependency information stored in server configuration through "flexible metadata"	• "Fine tune" the virtual server, e.g., – System parameters – Workload profile	• Capture changes to the fine-tuned virtual server by harvesting (i.e. creating) a new server configuration • Hand over tested and fine-tuned virtual server (and server configuration) to production
Platform support	• Instances *(virtual servers)* • Templates *(server configurations)*	• Instances • Flexible metadata (containing dependency information)	• Templates • Change management	• Harvesting an instance • Repository *(data storage)*

SOFTWARE CHANGE MANAGEMENT

Scenario: Upgrade of all existing production web servers to a new version of Apache



Best practice approach

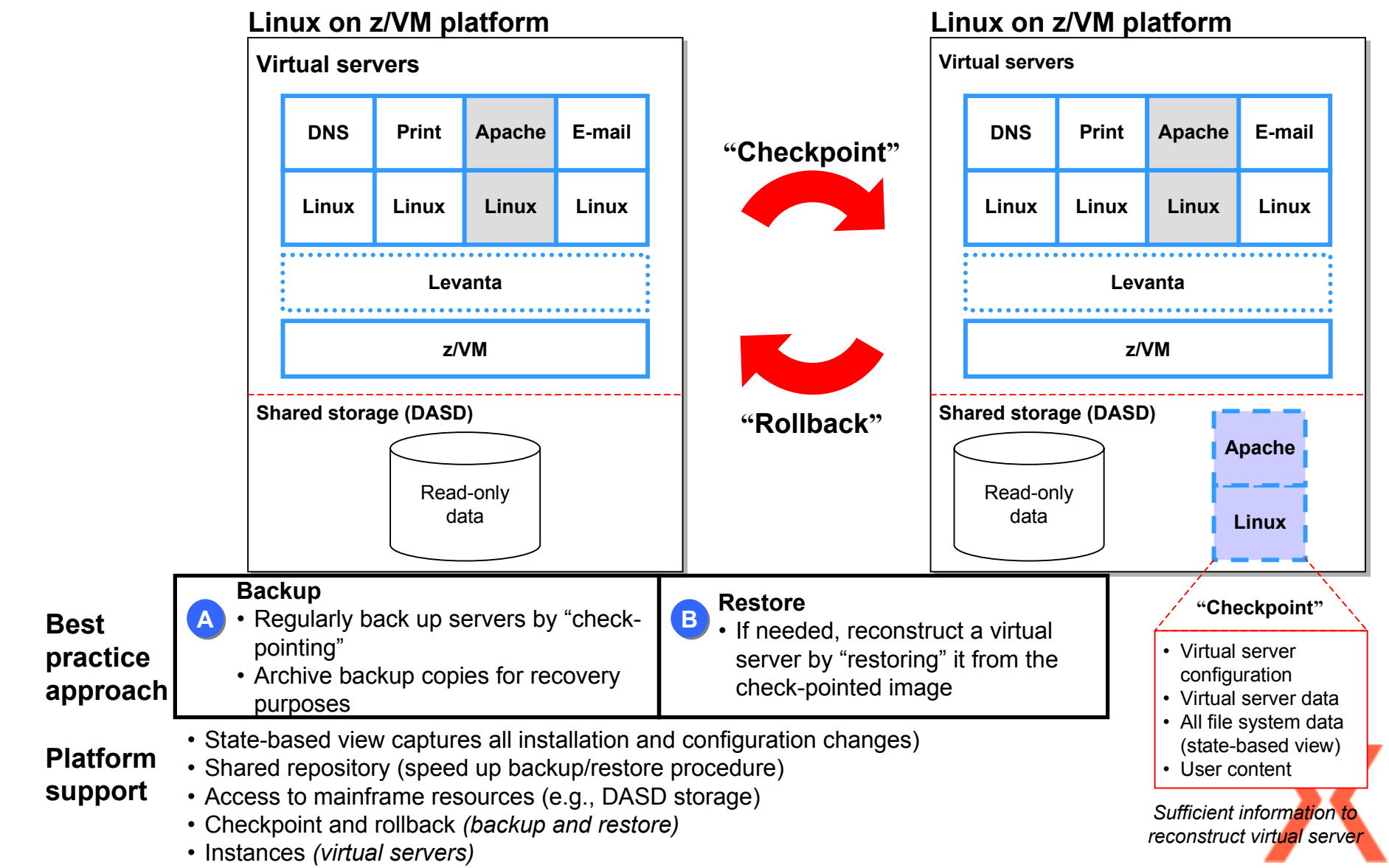
- New software version (e.g., patch, upgrade)
- Create new test server from existing server config
- Apply change/patch to that server only
- Test modified server, with standard test guidelines
- "Fine tune" the server
- Hand over tested tuned server to production
- Roll out (apply) a tested change to production servers
- Harvest / export a new server config from a changed server (for creation of future servers
- If needed, roll back a change

Platform support

- Templates *(server configs)*
- Instances *(virtual servers)*
- Change log
- Instances contain compatibility / dependency information
- Instance groups & harvesting
- Templates
- Repository *(data storage)*

VIRTUAL SERVER BACKUP AND RESTORE

Scenario: Weekly backup of all virtual production web servers



Best practice approach

A Backup
- Regularly back up servers by "check-pointing"
- Archive backup copies for recovery purposes

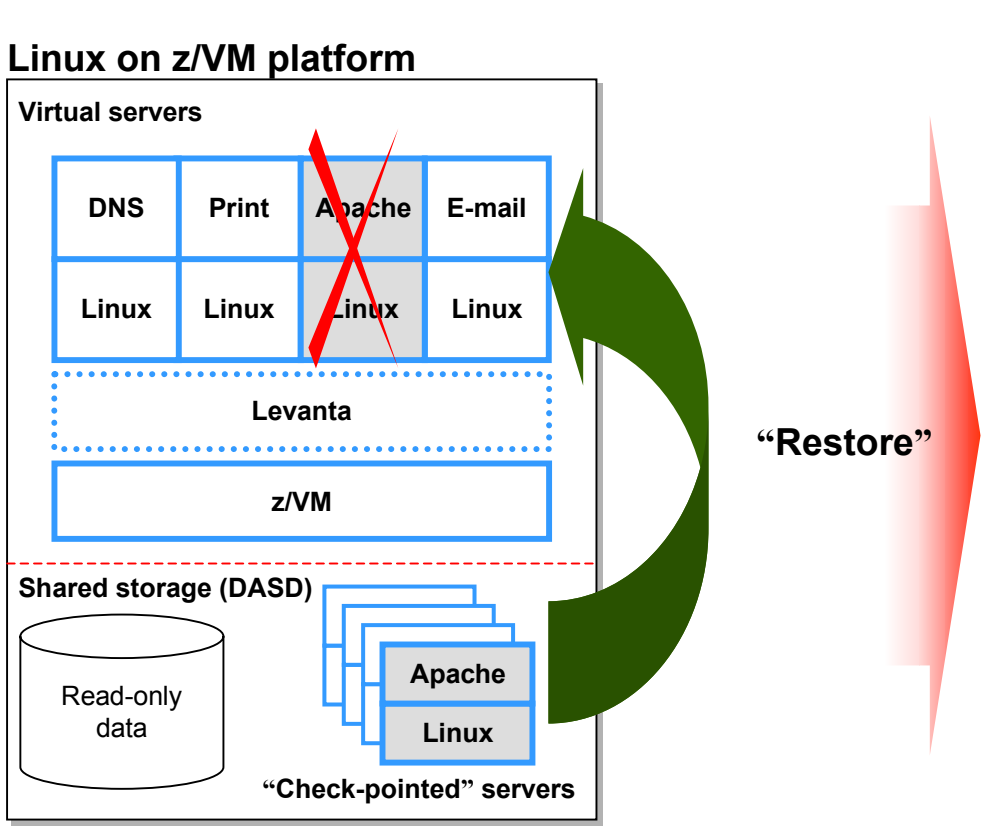
B Restore
- If needed, reconstruct a virtual server by "restoring" it from the check-pointed image

Platform support

- State-based view captures all installation and configuration changes)
- Shared repository (speed up backup/restore procedure)
- Access to mainframe resources (e.g., DASD storage)
- Checkpoint and rollback *(backup and restore)*
- Instances *(virtual servers)*
- Templates *(server configurations)*

VIRTUAL SERVER DISASTER RECOVERY

Scenario: Restore production web servers to a well-defined state after loss



- **"Free"** disaster recovery from:
  - State-based view – backup copy contains full state of a server, including all changes made, and
  - Rollback – possibie return a server to any prior check-pointed state
- **Fast** execution of DR enabled by
  - One centralized data storage (repository) that accessed by all servers
  - Optimal use of mainframe resources (e.g., DASD storage)
  - Predictable, reliable, and space-efficient rollback operation

Best practice approach

- Perform regular backups of running virtual servers (e.g., daily/weekly)
- Back up most critical virtual servers more frequently (e.g., daily/hourly)
- In case of a lost virtual server, restore its state from previously generated backup copy

Platform support

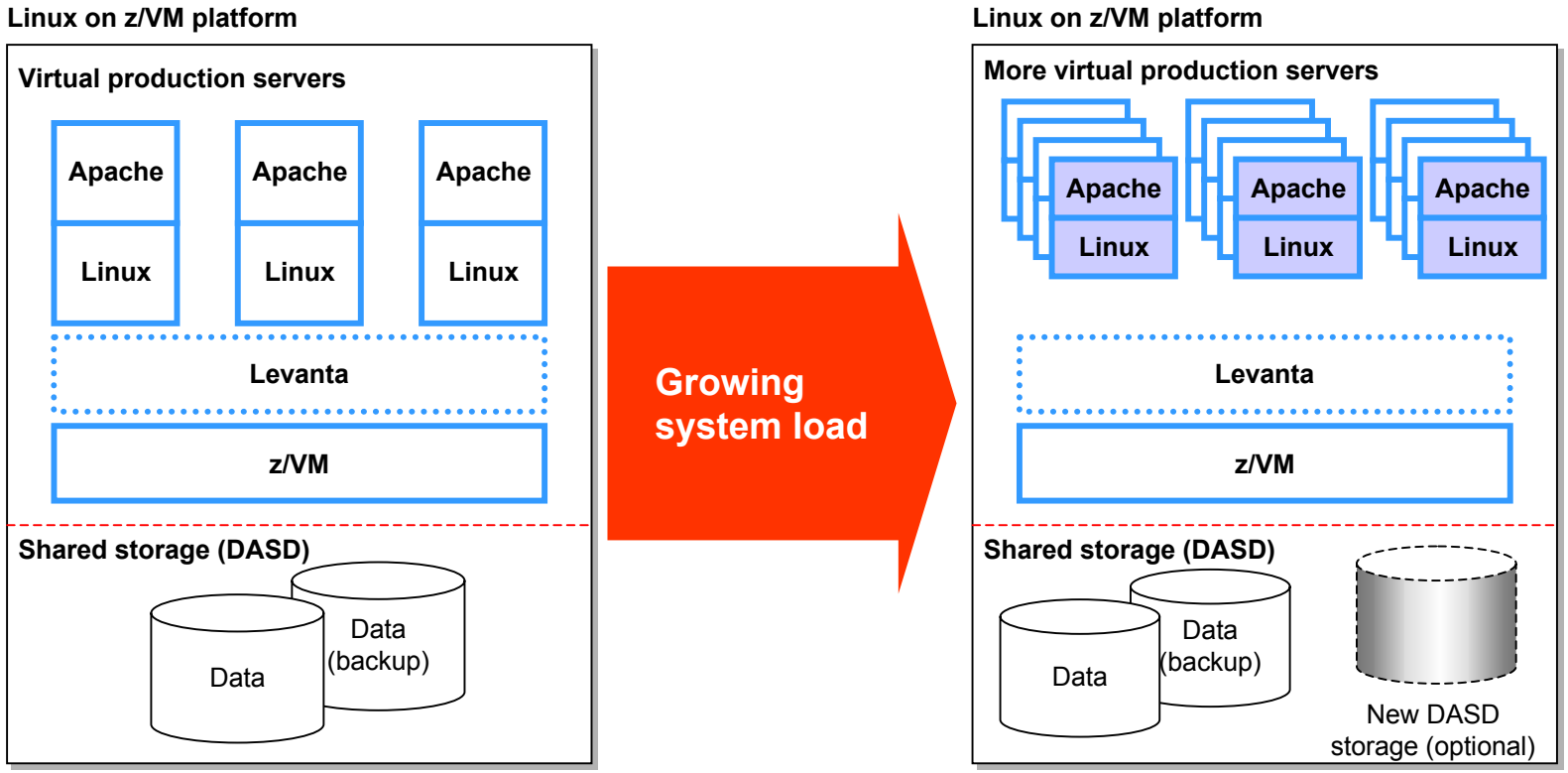
- Instances *(virtual servers)*
- Checkpoint and rollback *(backup and restore)*
- Shared repository *(shared storage)*

DISASTER RECOVERY ACTIVITIES BY ROLE

Platform role	Played by Whom	Activities
• Server User	• Linux systems administrator (day-to-day operations) • Database administrator	• Perform routine backups of individual virtual servers (e.g., daily checkpoint)
• Server Manager	• Linux systems manager (with advanced privileges) • Database manager	• Perform backup of the entire Linux on z/VM platform • Restore some or all virtual servers to a previously saved state (rollback)
• System Manager	• Mainframe system administrator	• Escalate to management in case of major problem (based on defined disaster scenarios)

RESOURCE AND CAPACITY MANAGEMENT

Scenario: Dynamically create new virtual production web servers, in response to growing system load



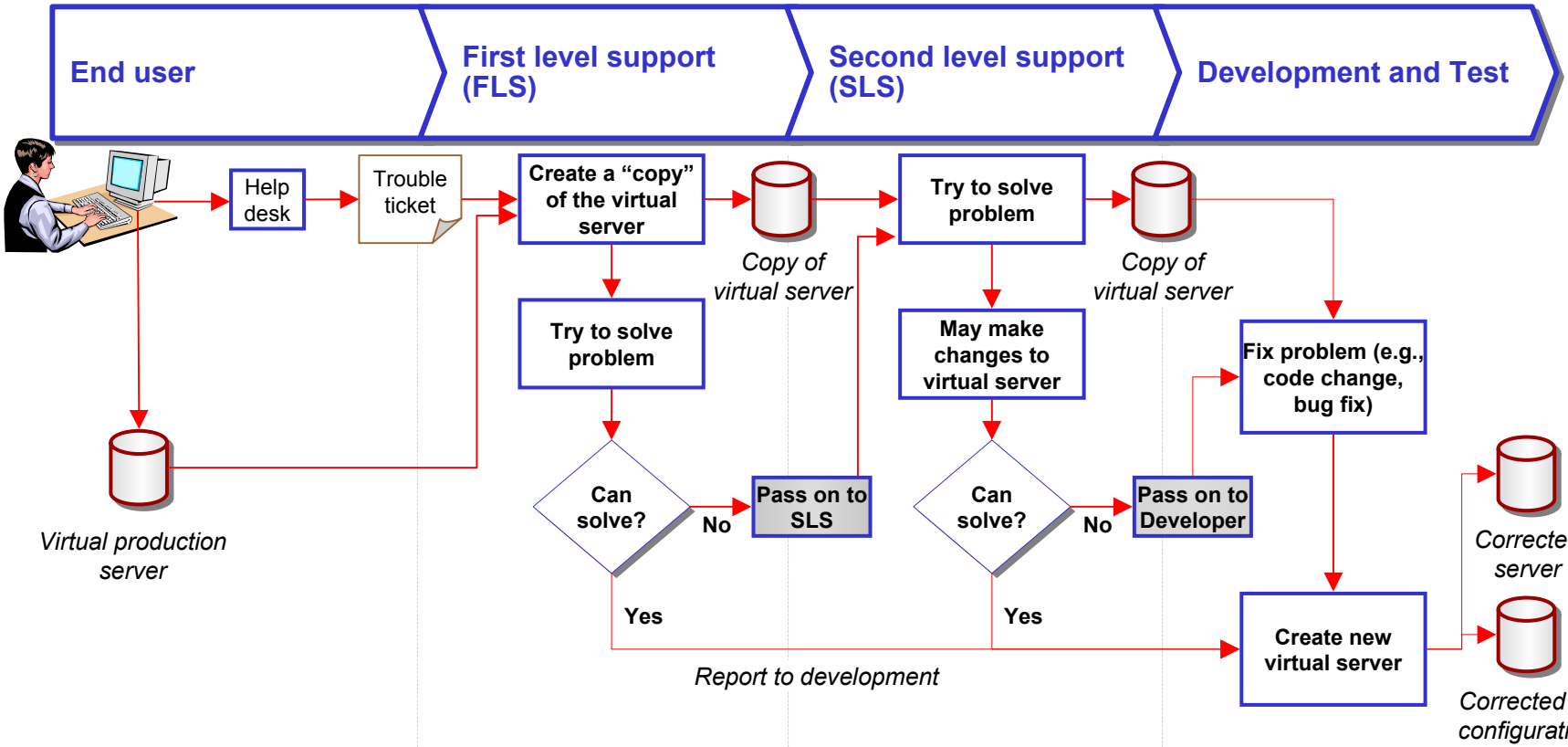
Best practice approach	• Provision enough virtual servers to meet estimated initial system load requirements • Monitor system load on an ongoing basis	• Ask for more virtual servers, in response to growing system load • Redistribute total system load to newly procured virtual servers	• Provision mainframe resources dynamically, e.g., DASD packs (do not need to shut down the platform when adding new resources), processing capabilities
Platform support	• Instances *(virtual servers)*	• Instance (creation) • Instance groups	• Templates *(server configurations)*, used to abstract various hardware components (through “template levels”)

CAPACITY MANAGEMENT ACTIVITIES BY ROLE

Platform role	Played by Whom	Activities
• System Manager	• Mainframe system administrator	• Maintain and configure mainframe/VM resources (e.g., add DASD storage)
• Resource Manager	• Mainframe resource manager	• Configure mainframe/VM resources for subsequent allocation/delegation • Provision and allocate mainframe resources for later self-service by Server Managers, based on their request (e.g., provide additional 10GB of disk space) • If all mainframe/VM resources are allocated, procure new mainframe/VM resources (subject to budget approval) • Define policies and guidelines for mainframe/VM resource usage
• Server Manager	• Linux system manager (with advanced privileges) • Database manager (with advanced privileges)	• Procure, create and administer virtual servers, using the mainframe resource pool previously allocated by the Policy Manager (i.e. through self-service) • If allocated resource pool is used up, request additional virtual servers/ resources from Resource Manager

PROBLEM MANAGEMENT

Scenario: User reports a problem with a virtual production server

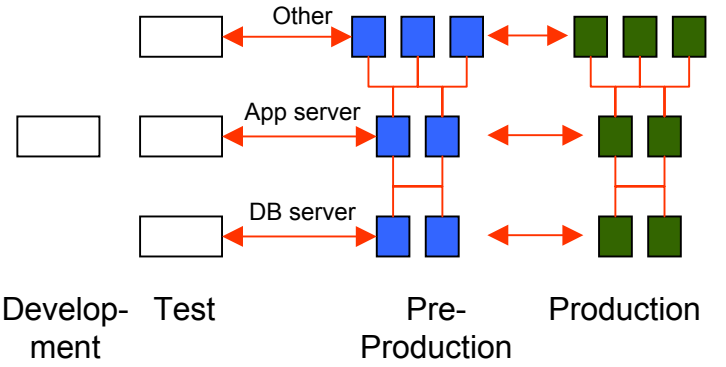


Best practice approach	• End user reports a problem with virtual production server (via helpdesk) to FLS (e.g., system administrator) • User describes the problem, specifies and "passes on" virtual server to FLS • Trouble ticket is generated	• Isolated copy of erroneous virtual server is generated (from running virtual server) • FLS (e.g., system admin) tries to solve the problem • FLS escalates the problem to SLS if necessary	• SLS (e.g., expert team) tries to solve the problem • SLS may have authority to make changes to the virtual server configuration • If needed, SLS "passes on" the problem (and the virtual server) to development	• Development team fixes the problem (e.g., code change) • The fix is captured by – creating a new version of the virtual server (e.g., a patch) – harvesting a new server configuration (for future creation of virtual servers)
Platform support	• Instance *(virtual server)*	• Harvesting of an instance (e.g., capture state of the virtual server just before and after the failure)	• Instance (being "passed on")	• Instance creation/update • Templates *(server configurations)*

COMBINATION OF DISTRIBUTED SERVERS

Scenario: Combine physical web servers into virtual servers on the z/VM platform

"Before" – Distributed systems

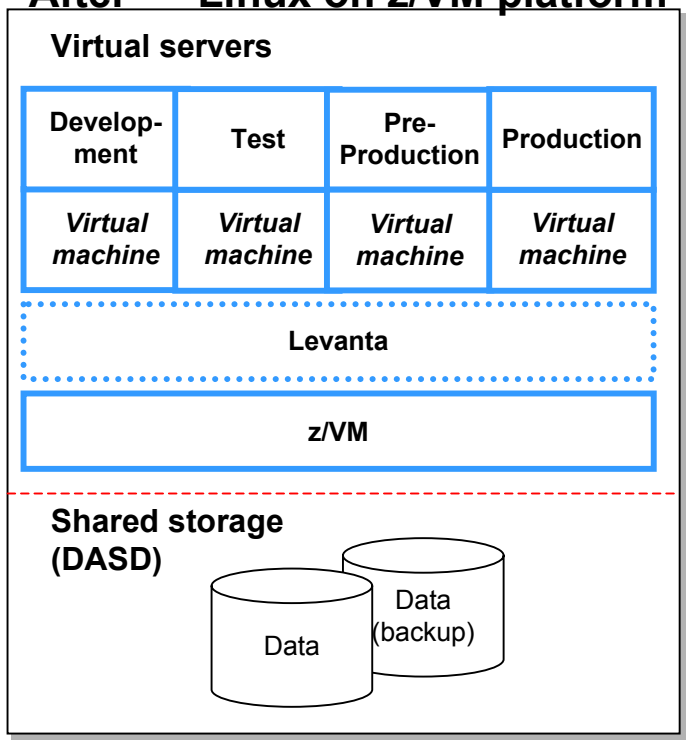


- Distributed servers (18 physical machines)
- Complex provisioning, staging, deployment, backup, and disaster recovery (DR)
- Expensive to maintain and operate

Server integration

- Isolation
- Encapsulation

"After" – Linux on z/VM platform



- Simplified administration
- More robust backup and DR
- Cost-effective operations

Best practice approach	• Ensure there mainframe resources available (to combine servers) • Capture config of the real/distributed servers by creating server configs • Create virtual servers on z/VM platform, using the captured server configs
Platform support	• Templates *(server configurations)* • Instances *(virtual servers)*

SERVER COMBINATION ACTIVITIES BY ROLE

Platform role	Played by Whom	Activities
• Supervisor	• Manager of data center operations	• Approve and oversee server combined project
• System Manager	• Mainframe system administrator	• Configure mainframe/VM resources, to meet expected additional work load requirements
• Resource Manager	• Mainframe resource manager	• Allocate mainframe/VM resources, to hold servers to be consolidated
• Server Manager	• Linux system manager • (with advanced privileges) • Database manager	• Create new virtual server configurations that match the previously distributed servers • Create virtual servers to replace the distributed servers (consolidation)

Banking Case Study

"Streamlining maintenance and administration of a growing Linux environment requires a software-based solution. We were impressed with its ability to bridge the technology differences between the mainframe and distributed system support environment, allowing our traditional mainframe system programmers to now maintain and administer Linux. As Linux use proliferates in our organization, we'll be ready."

SITUATION	• International multibank HQ'd in the midwest ~$35 B assets • Accessed mainframe CICS / DB2 data via of gateway servers • Desired improvements – Reduce complexity and points of failure – Reduce the number of servers managed – Error resolution too complex, touched too many parties – Shorten time for disaster recovery – Increase system reliability and system supportability – Reduce time on emergencies; address new business
BUSINESS SOLUTION	• Moved to DB2 Connect servers to mainframe • Reduced disaster recovery time • Increase scalability • Reduce support overhead • Trouble ticket resolution simplified (Mtgs 3x / week to none) • Retired numerous servers • IT now focused on revenue generating endeavors
LEVANTA VALUE	• New virtual server instances; change mgmt of multiple instances; maintain patches and updates • Easier Linux release upgrades; rapid new Linux instance create • Access to features and function of z/VM without a steep learning curve - able to share DASD and allocate network IP addresses
IT SOLUTION	• Bank has successfully migrated all to DB2 Connect related on z900, using zLinux, Levanta, and SuSe

Platform Operations Concept for zSeries Linux with z/VM

Art Olbert
Linuxcare, Inc
AOlbert@linuxcare.com
415-354-4346