PowerVM : Latest announcements

IBM

Power Systems

Open Innovation to Put Data to Work

Jean-Manuel Lenez

PowerVM Editions are tailored to client needs

PowerVM Editions offer a unified virtualization solution for all Power workloads

- **PowerVM Standard Edition**
  - *Production deployments*
  - *Server consolidation*
- **PowerVM Enterprise Edition**
  - *Multi-server deployments*
  - *Cloud infrastructure*
- **PowerVM PowerLinux Edition**
  - *Same Function as PowerVM EE*
  - *Restricted to Linux VMs*

PowerVM Editions	**Standard**	**Enterprise**
Concurrent VMs	**20 per core** (up to 1000)**	**20 per core** (up to 1000)**
Virtual I/O Server	**Multiple per server**	**Multiple per server**
NPIV	✓	✓
Linux Little Endian Distro Support✧	✓	✓
Shared Processor Pools	✓	✓
Shared Storage Pools	✓	✓
Thin Provisioning	✓	✓
Live Partition Mobility		✓
Active Memory Sharing		✓
PowerVP*		✓

AIX for Business Linux PowerVM

*** Requires eFW7.6 or higher*

** Requires eFW7.7 or higher*

✧ Requires wFW8.30 or higher

HMC news

New HMC Options

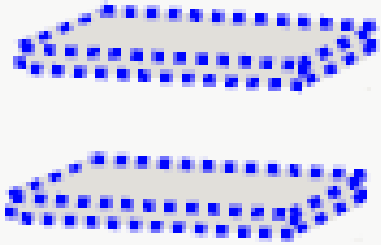
- Refreshed rack-mounted HMC
  - For POWER6 and later servers
  - 7042-CR9 — follow-on to CR8

- New HMC virtual appliance (vHMC)
  - For POWER6, POWER7, POWER8 servers
  - Same functionality as traditional HMC (version 8)
  - Runs as virtual machine on x86 server

Options for attachment to a server

One or two HMCs

One vHMC and one HMC



One or two vHMCs

HMC Virtual Appliance Details

- IBM Virtual HMC Complete Software offering
  - License plus software maintenance
  - PID – 5765-HMV
  - Approximately $3k price including 1 year SW maintenance
- Virtual HMC Runs
  - On x86 Hardware (provided by client)
  - Under KVM or VMware virtualization (provided by client)
- New vHMC Complete Software offering
  - Activation engine – provides configuration on first boot
  - Accept License, locale, network, SSH, NTP
- Manages any POWER6 or later Power servers
- Version 8 HMC firmware
- Can be used with or without hardware HMCs

LPM improvement

LPM – Even Better

Better Validation and Checking Prior to LPM Operation

NPIV Specific Improvements

Better use of Etherchannel or Link Aggregation

Even better spreading of I/O across multiple interfaces

Overall speed Improvements

LPM functionality when a VIO server in a pair becomes unavailable.

The ability to move without having to manually remove everything pointing to the VIO server that is down

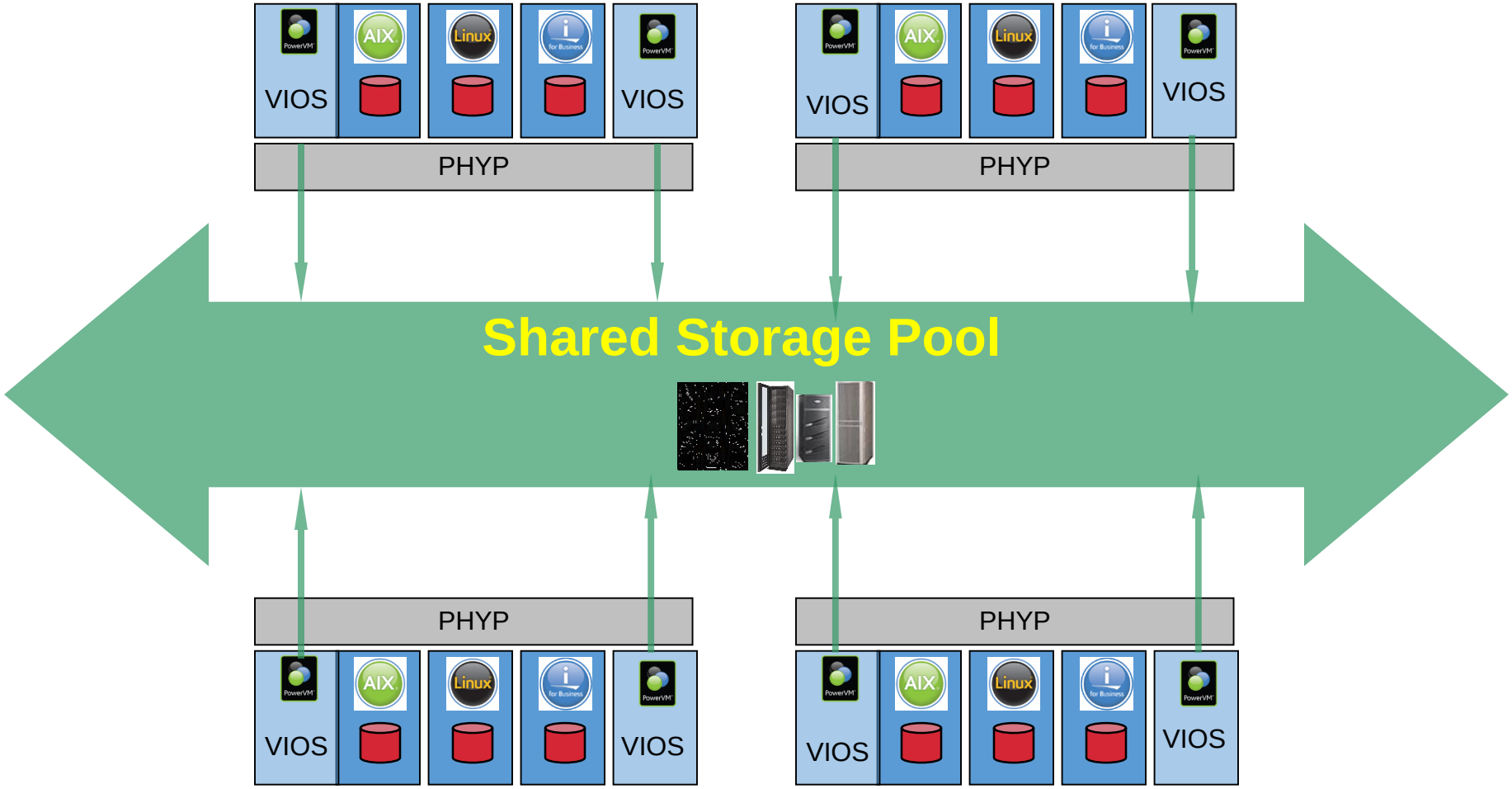
Target vSwitch can now be selected

Even better in the case where different systems have unique vSwitches

Shared storage pool

Shared Storage Pools Simplify Virtualization

PowerVM



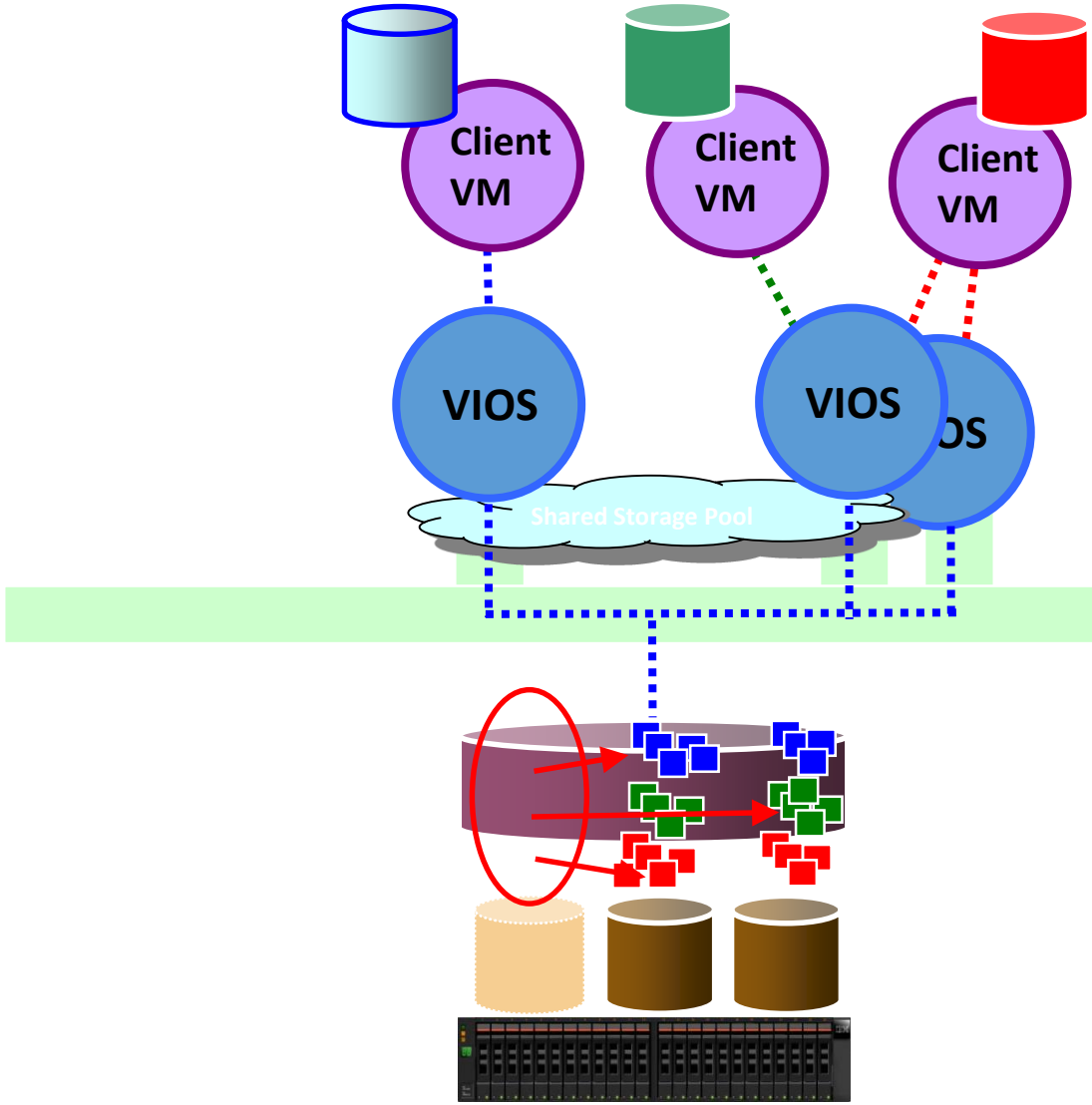
SSP4 – new feature in brief

1. Remove LUN from Shared Storage Pool

2. Pool Mirroring Option for Resilience
   - Disaster Recover option to handle
     - Disk-subsystem failure or LUN offline failures
   - Doubles the disk requirement – typical of DR
   - SSP4 manages the mirror I/O (not the client VMs OS)
   - SSP4 does the silver & recovery re-silvering is autonomic
   - Managed using the failgrp command

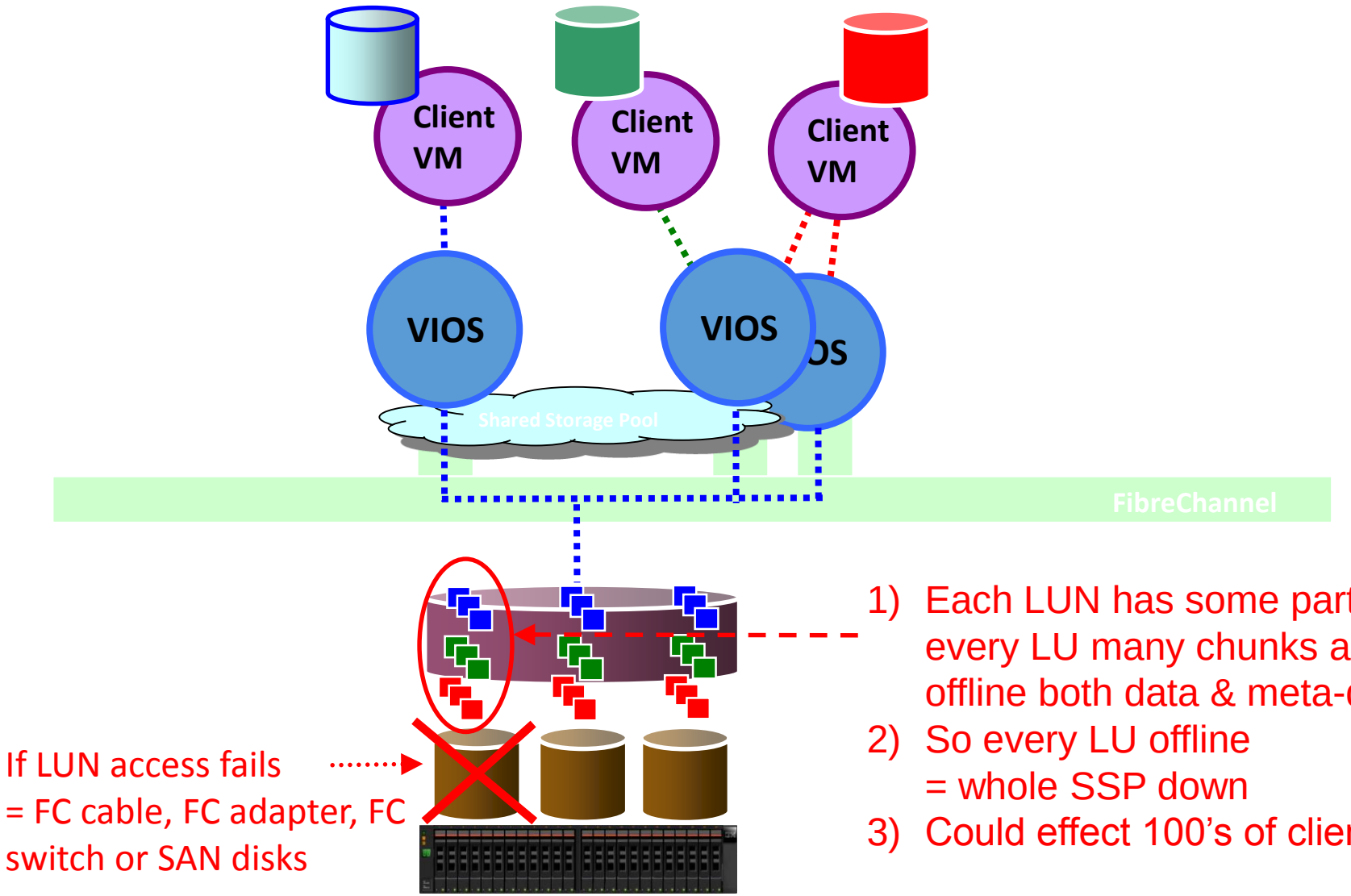
3. New simpler SSP commands
   - pv – Physical Volume control SSP LUNs in the pool
   - lu – Logical Unit control SSP virtual disk

SSP4 – New Remove LUN function can be tricky



1) Remove LUN: fault tolerant shrink & rebalance operation
2) Lots of I/O to redistribute chunks to remaining LUNs while still in use
3) “Fingers crossed” it will fit!

SSP3 – LUN access failure = Single Point of Failure



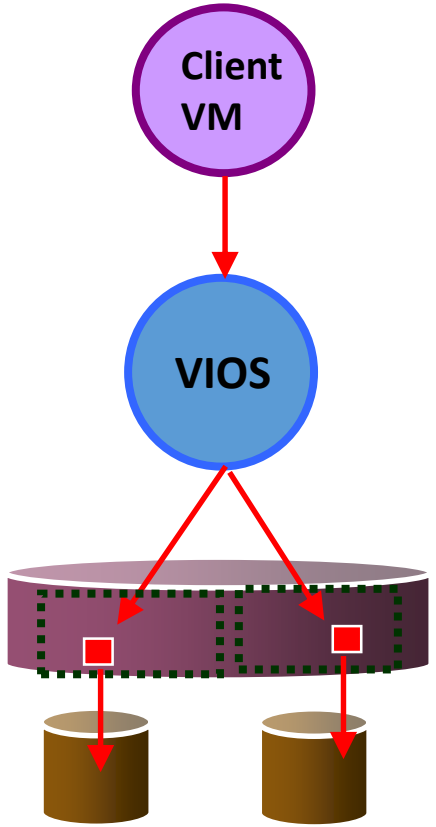
Protecting your SSP data from DISASTER

Client VM

- Only sees one copy & does one write
- Mirror is invisible → no admin work

VIOS

- Do the hard work on the VIOS
- VIOS duplicates the writes to mirrors
- It knows about LUN states & LU mapping
- Autonomic re-silver recovery



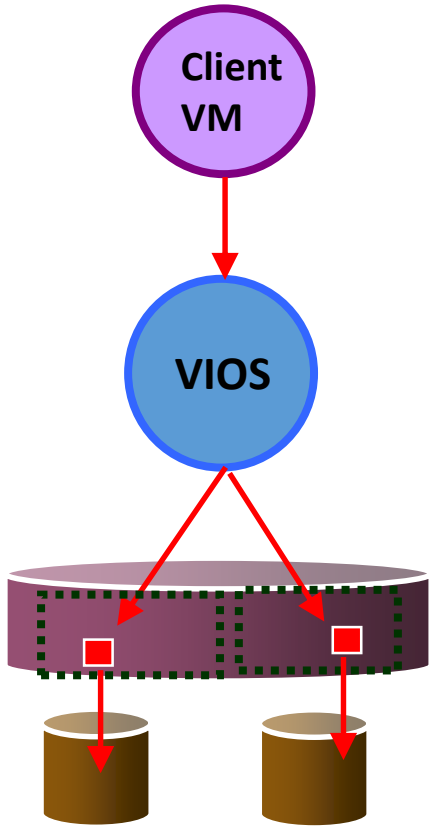
Protecting your SSP data from DISASTER

Client VM
- Only sees one copy & does one write
- Mirror is invisible → no admin work

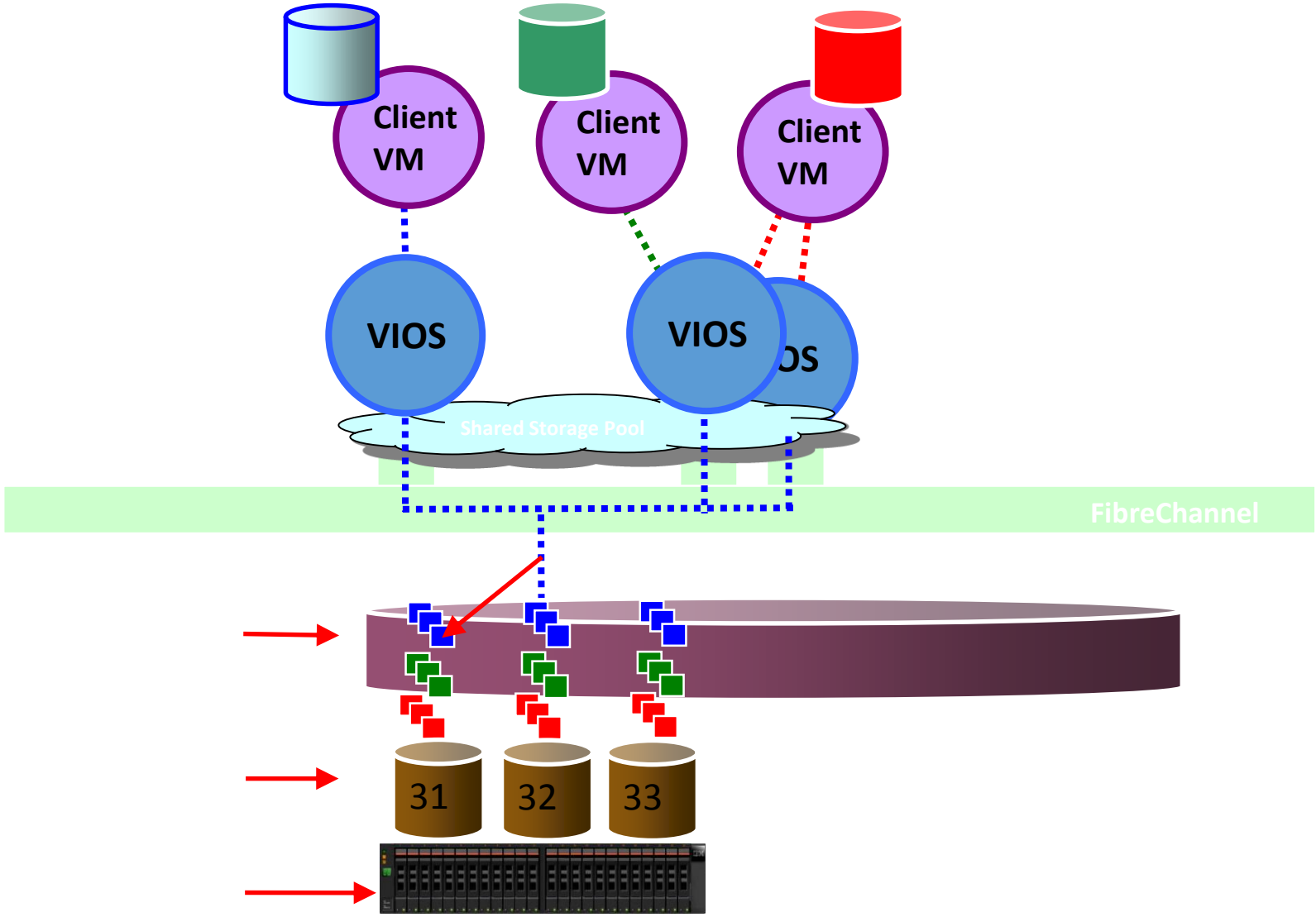
VIOS
- Do the hard work on the VIOS
- VIOS duplicates the writes to mirrors
- It knows about LUN states & LU mapping
- Autonomic re-silver recovery

New feature **failgrp** → "LUN failure groups"
- Two sets of LUNs on two different disk units*
- Whole pool is mirrored (not at LU level)

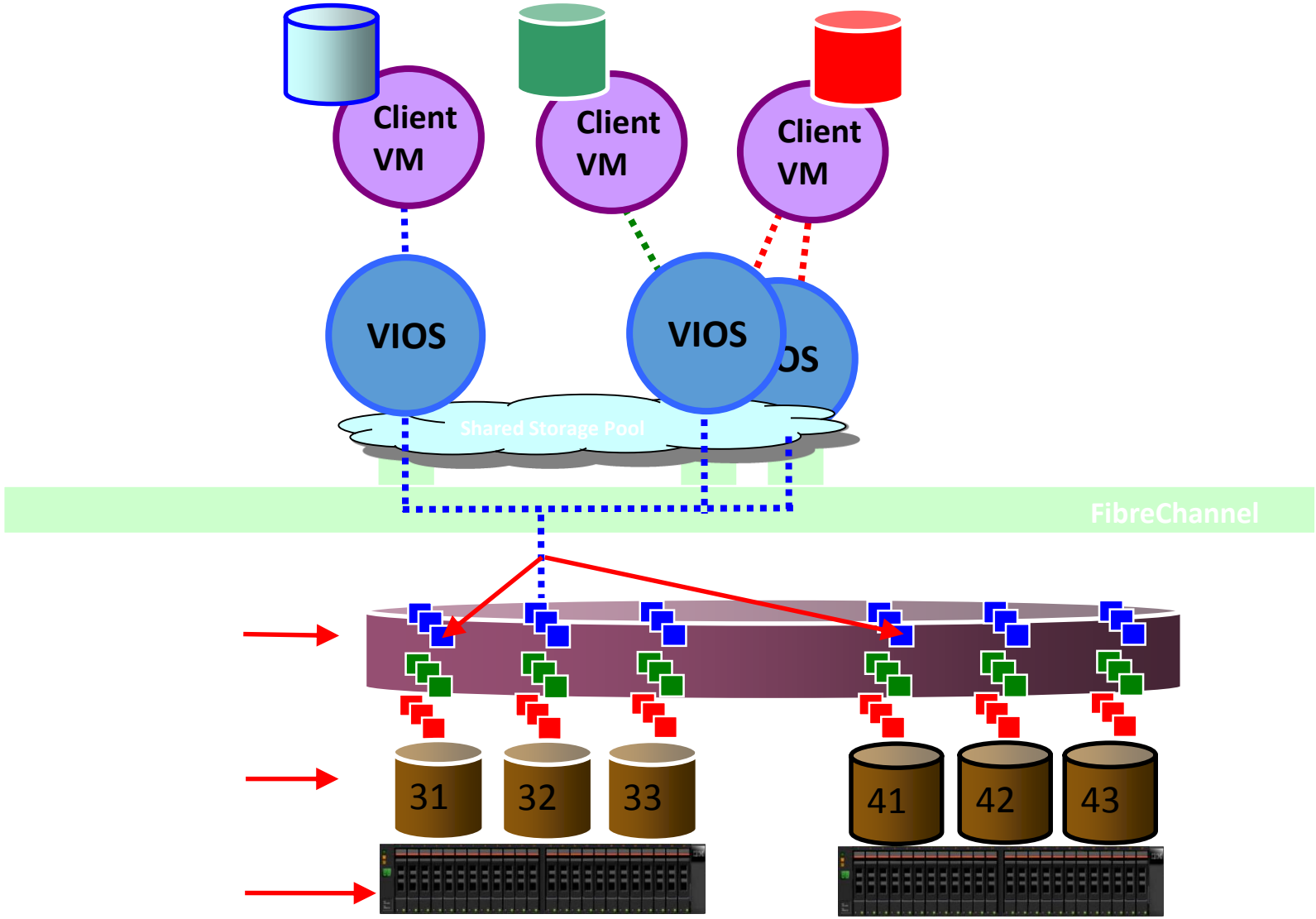
A. Default one failgrp called "Default" → no mirror
B. Optional: Adding a 2nd failgrp → adds the mirror



SSP4 – No extra failgrp = no Mirror



SSP4 – Add 2nd failgrp = add Mirror



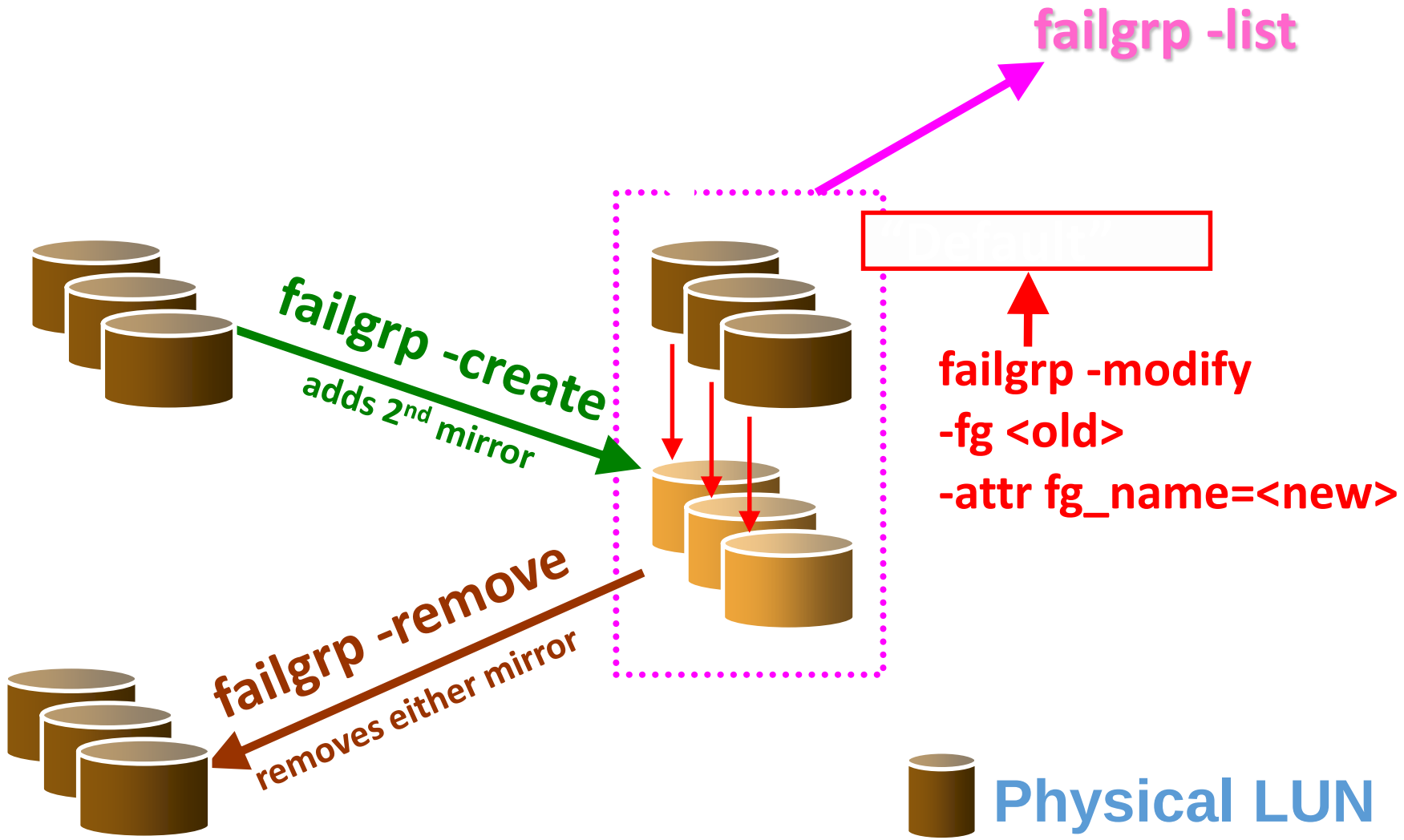
Three New SSP4 Command

New commands overview

- failgrp — Add a pool mirror copy command
  - create
  - remove
  - list
  - modify
- pv — SSP physical volume (LUN) command
  - add
  - remove
  - list
  - replace
- lu — SSP logical unit (virtual disk) command
  - create
  - remove
  - list
  - map
  - unmap

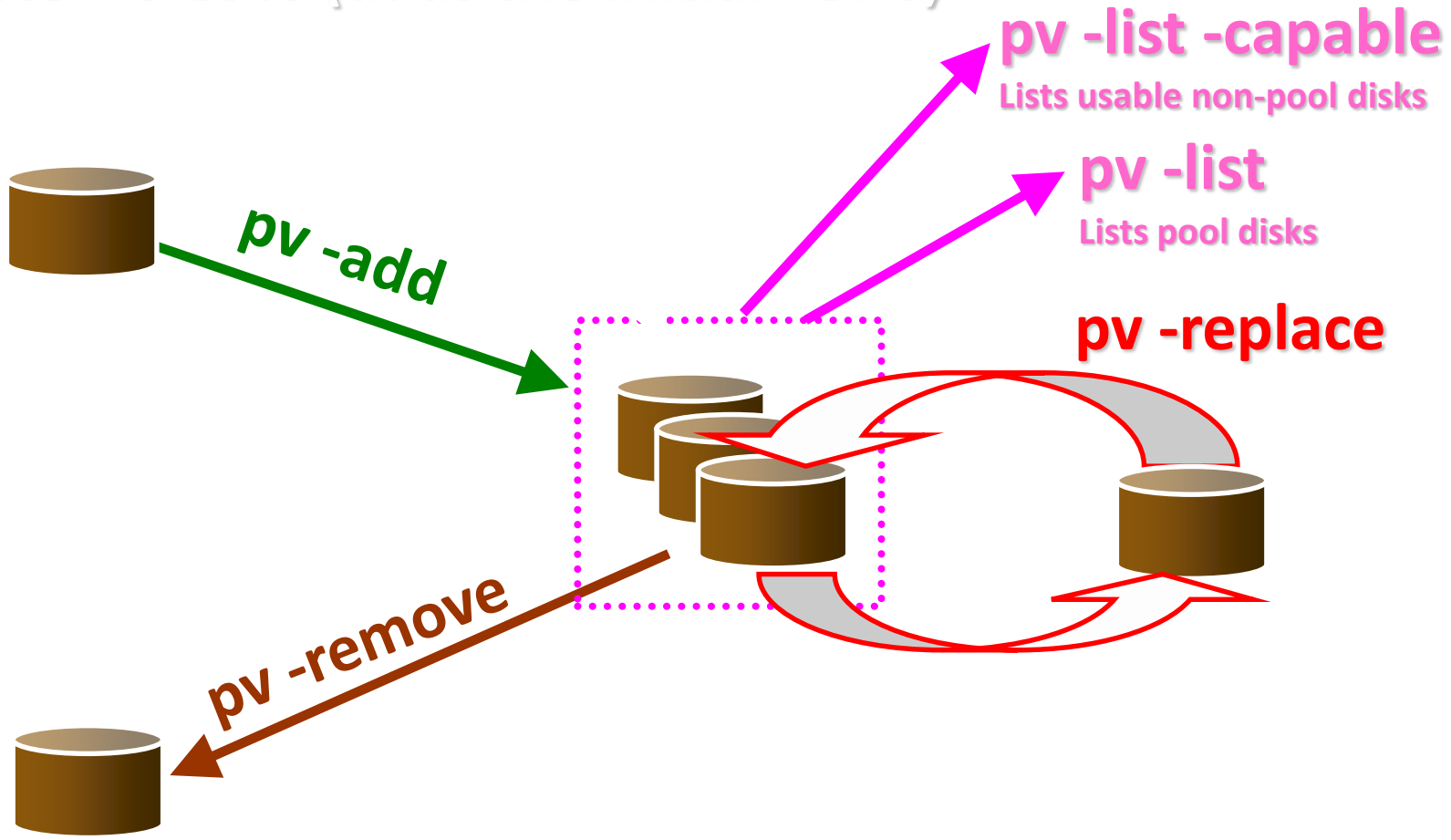
“failgrp” command / state map

cluster -create (adds the initial LUNs)



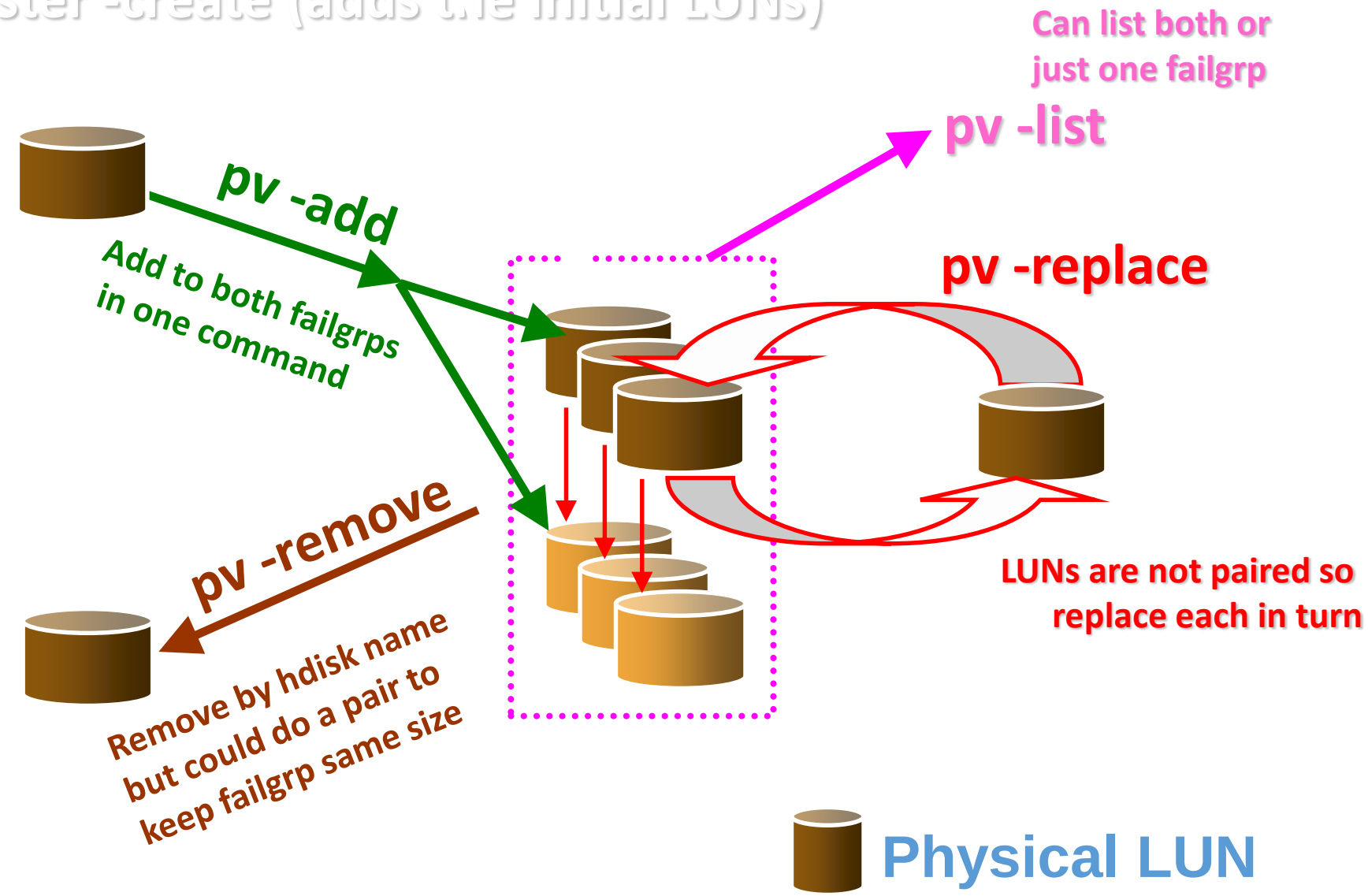
“pv” command / state map

cluster -create (adds the initial LUNs)

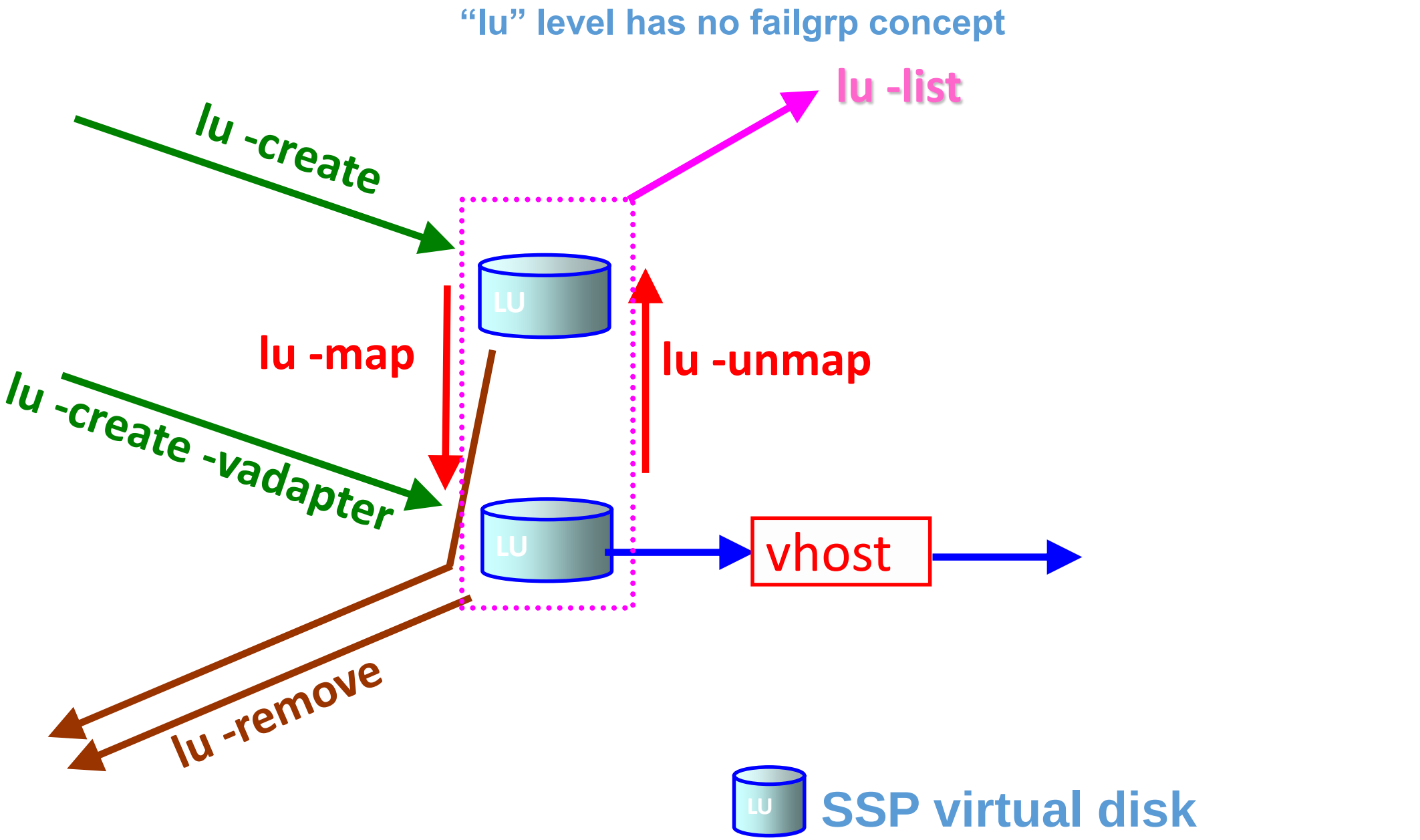


“pv” command / state map with mirror

cluster -create (adds the initial LUNs)

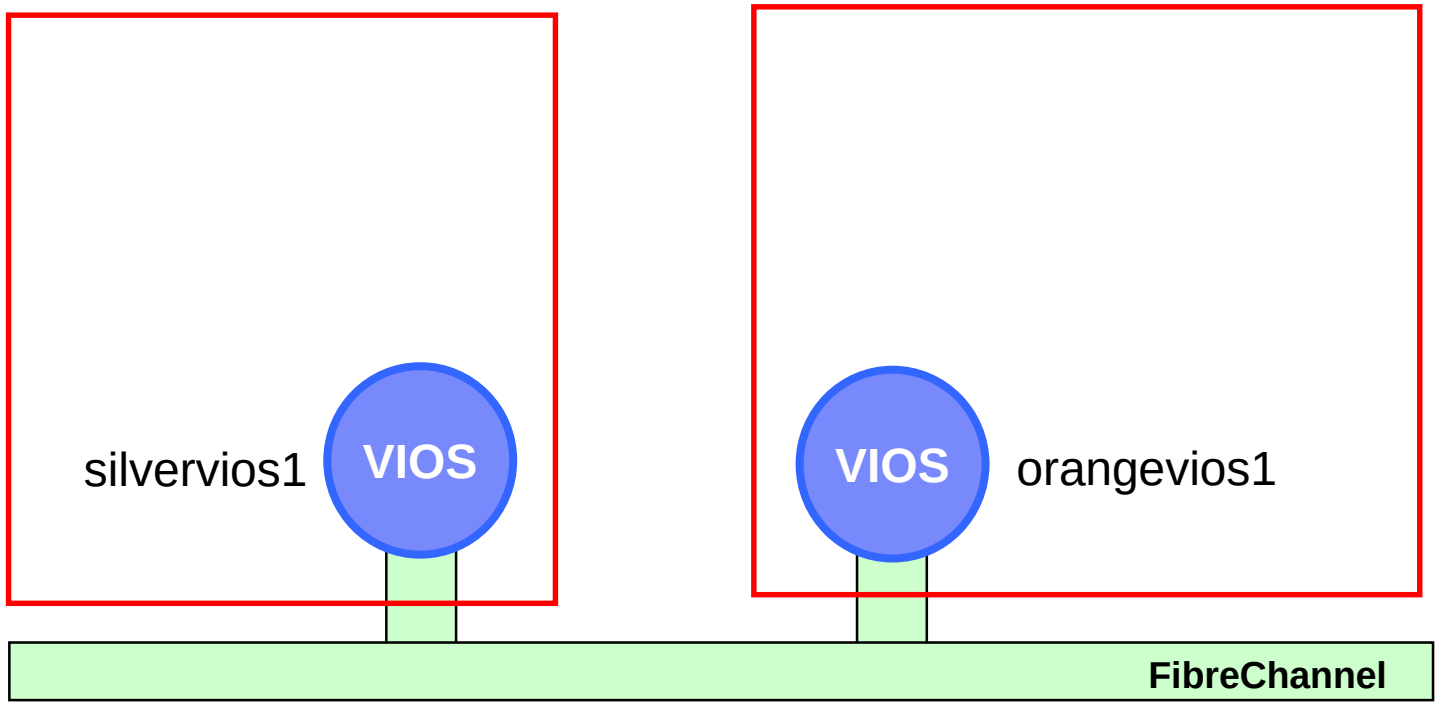


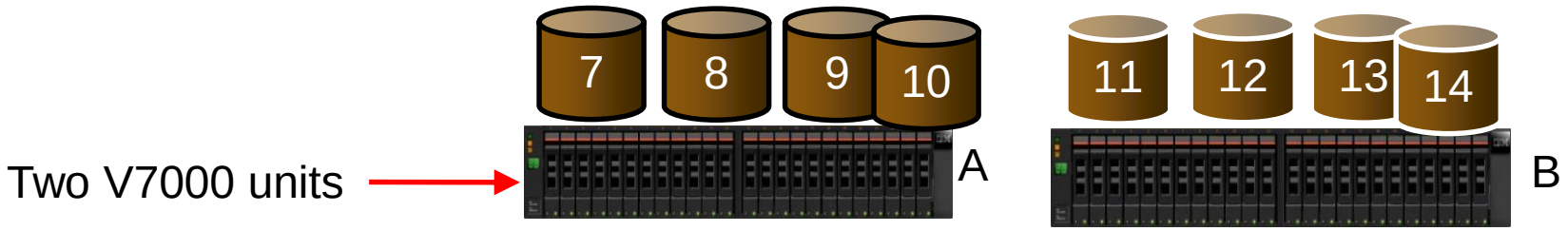
“lu” SSP virtual disk command / state map



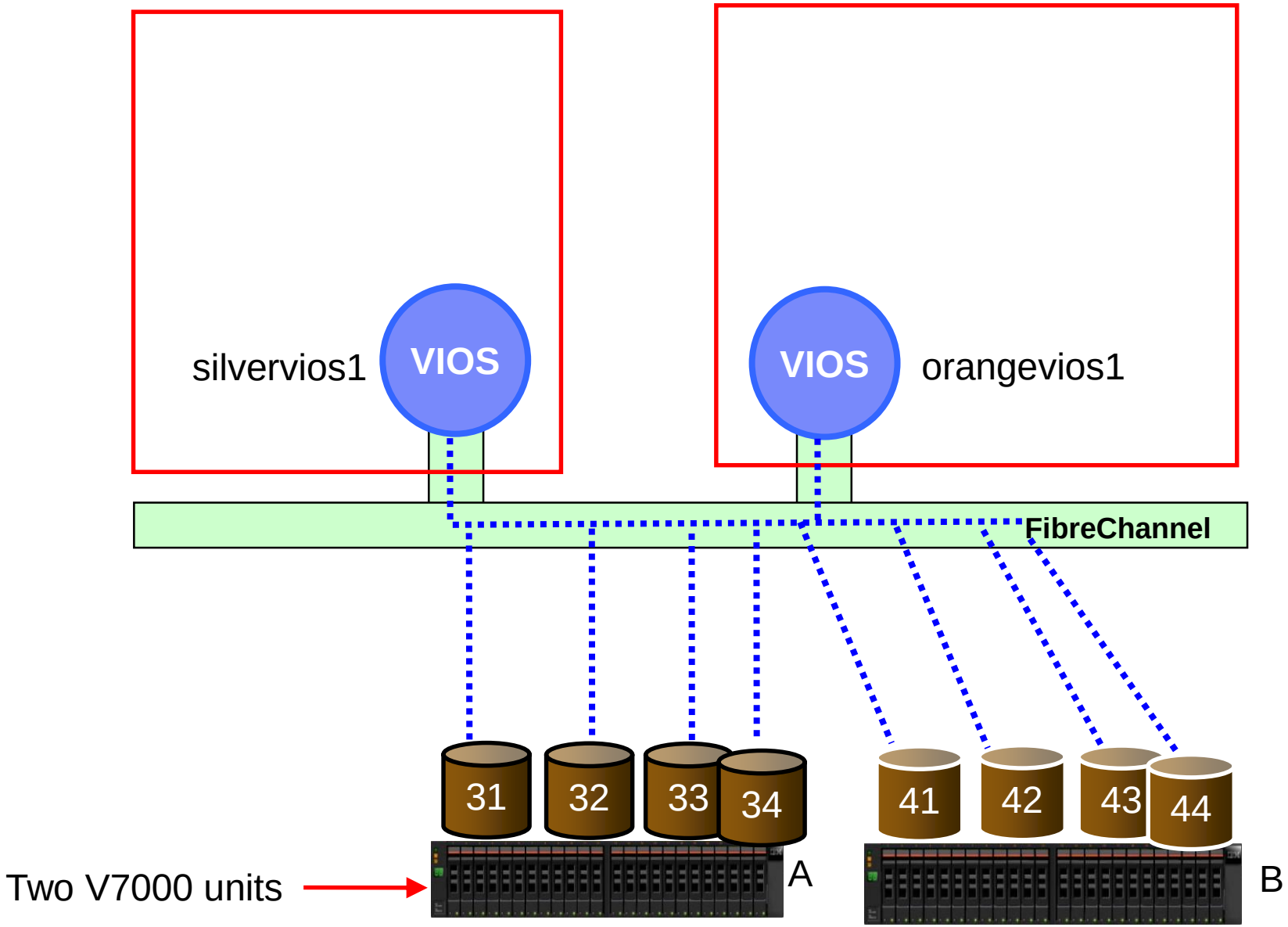
Demonstration

SSP4 – Demonstration Configuration

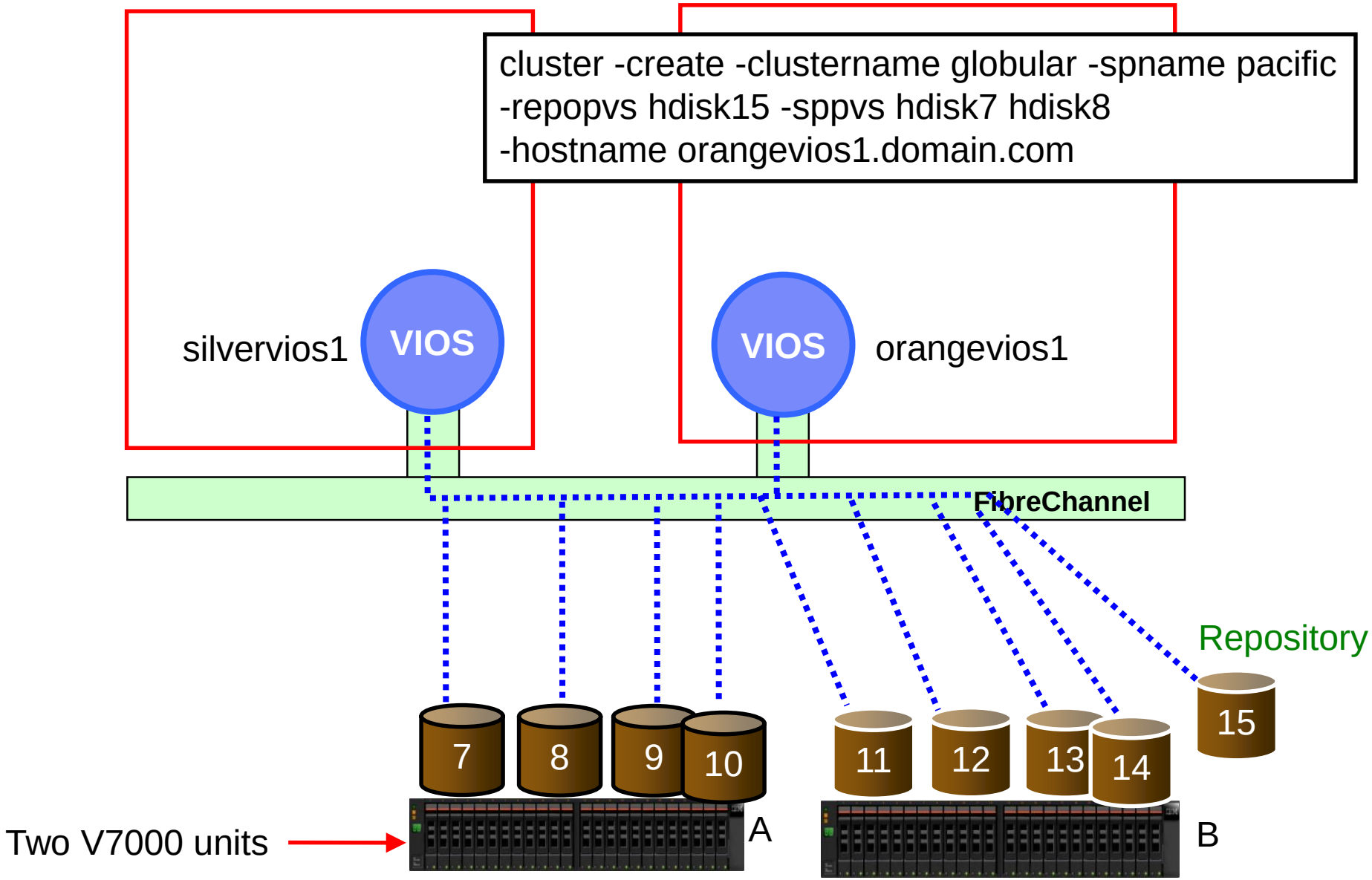




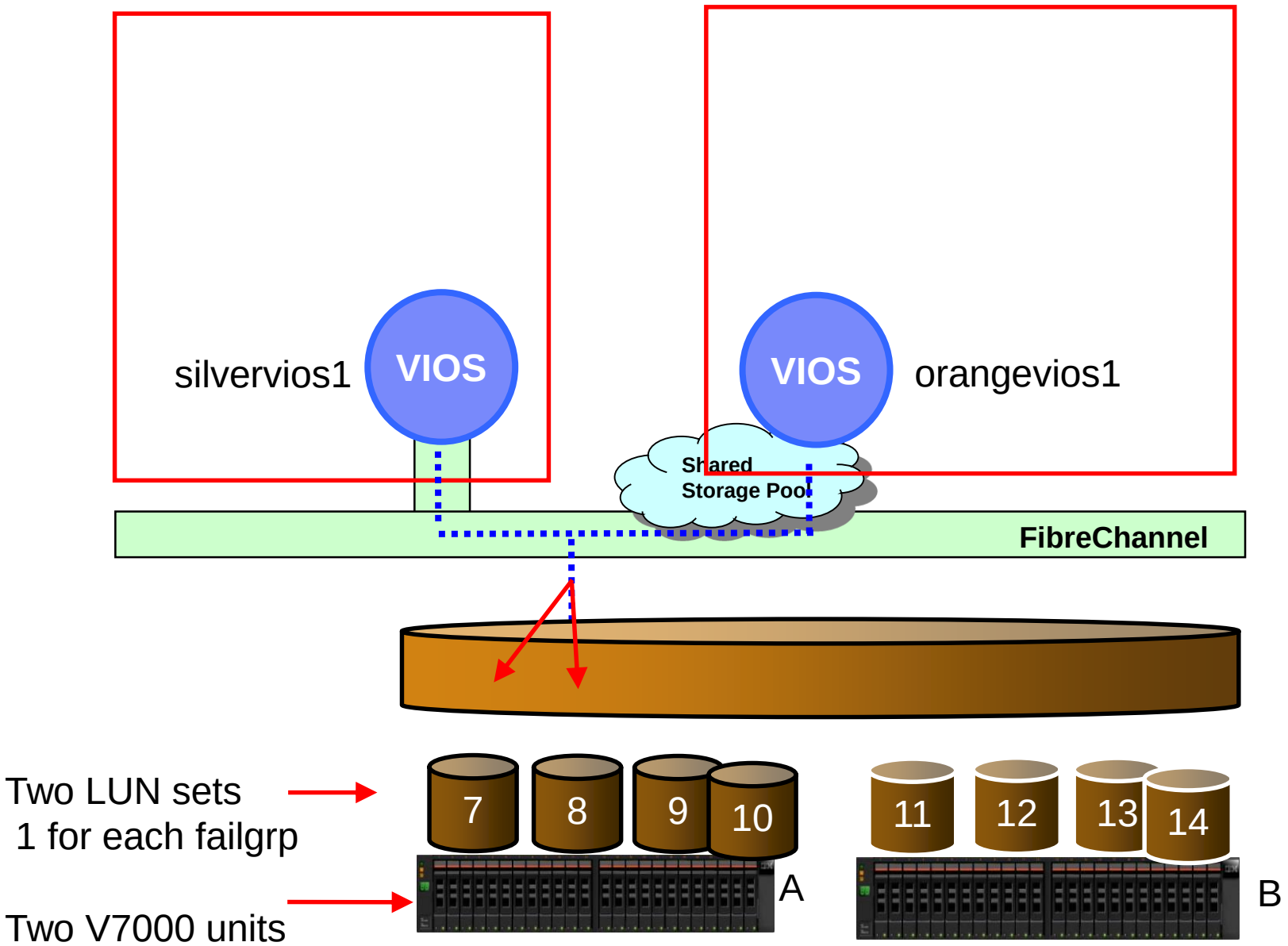
SSP4 – Demonstration Configuration



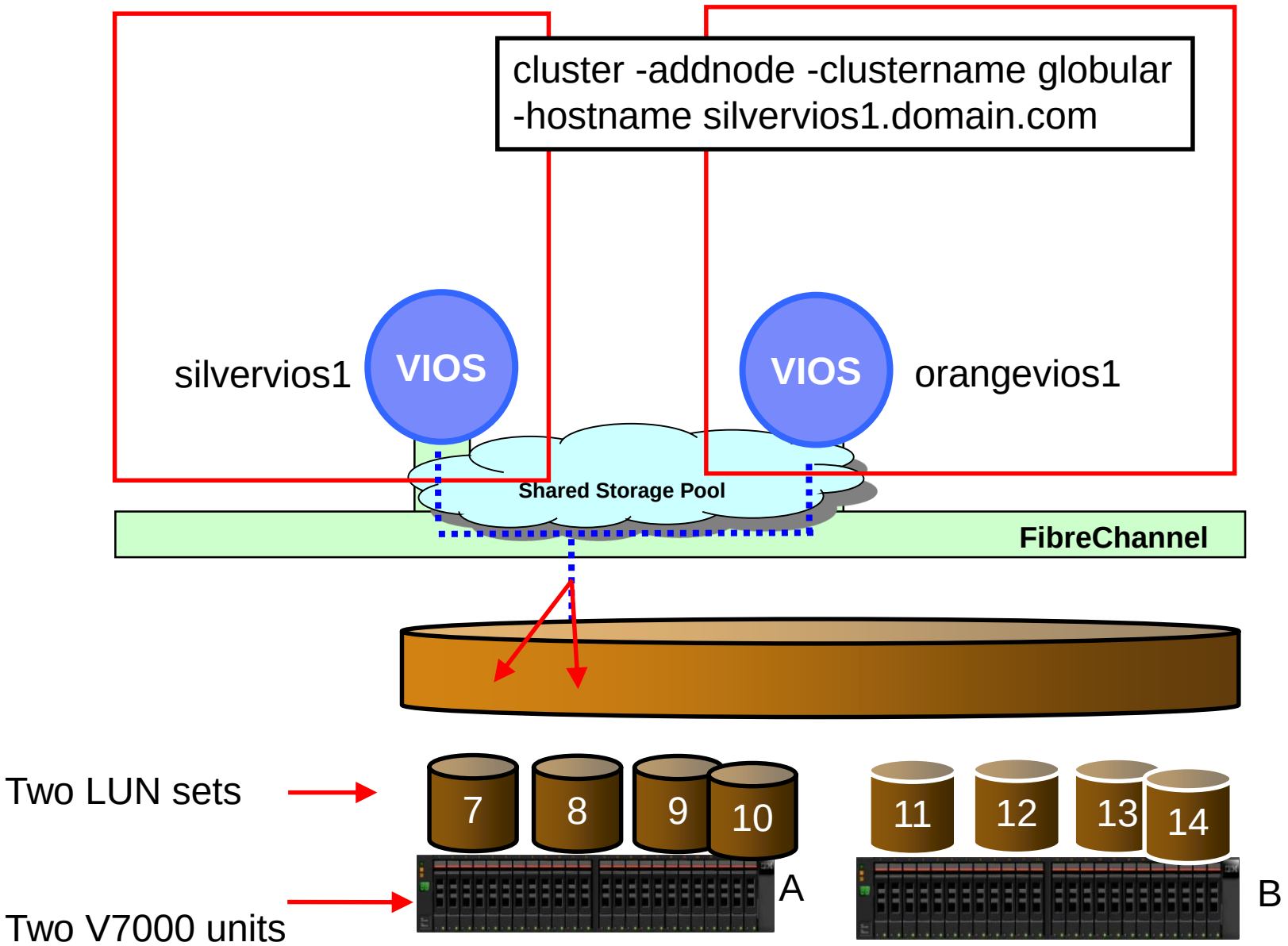
SSP4 – Demonstration Configuration



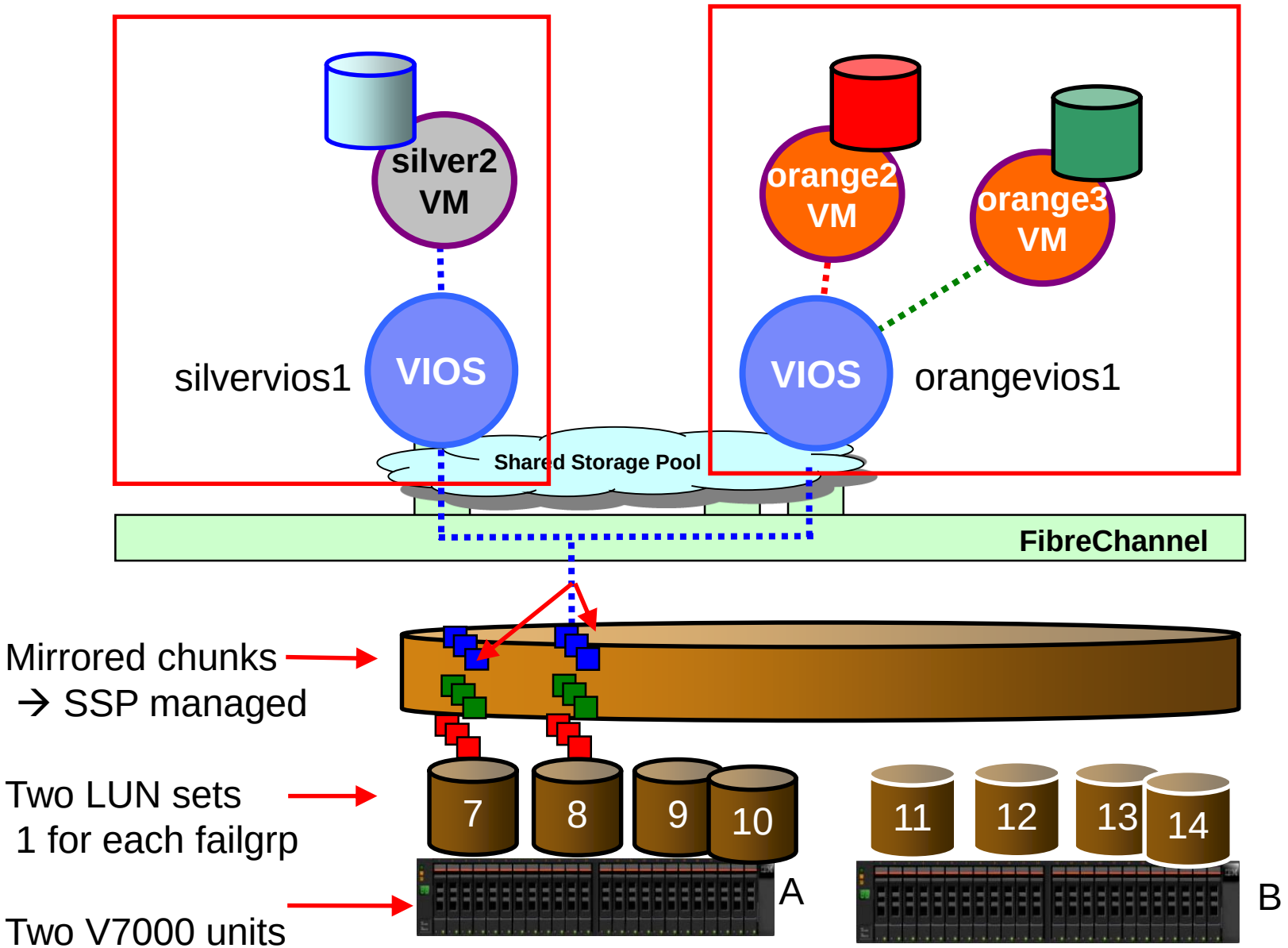
SSP4 – Demonstration Configuration



SSP4 – Demonstration Configuration

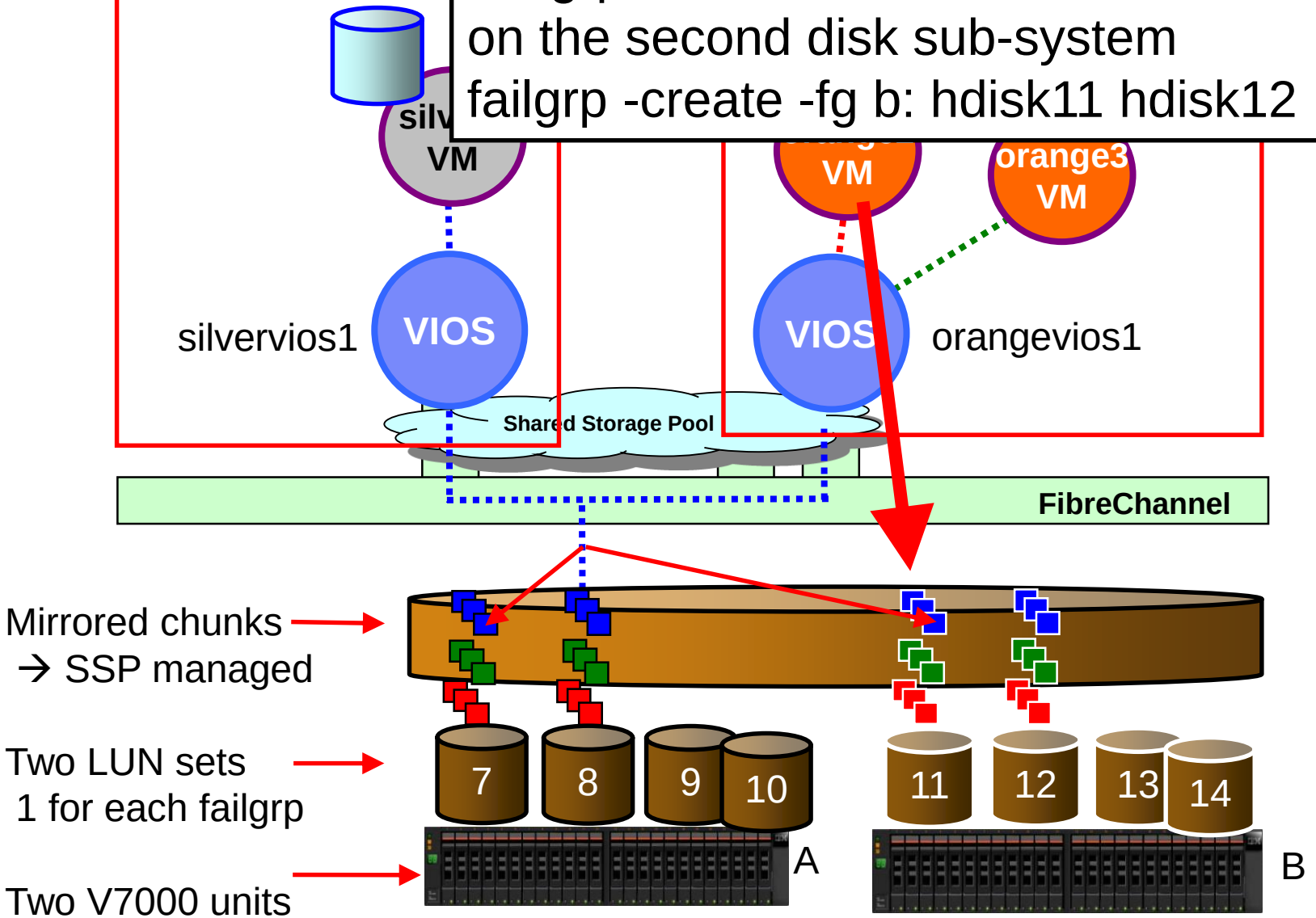


SSP4 – Demonstration Configuration

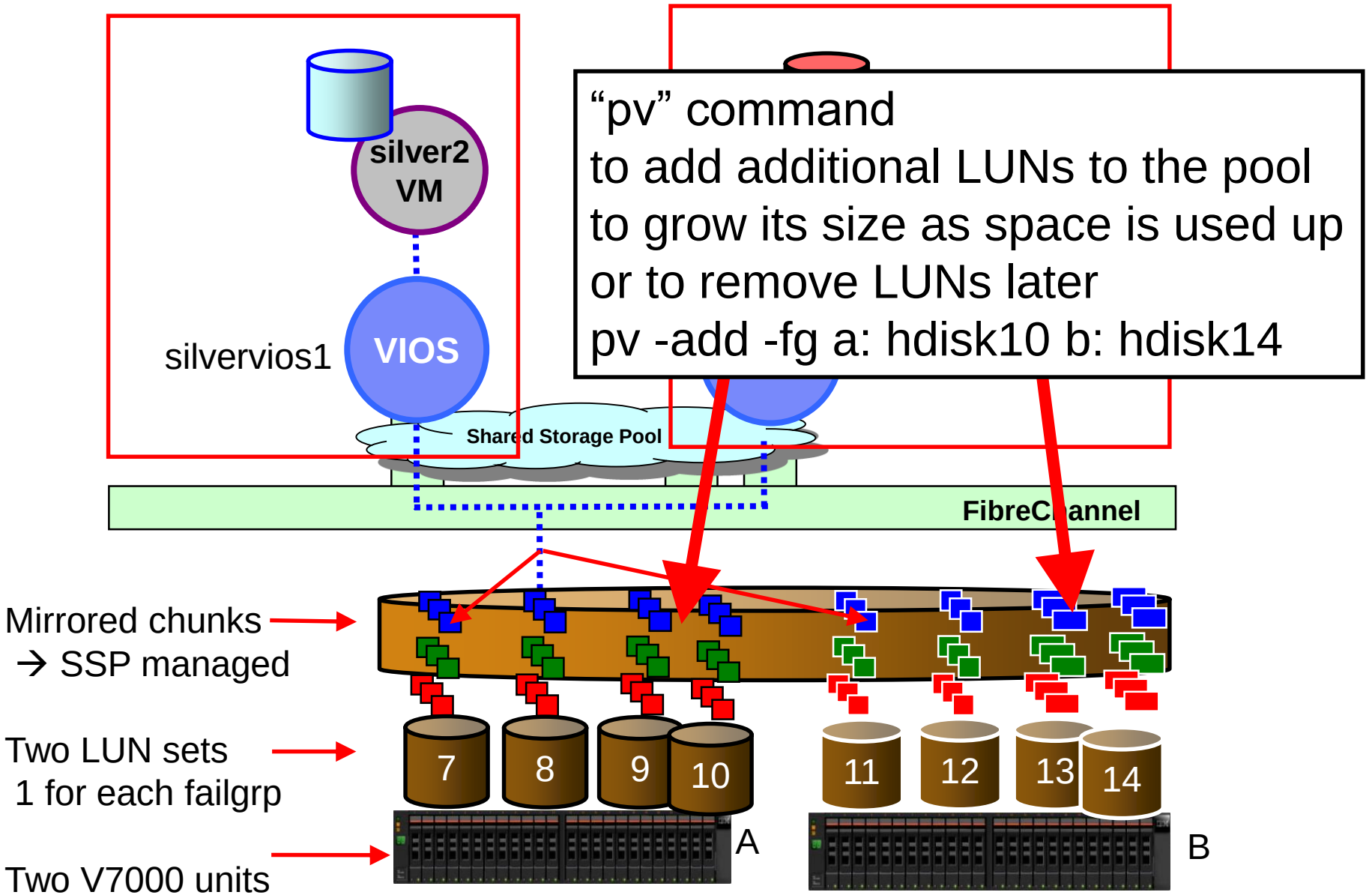


SSP4 – Demonstration Configuration

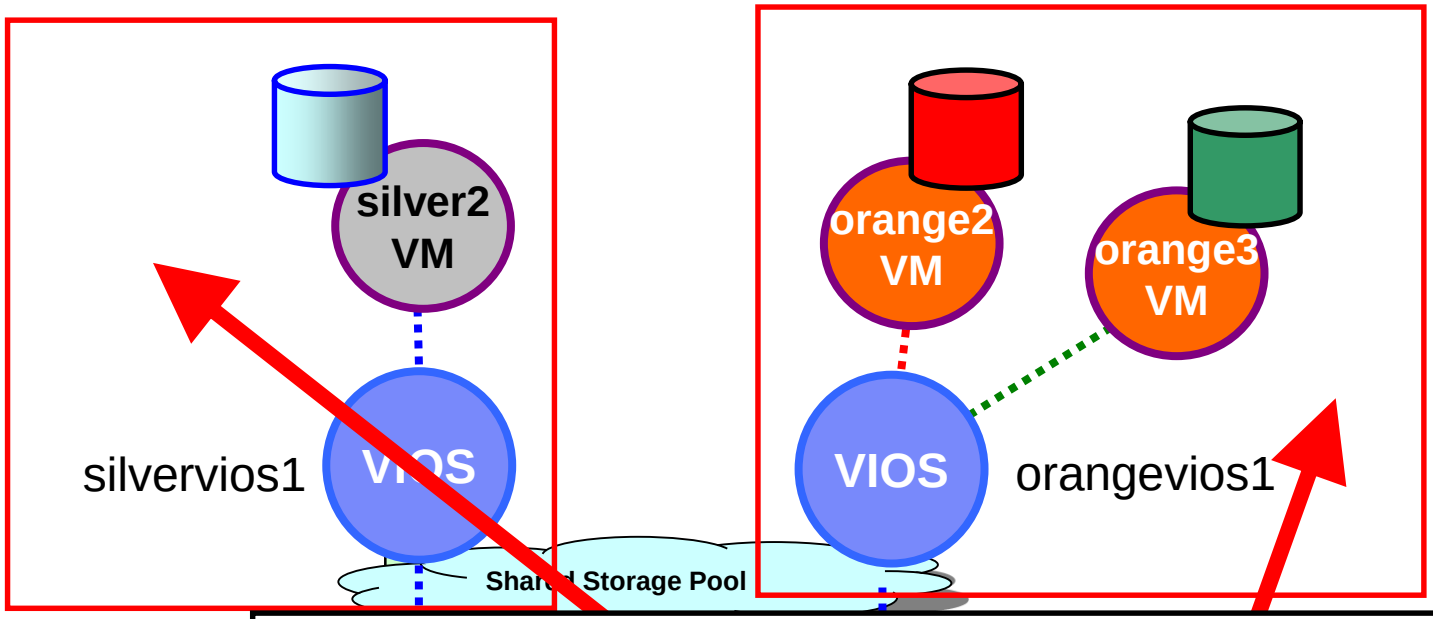
"failgrp" command to add the mirror disks on the second disk sub-system
failgrp -create -fg b: hdisk11 hdisk12



SSP4 – Demonstration Configuration



SSP4 – Demonstration Configuration

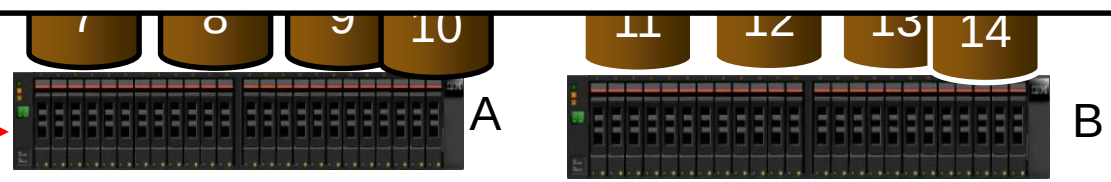


"lu" command
to allocate thin or think provisioned
space and map to the client VM
in a second
lu -create -lu xyz -size 64G -vadapter vhost42

Mirrored chunks
→ SSP managed

Two LUN sets
1 for each failgrp

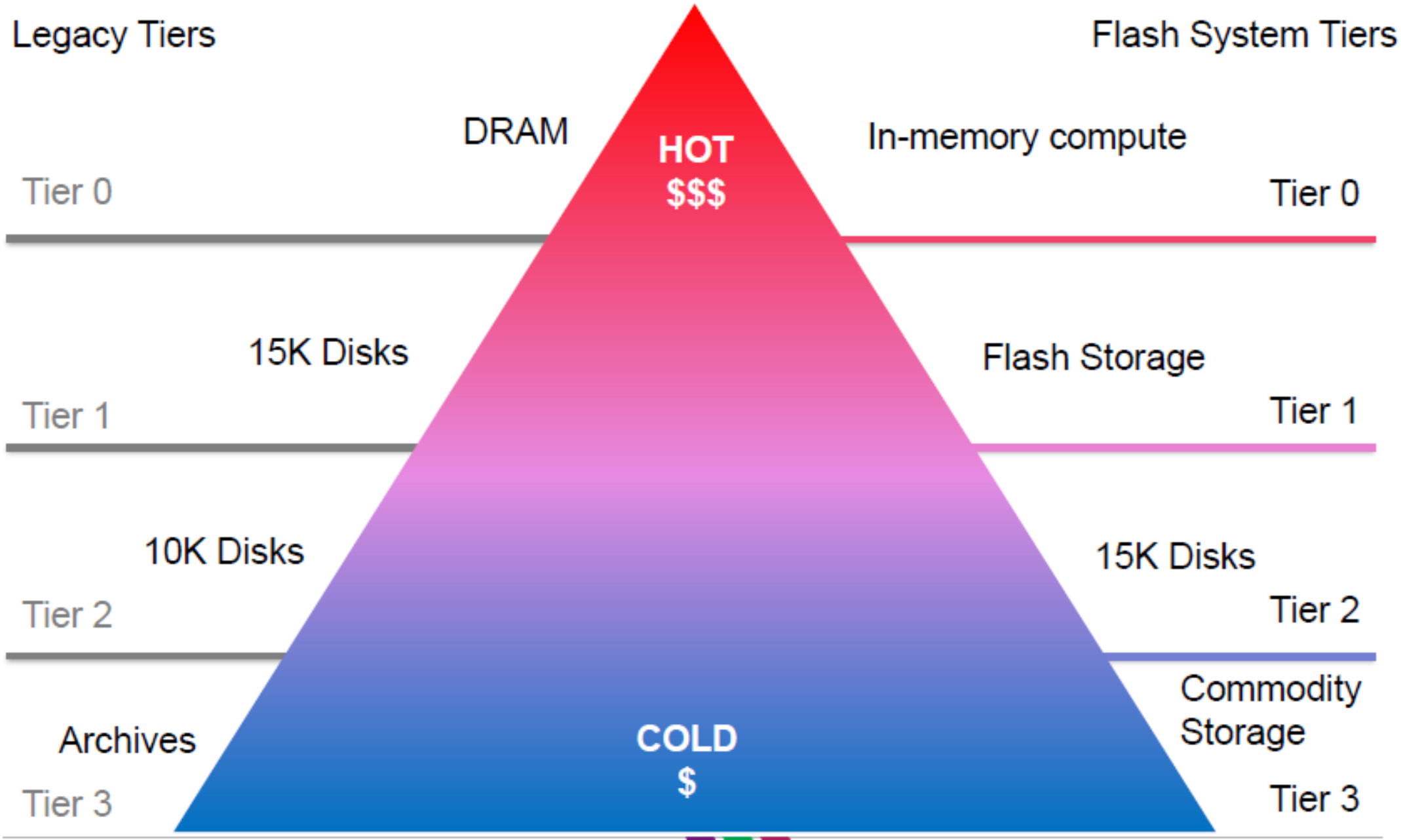
Two V7000 units



What's New in Phase 5 (December 2015)?

- HMC V8 Enhanced+ GUI support for SSP
- VIOS 2.2.4:
  - Storage Tiers
  - LUs can be moved between tiers
  - LU grow
  - LU create performance enhancements
  - SSP DB access performance enhancements
  - The **`lu -list`** command now sorts!

Storage Tiers

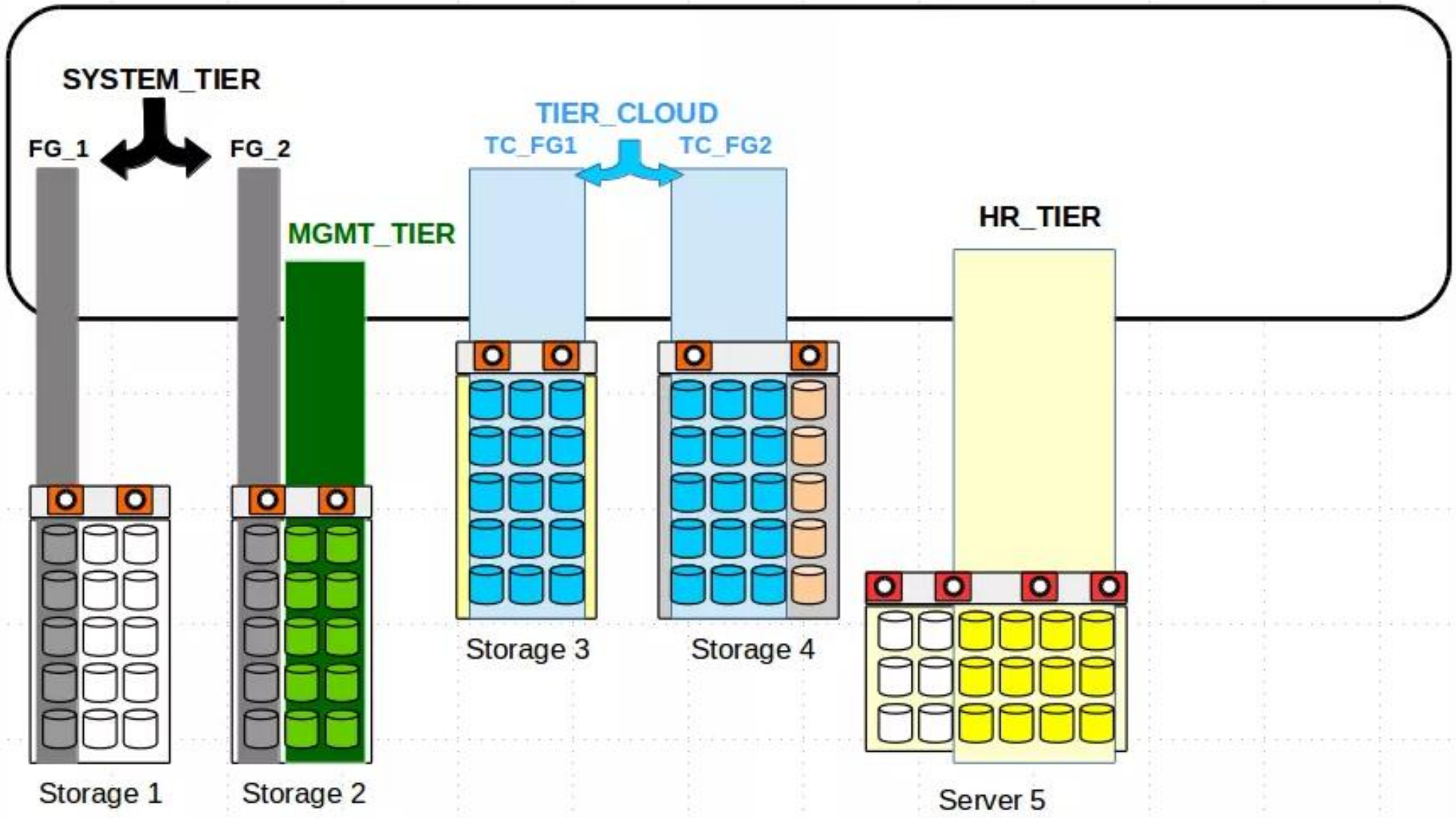


Storage tiers

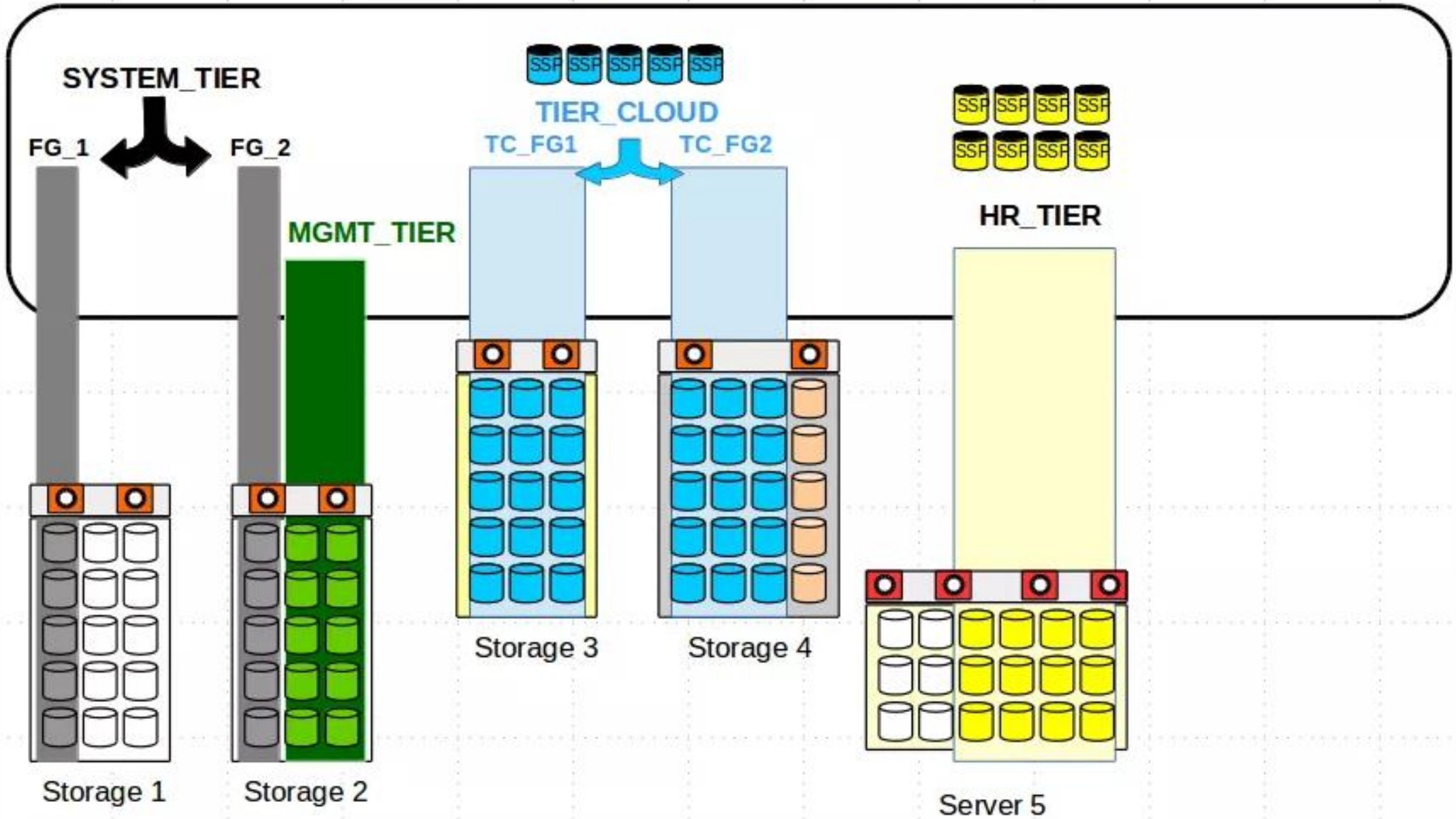
- Allows you to designate different types of storage: fast/slow, production/test, etc.
- Tiers can be individually mirrored

```
$ tier -help
Usage:
    tier -create  -tier TierName: PhysicalVolume ...
    tier -create -file -tier TierName: FileName
    tier -remove  -tier TierName
    tier -modify [-tier TierName] -attr Attribute=Value
    tier -list [-verbose | -field FieldName ...]
               [-fmt delimiter [-header]]
```

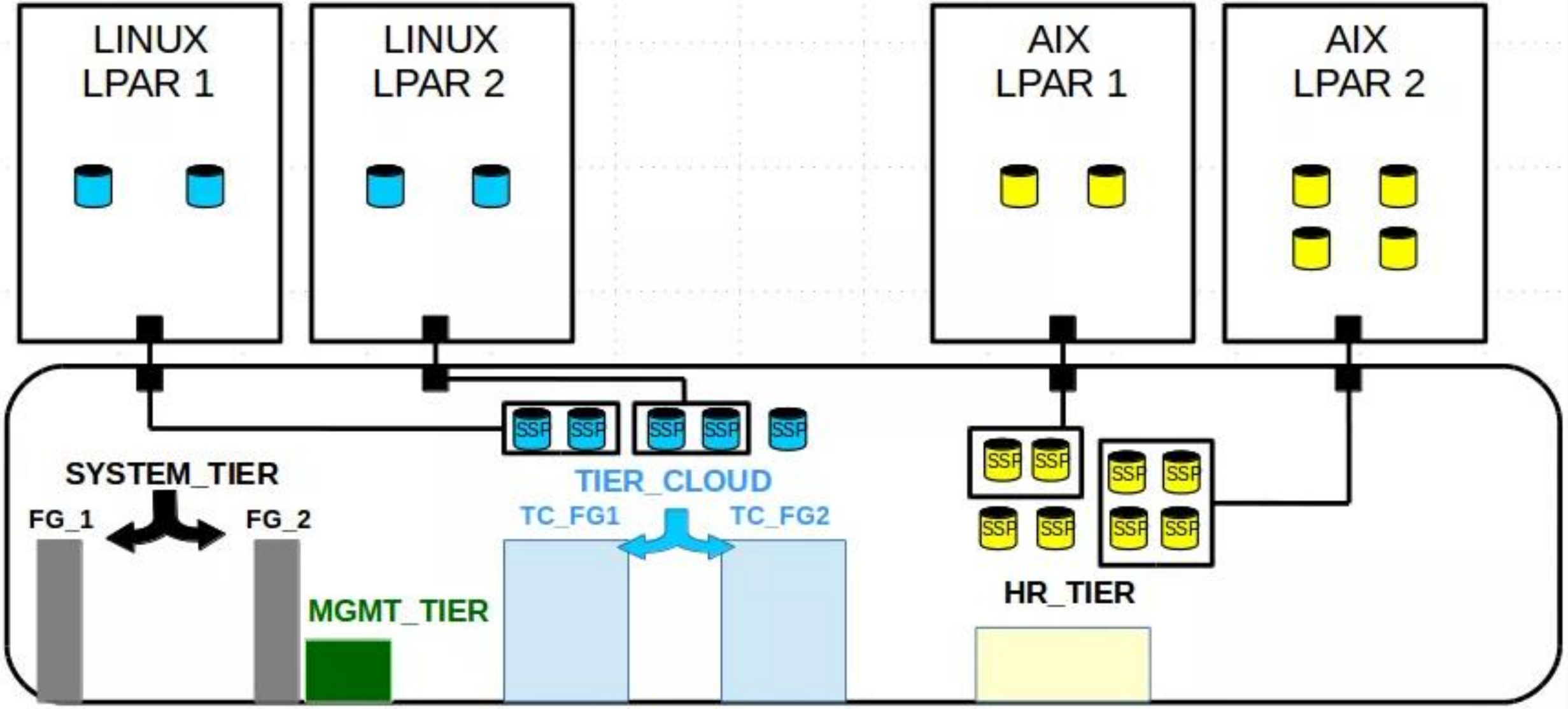
Shared Storage Pool - Tiers



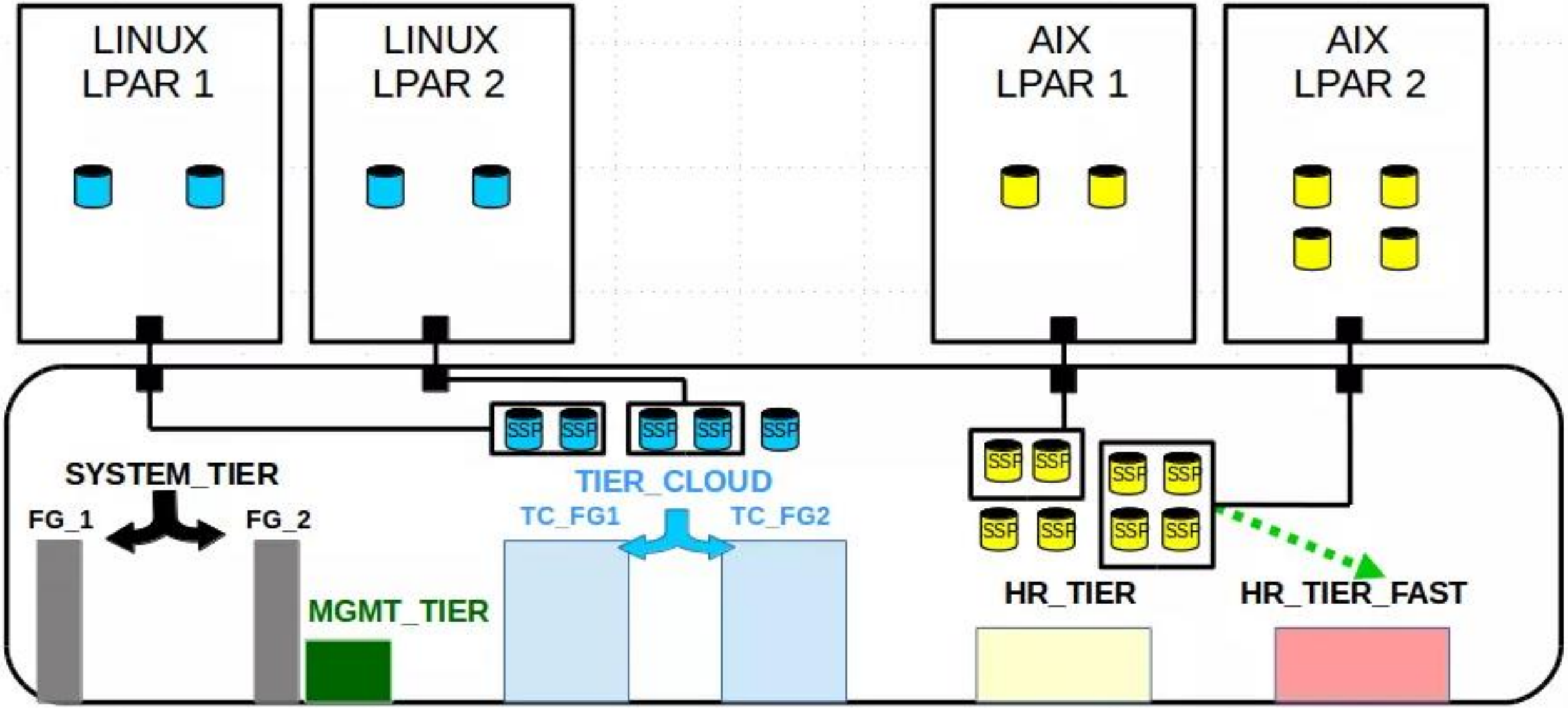
Shared Storage Pool – Logical Units created



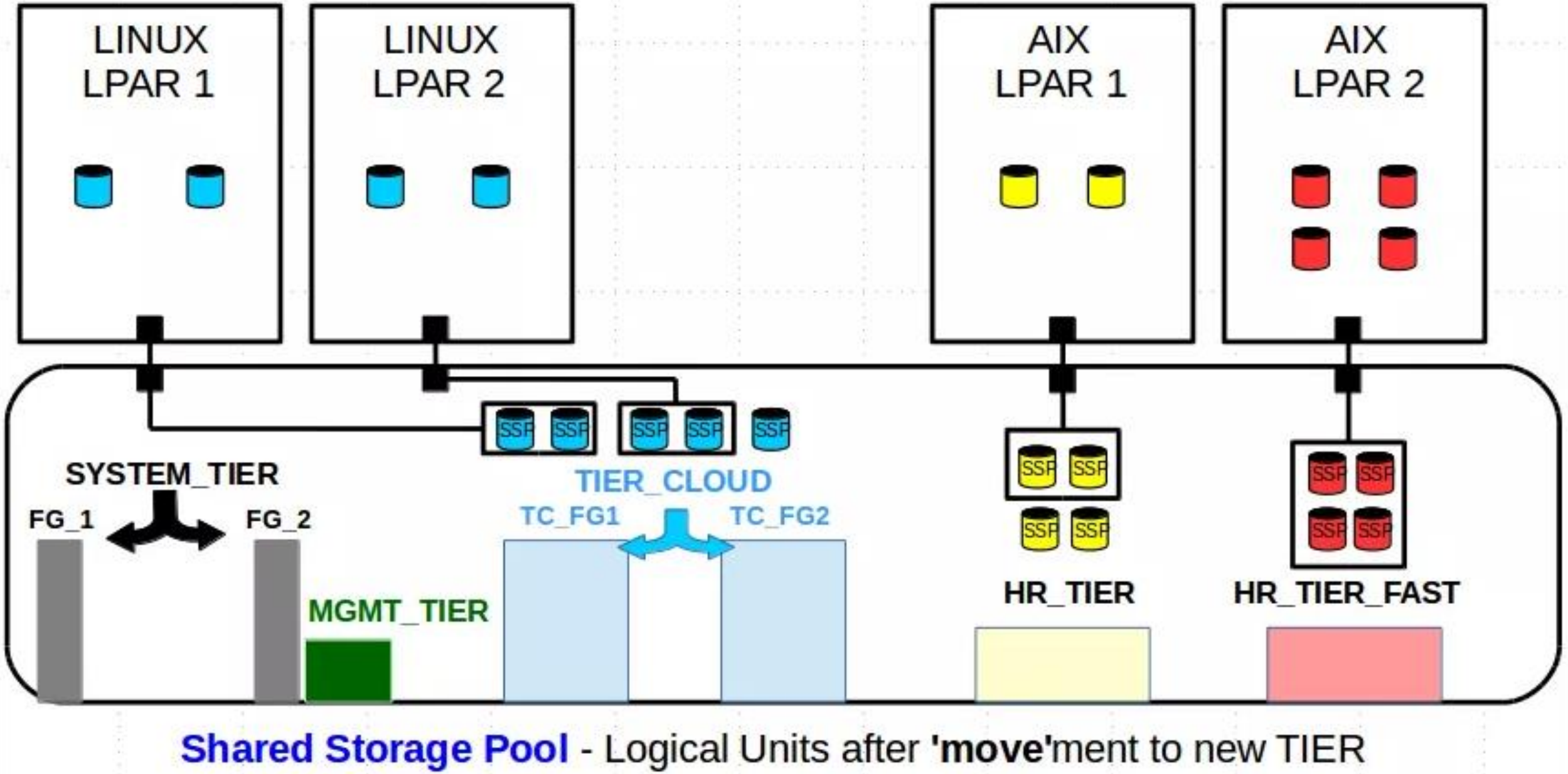
SSP LU created in Tier



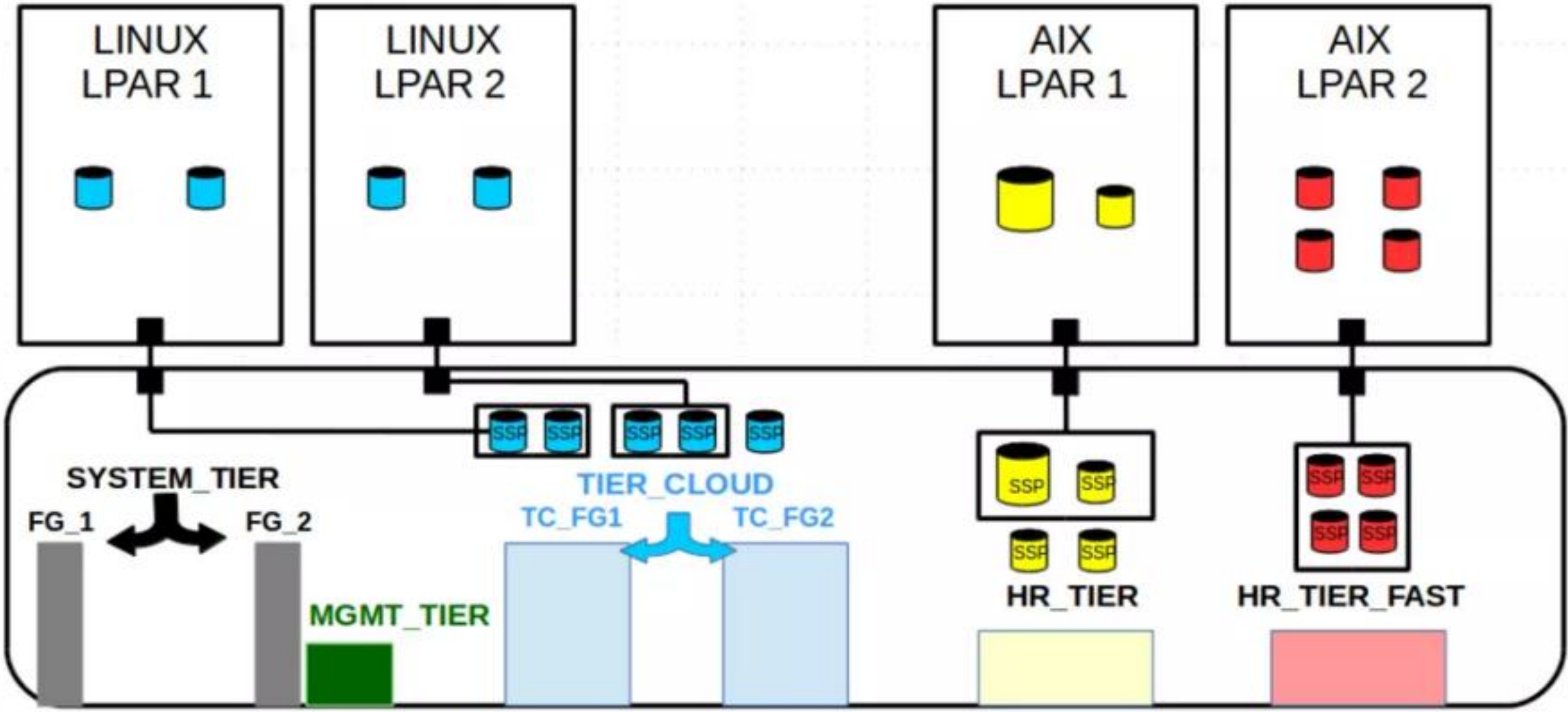
SSP LU mapped to LPAR vhost



SSP LU moved to a different tier



SSP LU after movement to new tier



Shared Storage Pool - Logical Unit being **'grow'**n

SSP LU resized(grown) in size

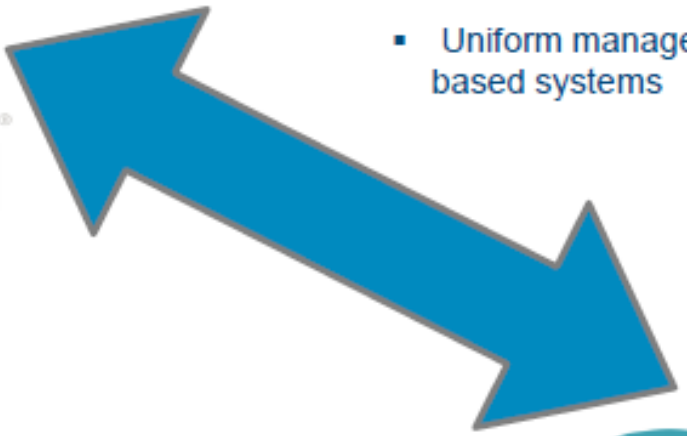
PowerVM NovaLink

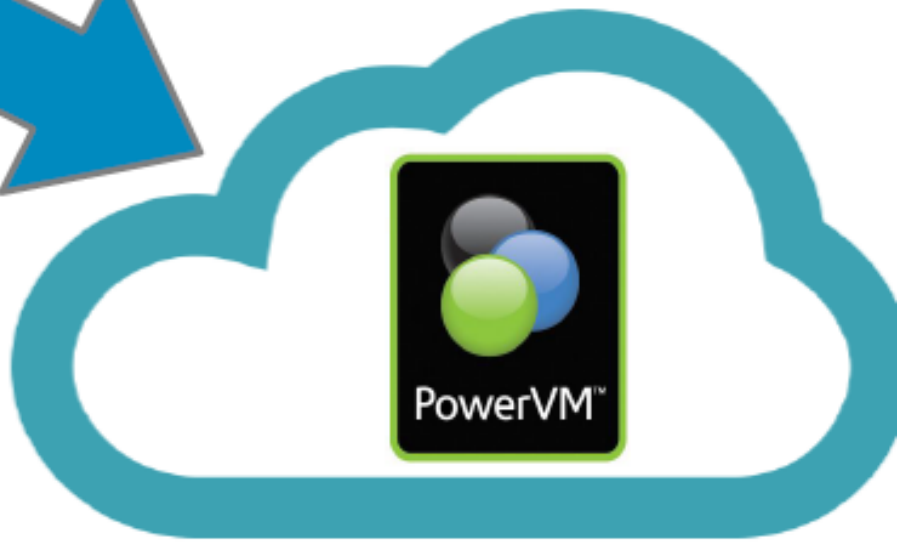
NovaLink Architecture

openstack
CLOUD SOFTWARE

Benefits

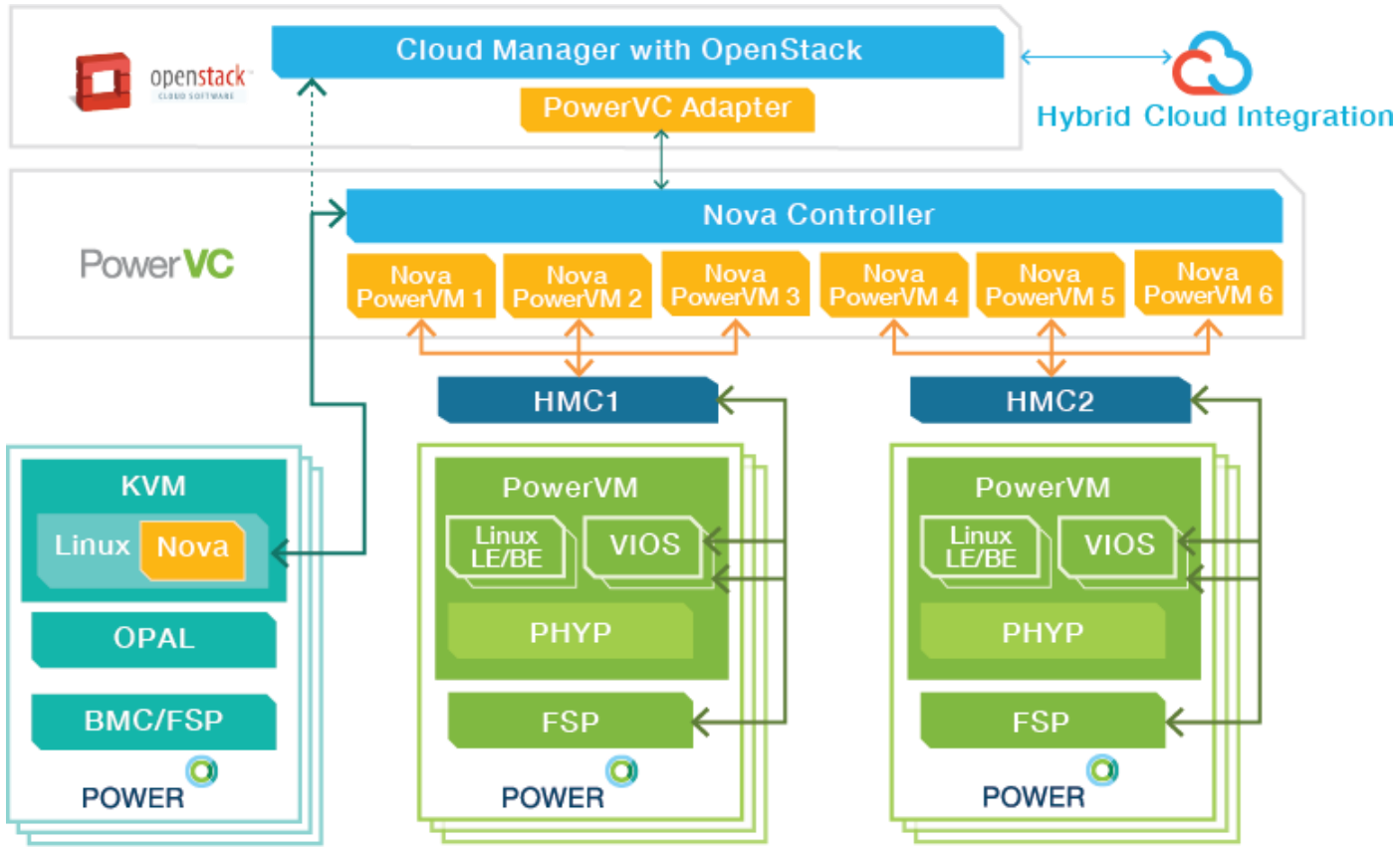
- Aligns PowerVM with the OpenStack community scale model – simplifying future OpenStack exploitation
- Simplifies management configuration – HMC not needed for virtual machine deployment and configuration
- Enables flexibility to use any OpenStack based manager to manage PowerVM
- Uniform management for PowerVM and PowerKVM based systems



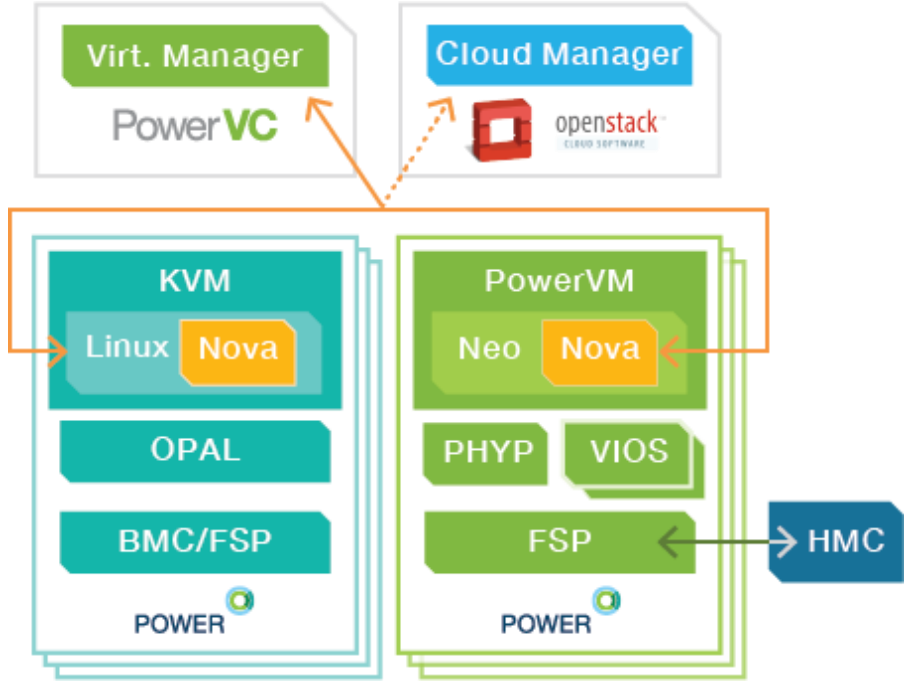


NovaLink Architecture

Current PowerVM Architecture



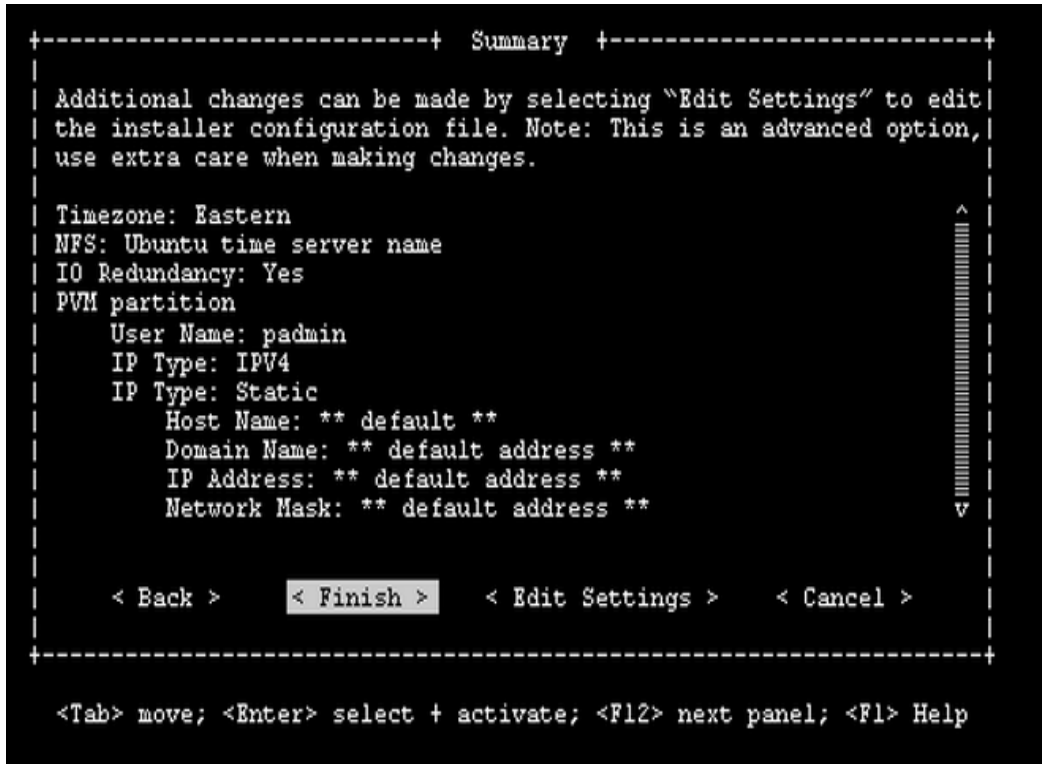
PowerVM/NovaLink Architecture



Simplified PowerVM / NovaLink Installation

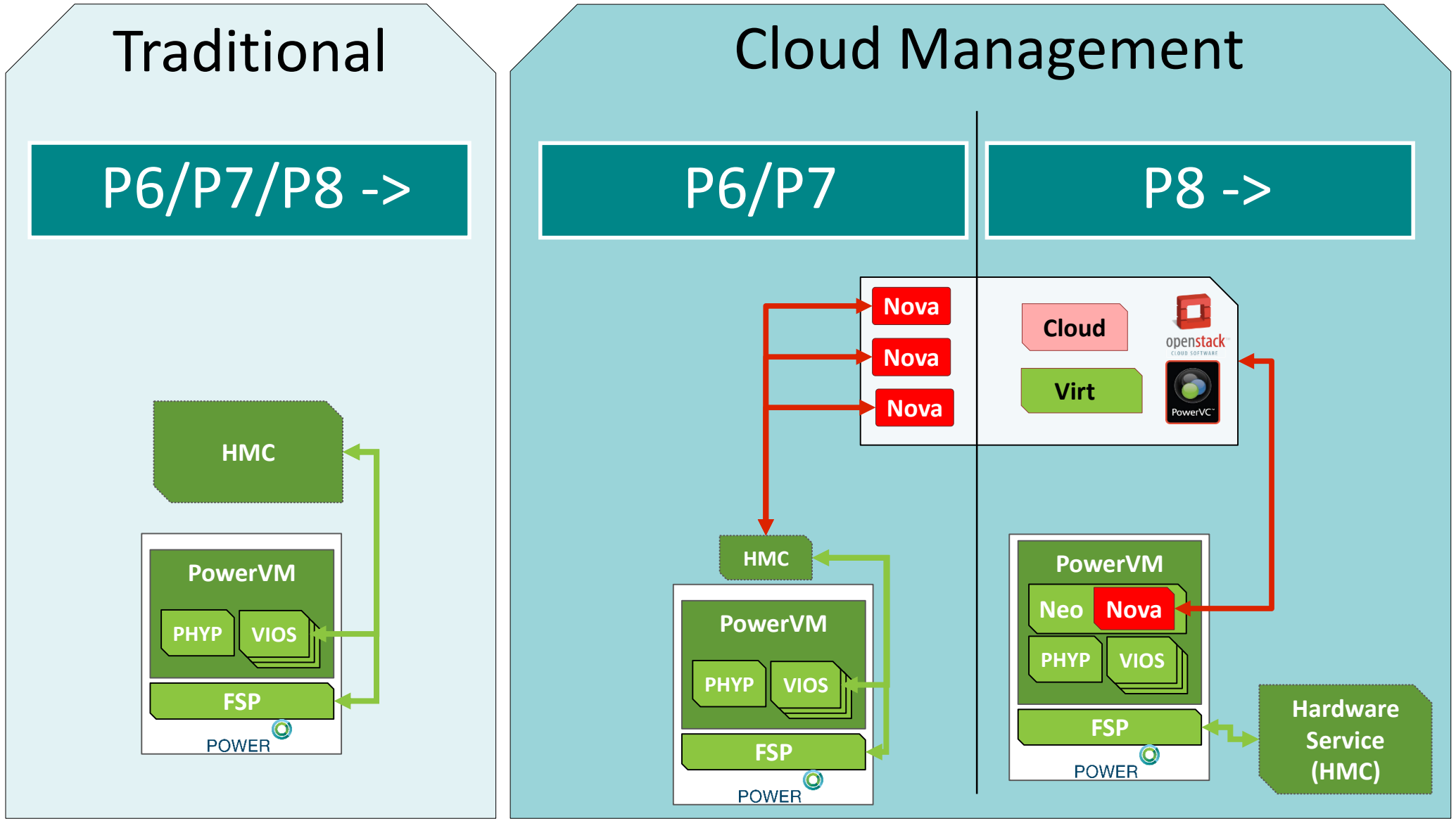
Installation prompts for

- Install Type
- VIOS & NovaLink Credentials
- NTP Server
- Network
- Storage

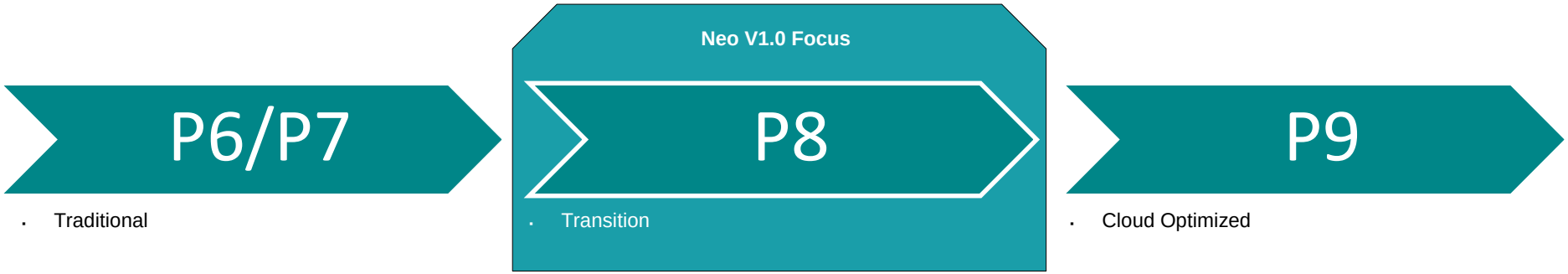


- ✓ **Bare Metal Install**
  VIOS & NovaLink are configured and ready for management
- ✓ **NovaLink Install on Existing System**
  NovaLink Partition is built and ready for management
- ✓ **Enables Install from USB Drive**
- ✓ **Works on POWER8 Systems with FW 840**

HMC and NovaLink



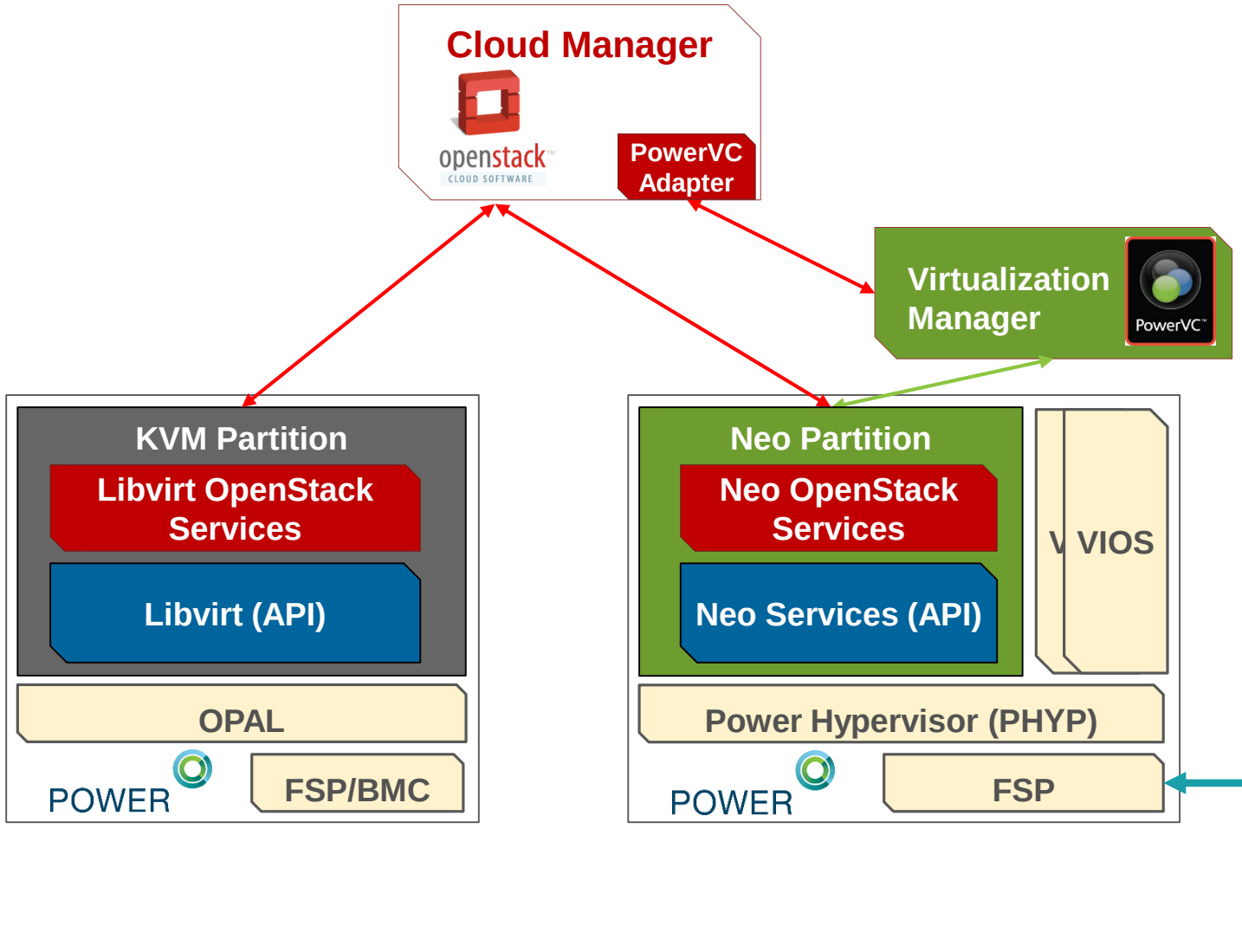
HMC and Neo



- Neo and HMC will fully coexist in the data center
  - Power8 systems can be Neo managed, HMC managed, or both Neo and HMC managed
  - An HMC co-managing Neo-P8 systems can manage existing Power6 and Power7 systems

- For Power6, 7, and 8 – Cross Neo/HMC Partition Mobility will be supported
  - Neo will support the same migration specification as the HMC
  - This allows a Neo-only Power8 system to migrate to and from an HMC-only system
- HMC remains the focus for hardware & systems management
  - Allows Neo to focus on improving common virtualization investments
  - No support for heterogeneous Global CoD or Remote Restart

Neo Platform View



Neo Services

- Neo delivers a new virtualization API
- Decouples virtualization management
- Supports decentralized architecture

OpenStack Services

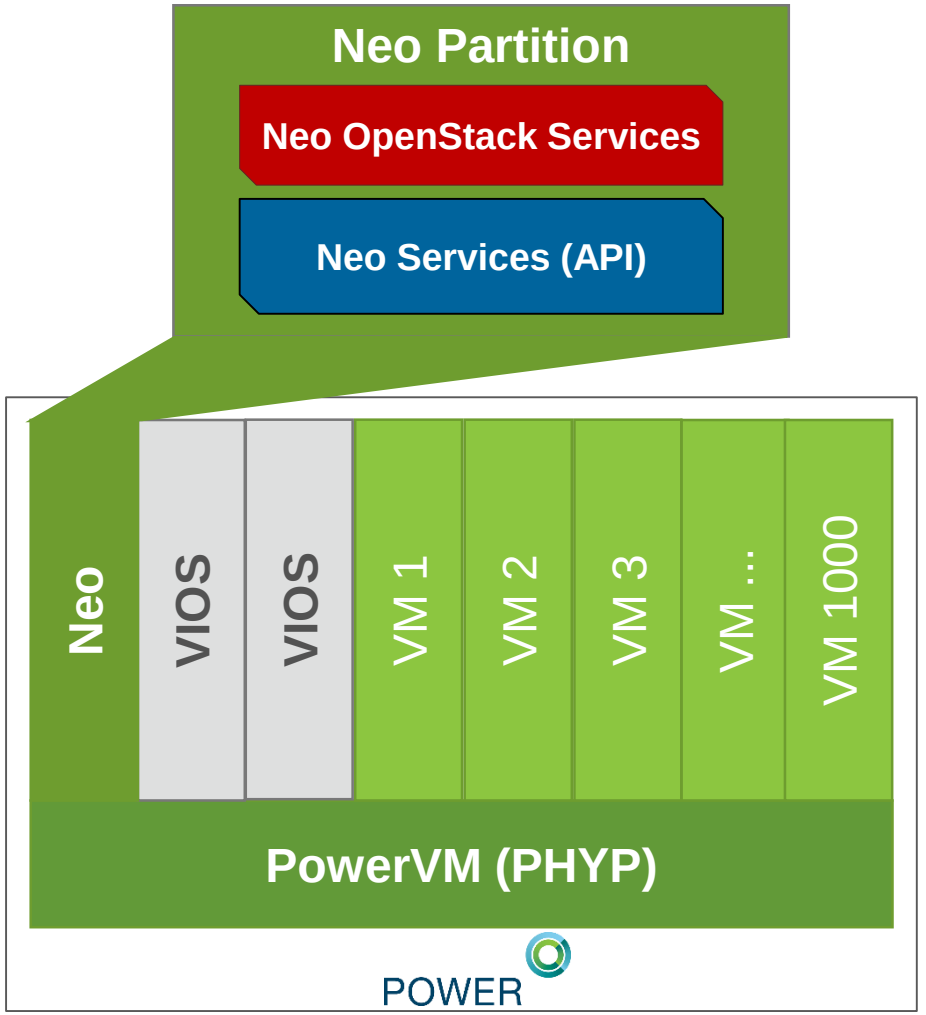
- Native OpenStack community driver
- Moves compute services into host
- Enriched upward integration

Hardware Services

- Improved management scalability
- Reduced infrastructure complexity
- Order of magnitude faster provisioning

Neo High Level Architecture

- Neo is a partition running on each system
  - Ubuntu PPC64 LE -> POWER8 only
  - Neo services provide two primary components:
    - Base level virtualization interaction with PHYP and VIOS
    - A virtualization interface with a K2-like schema - heavily optimized and re-implemented where necessary
  - Neo runs OpenStack Nova Compute, Neutron Agents, and Ceilometer Agents
- VIOSes and Neo Partition will be sized on installation to minimize customer impact



Neo Partition FAQ

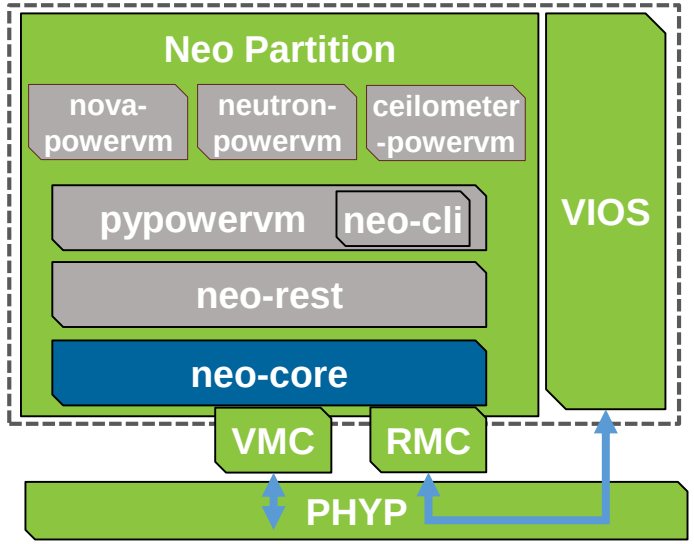
- **Is it an appliance?**
  - No – NovaLink is focused on giving customers control over their environments
  - Customers have varying requirements – PowerVC vs OpenStack, Chef vs Puppet – flexibility is key
  - We will have ***secure*** defaults defined for common scenarios, but many users have custom security modules
- **Will there be an additional cost?**
  - No – Neo will be included in your PowerVM edition (standard or Enterprise)
- **Why Ubuntu versus RHEL?**
  - Ubuntu has strong ties to the open industry standards
  - Canonical (Ubuntu's parent company) has a close partnership with IBM
- **I'd rather run on RHEL/Suse?**
  - Future versions **may** ship Neo as an installer that can be put on other Linux distributions
  - Focus for now is on simplicity, iterating on customer demand in the future
- **Can I still use the HMC?**
  - Yes – they can co-exist
  - However, the primary interface should be PowerVC
- **Do I still need an HMC?**
  - For smaller systems, the HMC is optional (as it always has been), but for larger systems, you still need the HMC for service and hardware management.
- **Do I have to use NovaLink? Is the HMC going away?**
  - No – NovaLink is focused on enabling Cloud like management function for PowerVM (via PowerVC). If you're not using PowerVC, then you can continue to use your HMC to drive your PowerVM management.
- **Does NovaLink only work with PowerVC?**
  - No – there will be community OpenStack drivers available, allowing you to use your own OPpnStack infrastructure to manage PowerVM.
  - However, PowerVC will continue to deliver significant value for PowerVM (and PowerKVM) above and beyond what vanilla open stack delivers.

Thank you

BACKUP SLIDES

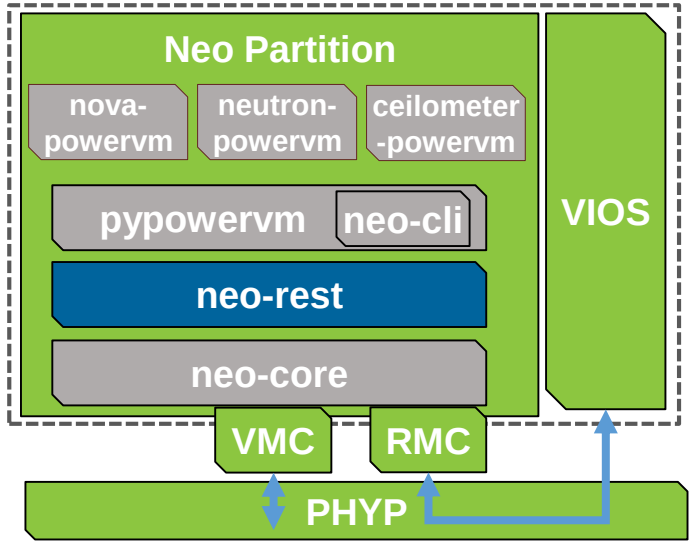
Neo Core

- **Neo Core** – Manages communications and core function of system virtualization resources
- The core is a new codebase utilizing the existing IVM model
  - C++ based - designed to be as lightweight as possible
  - Focused on the concept of minimal state
- Power Hypervisor (PHYP)
  - Neo communicates with PHYP using the Virtual Management Channel (VMC)
  - PHYP supports only one VMC connection
- Virtual I/O Server(s) – VIOS
  - Neo communicates with all VIOSes via RMC
  - Communications are over an internal, secure RMC network with the trunk adapter in the Neo partition



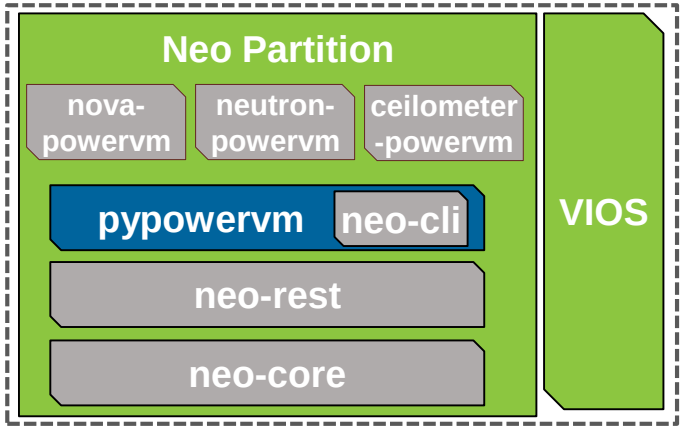
Neo REST – RESTful Interaction

- **Neo REST** provides a RESTful API that provides access to the virtualization functions of PHYP and VIOS
  - Based on K2 but with major revisions for optimizations
  - Breaks the schema in certain places to speed up operations
  - Communicates with the Neo Core via JNI
- Focused on speed and small footprint
  - Goal is to more closely match KVM operation return times
  - Minimizing the impact to system resources is key



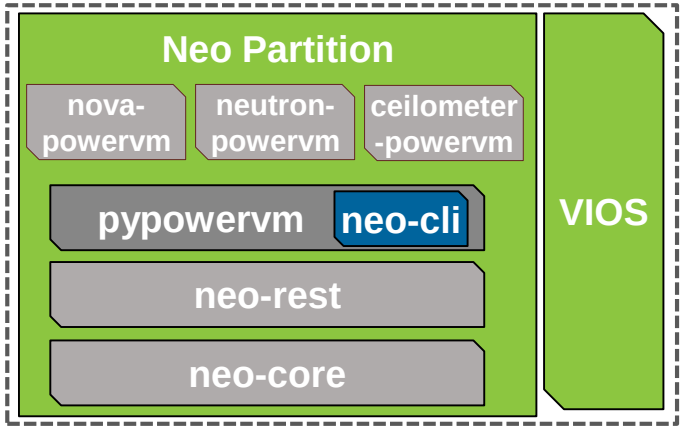
pypowervm – Interfacing Python with Neo

- **pypowervm** is an open source library for interfacing with Neo's REST API
- Open source and available on Github
  - https://github.com/pypowervm/pypowervm
- Provides the foundation for software function
  - Goal is to abstract away many "PowerVM-isms" from the end user of the API
  - Focus is on consumability, making it easy to write programs to interact with PowerVM



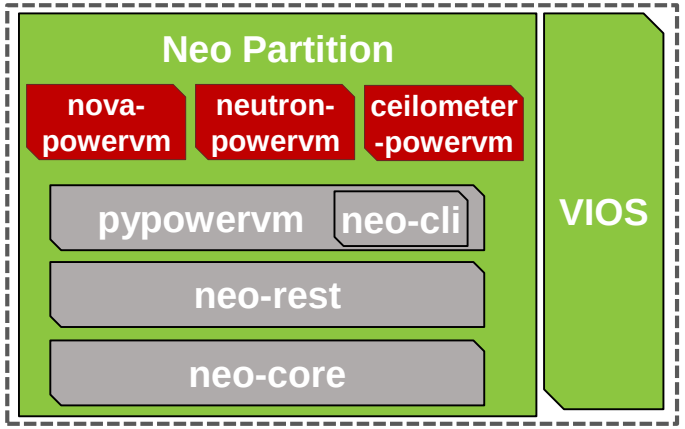
Neo CLI – Admin Management

- The **Neo CLI** is a Python-based CLI for admin functions on a Neo managed system
- Interfaces with the REST API entirely through pypowervm
- Goal is to provide a CLI focused entirely on human readability and usability
- Provides the foundation for infrastructure management in a Neo-only environment
  - Infrastructure management
- powervmctl
  - Obtain help on any command with 'help'. EG:
  - powervmctl lpar help
  - powervmctl lpar power-on help



Neo OpenStack - Community Support

- **Fully support community PowerVM compute drivers**
  - Focus is on following community standards
  - Driver is in development, working towards promotion to Group B status for OpenStack 'M' release
- Utilize the drivers as a way to show PowerVM use cases when interacting with the community
- Provide a full third-party CI system to catch community changes that would break our drivers
- Extend support for Enterprise Virtualization atop the community driver
- Currently in StackForge, an incubator for OpenStack projects
  - https://github.com/stackforge/nova-powervm
  - https://github.com/stackforge/neutron-powervm
  - https://github.com/stackforge/ceilometer-powervm



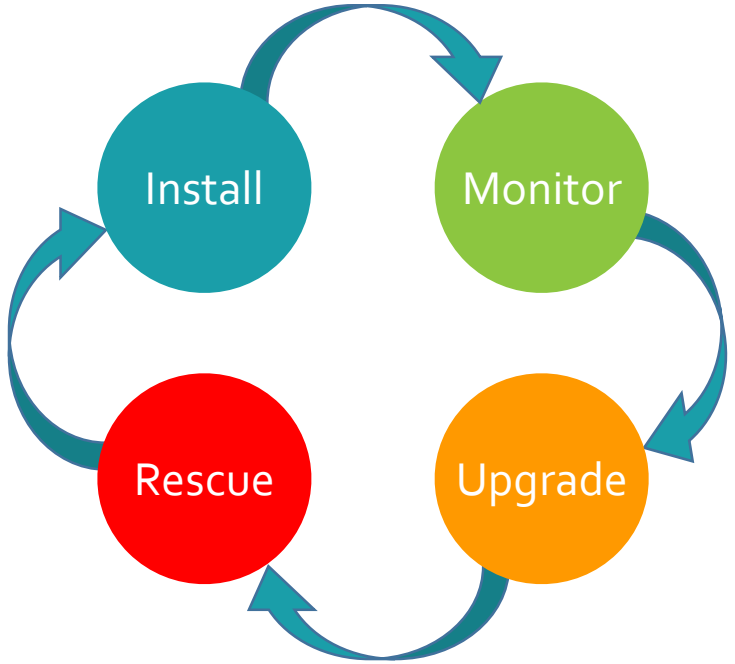
What is ‘OpenStack Compute’

- **Nova** – Compute Virt Driver
  - Must be able to deploy a VM using standard technologies quickly and repeatedly
- **Neutron** – Networking ML2 Agent
  - Must be able to support standard network use cases.
- **Cinder** - Storage Support
  - Build upon existing Community Cinder Drivers for Storage Connectivity
- **Ceilometer** – Performance Monitoring Agents
  - Gather Compute, Storage and Network metrics to report back to the core

Neo Lifecycle

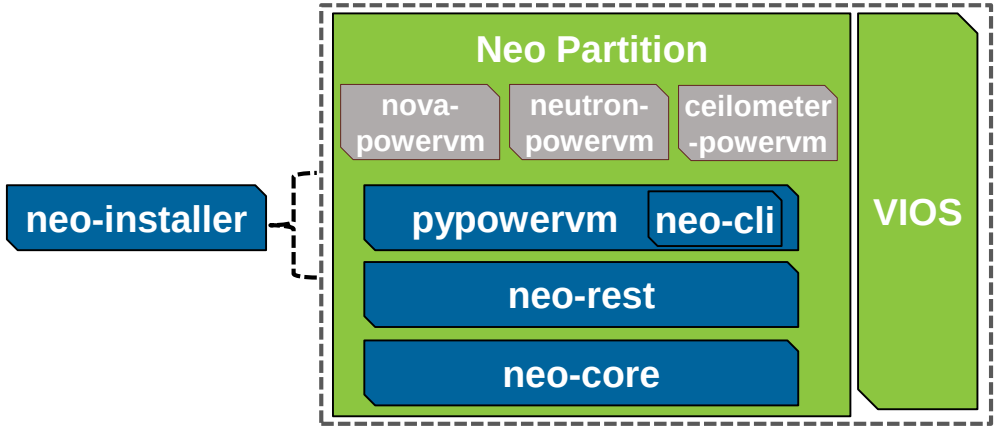
- **There are four major parts of the Neo lifecycle**
  - Installation & Setup
  - Monitoring & Maintenance
  - Upgrade
  - Rescue & Recovery

- **Focus is on ensuring sure these lifecycle processes are low impact**
  - Driving automation support wherever possible
  - Support industry standard tooling
  - Designing rapid recovery methods



Neo Install

- **Provides two models:**
  - A TTY-based install wizard
  - A fully automated installation with an advanced template for complex configuration
- **Two types of installations:**
  - In Greenfield, Neo will be installed onto a system either via netboot (automatable) or via USB
  - In an HMC environment, Neo will be created from the HMC using standard create commands. Then:
    - chcomgmt –m <SYS> -o setmaster –t norm
    - chsyscfg –m <SYS> -r lpar –I lpar_id=X,powervm_management_authority=1
    - Then netboot to the installer
- **Running the installer will result in:**
  - VIOS setup and configured (Greenfield only)
  - Ubuntu LPAR installed
  - Neo services and processes installed
  - Neo API ready for communications



Installation & Setup (cont.)

- Neo's installer is built upon the same Ubuntu image as the Neo partition
  - Focus is on resource reuse

- The Neo installer is designed to focus on customer ease of use:
  - Focused foremost on asking use case based questions
    - "Do you want redundancy" instead of "single or dual VIOS"
  - Inspects the existing system HW to generate a profile
  - Asks only what is needs to get the system operational, with intelligent defaults
  - Provides an 'advanced mode' for customers with specific needs

Compute Node Install

- **Community OpenStack**
  - Users and distributors will consume our drivers directly and integrate them
  - Able to use the existing compute node install model used for other hypervisors
- **IBM Cloud Manager**
  - Installation of our OpenStack compute services will be driven by Chef cookbooks
  - Allows users to deploy and update the OpenStack services in an automated fashion
- **PowerVC**
  - The PowerVC controller will push installation packages to the Neo partition
  - This includes the service packages and all necessary dependencies

Monitoring & Maintenance

- **Monitoring is driven through industry standards**
  - All Neo services are registered and run as standard Linux services
    - Dependent on each distribution
  - VIOS partitions can be checked through Neo or existing interfaces
  - Since Neo is an open partition, it can be monitored using standard software
- **Maintenance changes can be made through several interfaces**
  - Programmatically through the Neo REST API
  - An experimental Python-based Neo CLI, supporting a subset of Neo function
  - VIOS CLI for VIOS operations
  - HMC for Neo-HMC co-managed systems
- Services
  - Systemctl status pvm-core (core daemon)
  - Systemctl status pvm-rest (REST server
  - Lssrc (standard RSCT / IBM.LparCmdRM / etc)
- Logs
  - /var/log/pvm/ (pvm_restjni, pvm_ap, pvm_other, mobility)
  - /var/log/pvm/wlp (REST server logs – Audit, FFDC, etc.
  - journalctl (systemd, kernel module, etc.

Neo Updates

- **Neo upgrades**
  - Performed as standard package updates through each Linux distributions' standard package management
  - Package dependencies enforce prerequisite installation
  - Instructions and software necessary to set up private package repositories for secure networks will be available
- **PowerVC upgrades**
  - Handled by the PowerVC controller
  - PowerVC controller will push updates to the Neo partition for its services
- **OpenStack upgrades**
  - Performed using the standard OpenStack upgrade process

Rescue & Recovery

- Neo is designed to make rescue and recovery as simple as possible

- **Neo Internal Watchdog**
  - Code that runs within the Neo partition, restarting Neo service daemons if required
- **Hypervisor Heartbeat**
  - Hypervisor has a timer that watches for Neo activity
  - Gets reset with each message from Neo
  - Neo sends a VMC low level echo at a preset interval if there's no activity
  - If PHYP doesn't get a response within a certain time, PHYP restarts the Neo partition
  - If this fails more than a predefined number of times, creates a serviceable event

Project/Timeline

Feature Name – PowerVM Novalink

GA Date (Tentative) – 12-11-2015

Executive – Ann Funai, Power Software Enablement

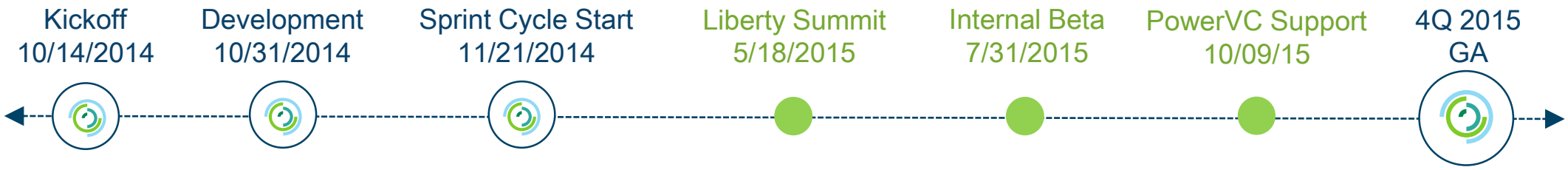
Offering Manager – T.R. Bosworth

Technical Overview

Manage-To Operating System – Ubuntu 15.10 PPC64EL on PowerVM

Availability – Core components delivered as part of Neo Installer, with public package repository for updates

OpenStack Integration - OpenStack Community, PowerVC (enhanced), IBM Cloud Manager (TBD)



Neo Design Thinking

Kicked off the IBM Design Thinking process with the workshop

•Focused on evaluating the scope of the project from a user perspective

•The workshop was broken into four primary parts:

- Determining areas for improvement
- Defining who our target users are
- Establishing our requirements and goals
- Creating our hills

Samira, an OpenStack user, can connect a PowerVM system to her OpenStack cloud and deploy her first workload in **20 minutes**.

Samira, a system administrator, can provision a new workload on an out-of-box, racked, and cabled Power system **within 1 hour**.

Rick, a PowerVC admin, can deploy **650 AIX enterprise workloads** across up to **200 hosts** in **60 minutes**.

Neo Sponsor Users

JP Morgan Chase
Simon Mortimer
Kieron Johnson

BNP Paribas
Benoit Creau
Jean-Luc Guyot

IBM Enterprise Pods
Ray Harrington

Lab Based Services
Vess Natchev
Kyle Wurgler

Connectria
Dave Wiseman

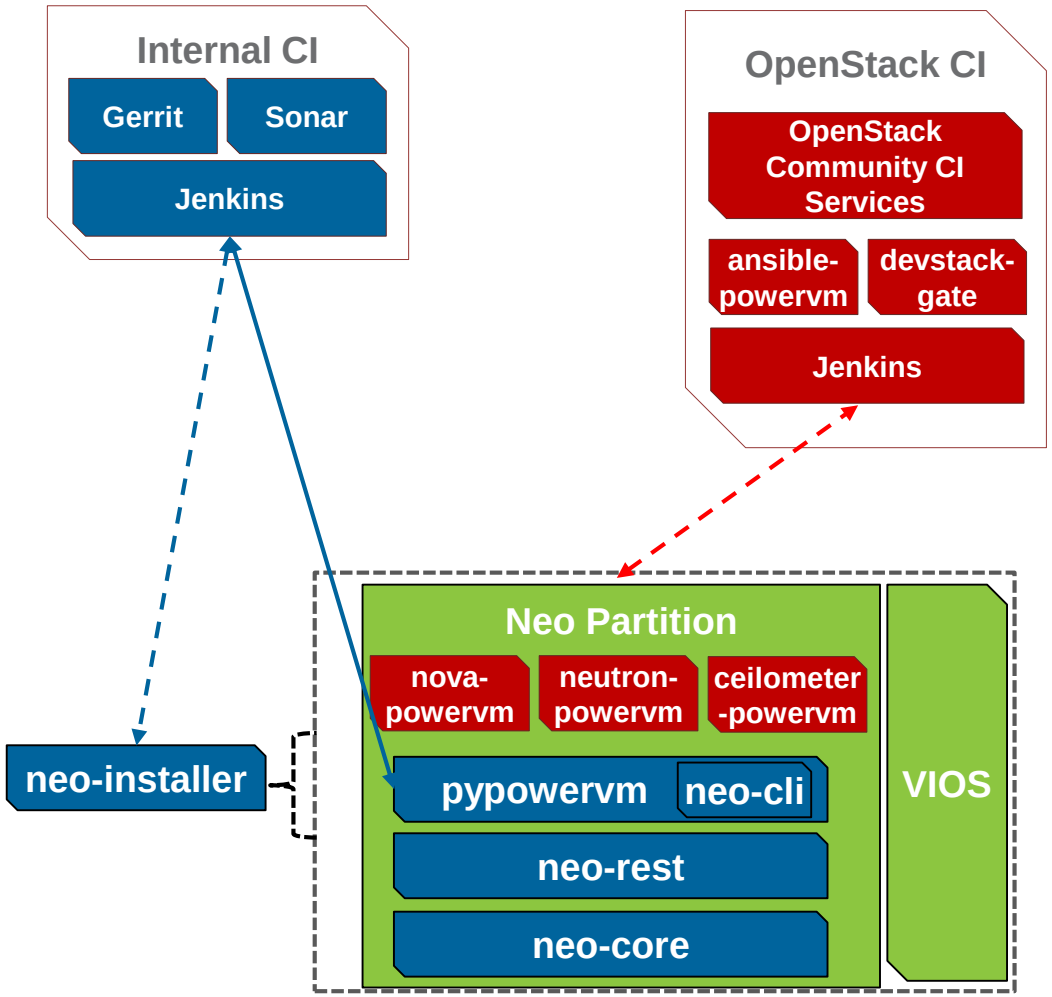
Joint effort between members of the development and design teams

- Establish relationships with existing or potential customers
- Collaborate - Meet 1-2 times a month to discuss topics related to the project
- Iterate - take their feedback to the team and make changes as needed
- Goal is to provide them with early access versions for testing

Neo Development - Continuous Integration

- OpenStack drivers require continuous integration (CI)
- CI means for every patch set that is ***proposed***
  - Build up a test environment and install the latest base code your patch relies on
  - Run all of the applicable integration and unit test cases
  - +1 or -1 the change in Gerrit
- For Neo we're extending this to all of our projects
  - Today we wait for changes to show up in a build, then for the appropriate test cycle
  - Moving to new levels is quite challenging. No checks or balances to validate the code
  - With CI we get provide feedback almost instantly, in an incredibly 'hands off' approach.
  - *Significant cost reduction will be seen when moving to new levels of code.*

Infrastructure Status



Glossary

Internal CI – Set of continuous integration services for all of our internal services

OpenStack CI – Set of continuous integration services required to test our driver in the OpenStack community

ansible-powervm – Automated delivery of nova/neutron/ceilometer drivers on to Neo system.

devstack-gate – Automated environment deployment for PowerVM Development

Neo Development – Open Source

- **nova-powervm**
  - Gerrit: https://review.openstack.org/#/q/status:open+project:stackforge/nova-powervm,n,z
  - Git: git.openstack.org/stackforge/nova-powervm
  - Open Source Bug Tracking: https://bugs.launchpad.net/nova-powervm
- **neutron-powervm**
  - Gerrit: https://review.openstack.org/#/q/status:open+project:stackforge/neutron-powervm,n,z
  - Git: git.openstack.org/stackforge/neutron-powervm
  - Open Source Bug Tracking: https://bugs.launchpad.net/neutron-powervm
- **ceilometer-powervm**
  - Gerrit: https://review.openstack.org/#/q/status:open+project:stackforge/ceilometer-powervm,n,z
  - Git: git.openstack.org/stackforge/ceilometer-powervm
  - Open Source Bug Tracking: https://bugs.launchpad.net/ceilometer-powervm
- **pypowervm**
  - Gerrit: http://morpheus.rch.stglabs.ibm.com:8080/#/q/project:pypowervm
  - Git: ssh://<userid>@morpheus.rch.stglabs.ibm.com:29418/pypowervm
  - Open Source Bug Tracking: https://bugs.launchpad.net/pypowervm

Neo Development – Bug Process

- A key community metric is ‘involvement’. This is shown in many ways – commits, IRC discussion, bug reports, etc...
- As such, bugs against the open source components will likely have an Open Source Bug in Launchpad.
- Bug reporters must follow the rules when opening bugs externally:
  - https://w3-connections.ibm.com/wikis/home?lang=en#!/wiki/W71527676a1d3_4c3f_9f75_43d3d96f3a2e/page/Contributing
  - No reference to IBM, PowerVC, product schedules, product names, etc...
- An internal tracking defect may be opened to the Neo Jazz Server to facilitate the internal, confidential discussions:
  - https://jazz07.rchland.ibm.com:13443/jazz/web/projects/NEO