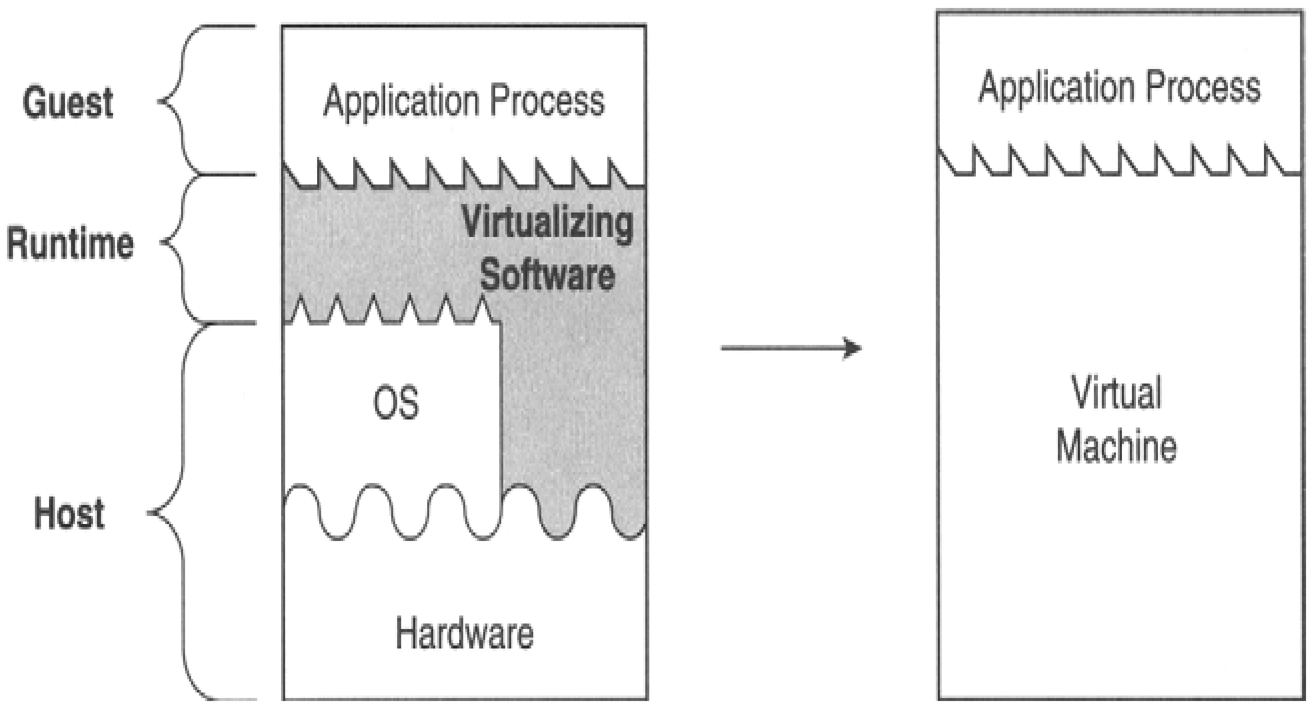
MODULE – IV
Types of Virtualization

Application virtualization - desktop virtualization - network virtualization - storage virtualization - comparing virtualization approaches

Application Virtualization

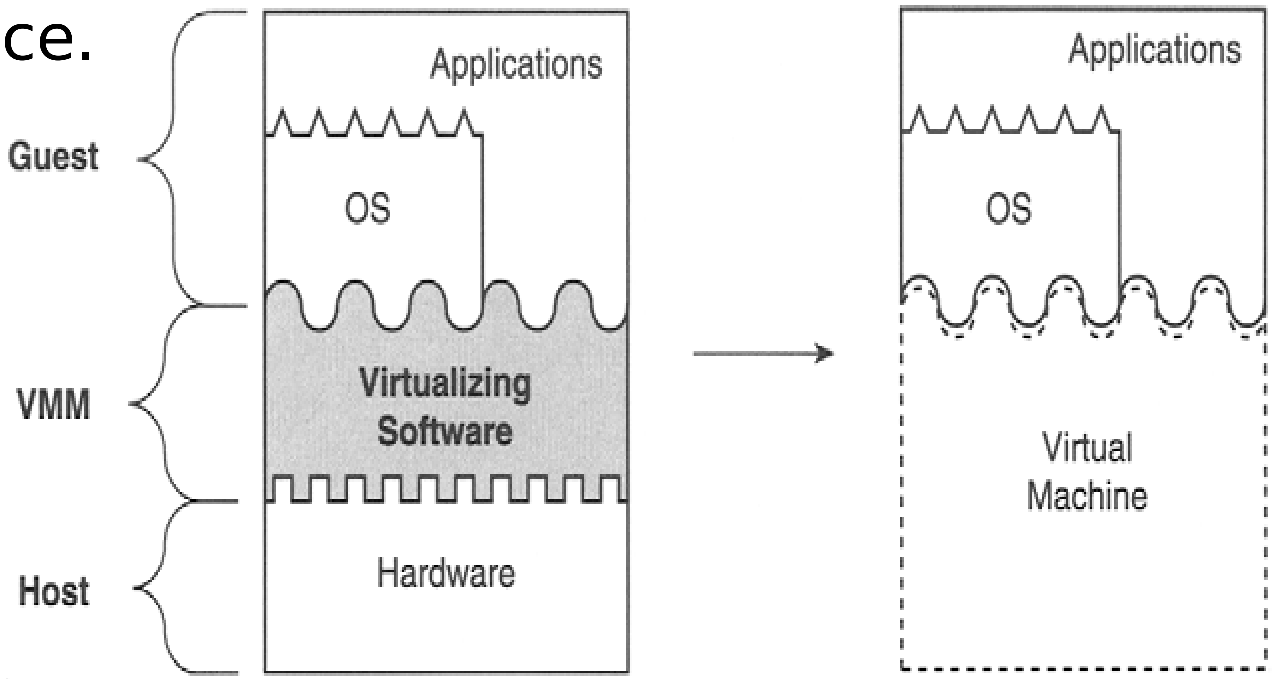
Process VM

Virtualizing software translates a set of OS and user-level instructions composing one platform to another, forming a process virtual machine capable of executing programs developed for a different OS and a different ISA.



System VM

Provides a complete system environment.
Supports an operating system along with its potentially many user processes.
Provides a guest operating system with access to underlying hardware resources, including networking, I/O, a display and graphical user interface.



Application Virtualization a.k.a Process Virtualization

- Software technology that **encapsulates (not abstracts)** computer programs from the underlying operating system on which it is executed

  Difference between abstraction and encapsulation

  - Hiding the details against hiding the access

- specifically designed to run a **process** or program **without being dependent on the platform environment**

Application Virtualization a.k.a Process Virtualization

Application behaves at runtime like it is **directly interfacing with the original operating system** and all the resources managed by it, but can be isolated or **sandboxed** to varying degrees

- Example
  - JVM for Java, Common Language Runtime for Windows and WINE software installed in Linux can run windows processes like Word

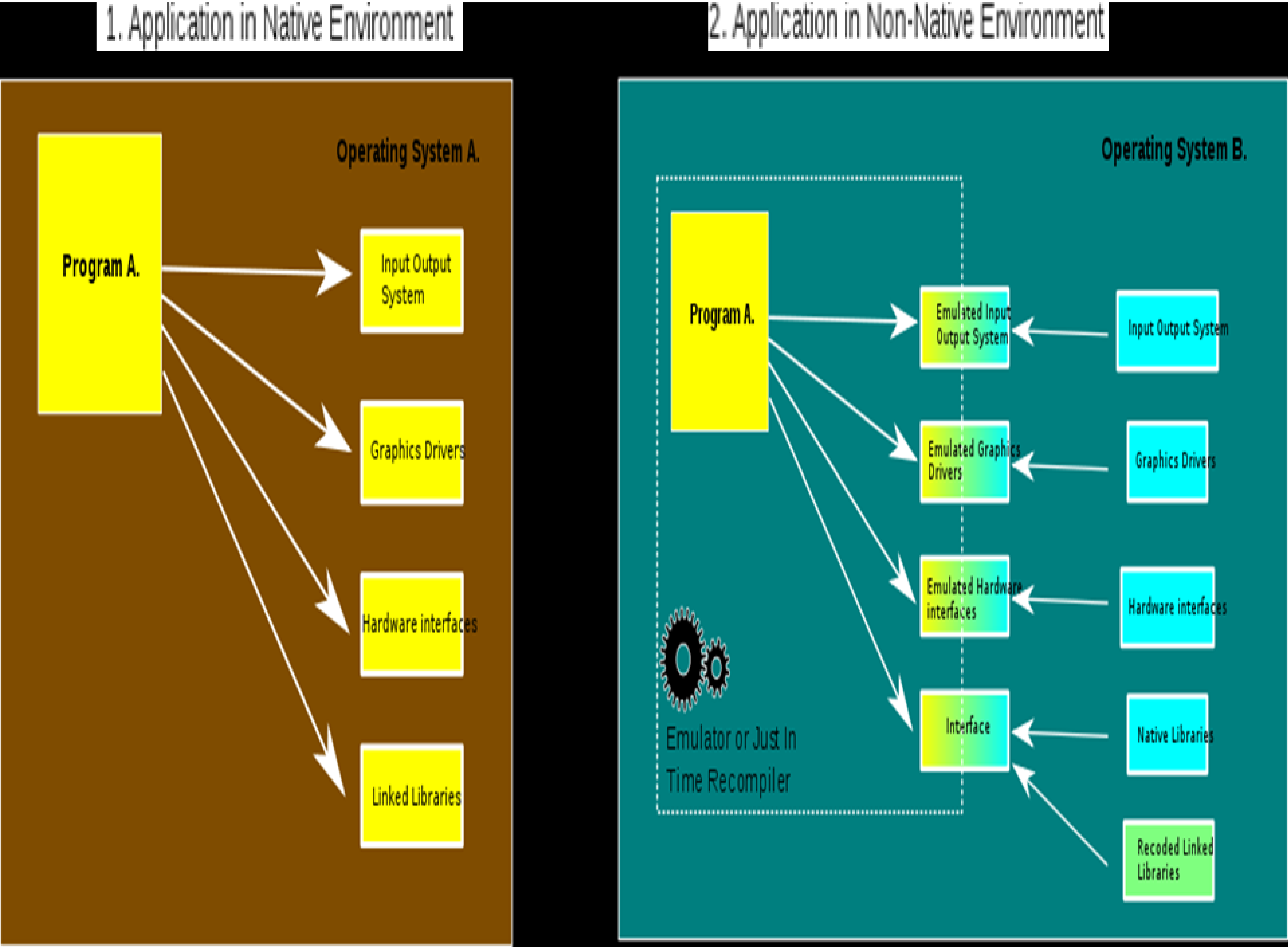
Runtime environment

- Runtime system that provides an environment in which programs run
  - layout of application memory,
  - how the program accesses variables,
  - mechanisms for passing parameters between procedures, interfacing with the operating system
  - setting up and managing the stack and heap
  - garbage collection, threads or other dynamic features

Runtime environment…

What are two choices for implementing application virtualization?

- **Full application virtualization**
  - Implements a complete **system VM** using **abstraction** (**virtualization layer**)
- **Process virtualization**
  - Virtualizes the **only interfaces** of runtime environment using **encapsulation** (**does not require virtualization layer**)


1. Application in Native Environment
Operating System A.
Program A.
Input Output System
Graphics Drivers
Hardware interfaces
Linked Libraries
2. Application in Non-Native Environment
Operating System B.
Program A.
Emulated Input Output System
Emulated Graphics Drivers
Emulated Hardware interfaces
Interface
Input Output System
Graphics Drivers
Hardware interfaces
Native Libraries
Recoded Linked Libraries
Emulator or Just In Time Recompiler

How application virtualization is implemented?

- **Packaging the application**
  - Application is installed within custom packager which records all files, registry and settings related to application
- **Delivering the app to the target system**
  - The packaged application is delivered to the target system through USB or web

How application virtualization is implemented?

- **Executing the app in virtual environment**
  - Finally the app is executed within the virtual environment which is completely isolated from other applications and underlying operating systems

Desktop Virtualization

Desktop Virtualization – in a brief

Desktop virtualization moves desktop computing to virtual machines hosted on servers.

- The end user interacts with a full desktop (typically Windows) environment while management of software, processing, and storage is removed from distributed desktop machines.

There is significant interest in desktop virtualization.

- About 39% of organizations are planning, piloting, or implementing solutions.
- Currently about 5% of organizations have already implemented.

Desktop virtualization is an emerging technology ready for prime-time and should be part of every organization's technology strategy.

- Desktop virtualization is a relative newcomer and it is still in its early days in terms of adoption.
- However, it has significant potential to address the end user needs of a larger proportion of the end users than the established presentation virtualization (Citrix XenApp, Microsoft Terminal Services).

Some limitations still remain.

- Desktop virtualization does not yet adequately meet the needs of truly mobile users or those who have high media processing requirements.

Virtualization is all about layers of abstraction

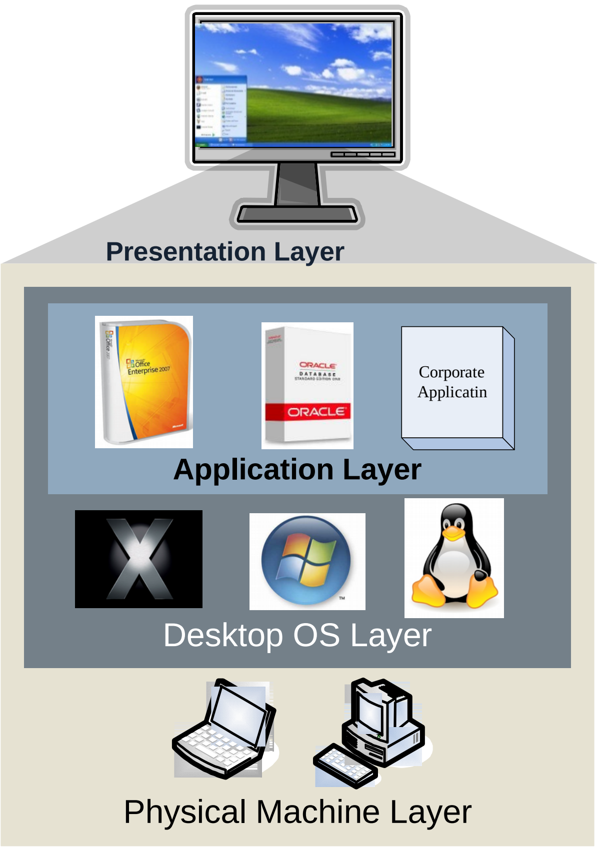
Key Definitions for Desktop Virtualization

A desktop computer is comprised of four system layers:

Physical Machine	The processor, memory, and storage that sits on the desk.
Operating System	Typically Microsoft Windows but could also be a desktop Linux variant or the Mac OS.
Applications	Such as an e-mail client, word processor, spreadsheet, or enterprise application client (example: client side application for a CRM system).
Presentation	The user interface including what is presented on the desktop monitor and interacted with via input devices (keyboard, mouse).

- Virtualization inserts a layer of abstraction between system layers. The system layer above interacts with an abstraction rather than something “real”.

Example: In presentation virtualization (such as Terminal Services) the user interacts with a presentation of an application on their desktop. But the “real” application, OS, and physical layers behind the presentation are not on the desktop but on a server located somewhere else.

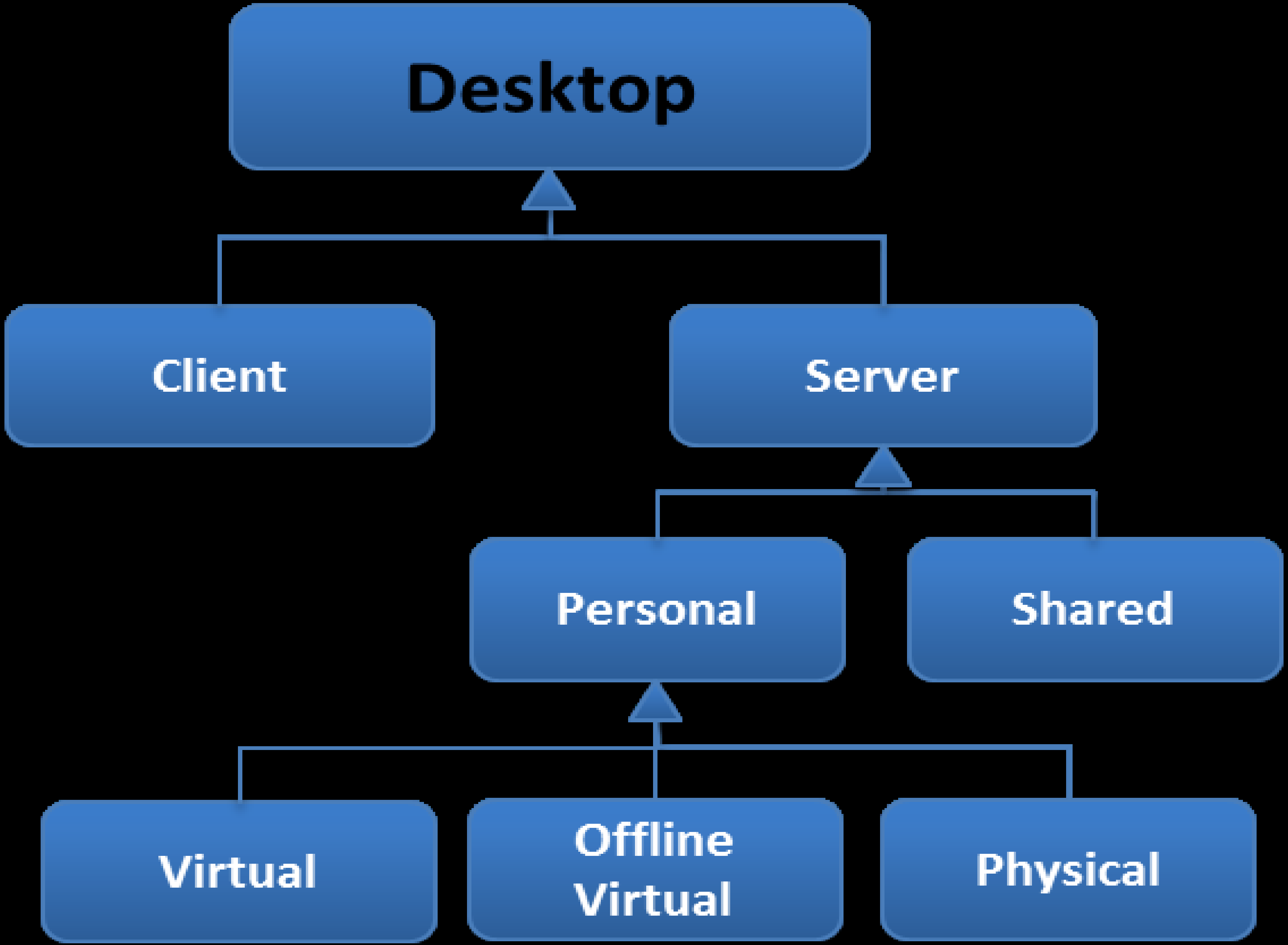


Layers of Desktop Computing

Desktop Virtualization

sktop Virtualization (DeskV)

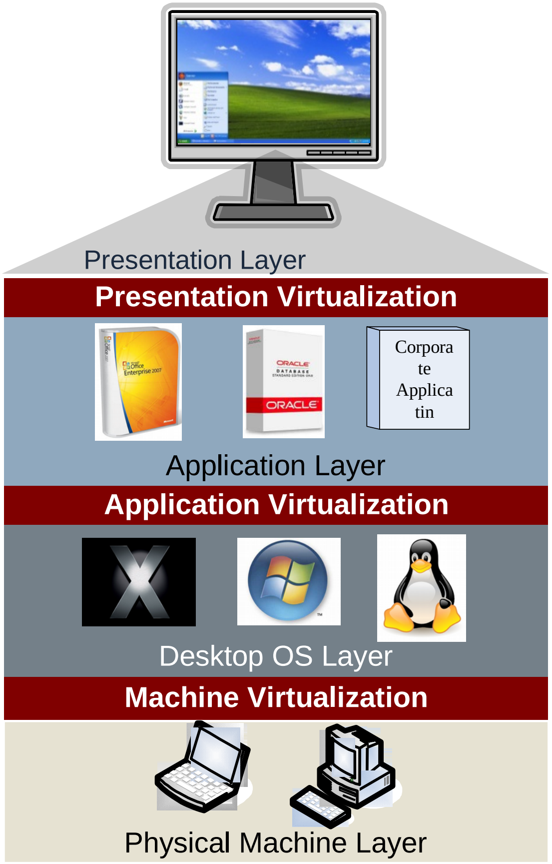
- Desktop Virtualization (DeskV) allows you to rely on virtual machines to **provision desktop systems**.
- Desktop virtualization has several advantages, the least of which is the **ability to centralize desktop deployments and reduce distributed management costs** because users access centralized desktops through a variety of thin or unmanaged devices.
  - **Client** desktop virtualization technologies are used to host virtual desktops (or virtual machines) locally on the clients’ computer.
  - Server desktop virtualization, two types, personal


Desktop
Client
Server
Personal
Shared
Virtual
Offline Virtual
Physical

Different abstractions for different kinds of virtualization

Presentation Virtualization

- In traditional remote access of server based applications, (Citrix Presentation Server, Microsoft Terminal Services) only presentation layer virtualization is used.
- Presentation

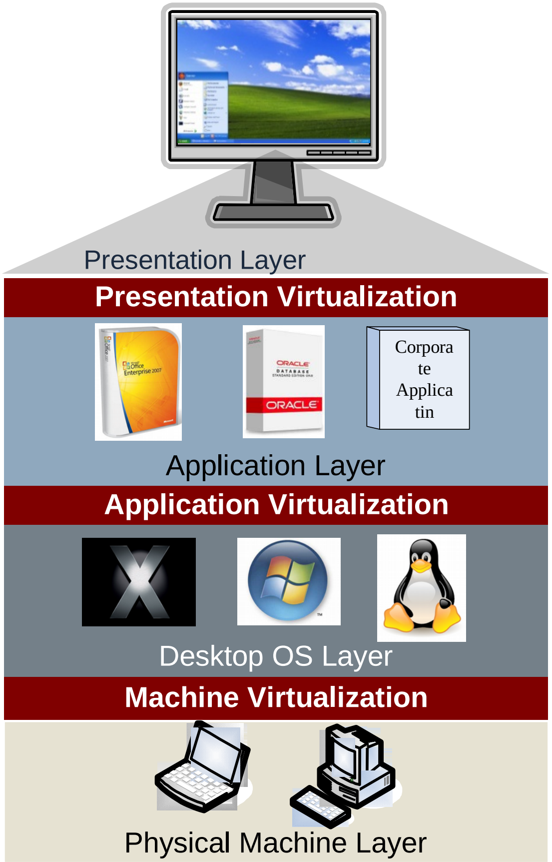


Abstraction Layers in Desktop Computing

Different abstractions for different kinds of virtualization

Application Virtualization

- In application virtualization, **an application can be downloaded and run locally** without special configuration of either the application or the client PC or its OS. The application interacts with an abstraction layer between it and the OS and machine layers.
- Examples of application virtualization solutions include VMware View (ThinView), Citrix

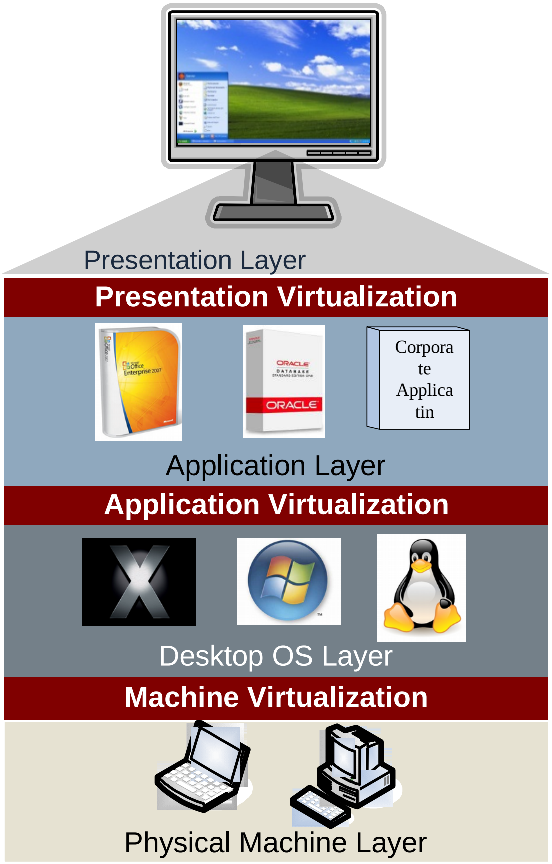


Abstraction Layers in Desktop Computing

Different abstractions for different kinds of virtualization

Desktop Virtualization

- Desktop Virtualization uses both a **Machine Virtualization** layer for hosting PC VMs on a server and **presentation virtualization** for remote access of those virtual machines from the desktop.



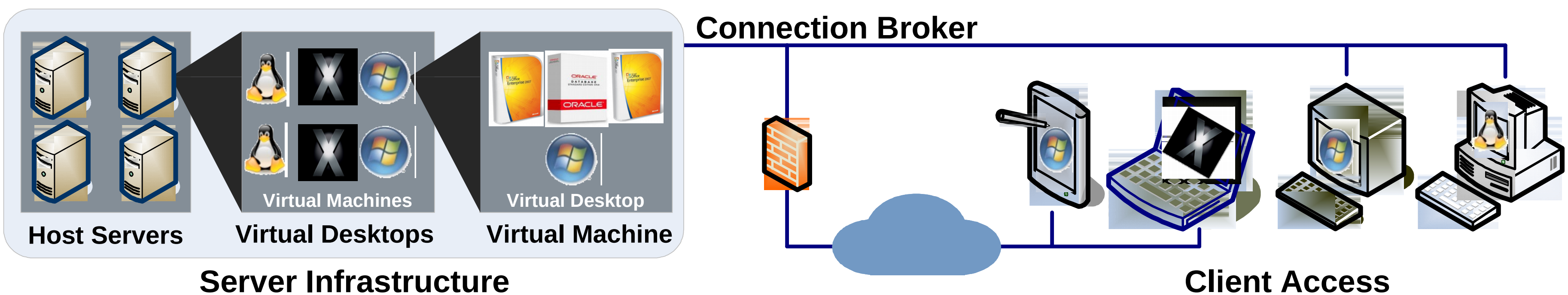
Abstraction Layers in Desktop Computing

Virtualization layers are not mutually exclusive

- The different layers of virtualization **can be combined** for different desktop computing solutions. For example:
  - Desktop virtualization solutions combine presentation virtualization and PC machine virtualization on a hosting server.
  - Application virtualization and application presentation can also be combined with desktop virtualization – **a virtual desktop machine can be dynamically provisioned with remote access and/or virtualized applications.**
  - Application virtualization can also be applied

Desktop virtualization moves desktop processing and storage off the desktop

- In desktop virtualization, a full instance of a desktop operating system (typically Windows) runs on a virtual machine that is hosted on a server and remote accessed from a client device (a PC, laptop, or thin client).
- For the end user, the experience of the virtual desktop should be the same as using a local PC. For example, they will see and interact with Windows and their familiar applications. They may not even be aware that the applications are no longer executing on a local CPU and OS instance.
- Access is managed through connection broker software. The broker connects the user with a specific virtual desktop. The most popular software for managing both virtual desktops and brokering are VMware View and Citrix XenDesktop.



Desktop virtualization works for a broad range of use cases

Using Virtual Desktop Solutions

Broad range of job types:

Knowledge workers (executive staff, analysts)
Process workers (customer service, claims and loan processing)
Data-entry workers (processing reservations, making order entries)

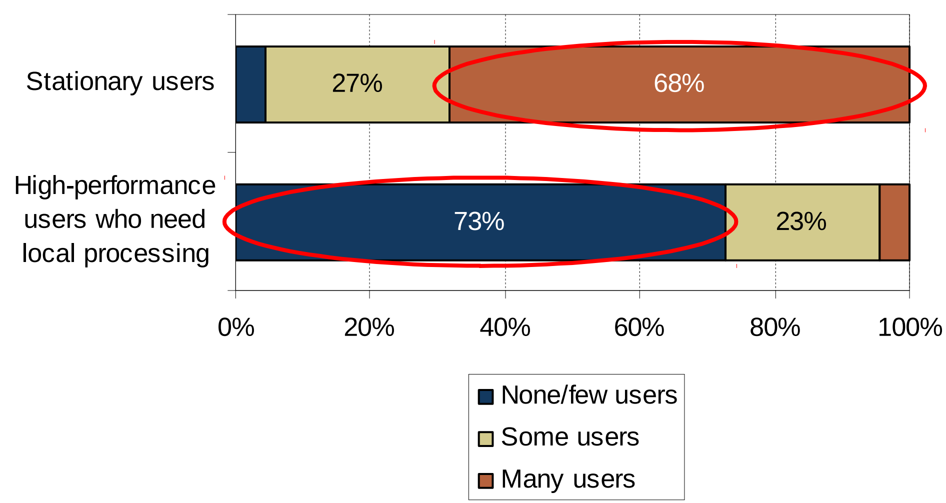
Wide range of connection types:

Stationary users (connecting using permanent LAN links)
Roaming users (on multiple permanent LANs)
Remote/Home users (use the same link/device to connect from multiple locations)

Not Using Virtual Desktop Solutions

High-performance users who need local graphics processing (video / graphic designers)
Mobile users (significant offline use while traveling, wide variety of connections – LAN, WAN, dial-up, public WiFi)

Deployment by User Type

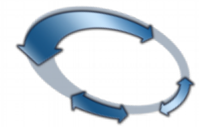


The Benefits of Desktop Virtualization

Centralized Management

- Manage physical and virtual desktops from a single console
- Centralized desktop lifecycle management

Anywhere Access for Connected Devices

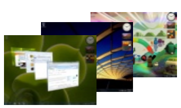


- Access desktops from any connected device
- Enable rich desktop experiences on thin clients and older PCs

Enhance Security and Compliance

- Data always locked in the data center
- Improved compliance through centralization

Increased Business Continuity



- Data center grade business continuity for the desktop
- Quicker resolution of desktop failures

There are limitations to the current model

- Desktop virtualization currently falls short of the ubiquitous desktop vision particularly in the area of **mobility of the virtual machine image**. In VDI models, VMs tend to be hosted on central servers and remote accessed from the desktop.
- Desktop virtualization has many of the same advantages and limitations of traditional presentation virtualization. For example, central management of applications is a benefit of both, while **latency of remote accessed applications and susceptibility to network outage is a limitation of both.**
- Moving an application that has **network latency issues** from terminal services to desktop virtualization will not improve the experience of that application.
- Desktop virtualization solutions also typically **require more storage and processing on the back end** than traditional terminal services solutions.
- Operating system/software licensing costs are also proving to be a challenge for adopters

How Persistent Virtual Desktops Will Save Money

In a large enterprise, PCs are purchased in bulk but not always from the same vendor. This makes it necessary to maintain multiple configurations of software and operating systems (one for the Dells, one for the HPs, one for the white boxes, etc.)

With fully virtualized desktops, the apps and OS are configured for one virtual machine. PCs are either VM hosts or remote access terminals. Differences in PC configurations are rendered irrelevant.

Presentation Virtualization

Presentation Virtualization

- With presentation virtualization, an application actually runs on another host and all that you see on the client is the screen from where it is run.

- Presentation Virtualization -- This is what Citrix Met frame (and the ICA protocol) as well as Microsoft Terminal Services (and RDP) are able to create.

Presentation Virtualization

- Presentation virtualization **consolidates** dozens or hundreds of users on a single server with one or more of the applications needed by those users installed to that server.

Reasons for Presentation Virtualization

- **Applications are installed only once**. In presentation virtualization, multiple users share the same application instance. So if you need to update an application, you can perform the update in one place and all users benefit immediately.
- **User resource needs are light**. The resource needs for a particular user in a presentation virtualization environment **are limited** to hosting a user's shell and what is needed to run a user's applications. Some platforms even provide mechanisms **to share these resources** among multiple users in order to increase the user density on a single server.

Reasons for Presentation Virtualization

- **Administration occurs at the individual server level**. Since multiple users share the resources of a single system, server administration is performed at the server level.
- **Presentation virtualization technologies are mature**. For more than 10 years, presentation virtualization technologies have been in the marketplace.
- Platforms like **Terminal Services** and **XenApp** are mature, stable, well known and easy to understand by technology generalists.

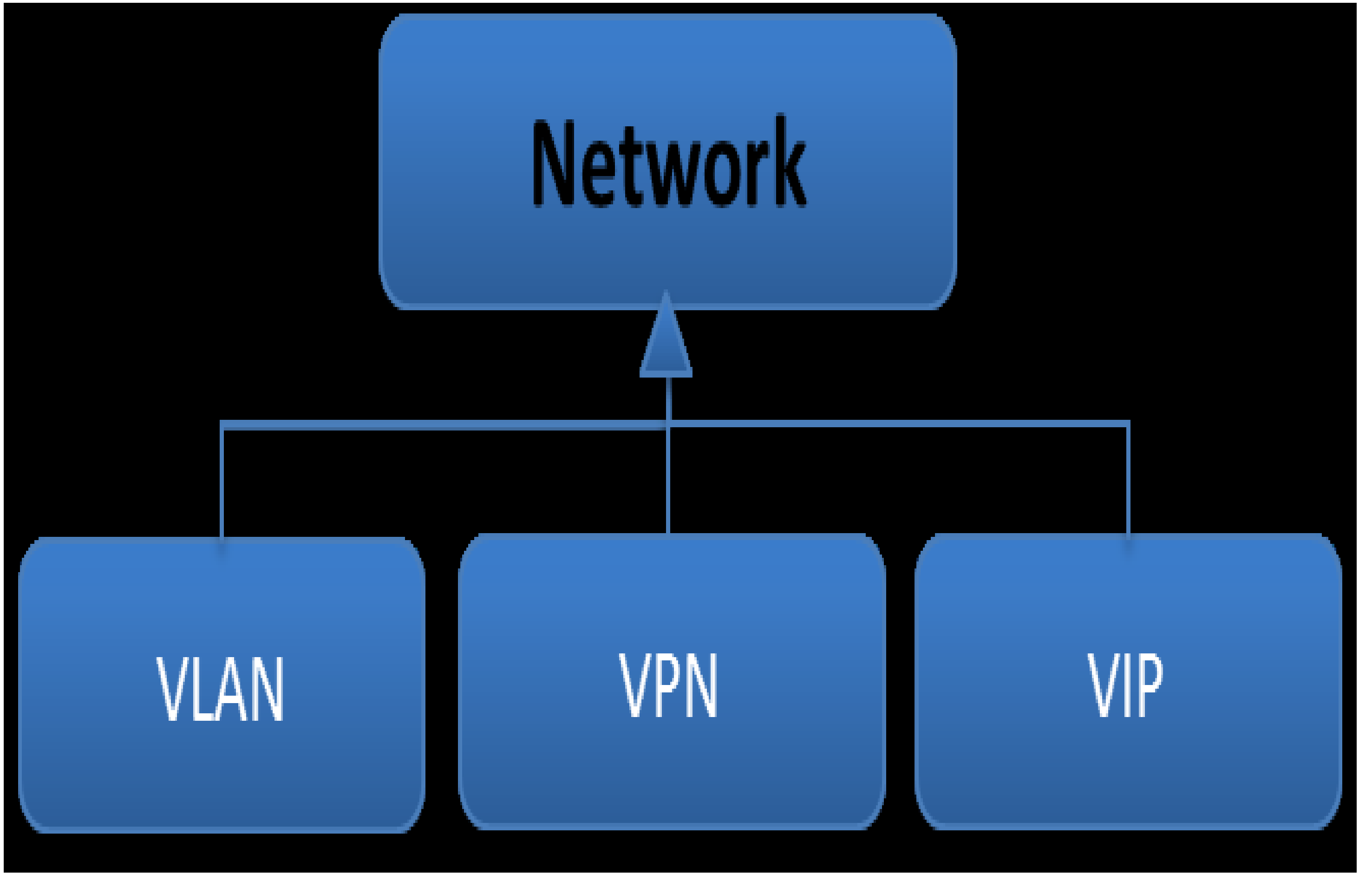
I/O Virtualization / Network Virtualization

Network or I/O Virtualization

twork Virtualization (NetV)

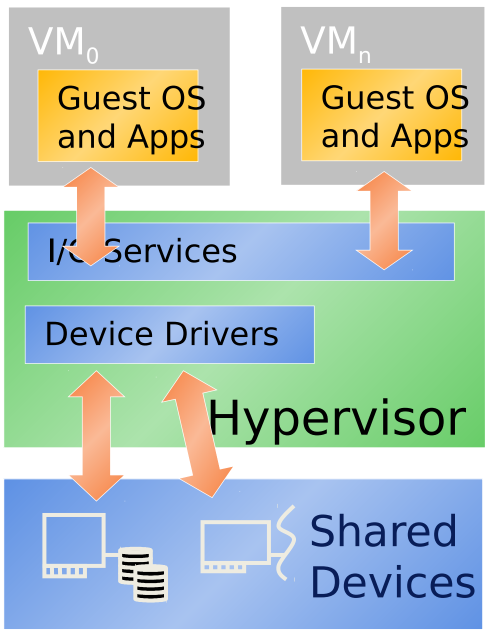
- Network Virtualization (NetV) lets you **control available bandwidth by splitting it into independent channels** that can be assigned to specific resources.
- For example, the simplest form of network virtualization is the **virtual local area network (VLAN),** which creates a logical segregation of a physical network.
- In addition, server virtualization products support the creation of virtual network layers within the product itself.
    - using this **virtual network layer** would let you place a perimeter network on the same host as

- Types of NetV
  - Virtual LAN (VLAN),
  - Virtual IP (VIP) and
  - Virtual Private



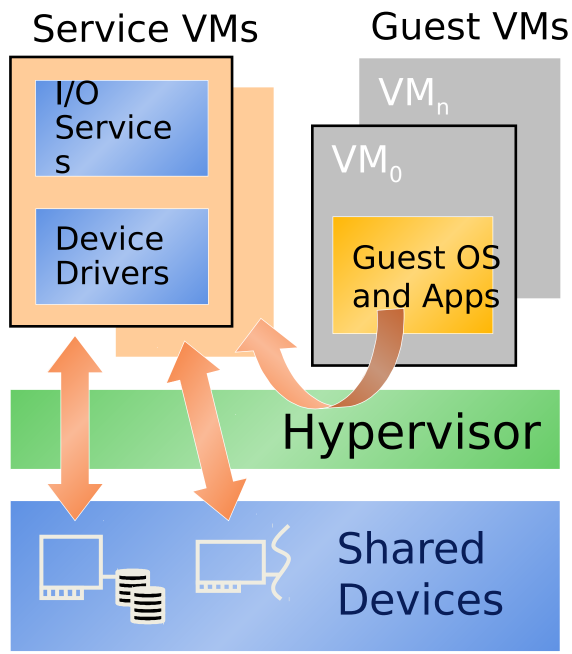
Virtualized I/O Models

Monolithic Model



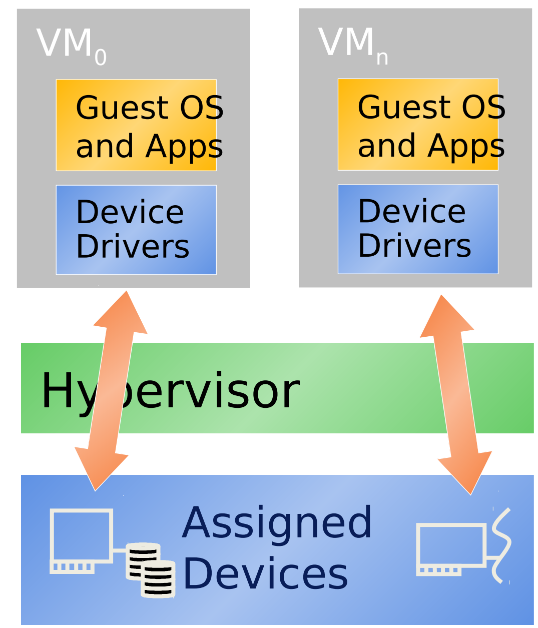
- Pro: Higher Performance
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Larger Hypervisor

Service VM Model



- Pro: High Security
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Lower Performance

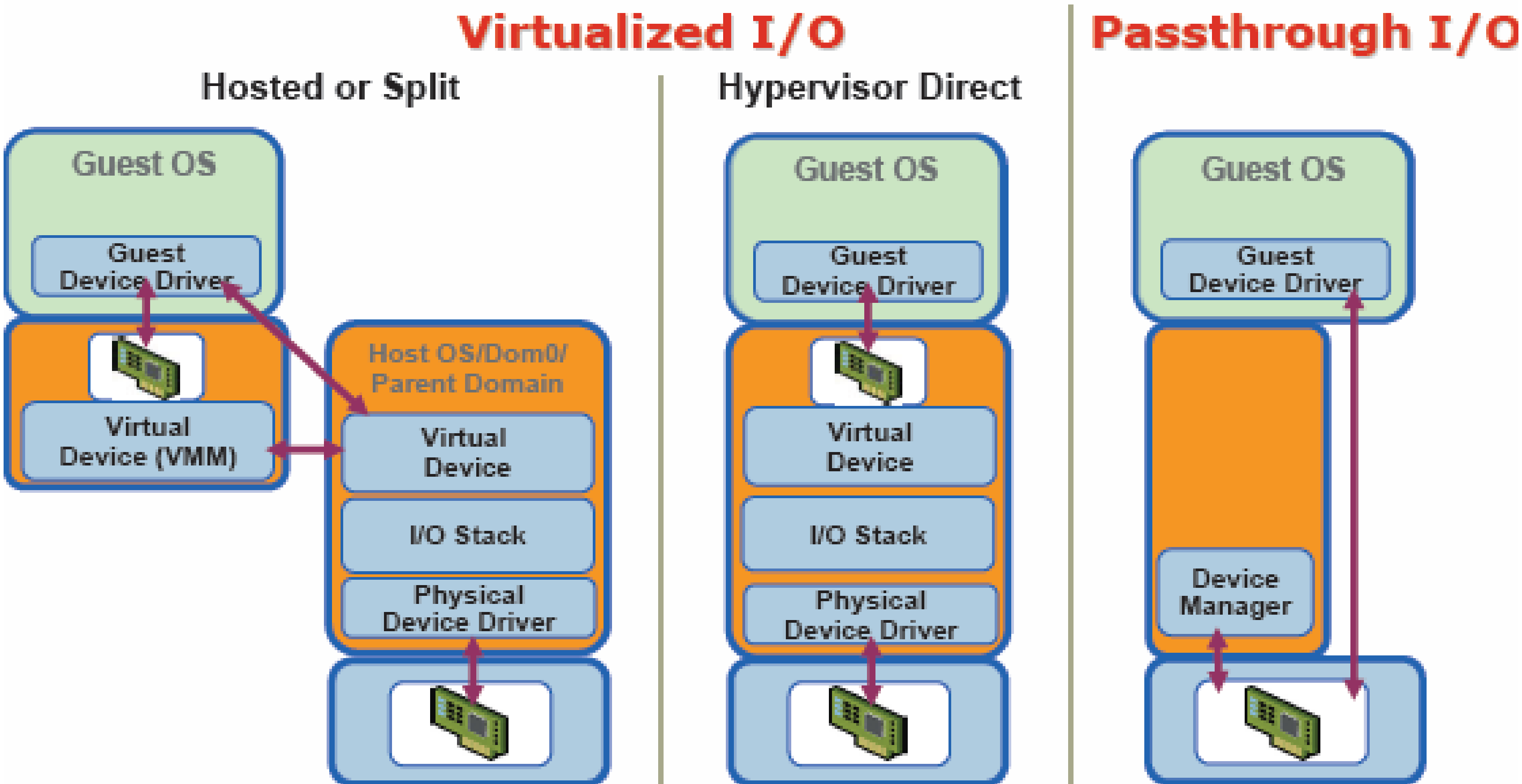
Pass-through Model



- Pro: Highest Performance
- Pro: Smaller Hypervisor
- Pro: Device assisted sharing
- Con: Migration Challenges

VT-d Goal: Support all Models

I/O Virtualization Implementations

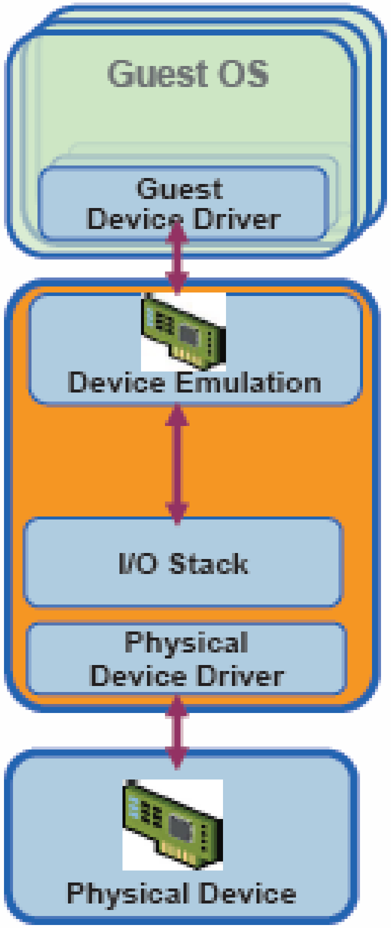


VMware Workstation, VMware Server,
VMware ESX Server
Microsoft Viridian & Virtual Server, Xen

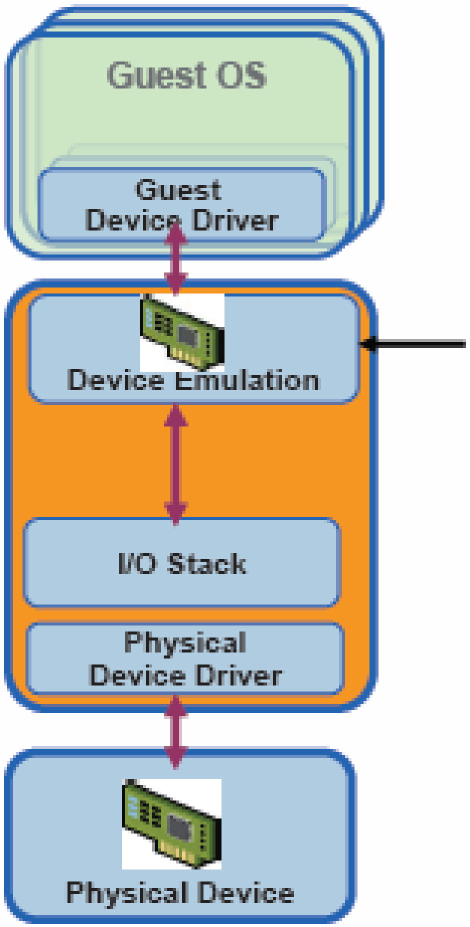
VMware ESX Server
(storage and network)

A Future Option

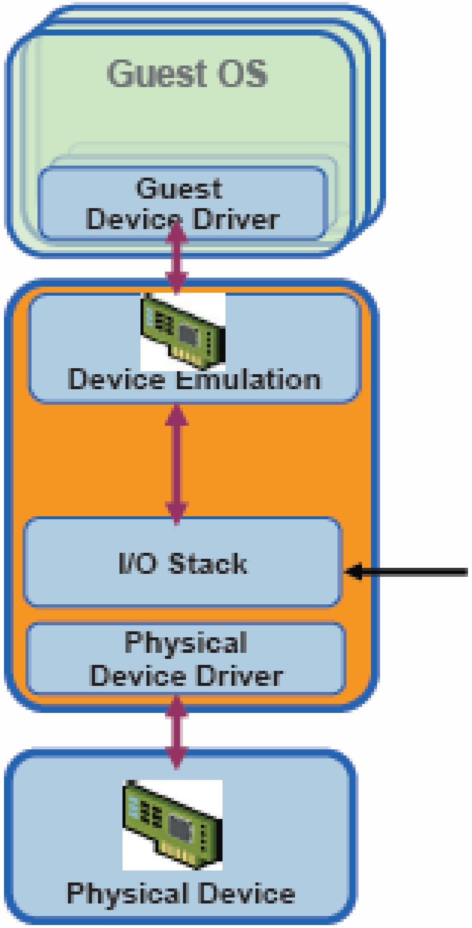
I/O Virtualization Architecture



- **I/O Virtualization architecture consists of**
  - Guest driver
  - Virtual device
  - Communication mechanism between virtual device and virtualization stack
  - Virtualization I/O stack
  - Physical device driver
  - Real device

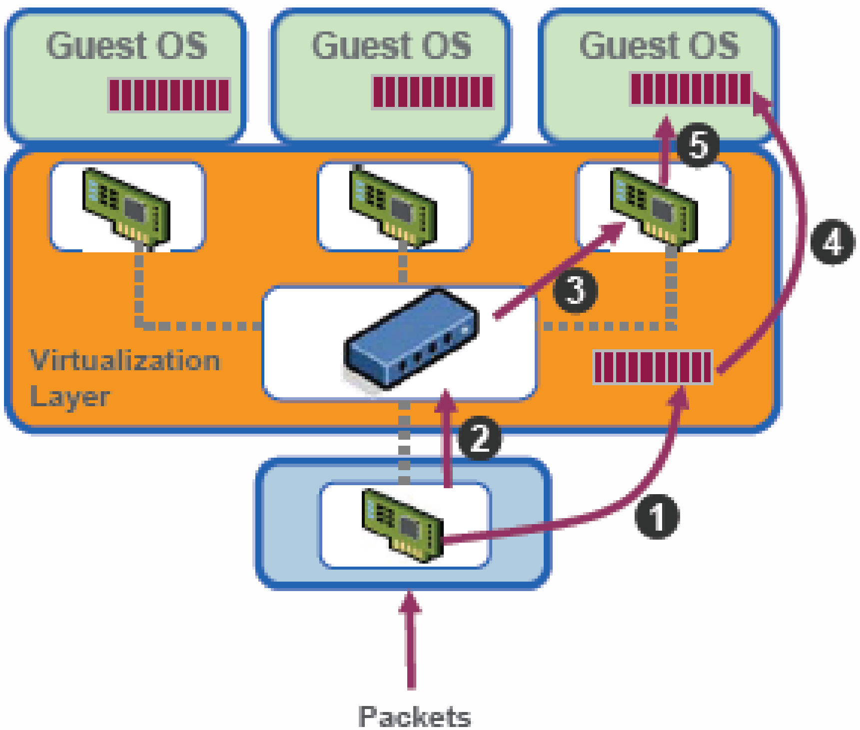


- **Virtual device**
  - model a real device
    - e.g., Intel e1000, LSI mptscsi
  - model a simple virtualization friendly device
    - e.g., VMware vmxnet



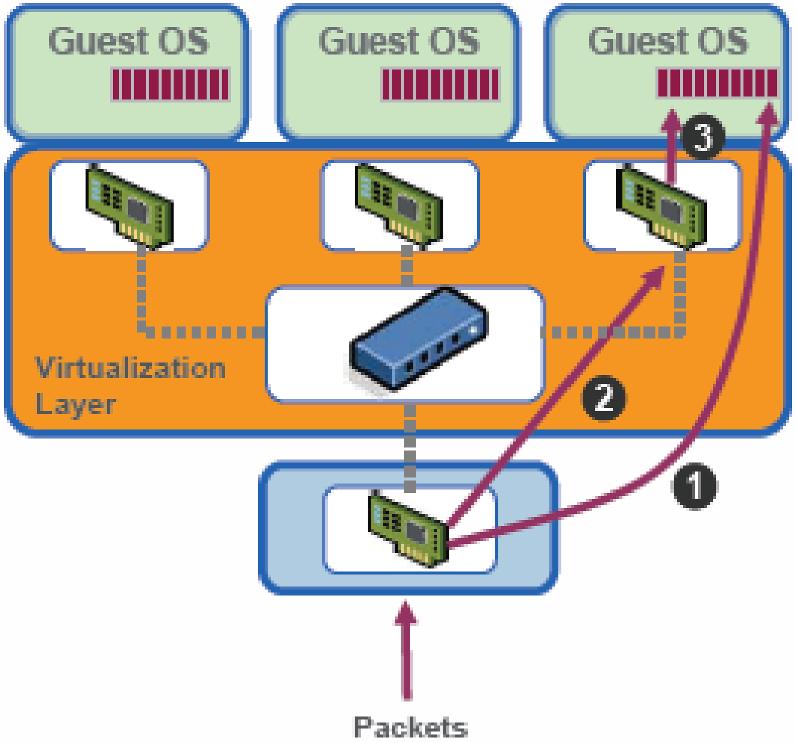
- **Virtualization I/O stack**
  - translates guest I/O addresses to host addresses
  - handles inter VM communication
  - multiplexes I/O requests from/to the physical device
  - provides enterprise-class I/O features

Packet Receive in Virtualized I/O

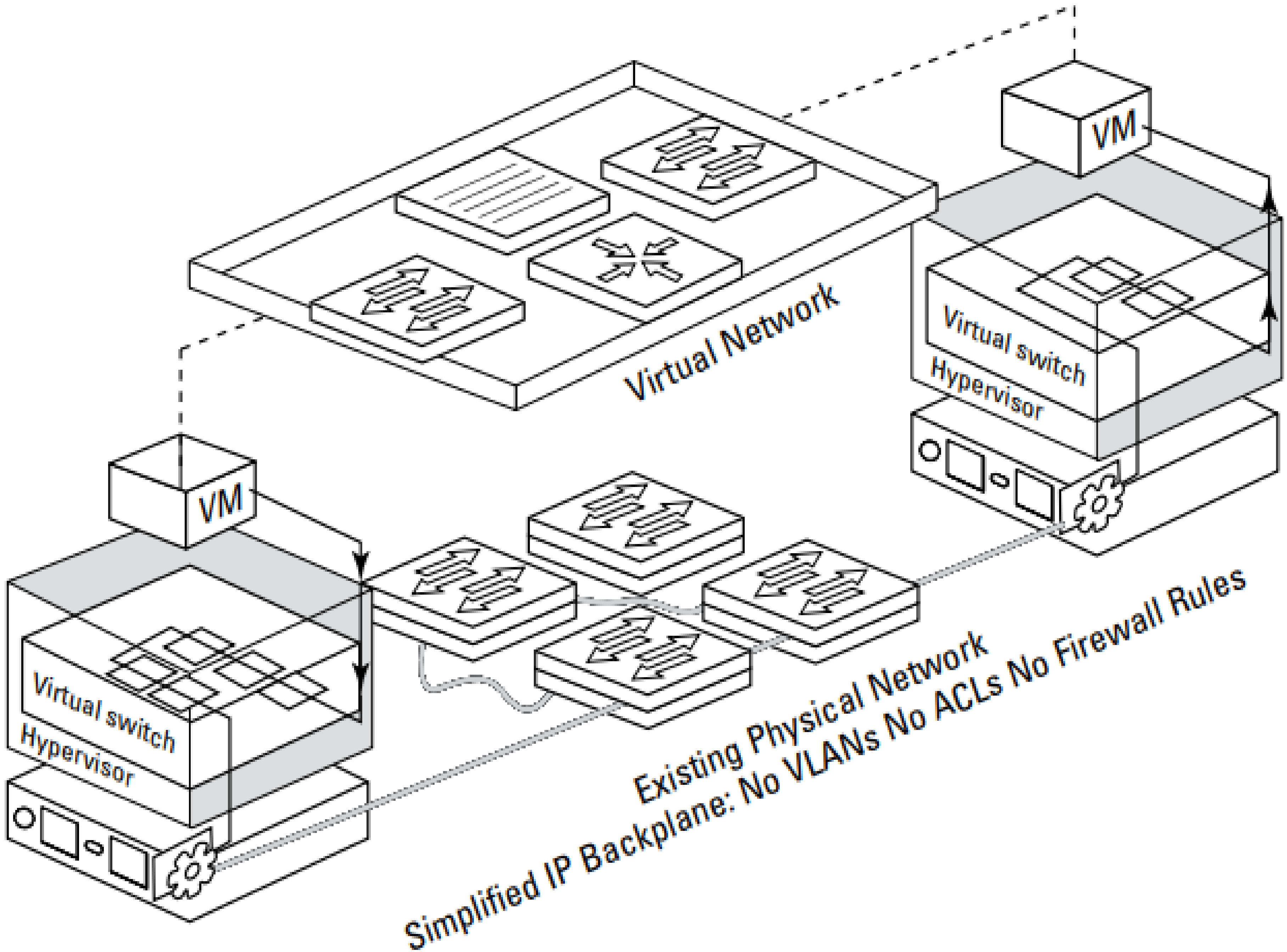


1. DMA packet into receive buffers owned by VMkernel
2. Raise physical Interrupt
3. Parse packet to find destination VM
4. Copy packet into guest receive queue
5. Raise virtual NIC interrupt

Packet Receive in Pass through I/O

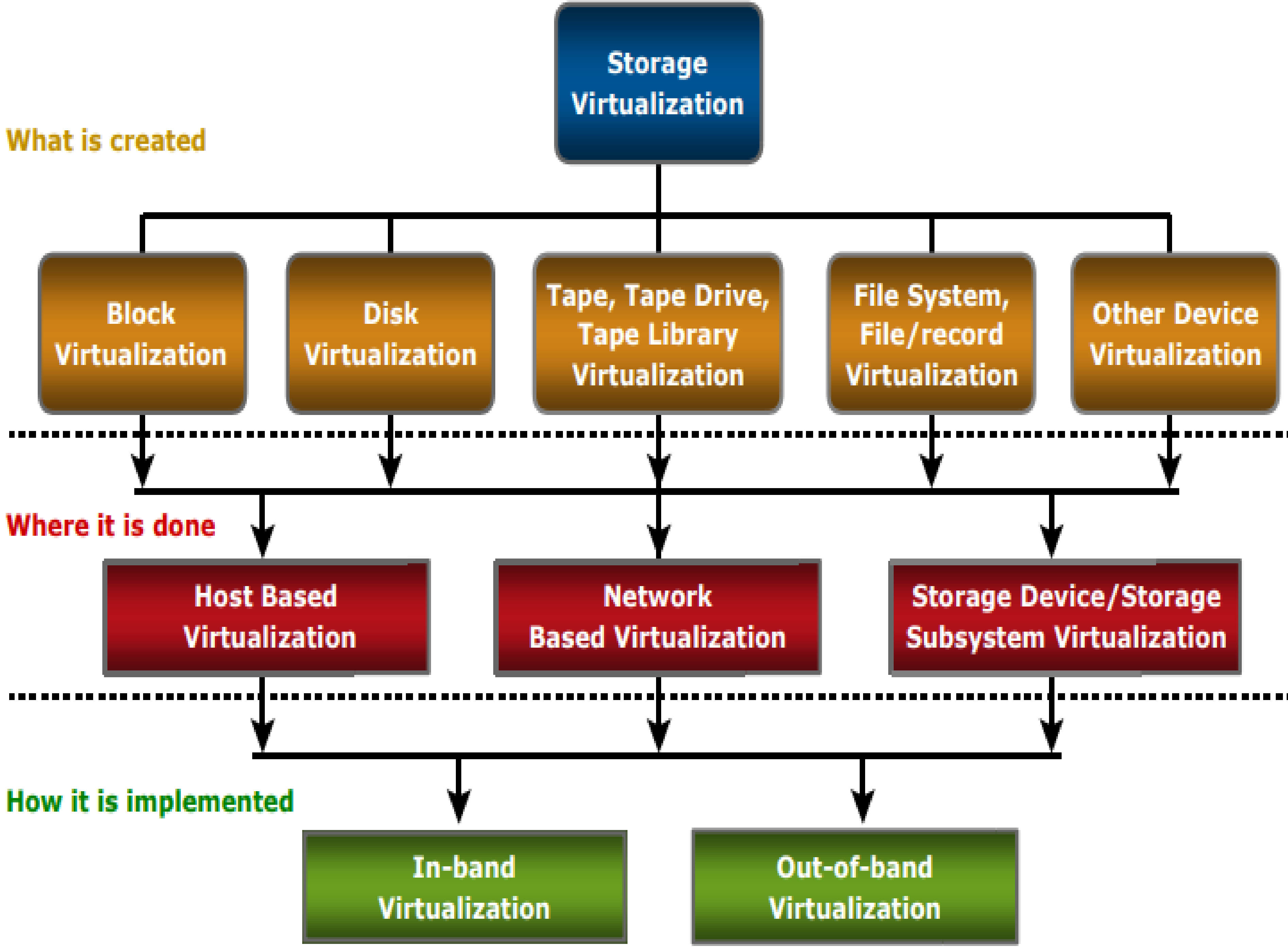


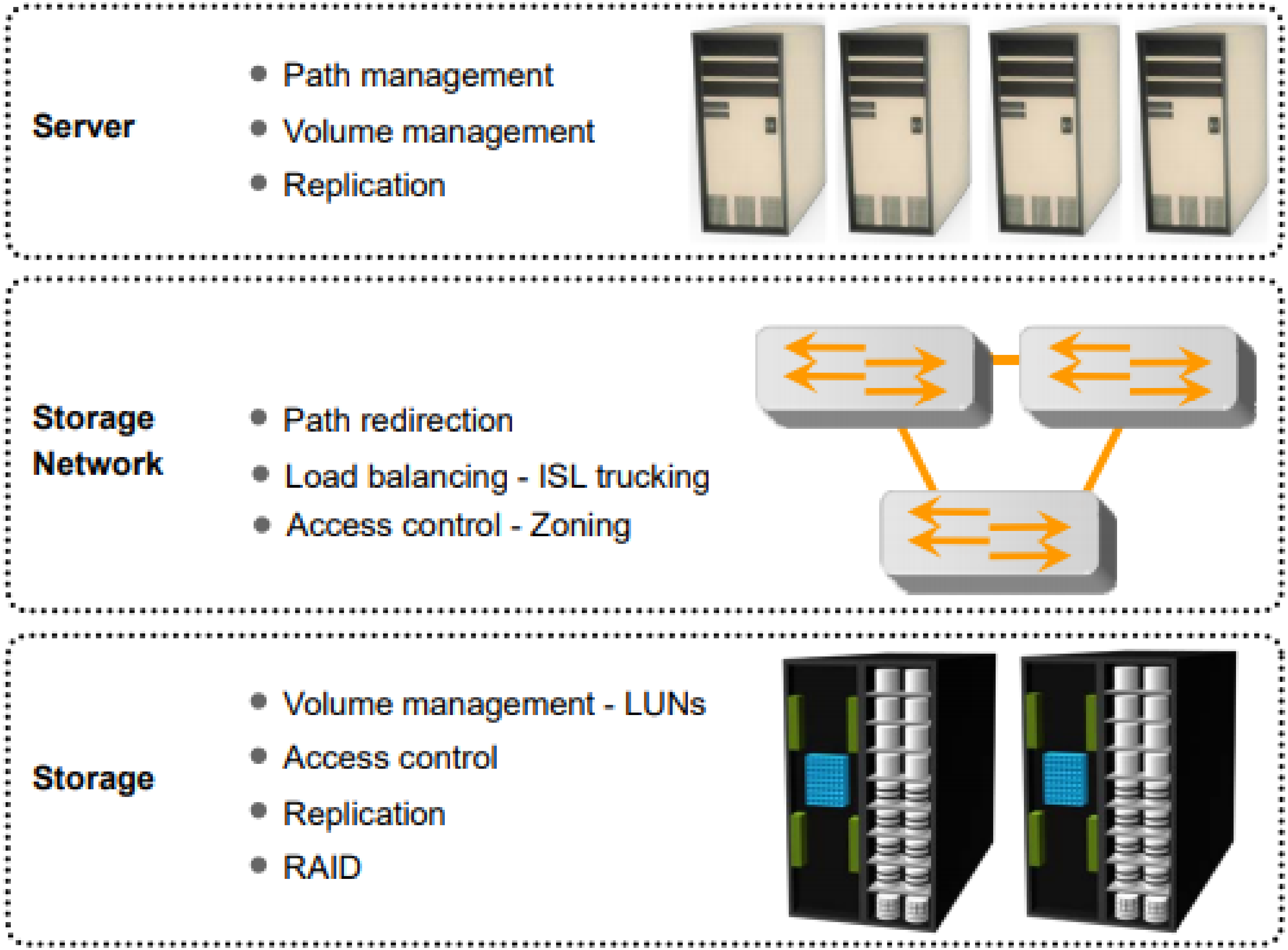
1. DMA packet into guest receive queue
2. Raise NIC interrupt at guest CPU
3. Raise virtual NIC interrupt

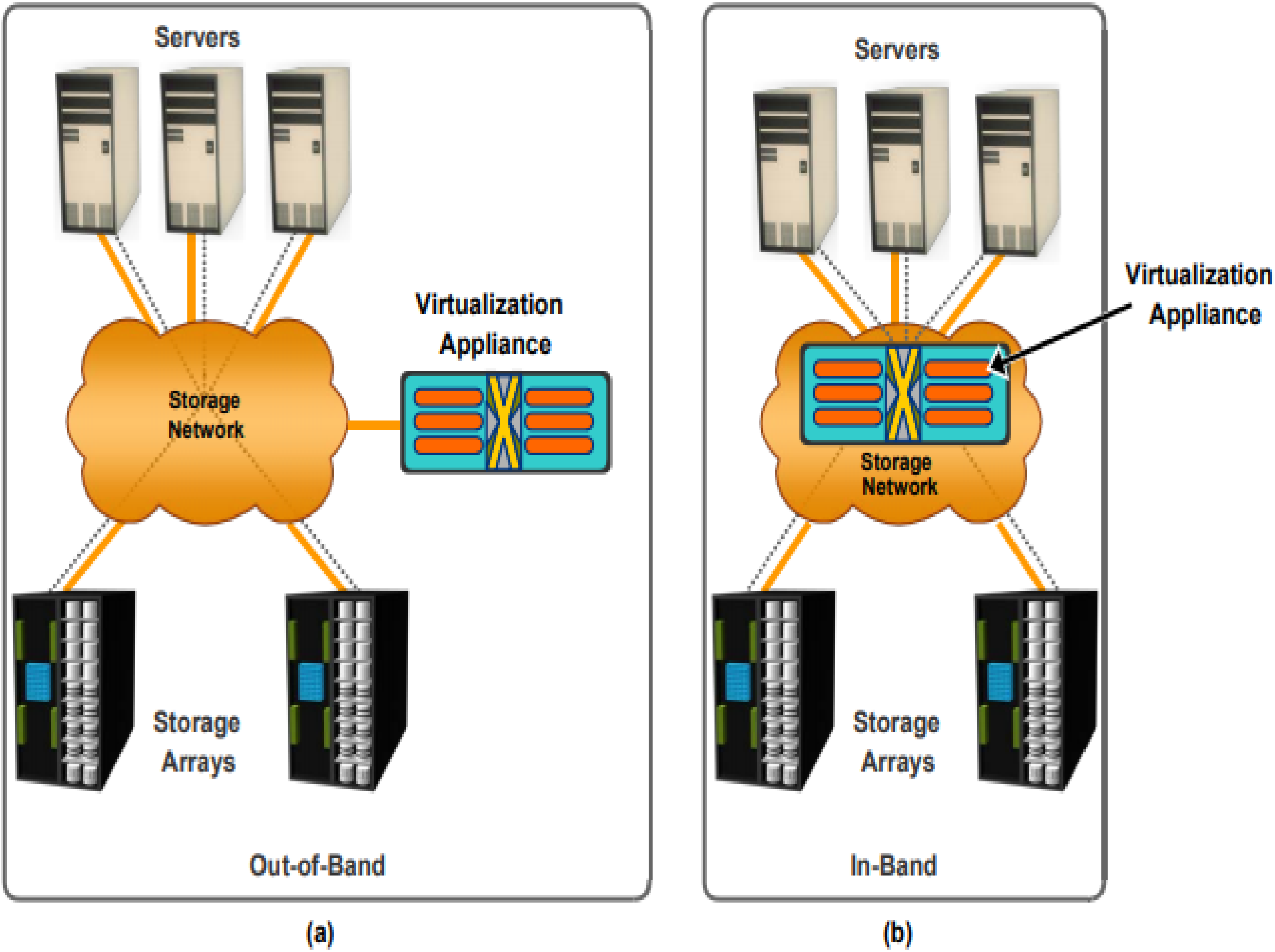


Virtual Network
VM
Virtual switch
Hypervisor
VM
Virtual switch
Hypervisor
Existing Physical Network
Simplified IP Backplane: No VLANs No ACLs No Firewall Rules

Storage Virtualization


Storage Virtualization
What is created
Block Virtualization
Disk Virtualization
Tape, Tape Drive, Tape Library Virtualization
File System, File/record Virtualization
Other Device Virtualization
Where it is done
Host Based Virtualization
Network Based Virtualization
Storage Device/Storage Subsystem Virtualization
How it is implemented
In-band Virtualization
Out-of-band Virtualization


Server
Path management
Volume management
Replication
Storage Network
Path redirection
Load balancing - ISL trucking
Access control - Zoning
Storage
Volume management - LUNs
Access control
Replication
RAID



(a) In out-of-band implementation, the virtualized environment configuration is stored external to the data path

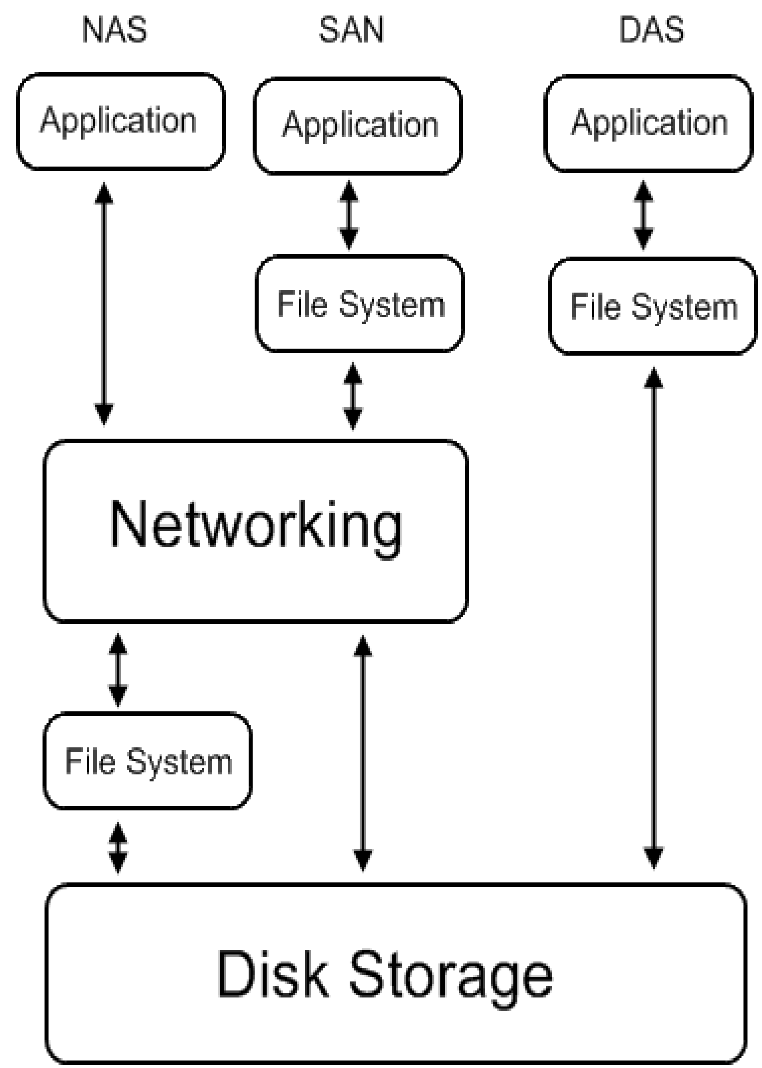
(b) The in-band implementation places the virtualization function in the data path

- Introduction
- What to be virtualized
- Where to be virtualized
- How to be virtualized
- Case study

STORAGE VIRTUALIZATION

Introduction

- Common storage architecture :
  - DAS - Direct Attached Storage
    - Storage device was directly attached to a server or workstation, without a storage network in between.
  - NAS - Network Attached Storage
    - File-level computer data storage connected to a computer network providing data access to heterogeneous clients.
  - SAN - Storage Area Network
    - Attach remote storage



Introduction

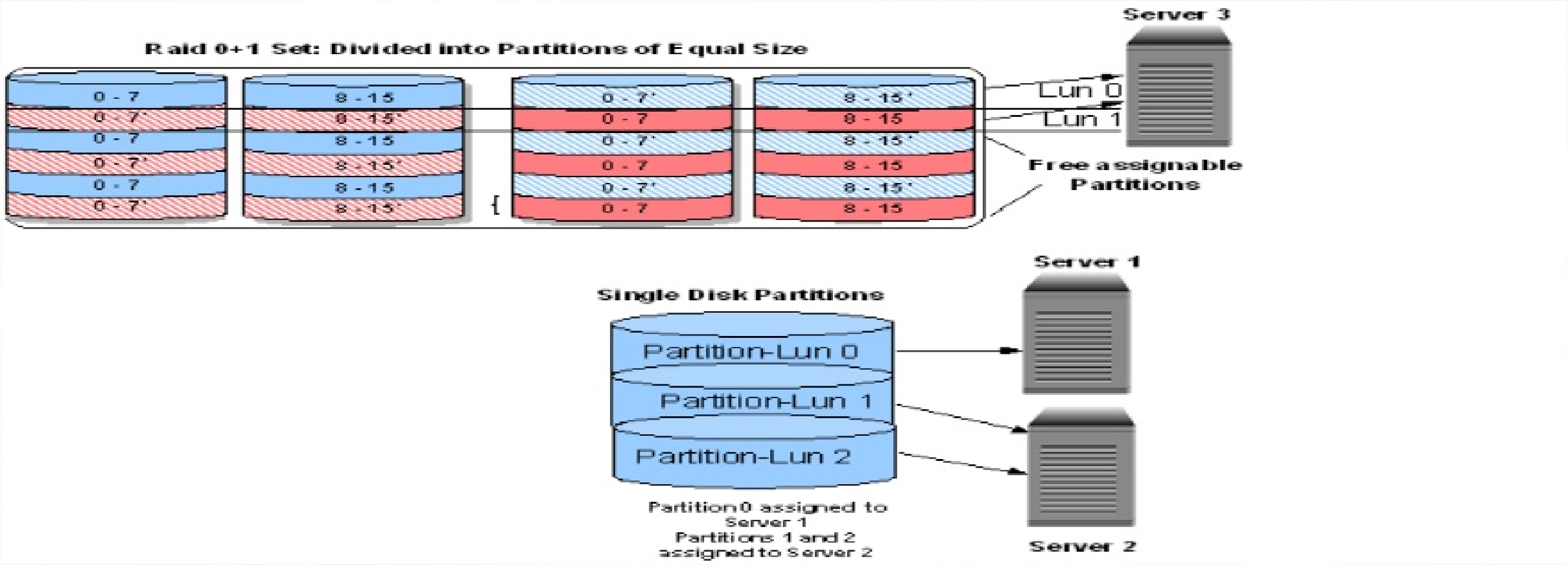
- Desirable properties of storage virtualization:
  - Manageability
    - Storage resource should be easily configured and deployed.
  - Availability
    - Storage hardware failures should not affect the application.
  - Scalability
    - Storage resource can easily scale up and down.
  - Security
    - Storage resource should be securely isolated.

Introduction

- Storage concept and technique
  - Storage resource mapping table
    - The virtualization software or device is responsible for maintaining a consistent view of all the mapping information for the virtualized storage. This mapping information is often called meta-data and is stored as a mapping table.
    - The address space may be limited by the capacity needed to maintain the mapping table. The level of granularity, and the total addressable space both directly impact the size of the meta-data, and hence the mapping table.
  - Redundant data
  - Multi-path
  - Data sharing
  - Tiering

LUN

- A LUN refers to the individual piece in the storage system that is being accessed. Each disk in an array, for example, has a LUN. Disk partitions may also be assigned a LUN



Storage Virtualization

rage Virtualization (StoreV)

- Storage Virtualization (StoreV) is used to **merge physical storage from multiple devices so that they appear as one single storage pool**.

- The storage in this pool can take several forms: direct attached storage **(DAS)**, network attached storage **(NAS)**, or storage area networks **(SANs)**.

- Though storage virtualization is not a requirement for server virtualization, one of the key strengths you will be able to obtain from storage virtualization is the ability to rely on thin provisioning or the assignation

rage Virtualization (StoreV)

For example, if you create a LUN of 100 gigabytes (GB) and you are only using 12GB, only 12GB of actual storage is provisioned. **This significantly reduces the cost of storage since**

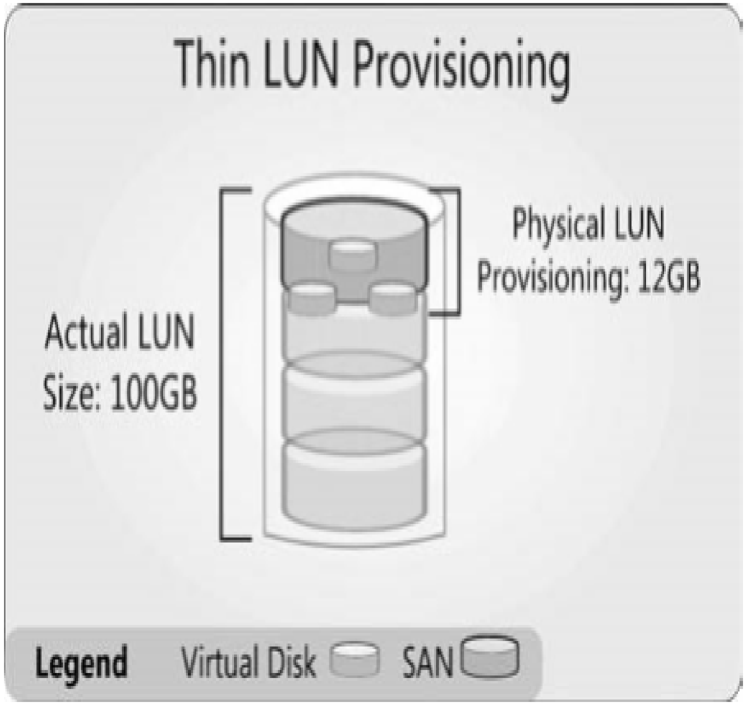
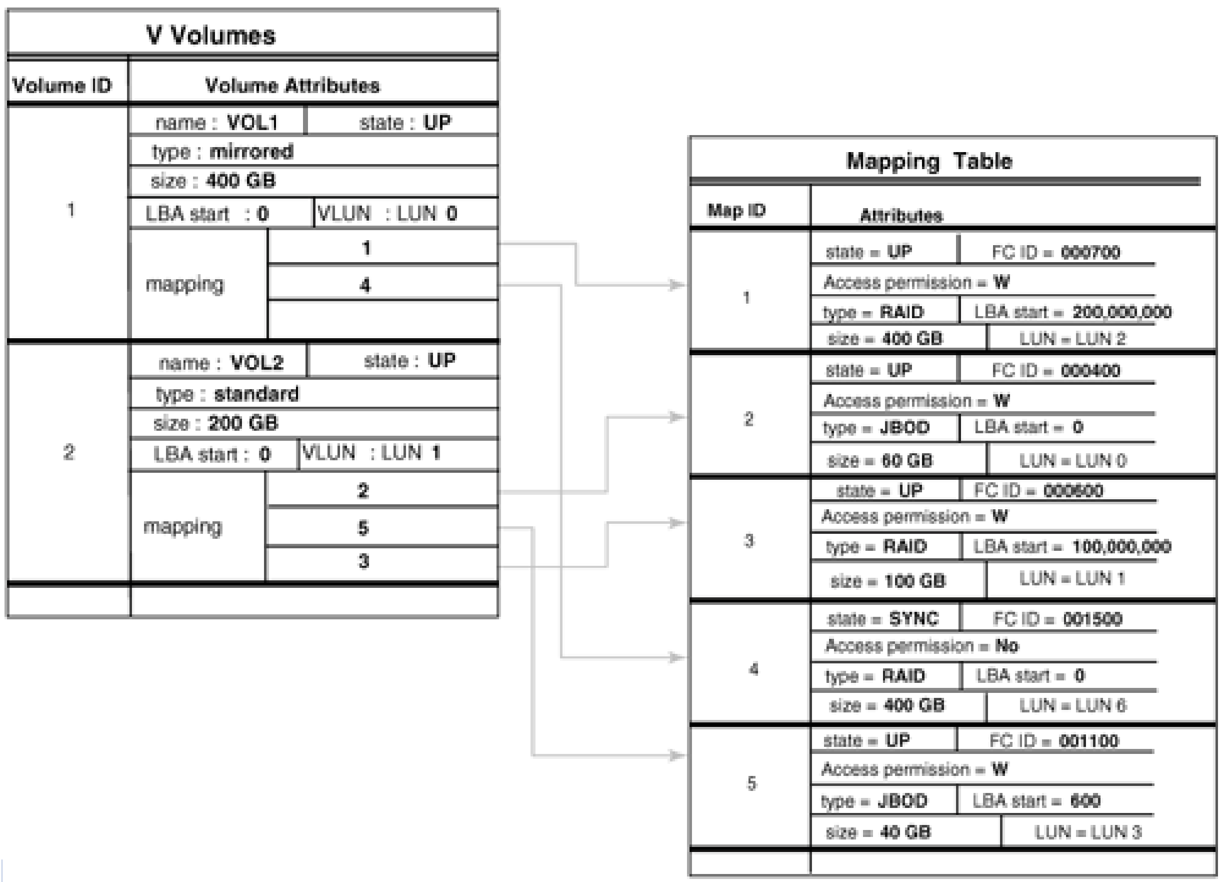


Figure 2-8. Using thin provisioning through Storage Virtualization can significantly reduce storage costs.

Concept and Technique

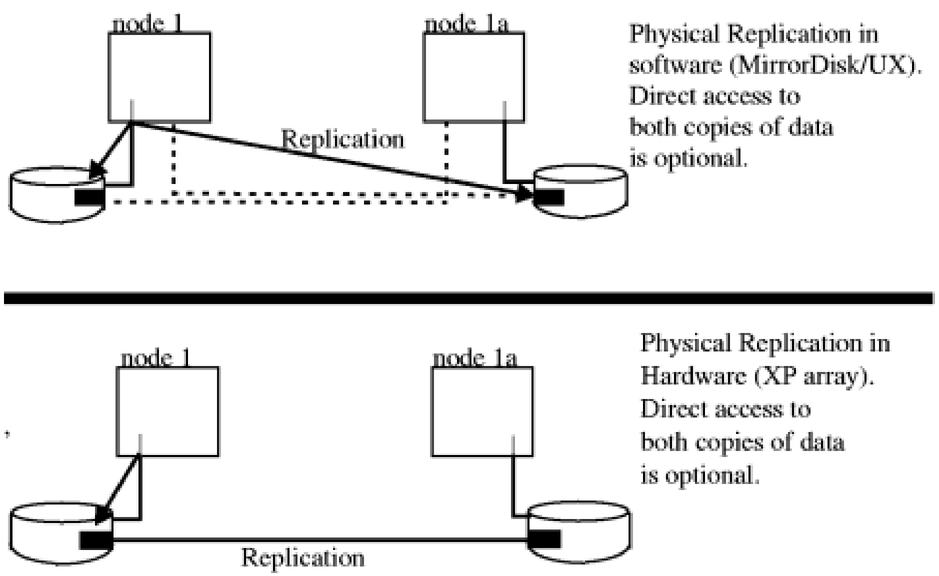
- Storage resource mapping table
  - Maintain tables to map storage resource to target.
  - Dynamic modify table entries for thin provisioning.
  - Use table to isolate different storage address space.

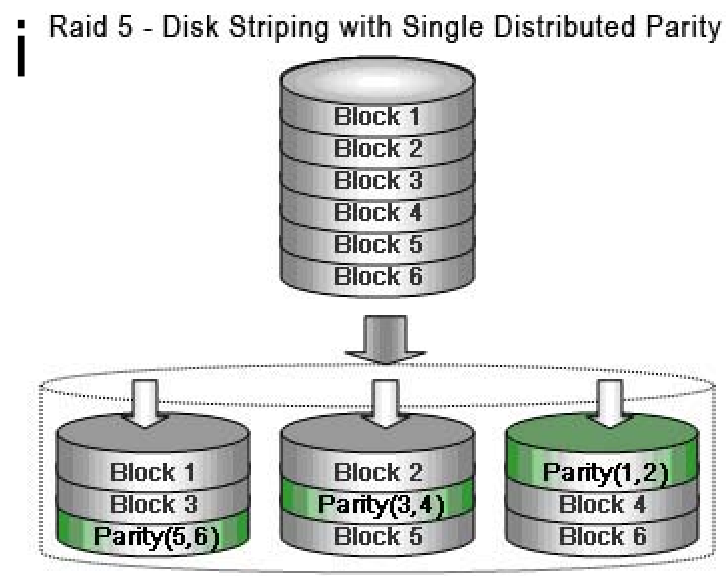
#LBA – Logical Block Addressing
#LUN – Logical Unit Number



Concept and Technique

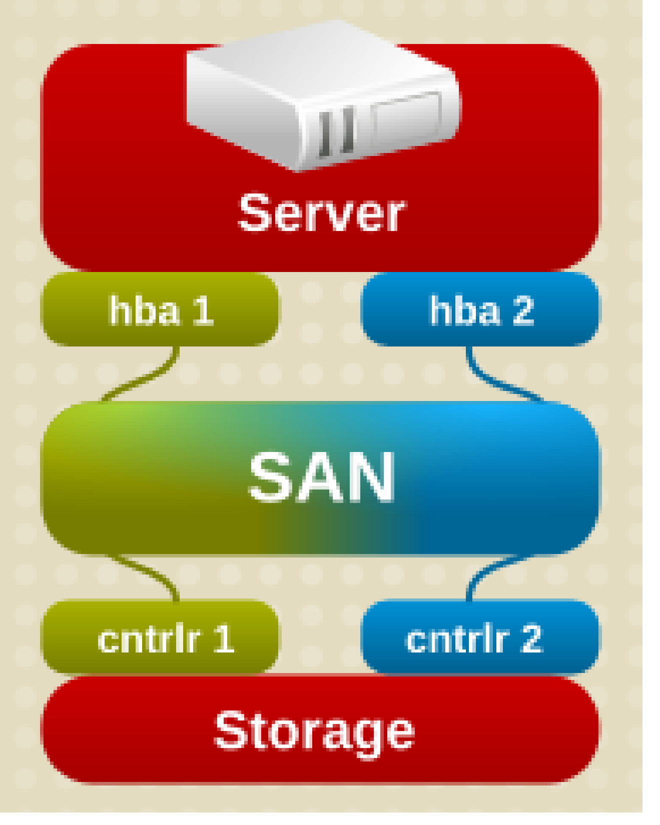
- Redundant data
  - Maintain replicas to provide high availability.
  - Use RAID technique to improve





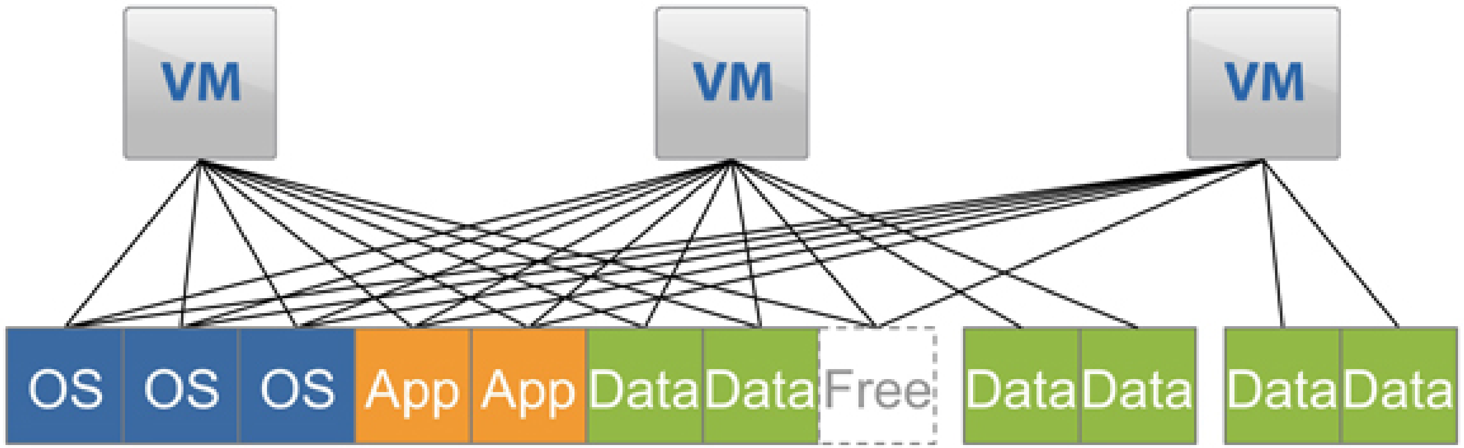
Concept and Technique

- Multi-path
  - A fault-tolerance and performance enhancement technique.
  - There is more than one physical path between the host and storage devices through the buses, controllers, switches, and bridge devices connecting them.



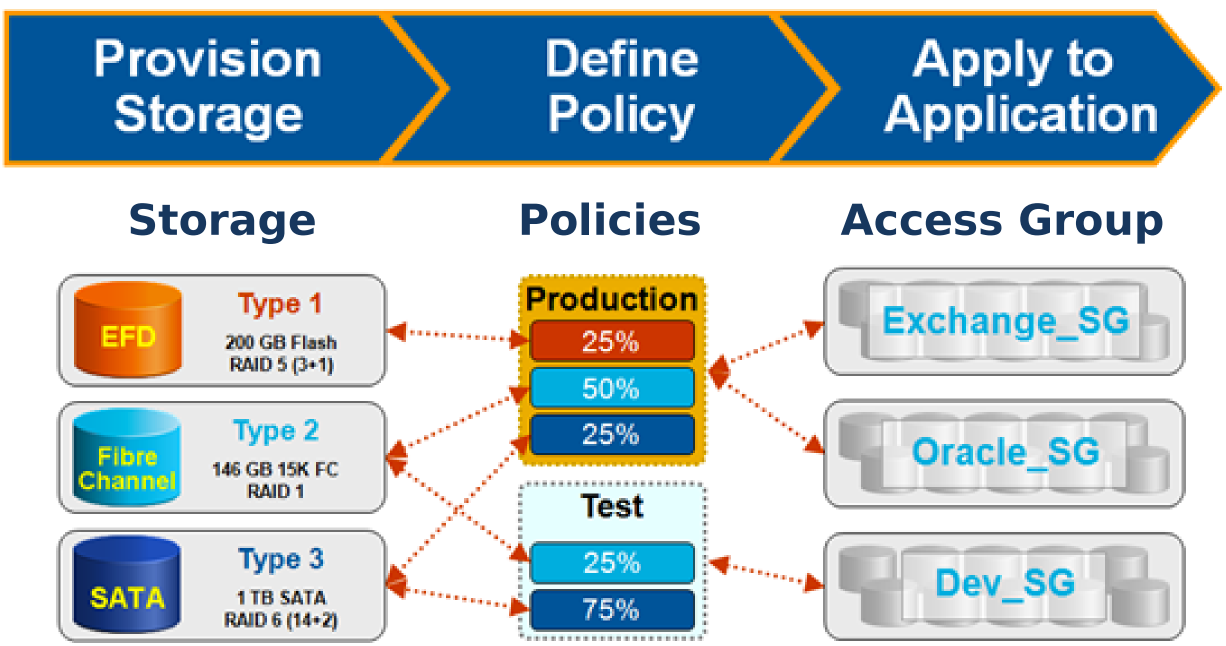
Concept and Technique

- Data sharing
  - Use data de-duplication technique to eliminate duplicated data.
  - Save and improve the usage of storage space



Concept and Technique

- Tiering
  - Automatic migrate data across storage resources with different properties according to the significance or access frequency of data.
  - Example: iMac fusion drive

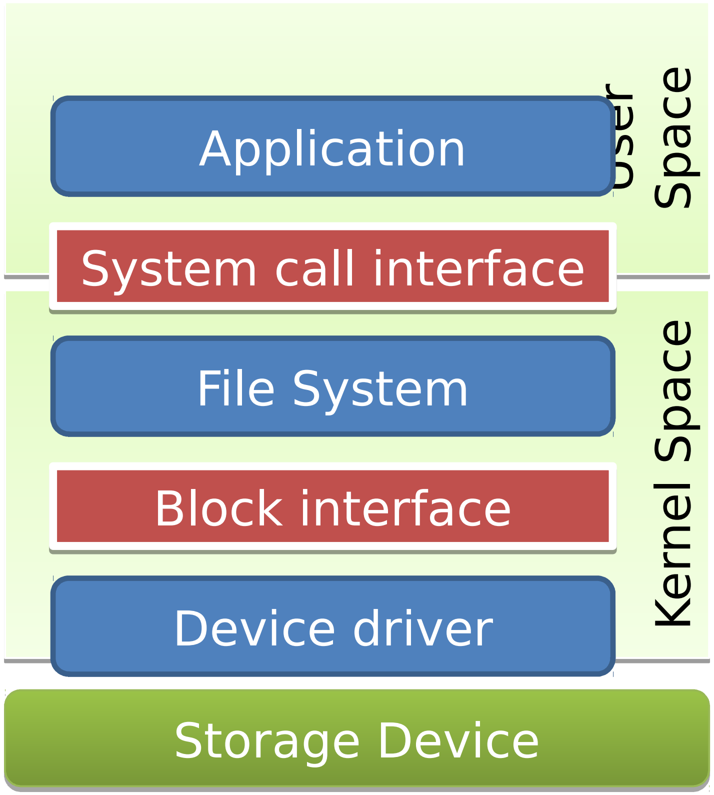


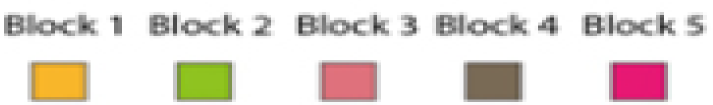
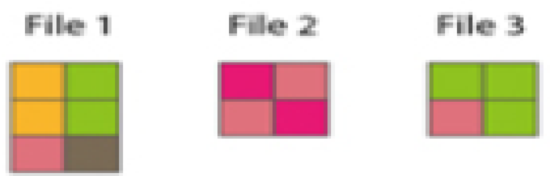
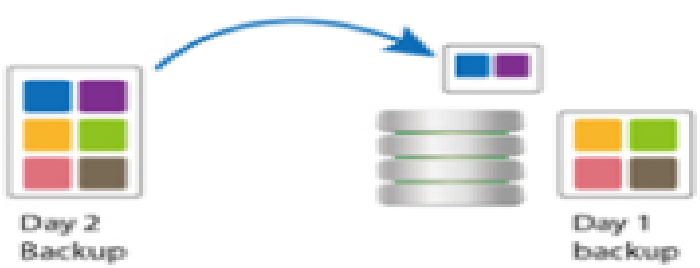
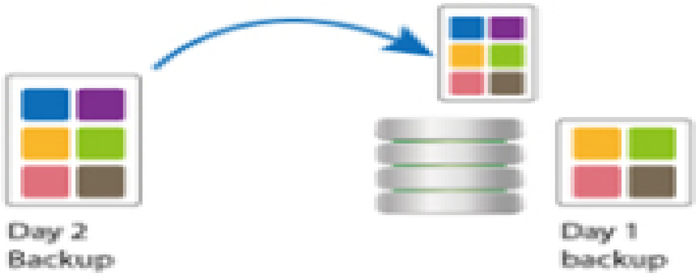
- Introduction
- What to be virtualized
- Where to be virtualized
- How to be virtualized
- Case study

STORAGE VIRTUALIZATION

What To Be Virtualized

- Layers can be virtualized
  - File system
    - Provide compatible system call interface to user space applications.
  - Block device
    - Provide compatible block device interface to file system.
    - Through the interface such as SCSI, SAS, ATA, SATA, etc.



	Block-Level	File-Level
Protocols	Fibre Channel/ iSCSI	Samba/ CIFS
Applications	*A LUN* can be *targeted as a physical drive.* It can be a ... *Logical Disk* *operation system* *file storage* *database* You can *format it* *install OS/applications on it* *boot-up of systems* *build file-system (NFS, NTFS, SMB, VMFS)*	*A shared folder* can be *mounted as a network drive.* It can be a ... *file storage* You can *set access priviledge* *integrate with corporate directories*
Pros	*Data transportation is much more efficient and reliable.* *Support individual formatting of file systems.*	*Simple to use and implement.* *Inexpensive to maintain.*
Data Deduplication	 * It saves network bandwidth usage, but may require additional time for comparing data.	
Incremental Backup		

File System Level

- Data and Files
  - What is data ?
    - Data is information that has been converted to a machine-readable, digital binary format.
    - Control information indicates how data should be processed.
    - Applications may embed control information in user data for formatting or presentation.
    - Data and its associated control information is organized into discrete units as files or records.
  - What is file ?
    - Files are the common containers for user data, application code, and operating system executables and parameters.

File System Level

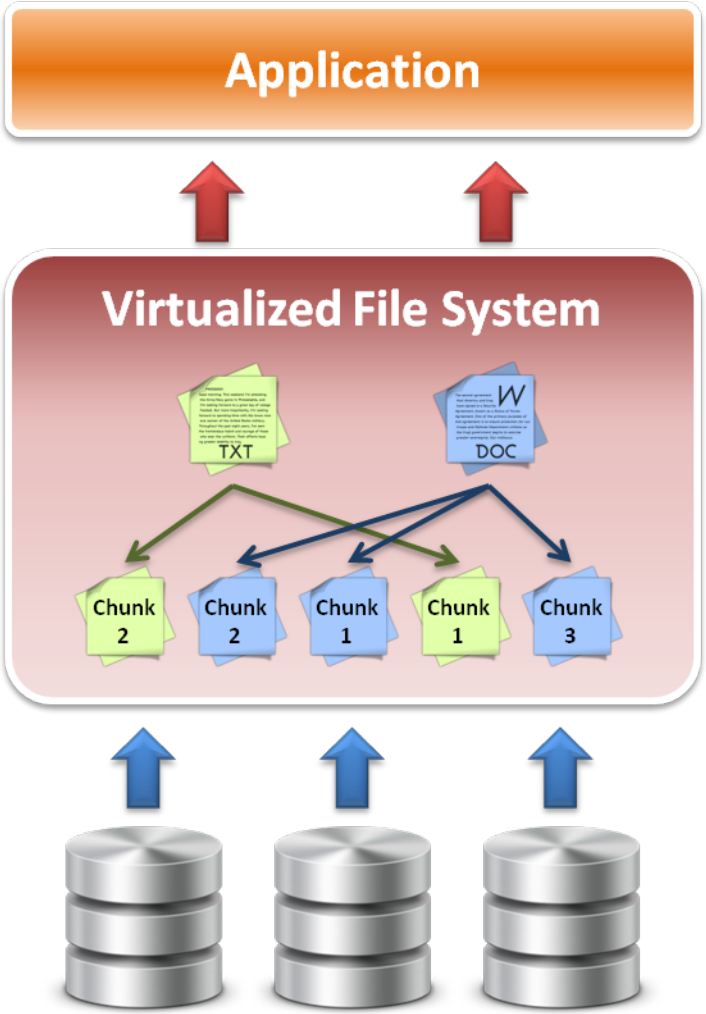
- About the files
  - Metadata
    - The control information for file management is known as metadata.
    - File metadata includes file attributes and pointers to the location of file data content.
    - File metadata may be segregated from a file's data content.
    - Metadata on file ownership and permissions is used in file access.
    - File timestamp metadata facilitates automated processes such as backup and life cycle management.
  - Different file systems
    - In Unix systems, file metadata is contained in the *i*-node structure.
    - In Windows systems, file metadata is contained in records of file attributes.

File System Level

- File system
  - What is file system ?
    - A file system is a software layer responsible for organizing and policing the creation, modification, and deletion of files.
    - File systems provide a hierarchical organization of files into directories and subdirectories.
    - The *B*-tree algorithm facilitates more rapid search and retrieval of files by name.
    - File system integrity is maintained through duplication of master tables, change logs, and immediate writes off file changes.
  - Different file systems
    - In Unix, the super block contains information on the current state of the file system and its resources.
    - In Windows NTFS, the master file table contains information

File System Level

- File system level virtualization
  - File system maintains metadata (*i*-node) of each file.
  - Translate file access requests to underlining file system.
  - Sometime divide large file into small sub-files (chunks) for parallel access, which improves the performance

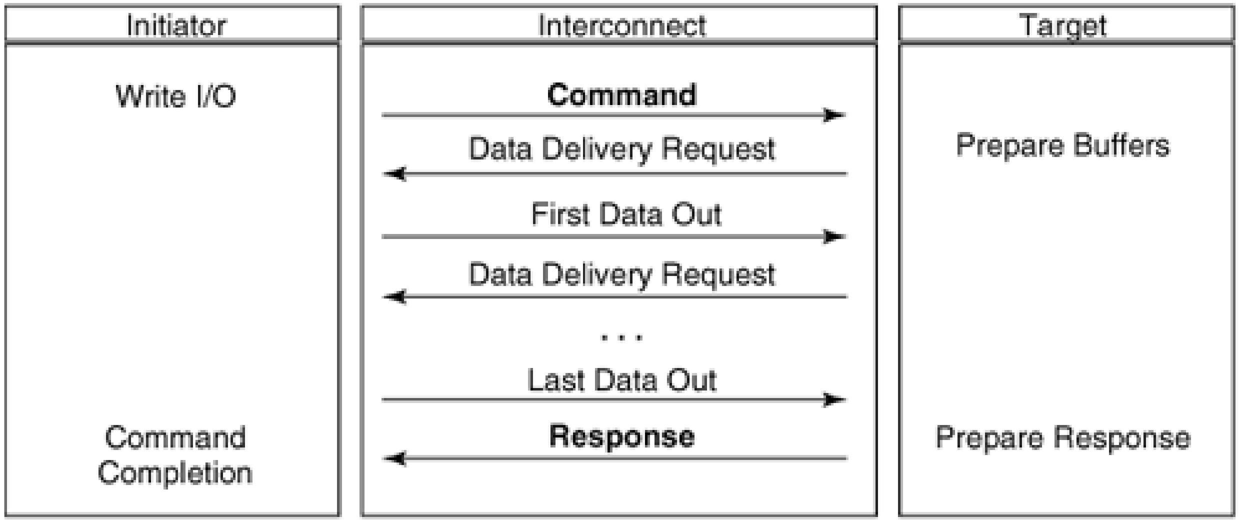


Block Device Level

- Block level data
  - The file system block
    - The atomic unit of file system management is the file system block.
    - A file's data may span multiple file system blocks.
    - A file system block is composed of a consecutive range of disk block addresses.
  - Data in disk
    - Disk drives read and write data to media through cylinder, head, and sector geometry.
    - Microcode on a disk translates between disk block numbers and cylinder/head/sector locations.
    - This translation is an elementary form of

Block Device Level

- Block device interface
  - SCSI (*Small Computer System Interface*)
    - The exchange of data blocks between the host system and storage is governed by the SCSI protocol.
    - The SCSI protocol is implemented in a client/server model.
    - The SCSI protocol is responsible for block exchange but does not define how data blocks will be placed on disk.
    - Multiple instances of SCSI client/server sessions may run concurrently between a server and storage.

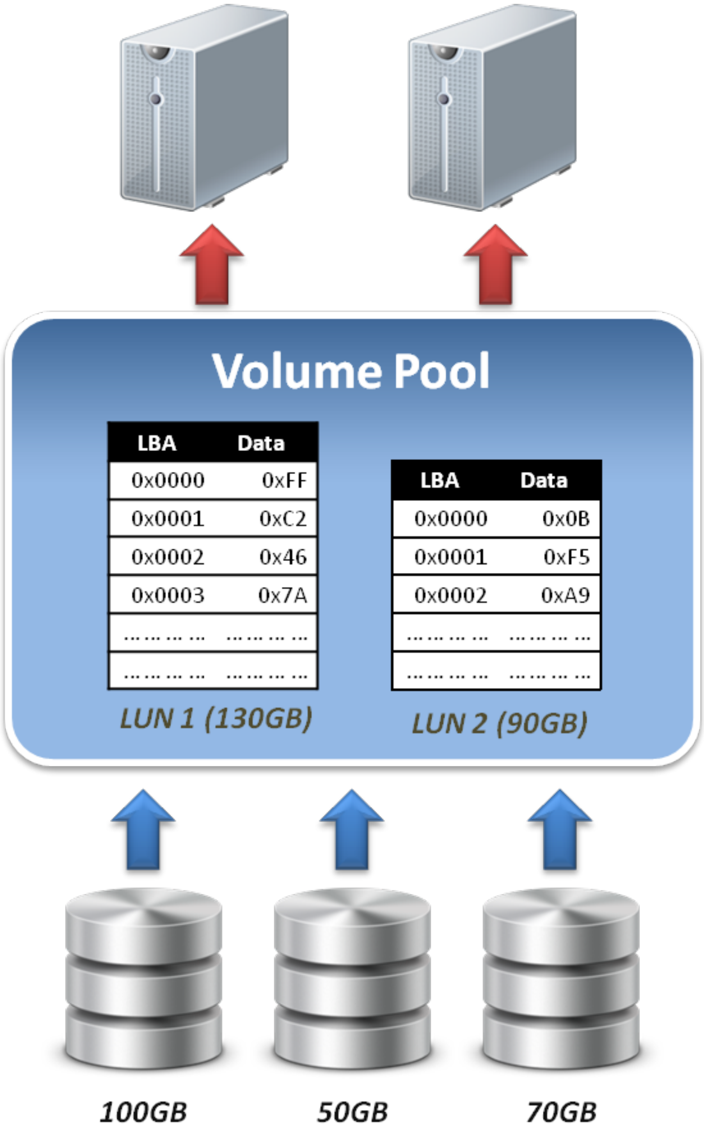


Block Device Level

- Logical unit and Logical volume
  - Logical unit
    - The SCSI command processing entity within the storage target represents a logical unit (LU) and is assigned a logical unit number (LUN) for identification by the host platform.
    - LUN assignment can be manipulated through LUN mapping, which substitutes virtual LUN numbers for actual ones.
  - Logical volume
    - A volume represents the storage capacity of one or more disk drives.
    - Logical volume management may sit between the file system and the device drivers that control system I/O.
    - Volume management is responsible for creating

Block Device Level

- Data block level virtualization
  - LUN & LBA
    - A single block of information is addressed using a logical unit identifier (LUN) and an offset within that LUN, which known as a Logical Block Address (LBA).
  - Apply address space remapping
    - The address space mapping is between a logical disk and a logical unit presented by one or more storage controllers.

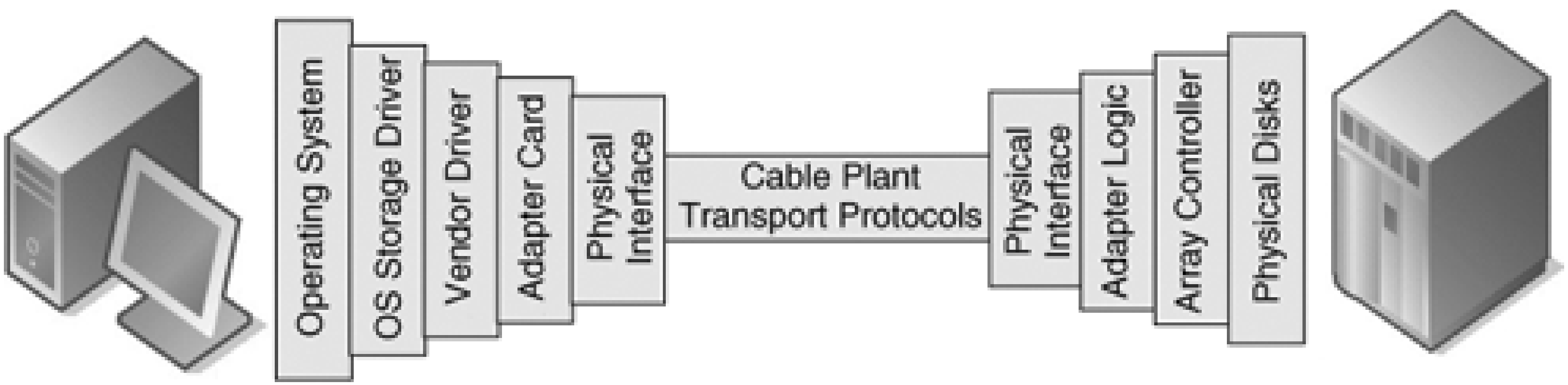


- Introduction
- What to be virtualized
- Where to be virtualized
- How to be virtualized
- Case study

STORAGE VIRTUALIZATION

Where To Be Virtualized

- Storage interconnection
  - The path to storage
    - The storage interconnection provides the data path between servers and storage.
    - The storage interconnection is composed of both hardware and software components.
    - Operating systems provide drivers for I/O to storage assets.
    - Storage connectivity for hosts is provided by host bus adapters (HBAs) or network interface cards (NICs).

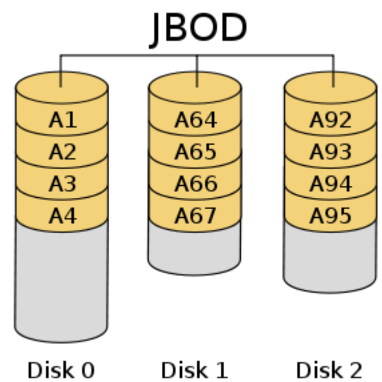


Where To Be Virtualized

- Storage interconnection protocol
  - Fibre Channel
    - Usually for high performance requirements.
    - Supports point-to-point, arbitrated loop, and fabric interconnects.
    - Device discovery is provided by the simple name server (SNS).
    - Fibre Channel fabrics are self-configuring via fabric protocols.
  - iSCSI (*internet SCSI*)
    - For moderate performance requirements.
    - Encapsulates SCSI commands, status and data in TCP/IP.
    - Device discovery by the Internet Storage Name Service (iSNS).

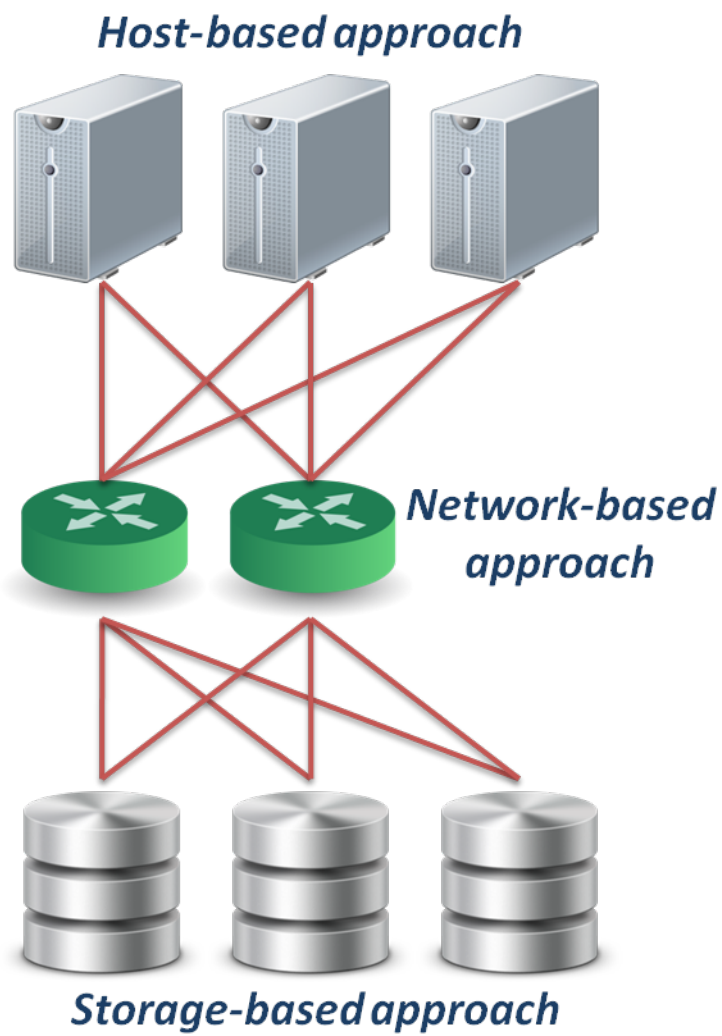
Where To Be Virtualized

- Abstraction of physical storage
  - Physical to virtual
    - The cylinder, head and sector geometry of individual disks is virtualized into logical block addresses (LBAs).
    - For storage networks, the physical storage system is identified by a network address / LUN pair.
    - Combining RAID and JBOD assets to create a virtualized mirror must accommodate performance differences.
  - Metadata integrity
    - Storage metadata integrity requires redundancy for failover or load balancing.



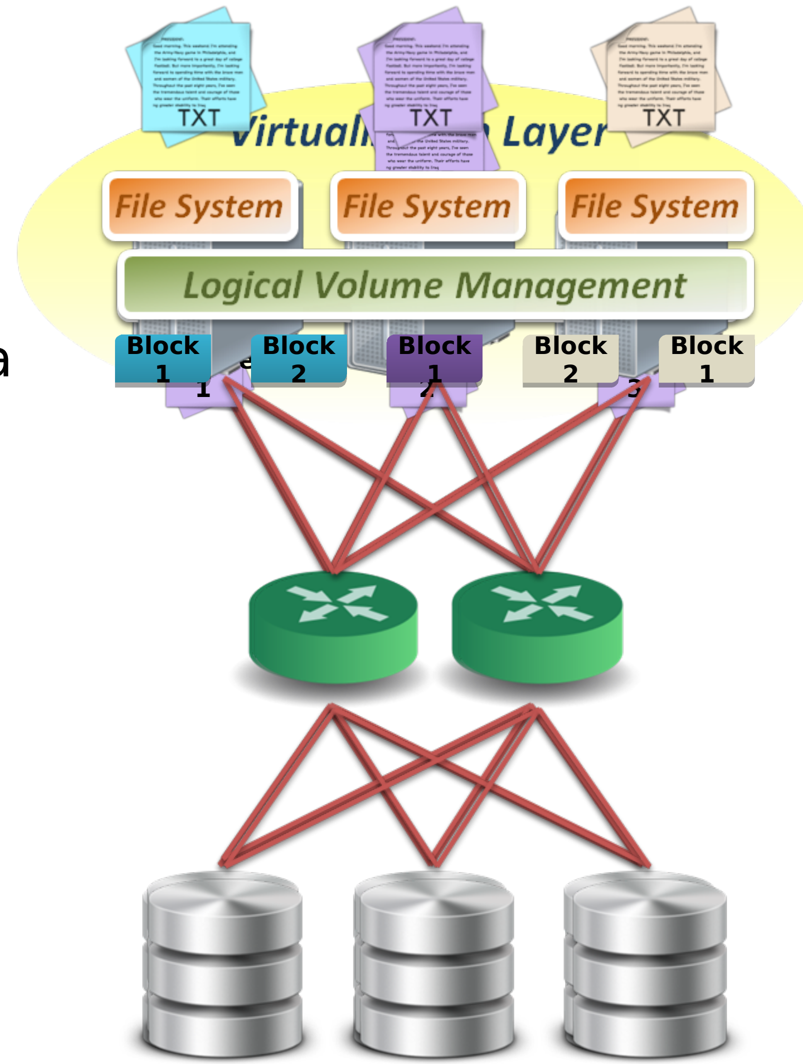
Where To Be Virtualized

- Different approaches :
  - Host-based approach
    - Implemented as a software running on host systems.
  - Network-based approach
    - Implemented on network devices.
  - Storage-based approach
    - Implemented on storage target subsystem.



Host-based Virtualization

- Host-based approach
  - File level
    - Run virtualized file system on the host to map files into data blocks, which distributed among several storage devices.
  - Block level
    - Run logical volume management software on the host to intercept I/O requests and redirect them to storage devices.
  - Provide services
    - Software RAID

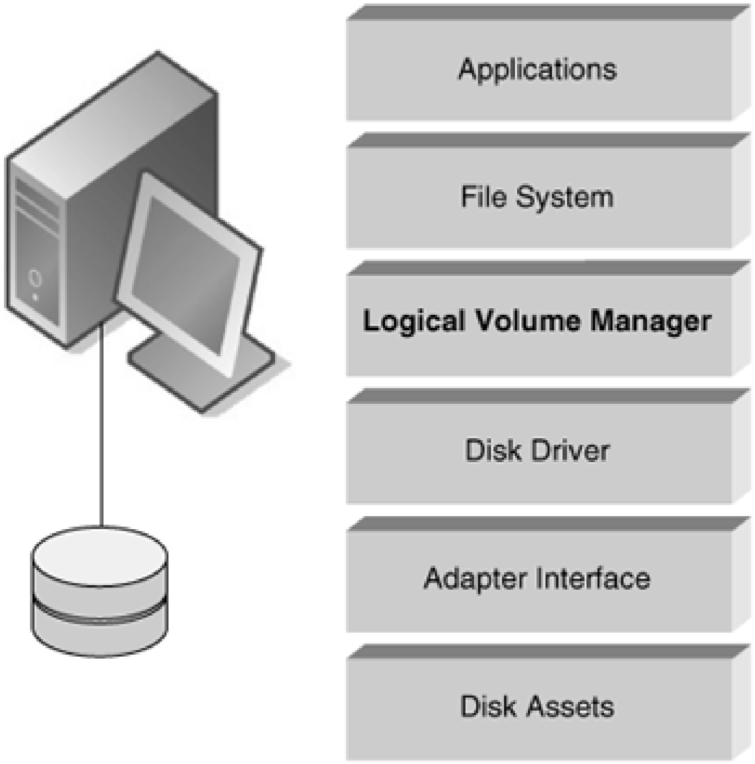


Host-based Virtualization

- Important issues
  - Storage metadata servers
    - Storage metadata may be shared by multiple servers.
    - Shared metadata enables a SAN file system view for multiple servers.
    - Provides virtual to real logical block address mapping for client.
    - A distributed SAN file system requires file locking mechanisms to preserve data integrity.
  - Host-based storage APIs
    - May be implemented by the operating system to provide a common interface to disparate virtualized resources.
    - Microsoft's virtual disk service (VDS) provides a management interface for dynamic generation of virtualized storage.

Host-based Virtualization

- A typical example :
  - LVM
    - Software layer between the file system and the disk driver.
    - Executed by the host CPU.
    - Lack hardware-assist for functions such as software RAID.
    - Independence from vendor-specific storage architectures.
    - Dynamic capacity allocation to expand or shrink volumes.
    - Support alternate pathing for high availability.

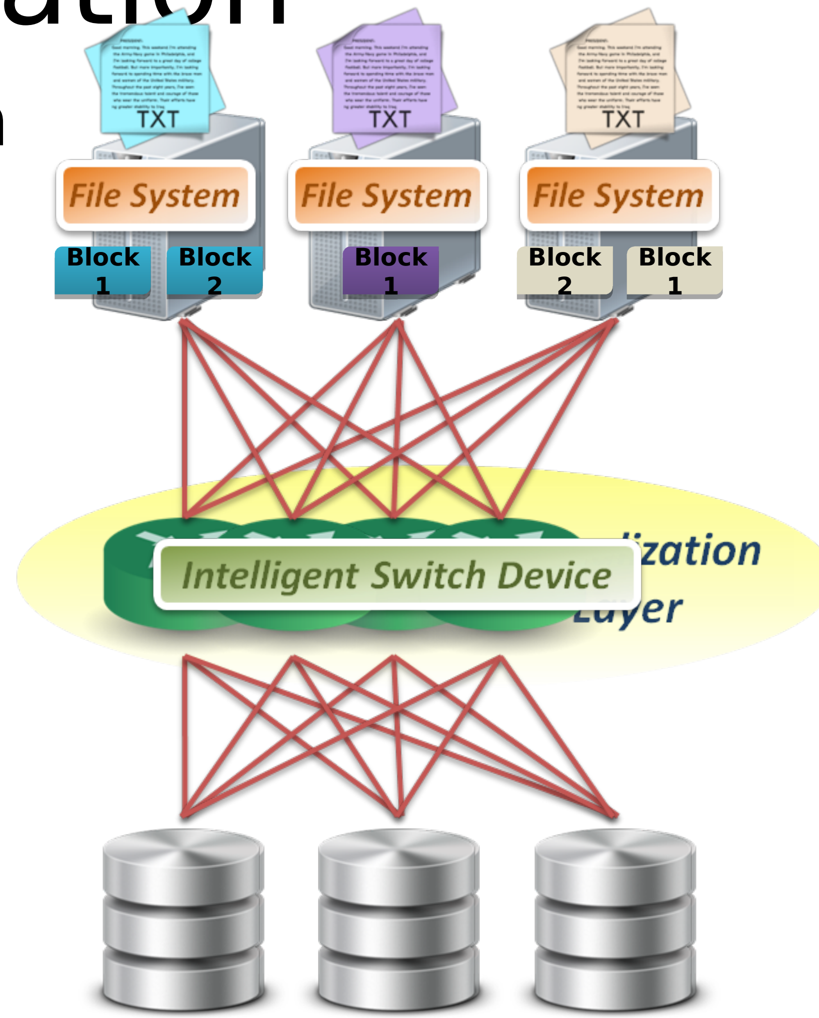


Host-based Virtualization

- Host-based implementation
  - Pros
    - No additional hardware or infrastructure requirements
    - Simple to design and implement
    - Improve storage utilization
  - Cons
    - Storage utilization optimized only on a per host base
    - Software implementation is dependent to each operating system
    - Consume CPU clock cycle for virtualization
  - Examples
    - LVM, NFS

Network-based Virtualization

- Network-based approach
  - File level
    - Seldom implement file level virtualization on network device.
  - Block level
    - Run software on dedicated appliances or intelligent switches and routers.
  - Provide services
    - Multi-path
    - Storage pooling

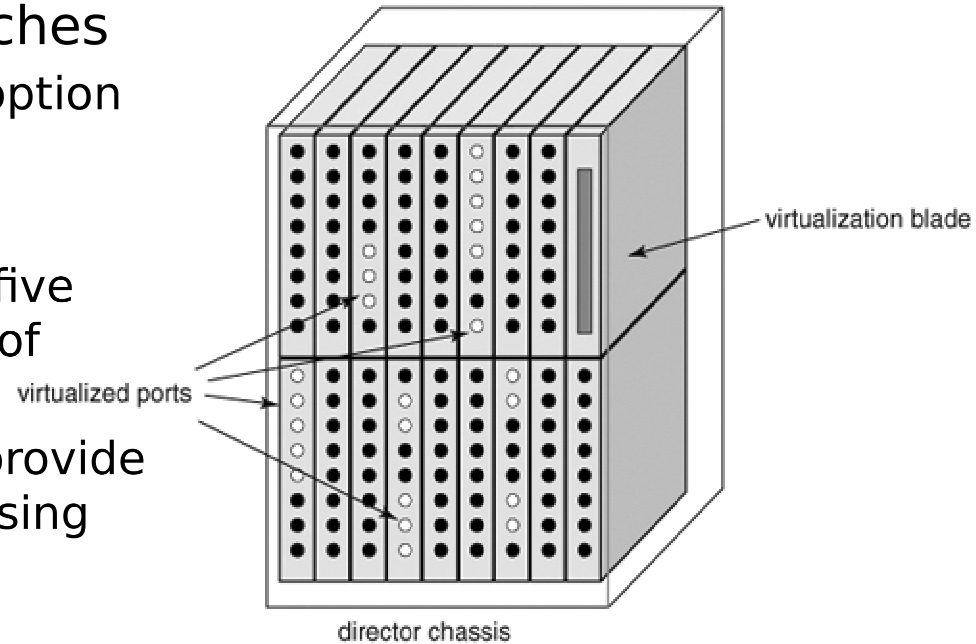


Network-based Virtualization

- Requirements of storage network
  - Intelligent services
    - Logon services
    - Simple name server
    - Change notification
    - Network address assignment
    - Zoning
  - Fabric switch should provide
    - Connectivity for all storage transactions
    - Interoperability between disparate servers, operating systems, and target devices

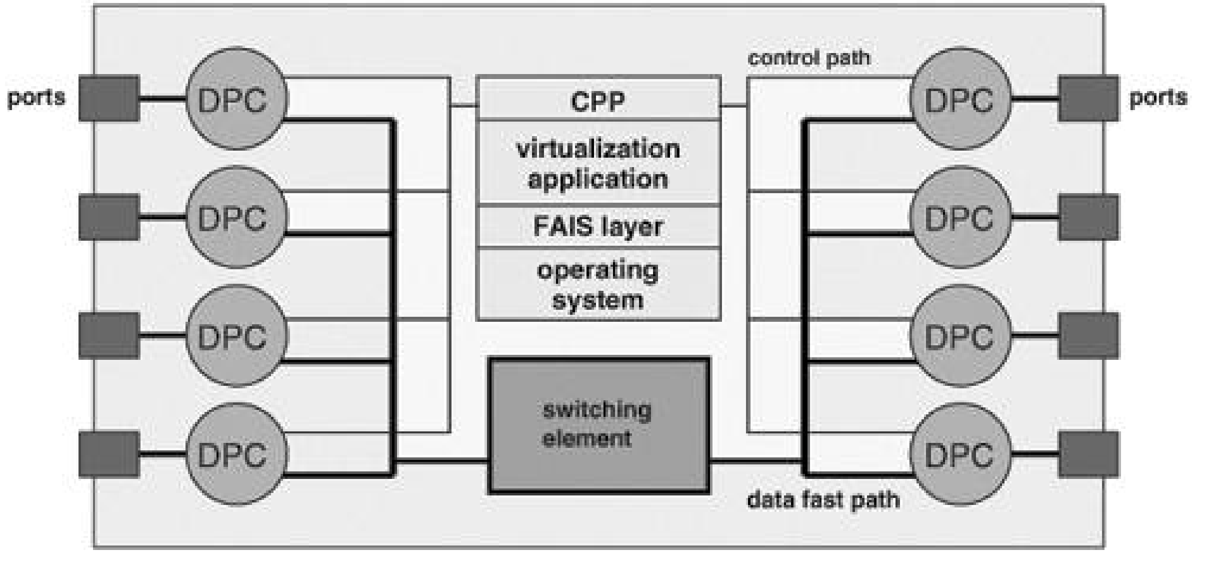
Network-based Virtualization

- Techniques for fabric switch virtualization
  - Hosted on departmental switches
    - A PC engine provisioned as an option blade.
  - Data center directors
    - Should be able to preserve the five nines availability characteristic of director-class switches.
    - Dedicated virtualization ASICs provide high-performance frame processing and block address mapping.
  - Interoperability between different implementations will become a priority.



Network-based Virtualization

- Interoperability issue
  - FAIS (*Fabric Application Interface Standard*)
    - Define a set of standard APIs to integrate applications and switches.
    - FAIS separates control information and data paths.
    - The control path processor (CPP) supports the FAIS APIs and upper layer storage virtualization application.
    - The data path controller (DPC) executes the virtualized SCSI I/Os under the management of one or more CPPs

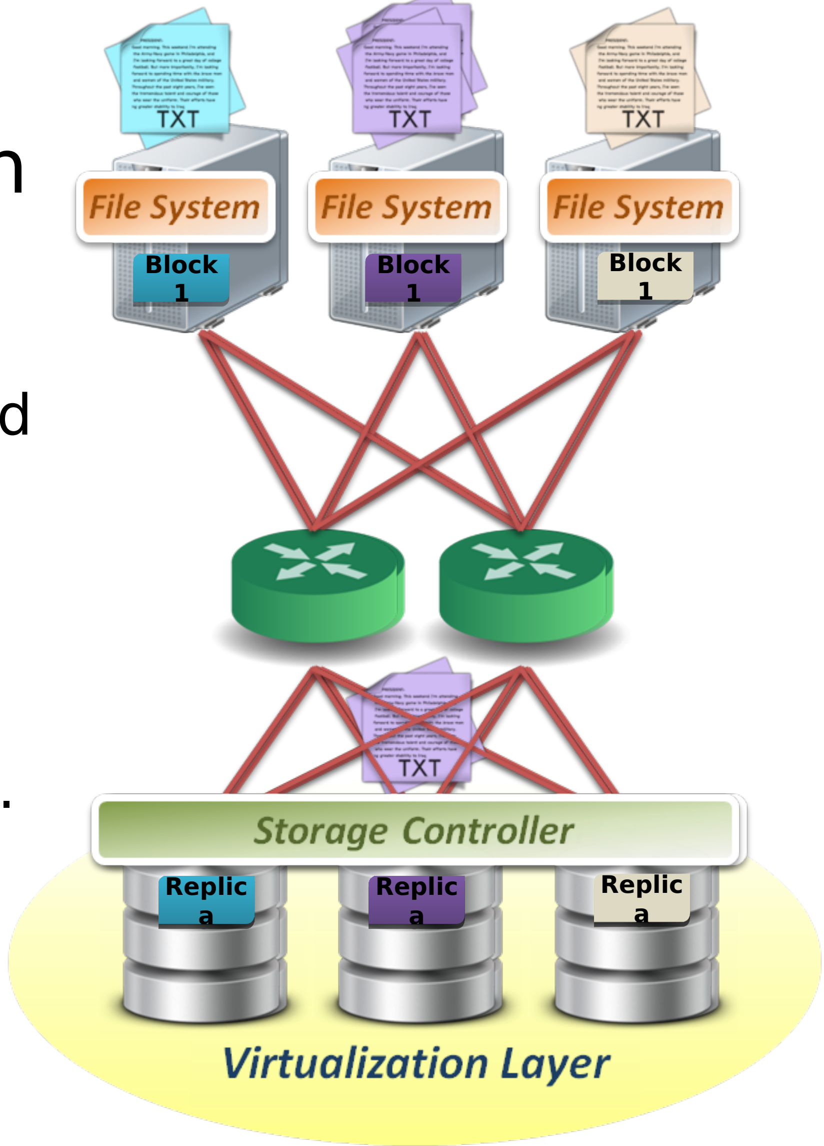


Network-based Virtualization

- Network-based implementation
  - Pros
    - True heterogeneous storage virtualization
    - No need for modification of host or storage system
    - Multi-path technique improve the access performance
  - Cons
    - Complex interoperability matrices - limited by vendors support
    - Difficult to implement fast metadata updates in switch device
    - Usually require to build specific network equipments (e.g., Fibre Channel)
  - Examples

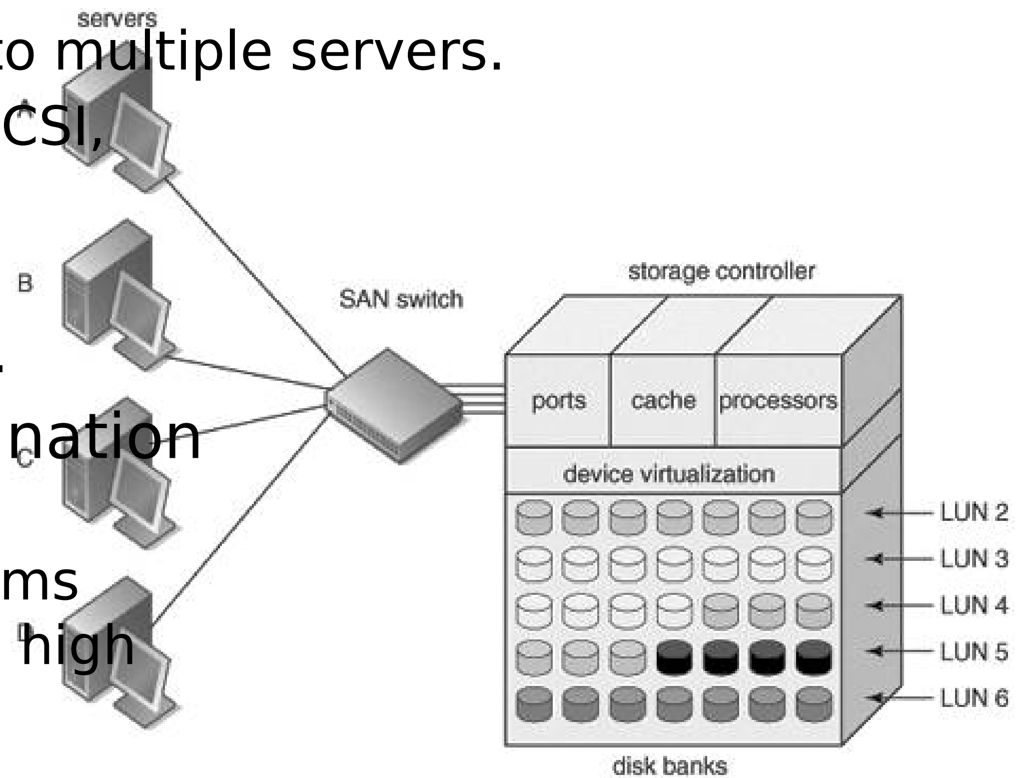
Storage-based Virtualization

- Storage-based approach
  - File level
    - Run software on storage device to provide file based data storage services to host through network.
  - Block level
    - Embeds the technology in the target storage devices.
  - Provide services
    - Storage pooling
    - Replication and RAID
    - Data sharing and tiering



Storage-based Virtualization

- Array-based virtualization
  - Storage controller
    - Provide basic disk virtualization in the form of RAID management, mirroring, and LUN mapping or masking.
    - Allocate a single LUN to multiple servers.
    - Offer Fibre Channel, iSCSI, and SCSI protocol.
  - Cache memory
    - Enhance performance.
  - Storage assets coordination
    - Coordination between multiple storage systems is necessary to ensure high availability.



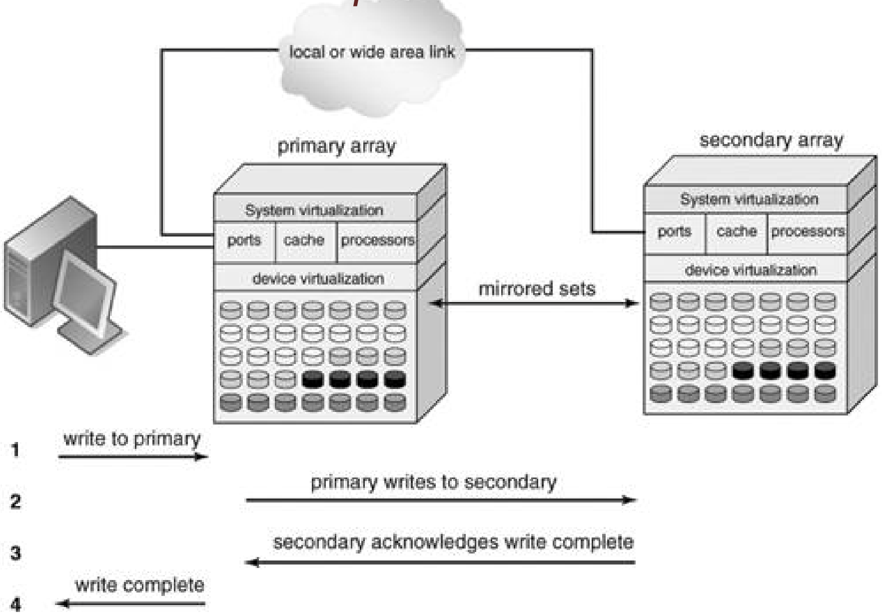
Storage-based Virtualization

- Data replication
  - Array-based data replication
    - Referred to as disk-to-disk replication.
    - Requires that a storage controller function concurrently as both an initiator and target.
  - Synchronous vs. Asynchronous
    - Synchronous data replication ensures that a write operation to a secondary disk array is completed before the primary array acknowledges task completion to the server.
    - Asynchronous data replication provides write completion by the primary array, although the transaction may still be pending to the secondary array.

Storage-based Virtualization

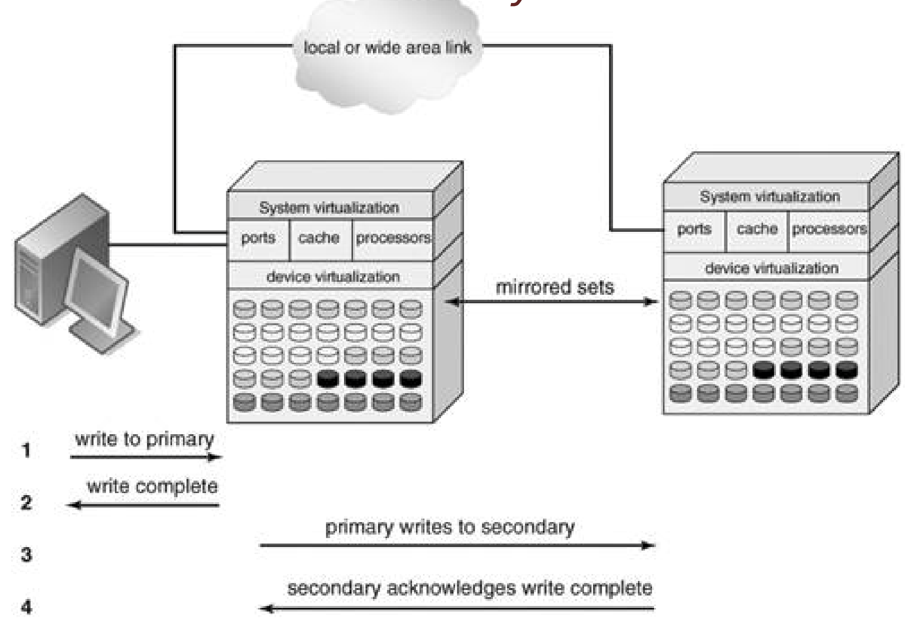
Synchronous

To preserve performance, synchronous data replication is limited to metropolitan distances



Asynchronous

Asynchronous data replication is largely immune to transmission latency



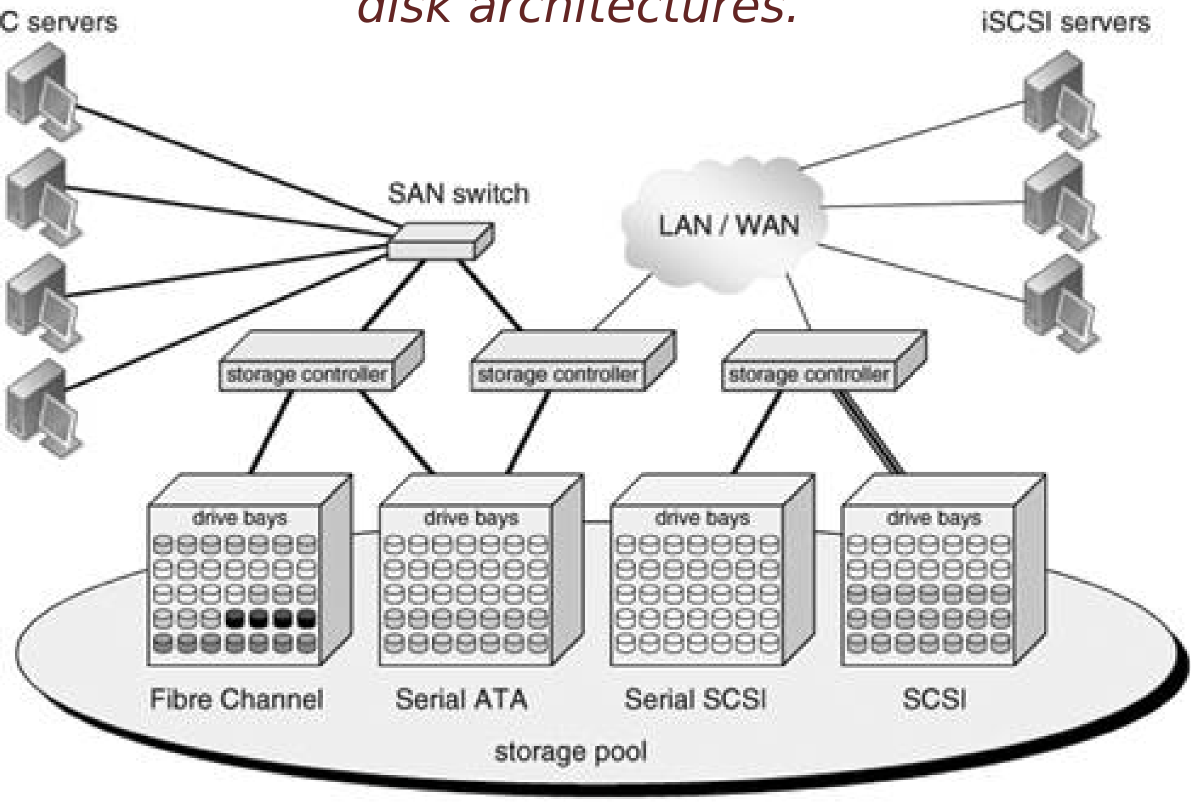
Storage-based Virtualization

- Other features
  - Point-in-time copy (snapshot)
    - Provide point-in-time copies of an entire storage volume.
    - Snapshot copies may be written to secondary storage arrays.
    - Provide an efficient means to quickly recover a known good volume state in the event of data from the host.
  - Distributed modular virtualization
    - Decoupling storage controller logic from physical disk banks provides flexibility for supporting heterogeneous disk assets and facilitates distributed virtualization intelligence.
    - Accommodates class of storage services and data lifecycle management.

Storage-based Virtualization

Distributed Modular Virtualization

Decoupling storage controller intelligence and virtualization engines from physical disk banks facilitates multi-protocol block data access and accommodation of a broad range of disk architectures.



Storage-based Virtualization

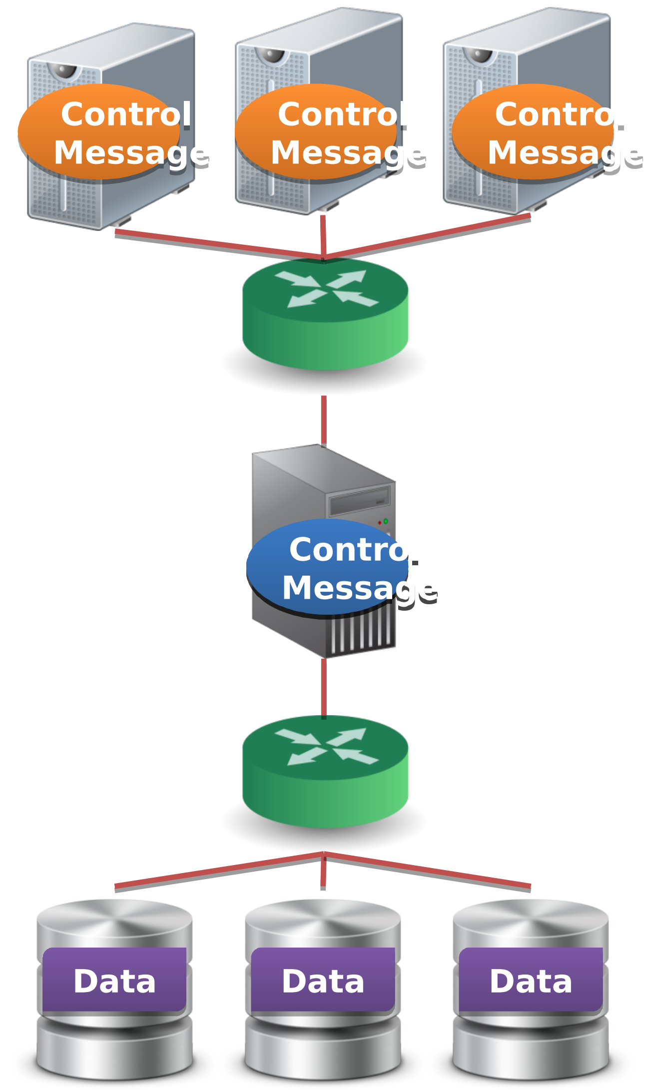
- Storage-based implementation
  - Pros
    - Provide most of the benefits of storage virtualization
    - Reduce additional latency to individual IO
  - Cons
    - Storage utilization optimized only across the connected controllers
    - Replication and data migration only possible across the connected controllers and the same vendors devices
  - Examples
    - Disk array products

- Introduction
- What to be virtualized
- Where to be virtualized
- How to be virtualized

STORAGE VIRTUALIZATION

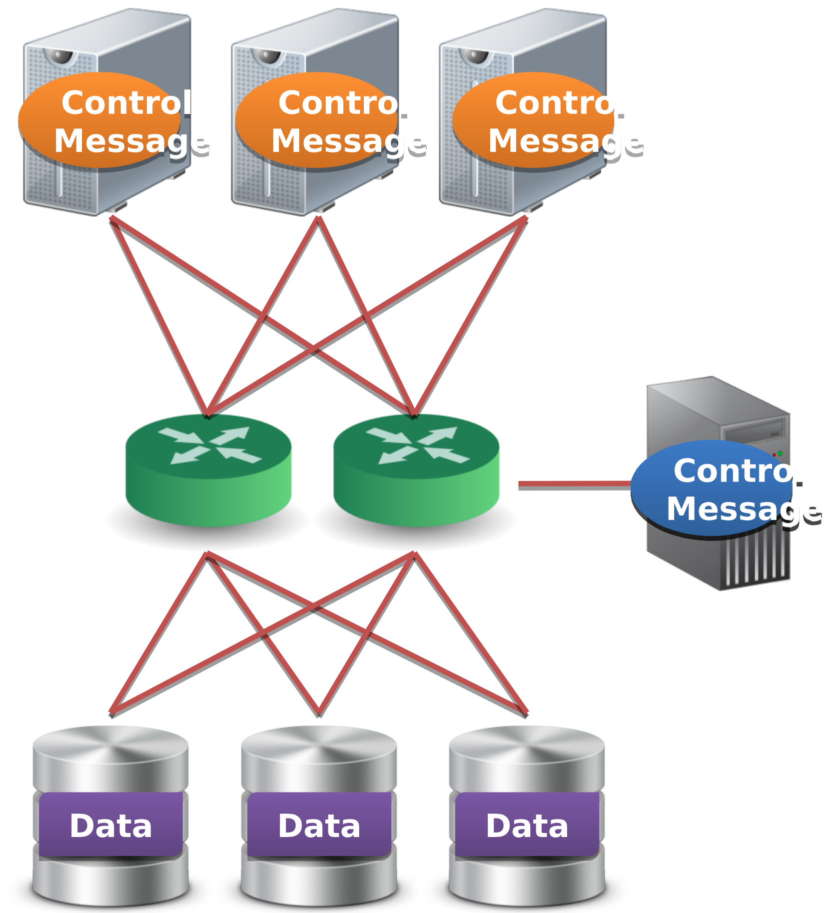
In-band Virtualization

- Implementation methods :
  - In-band
    - Also known as ***symmetric***, virtualization devices actually sit in the data path between the host and storage.
    - Hosts perform IO to the virtualized device and never interact with the actual storage device.
  - Pros
    - Easy to implement
  - Cons
    - Bad scalability & Bottle neck



Out-of-band Virtualization

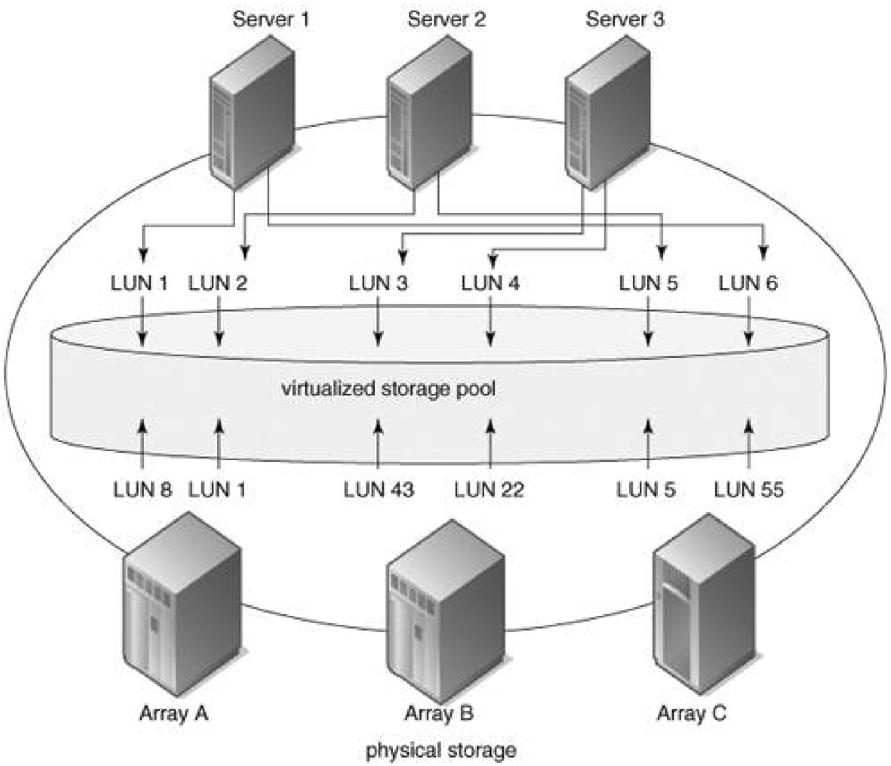
- Implementation methods :
  - Out-of-band
    - Also known as ***asymmetric***, virtualization devices are sometimes called metadata servers.
    - Require additional software in the host which knows the first request location of the actual data.
  - Pros
    - Scalability & Performance
  - Cons
    - Hard to implement



Other Virtualization Services

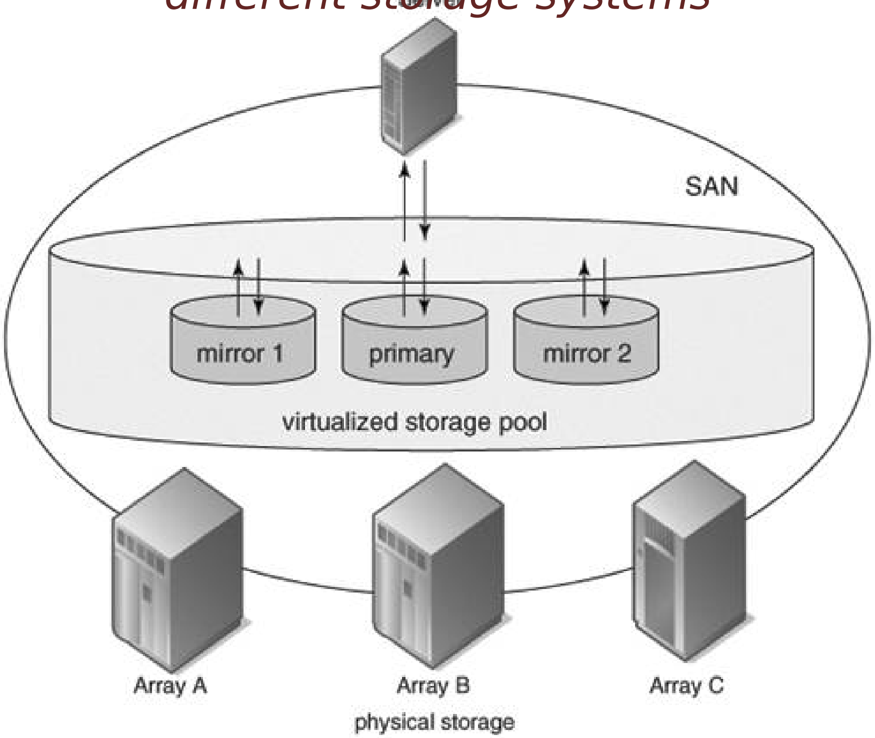
Pooling Heterogeneous Storage Assets

In a virtualized storage pool, virtual assets may be dynamically resized and allocated to servers by drawing on the total storage capacity of the SAN



Heterogeneous Mirroring

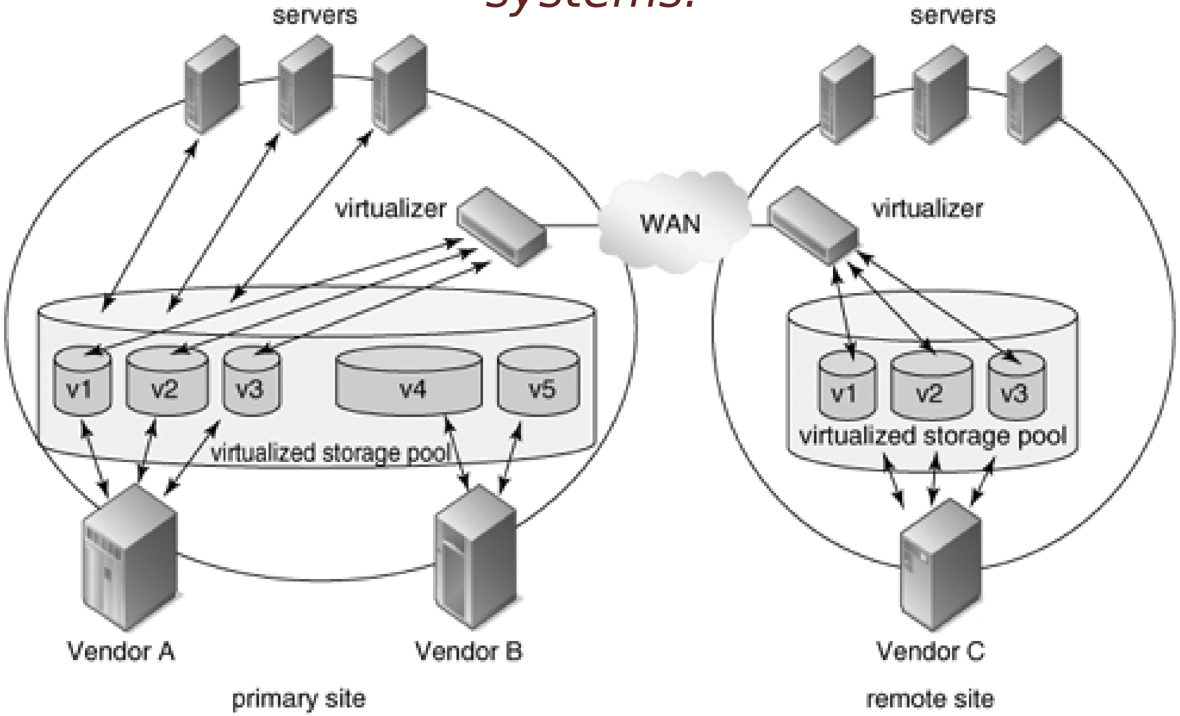
Heterogeneous mirroring offers more flexible options than conventional mirroring, including three-way mirroring within storage capacity carved from different storage systems



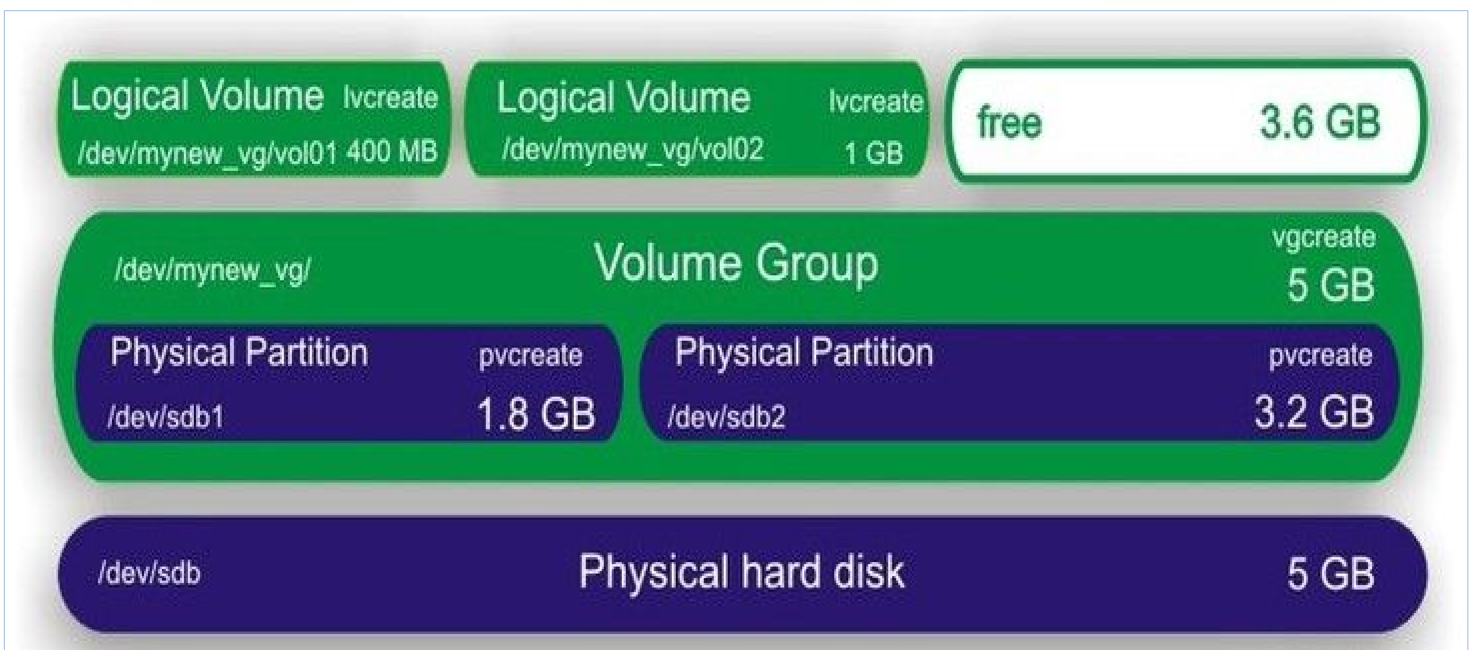
Other Virtualization Services

Heterogeneous Data Replication

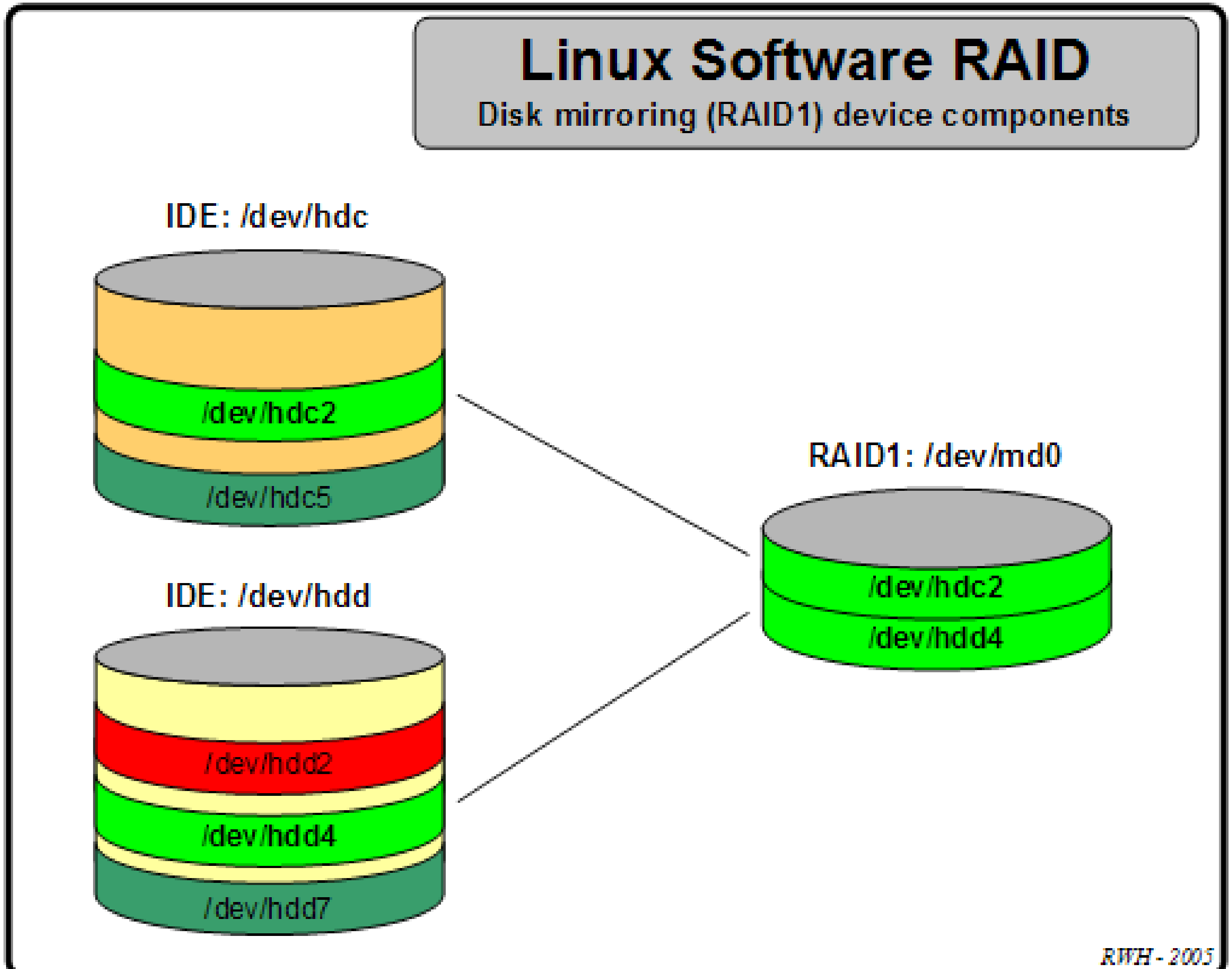
Heterogeneous data replication enables duplication of storage data between otherwise incompatible storage systems.



Software Technique – LVM creation



Software Technique – RAID creation



Summary

- Storage virtualization technique :
  - Virtualization layer
    - File level and block leve
  - Virtualization location
    - Host, network and stora
  - Virtualization method
    - In-band and out-of-ban
- Storage virtualization
  - Storage pooling and sharing
  - Data replication and mirroring
  - Snapshot and multi-pathing