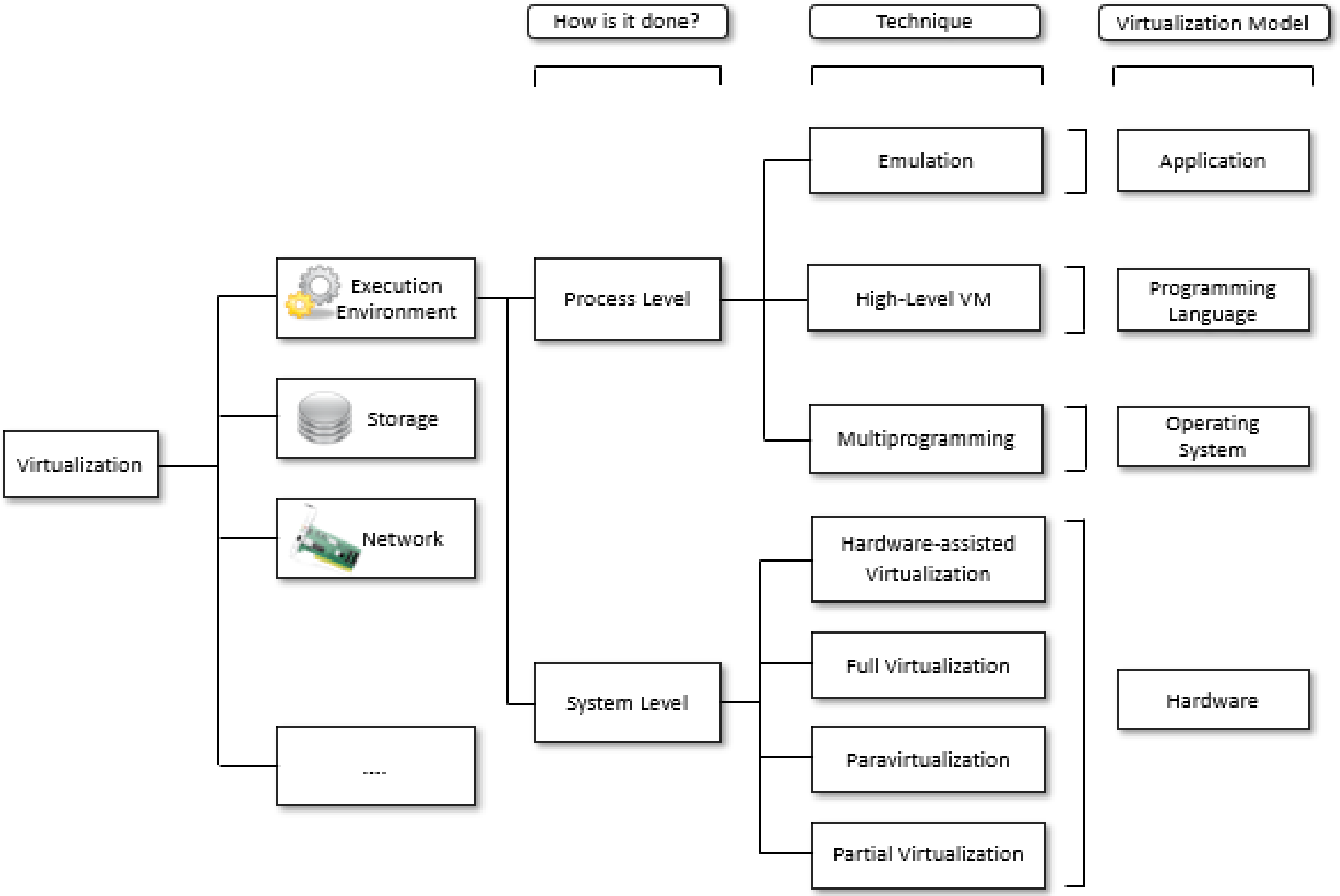
UNIT 2:
TYPES OF VIRTUALIZATION

Taxonomy of Virtualization

Virtualization covers a wide range of emulation techniques that are applied to different areas of computing. A classification of these techniques helps to better understand their characteristics and use.

- Virtualization is mainly used to emulate execution environments, storage, and networks.
- Among these categories ***execution virtualization*** constitutes the oldest, most popular, and most developed area.
- We can divide these execution virtualization techniques into two major categories by considering the type of host they require.
  - **Process level techniques** are implemented on top of an existing operating system.
  - **System level techniques** are implemented directly on hardware and do not require or require a minimum support from an existing operating system.

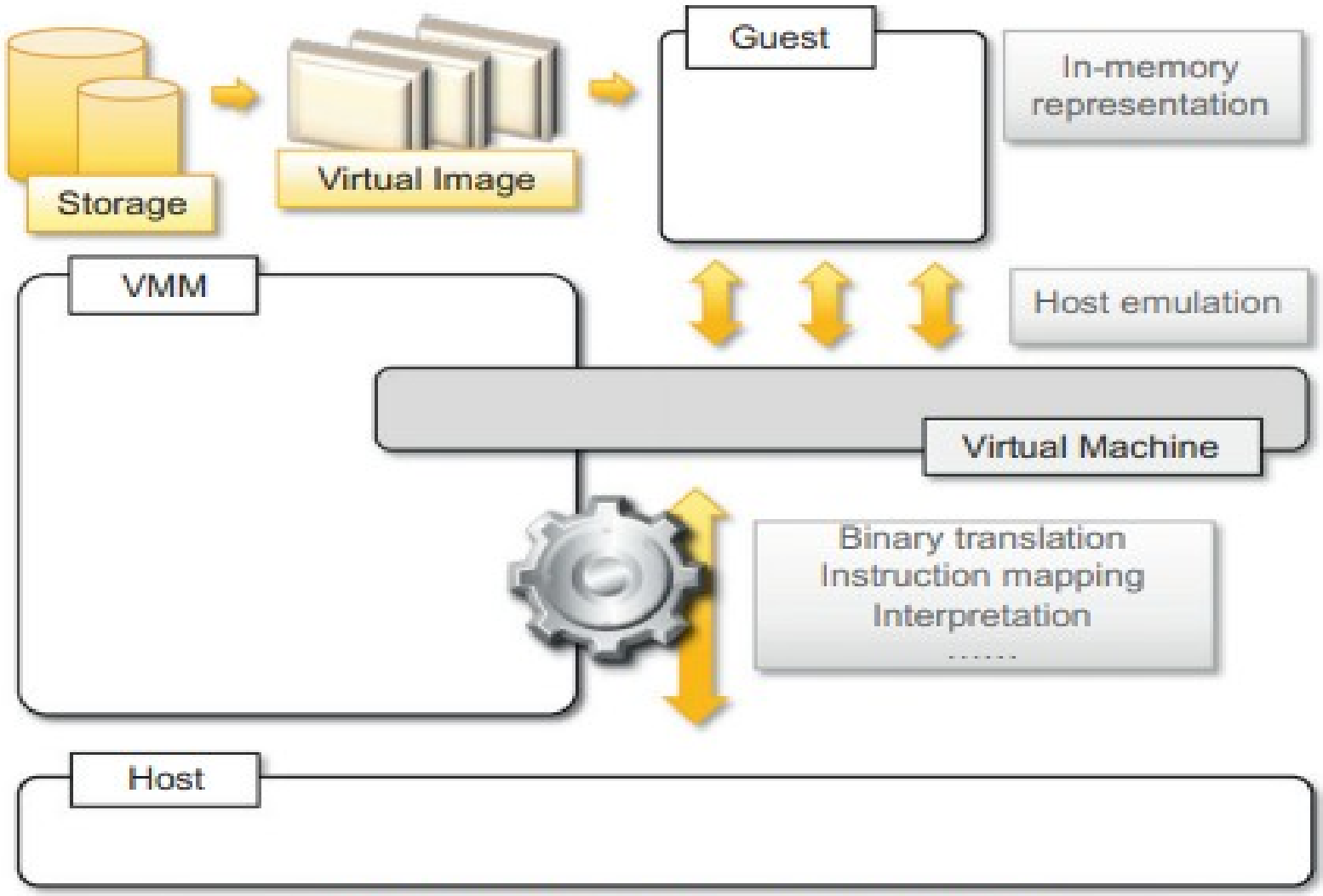
Taxonomy of Virtualization



Hardware Level Virtualization

- Hardware-level virtualization is a virtualization technique that provides an abstract execution environment in terms of computer hardware on top of which a guest operating system can be run. In this model
  - Guest is represented by the operating system
  - Host by the physical computer hardware
  - Virtual machine by its emulation
  - Virtual machine manager by the hypervisor
- The hypervisor is generally a program or a combination of software and hardware that allows the abstraction of the underlying physical hardware. Hypervisor recreates a hardware environment in which guest operating systems are installed.
- Hardware-level virtualization is also called **system level virtualization**, since it provides **ISA** to virtual machines, which is the representation of the hardware interface of a system. This is to differentiate it from **process level virtualization**, which expose **ABI** to virtual machines.

Hardware Level Virtualization



A hardware virtualization reference model.

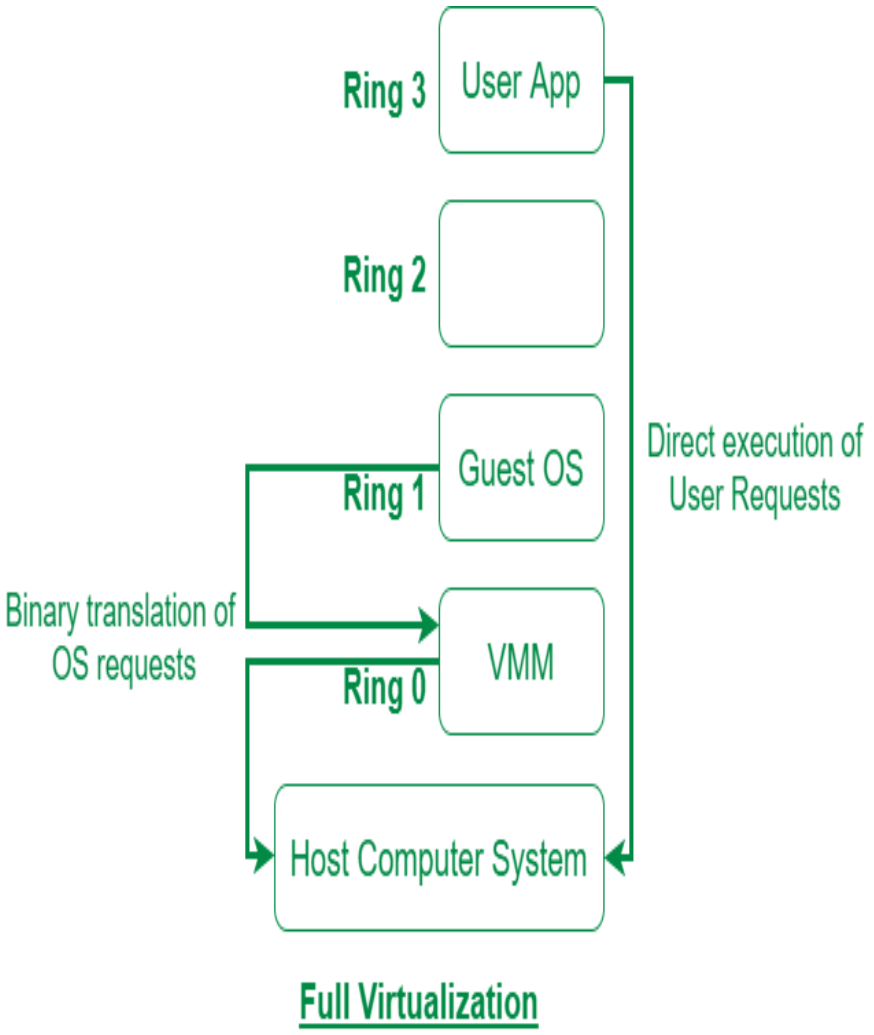
Types of Hardware Virtualization

Based on the hardware utilization, following are the types of virtualization

- Full virtualization
- Para virtualization
- Hardware Assisted Virtualization
- Partial virtualization

Full Virtualization

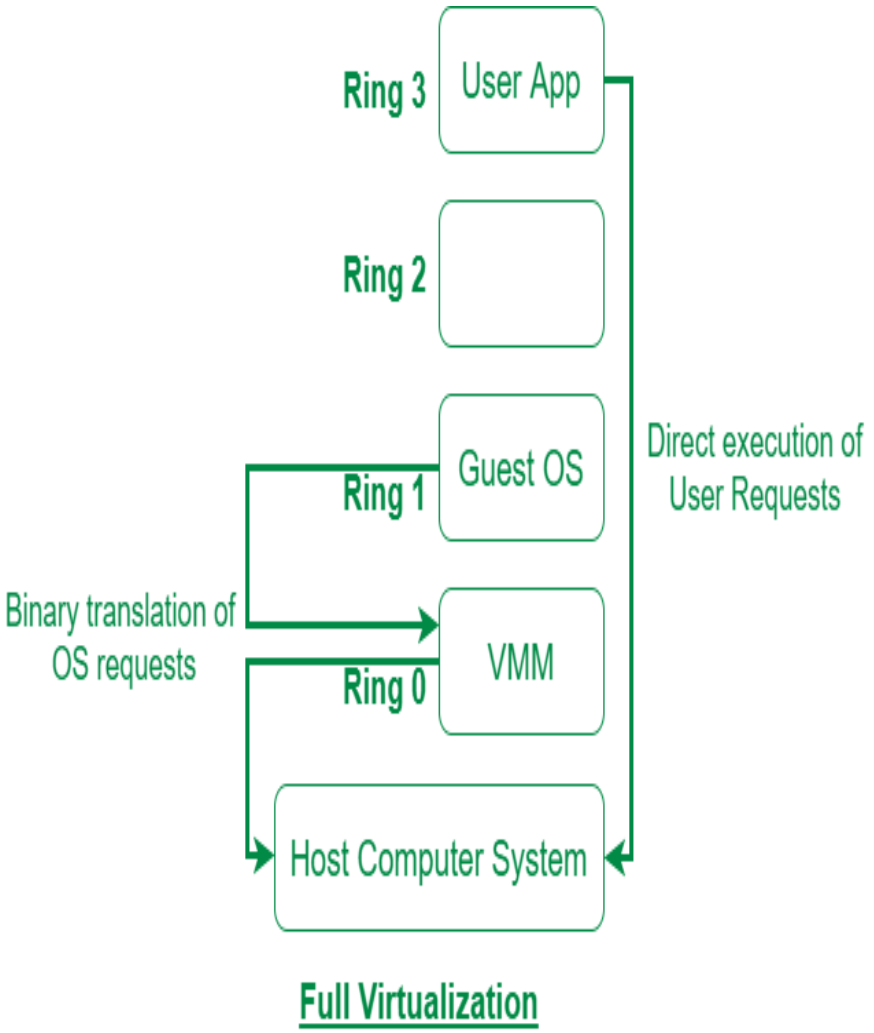
- Full virtualization refers to the ability to run an operating system, directly on top of a virtual machine without any modification, as though it were run on the raw hardware.
- To make this possible, virtual machine managers are required to provide a complete emulation of the entire underlying hardware.



Full Virtualization

Full Virtualization

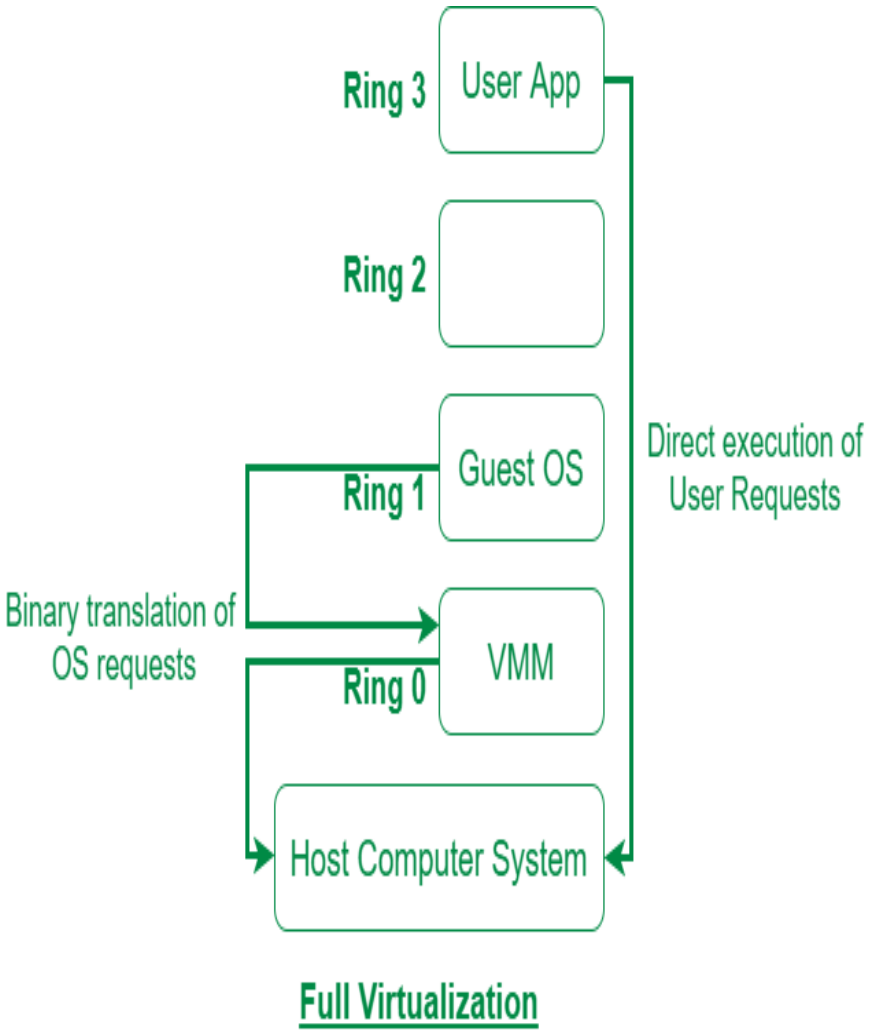
- VM's are unaware that they are working in the shared environment.
- The hypervisor traps and effectively translates any machine instructions for IO operations or any other instructions that require direct or privileged access to the underlying physical hardware.



Full Virtualization

Advantages of Full Virtualization

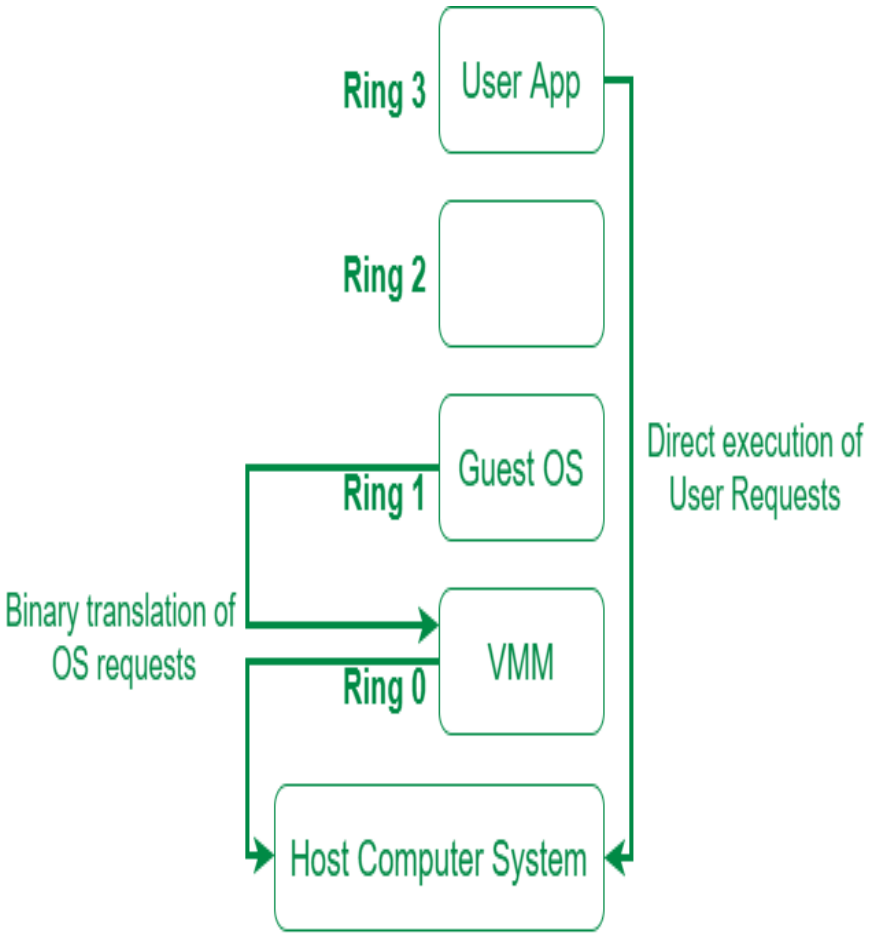
- It does not require modification in the guest OS and provide complete secure isolation between different VM's.
- VM runs independently, each with its own OS and configuration.
- VM's don't communicate and share resources.
- The requests from one VM are not allowed to alter the state of another VM.
- Full virtualization is de facto standard for most of the industries.



Full Virtualization

Disadvantages of Full Virtualization

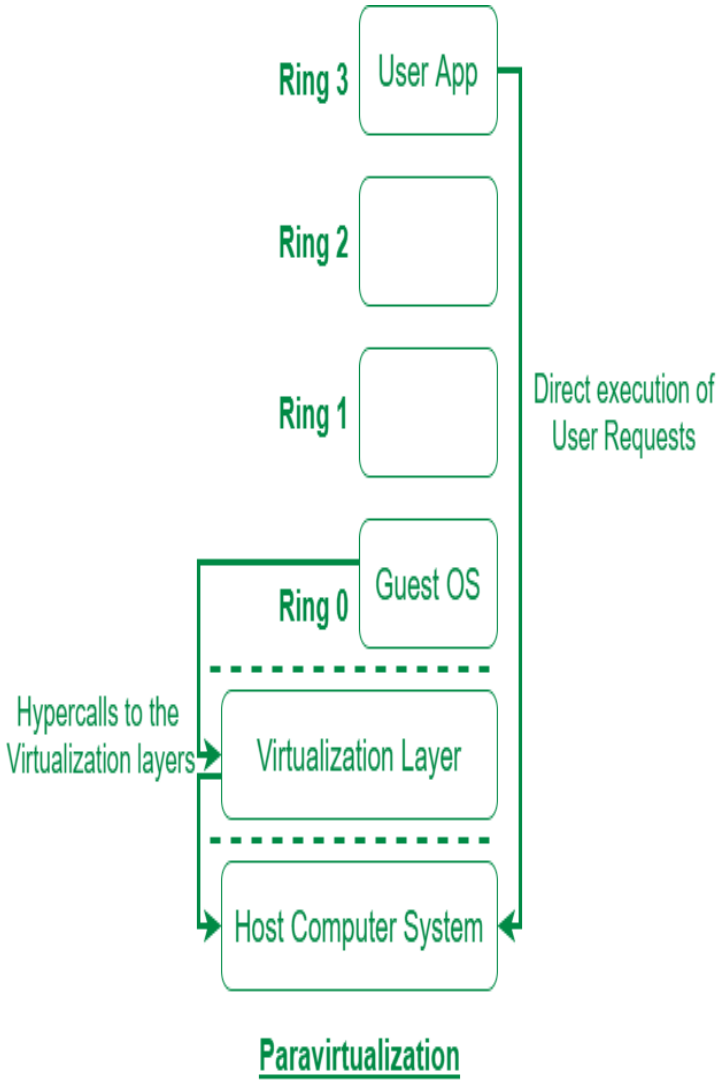
- Time critical applications that require instantly direct access to the hardware would not function properly in a VM.
- A key challenge is the interception of privileged instructions such as I/O instructions: Since they change the state of the resources exposed by the host, they must be contained within the virtual machine manager.
- Physical server fault can affect every VM.



Full Virtualization

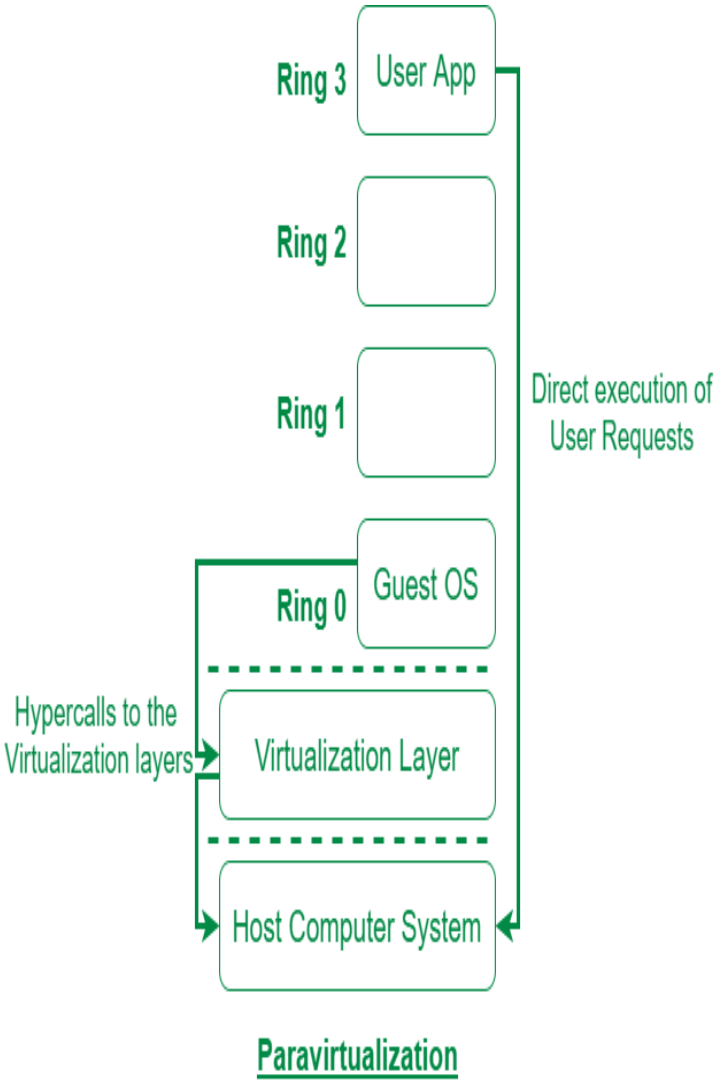
Para Virtualization

- The aim of paravirtualization is to provide the capability to execute the performance-critical operations directly on the host, thus preventing performance losses.
- This allows a simpler implementation of virtual machine managers that must simply transfer the execution of these operations, which were hard to virtualize, directly to the host.



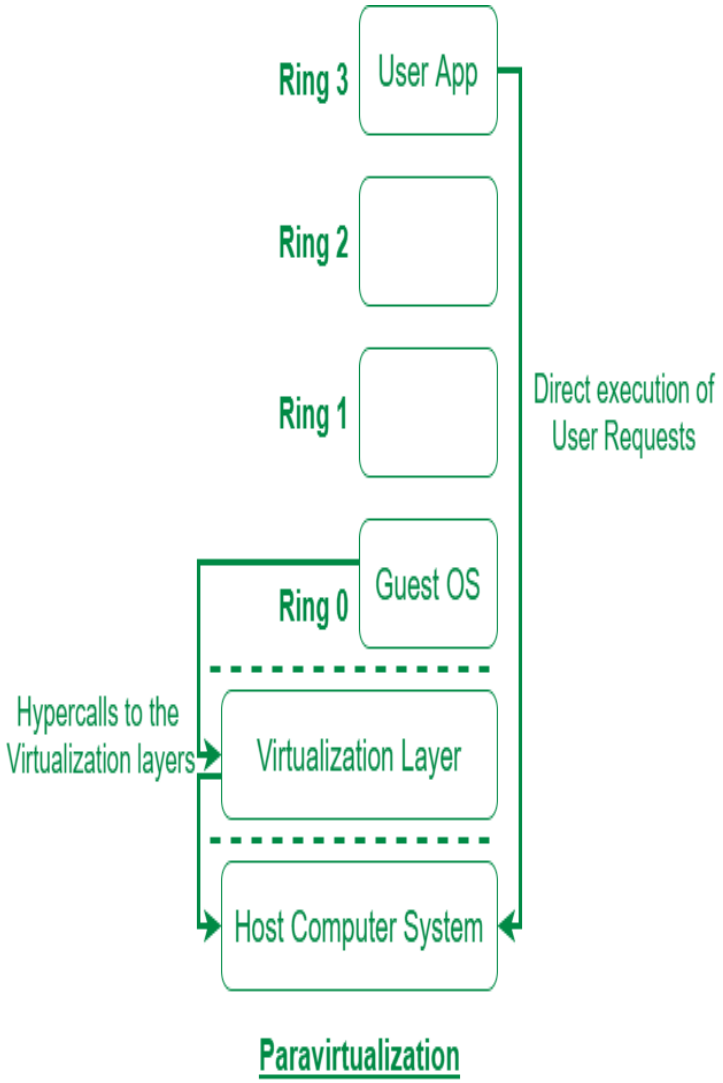
Para Virtualization

- Guest OS need to be modified and explicitly ported by remapping the performance-critical operations through the virtual machine software interface.
- This is possible when the source code of the operating system is available, and this is the reason that paravirtualization was mostly explored in the open source and academic environment.



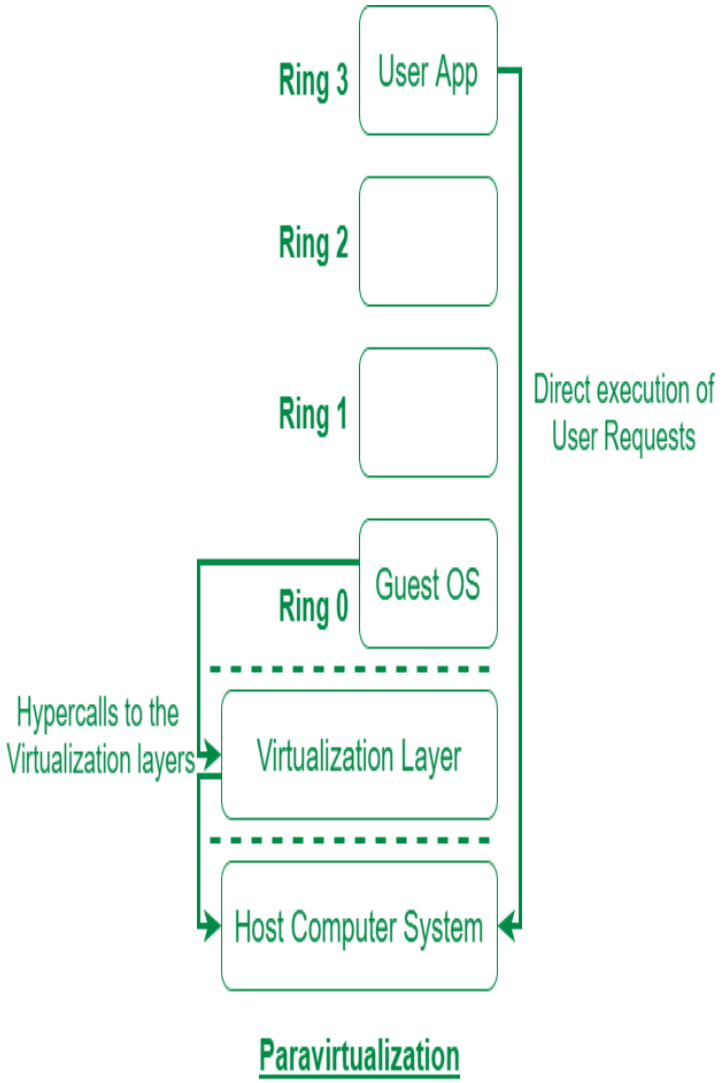
Para Virtualization

- Para Virtualization requires specialized hypervisor support in the form of an application programming interface (API).
- This specialized support from the hypervisor is known as the Hyper-call or Para-API.
- API is used by virtual machines to request services from the hypervisor.
- The virtual machine operations could be run in the virtual context (simulation) or the Hyper-call context.



Para Virtualization

- The performance of operations in the virtual context is slower and suffers from performance degradation.
- The guest operating systems hosted in a virtual machine must be modified to call these APIs instead of the regular machine instructions.
- VM's are aware that they are working in the shared environment.



Paravirtualization

Advantages of Para Virtualization

- It offers improved performance as performance critical operation can be directly executed on the host.
- Overhead also get minimized due to direct access of system resources.
- This technique has been successfully used by Xen (an open-source hypervisor) for providing virtualization solutions for Linux-based operating systems specifically ported to run on Xen hypervisors.

Disadvantages of Para Virtualization

- Admin must modify the guest OS which makes it less portable and provide limited OS options.
- Partial isolation also cause security risk.
- There is no reliable way to predict performance gains.
- Significant support and maintenance issues may arise since the production environment requires complex guest kernel modifications.

Hardware Assisted Virtualization (HAV)

- Hardware Assisted Virtualization (HAV) use features provided by the hardware to improve performance of the simulated virtual machines.
- Support of virtualization is embedded in the hardware.
- It is a kind of Full-Virtualization, where VMs run in isolation without altering the state of another VM.

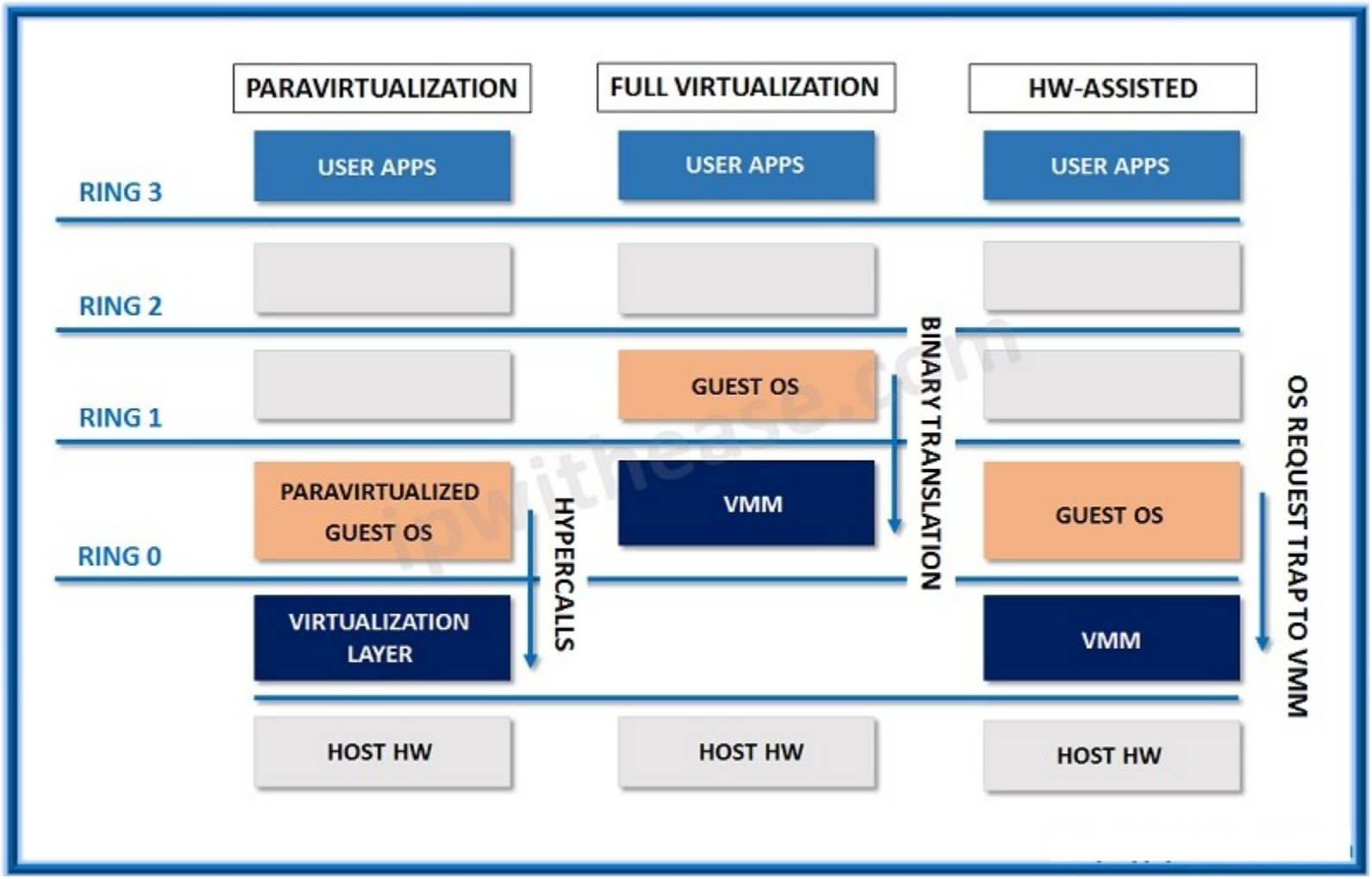
Hardware Assisted Virtualization (HAV)

- Hardware vendors have developed the processors supporting the virtualization through the hardware extension.
- VT-x and AMD-v are the examples of virtualization enabled processors provided by Intel and AMD respectively.
- After the introduction of Intel VT-x and AMD-V the hypervisor could rely on the hardware, thereby cutting down on the time required to virtualize and simulate.
- This improves efficiency and performance of the virtual machines.

(HAV)-Adv. & Dis.

- This approach eliminates the overhead of binary translation in full virtualization and Hypercalls in paravirtualization.
- A common drawback of this technique was that not all hardware supported HAV.
- Hence, the widespread deployment of this was limited in presence of legacy or old hardware.
- But today, almost all new processors supported HAV, hence the hypervisors could make use of it at a larger scale.

Full vs Para vs HAV



Full vs Para vs HAV

Full virtualization vs. paravirtualization

Full virtualization	Paravirtualization
The VM enables execution of instructions from unmodified OSes	The VM uses an API to port an OS to the hypervisor
OSes require no modification	OSes require modifications
Provides complete logical isolation	Not fully isolated; poses security risks in the API
Highly portable and compatible	Less portable and compatible
Uses binary translation and direct calls	Uses hypercalls through an API

Partial Virtualization

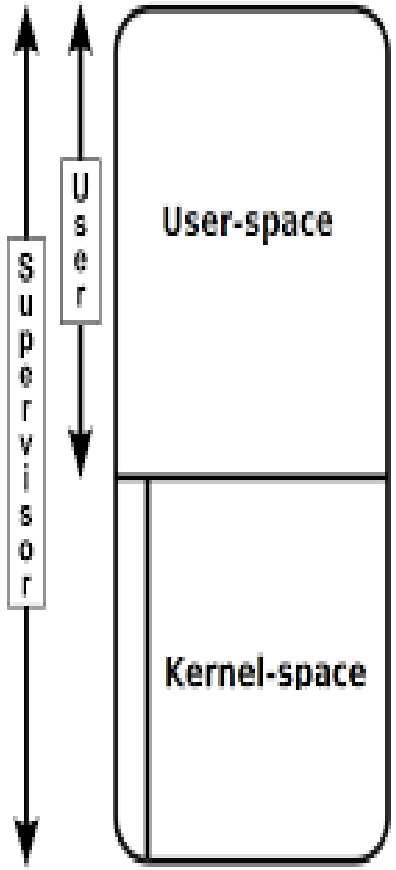
- When entire operating systems cannot run in the virtual machine, but some or many applications can, it is known as Partial Virtualization.
- Partial virtualization provides a partial emulation of the underlying hardware.
- Partial virtualization allows many applications to run transparently but not all the features of the operating system can be supported as happens with full virtualization.
- This type of virtualization is far easier to execute than full virtualization.
- This is very successful when computer resources are shared amongst multiple users.

Partial Virtualization

- **Address space virtualization** is a key form of partial virtualization.
- Here, each virtual machine consists of an independent address space, but they still share the same hardware resources (disk, processor, and network).
- This needs address relocation hardware and is present in the most practical examples of partial virtualization.
- Partial virtualization was implemented on the experimental IBM M44/44X.

Operating System Level Virtualization

- **User space** is that portion of system memory in which user processes run. These processes can't access Kernal space directly. Some part of Kernal space can be accessed via system calls.
- **Kernel space** is that area of memory where the kernel (i.e., the core of the operating system) runs and provides its services.
  - It's something that the user is not allowed to interfere with.
  - Role of Kernal space is to manage applications/ processes running in user space.
  - If a process running in user space performs a system call, a software interrupt is sent to kernal which then dispatches an appropriate interrupt handler.

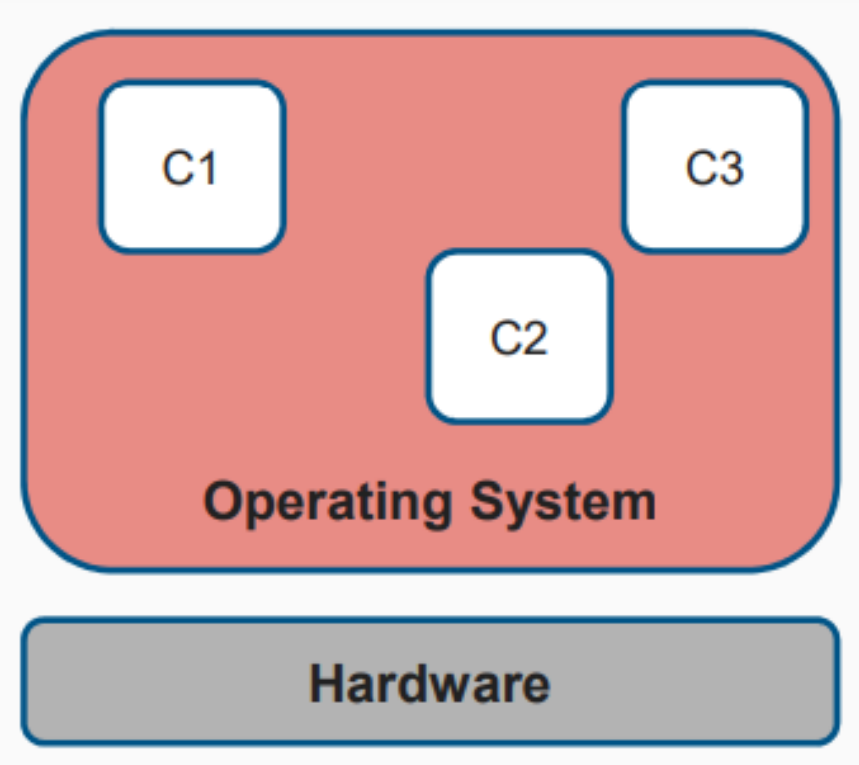


Applications run in User mode with limited privileges. Each Application has its own virtual memory map.

Kernel runs in Supervisor mode and has access to the User section(s) as well. The Kernels memory map is fixed.

Operating System Level Virtualization

- **Containers** are Multiple isolated user-space instances (instead of one) on a single kernel of the operating system.

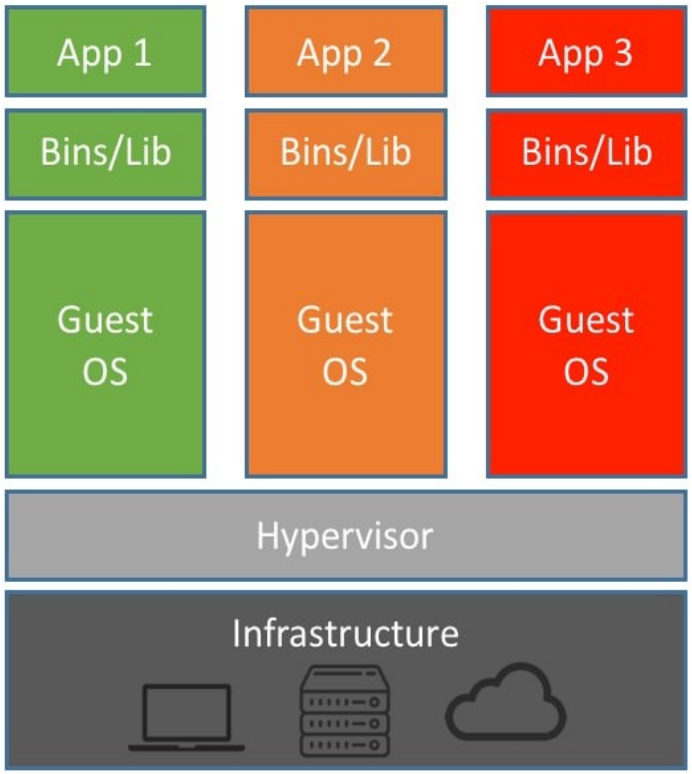


Operating System Level Virtualization

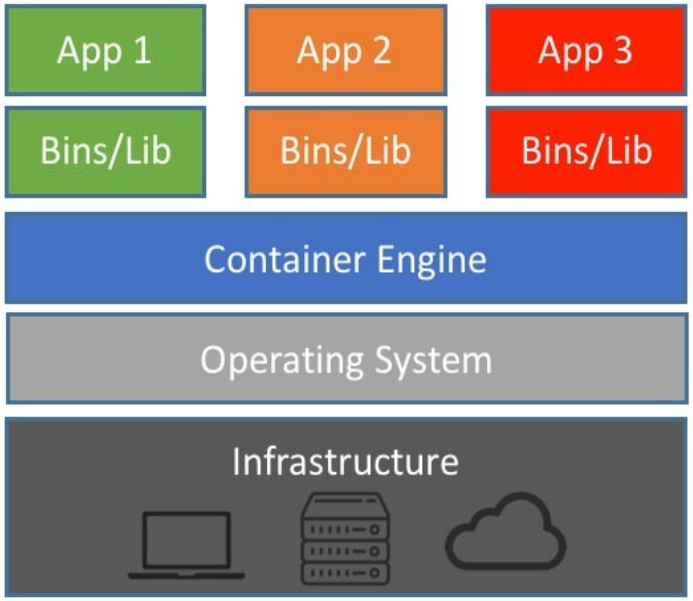
- How Containers are different from VMs?
  - Containers provide a way to virtualize an OS so that multiple user space instances can run on a single OS instance. With VMs, the hardware is being virtualized to run multiple OS instances.
  - VM's are heterogeneous in nature as they may have different guest OS while containers are homogeneous in nature as all runs under a single OS.
  - Due to light weight, Containers are fast agile and portable in nature while lot of overhead are there in VM due to its heavy weight.

Operating System Level Virtualization

- How Containers are different from VMs?



Machine Virtualization



Containers

Operating System Level Virtualization

- In OS level virtualization, virtualization technology works on the OS layer.
- Kernel of the OS allows more than one isolated user space instances to exist. Such instances are called container.
- In other words, OS kernel will run a single OS and provide the operating system's functionality to replicate on each of the isolated partitions.
- In OS level virtualization, a user installs the virtualization software called container engine on the operating system of his system like any other program and utilizes this application to operate and generate various containers. The operating system over which virtualization software is installed is called the host operating system.
- In the OS level virtualization, nothing is required to be pre-installed on the local storage device. Everything runs from the network using a kind of virtual disk.

Operating System Level Virtualization

- Advantages: As all containers use the same OS underneath, it leads to the following benefits:
  - They use less CPU and Memory for running the same workloads as compared to virtual machines.
  - The time to initiate a container is smaller as compared to VMs.
  - On a machine, we can have many more user containers (~100) as compared to a small number of user VMs (~10)

Operating System Level Virtualization

- Limitations:
  - Less Flexibility: Cannot host a guest OS different from the host, or a different guest kernel
  - Security: As containers run on top of same OS, security issues exist from adjacent containers

Operating System Level Virtualization

- The client will be connected via the network to the virtual disk & will boot the OS installed on virtual disk. Two types of virtual disks are there for implementation.
  - Private Virtual Disk: is used by one client only like that of a local hard disk. Users can save information on the virtual disk based on the rights assigned. So as the client restart the system, the settings are retained just like working with physical local hard disk.
  - Shared/Common Virtual Disk: It is used by multiple clients at the same time. The changes are saved in a special cache & these caches gets cleaned as the user restarts or shutdowns the system. In other words, when a client is booting up, it will use the default configuration available on the virtual disk.

Operating System Level Virtualization

- General Type of OS Virtualization.
  - Linux OS Virtualization
  - Window OS Virtualization

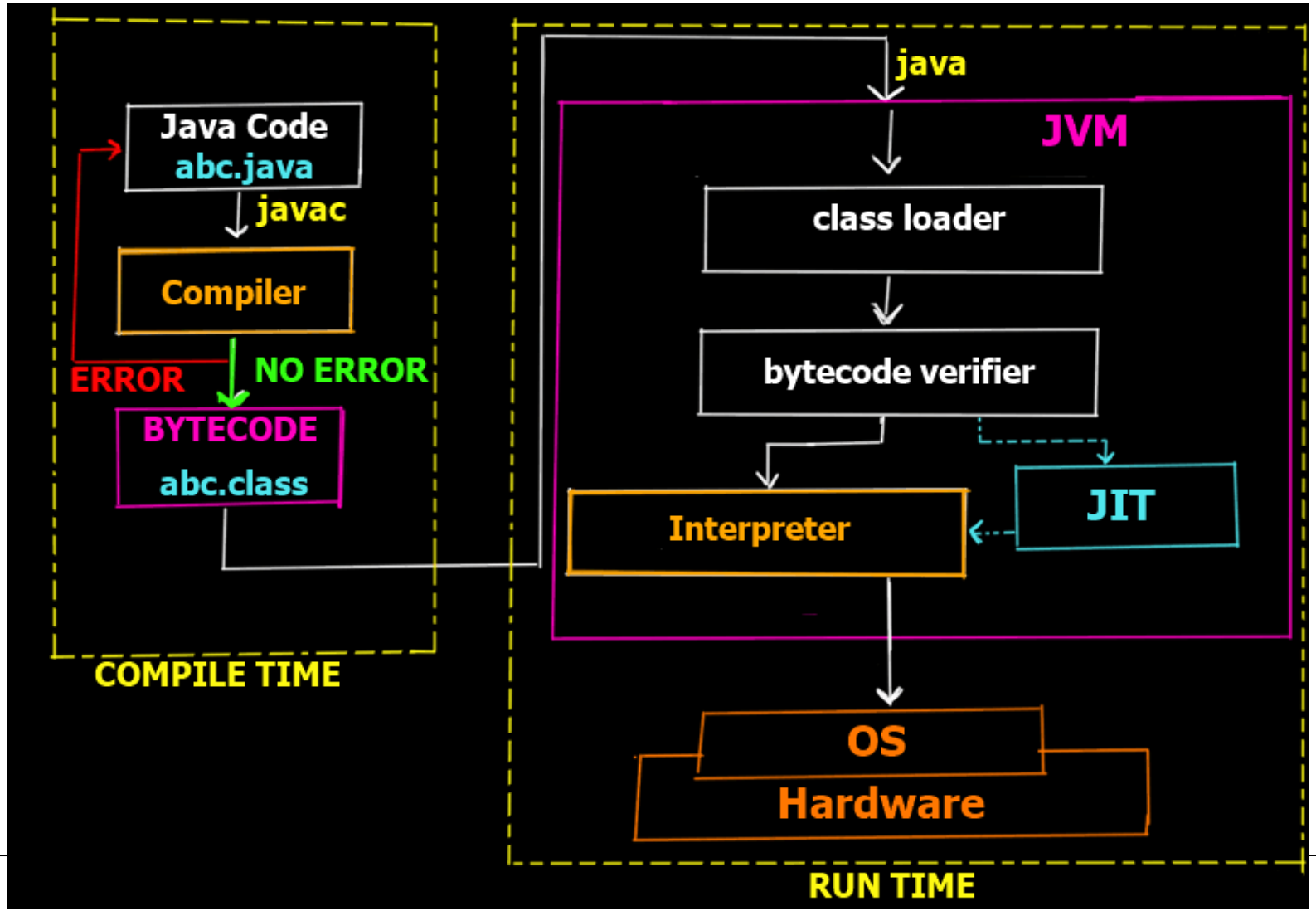
Operating System Level Virtualization

- Examples
  - FreeBSD Jails
  - IBM Logical Partition (LPAR)
  - SolarisZones and Containers
  - Parallels Virtuozzo Containers
  - OpenVZ
  - iCore Virtual Accounts
  - Free Virtual Private Server (FreeVPS)

Programming Level Virtualization

- Programming language level virtualization is mostly used for achieving ease of deployment of applications, managed execution, and portability across different platforms and operating systems.
- It consists of a virtual machine executing the byte code of a program, which is the result of the compilation process.
- Compilers implemented using this technology produce a binary format representing the machine code for an abstract architecture.
- The main advantage of programming-level virtual machines, also called process virtual machines, is the ability of providing a uniform execution environment across different platforms.
- Programs compiled into byte code can be executed on any operating system and platform for which a virtual machine able to execute that code has been provided.
- As an example, Java provide an infrastructure for pluggable security policies and code access security frameworks.

Programming Level Virtualization



Application-Level Virtualization

- Virtualization of an application involves encapsulation of an application into a self-contained distributable package which is isolated from the underlying OS and other applications.
- This allow applications to be run in an environment which do not natively support all the features required by such applications.
- In this scenario, applications are not installed in the expected run time environment but run as if they were.
- Such emulation is performed by a thin layer—a program or an operating system component—that oversees executing application.

Application-Level Virtualization-Adv.

- No need to install the application on the user system.
- Incompatible applications can be made to run
- Easy disaster recovery
- Virtual machine manager is much lighter since it provides a partial emulation of the run time environment if compared to hardware or OS virtualization.
- Application virtualization is a good solution in the case of missing libraries in the host operating system: in this case a replacement library can be linked with the application or library calls can be remapped to existing functions available in the host system.

Application-Level Virtualization-Ex.

- One of the most popular solution implementing application virtualization is Wine, which is a software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform.

Desktop Virtualization

- Desktop virtualization is a software-based technology that enables the creation of a virtualized desktop environment on remote servers or on premise.
- Virtualization removes the use of bulky physical desktops in the local premises, resulting in reduced IT hassles and expenses.

Desktop Virtualization

- Desktop virtualization is like **application virtualization**, but the apps are now replaced with whole desktop environments.
- It refers to the technique of abstracting the desktop environment available on a personal computer/server/VM in order to provide access to it by using a client server approach.
- The end user has an option to access their virtual desktops hosted on another machine using remote desktop client.

Desktop Virtualization

- Desktop virtualization (DV) provides the same outcome as hardware virtualization **but serves a different purpose**.
- Desktop virtualization is a response to increasing numbers of employees working remotely and from multiple devices, whereas hardware virtualization is an answer for companies that need to diversify workloads and maximize hardware efficiency.

How Desktop Virtualization Work?

- The concept of DV works around client server model.
- An end user using a thin client can access their respective desktop environment on the centralized server.
- The centralized server is designed to host desktop environments for multiple end users.
- DV is achieved by using a set of h/w & s/w technologies.
- The desktop environment can be provisioned or deprovisioned on demand.

Different Tech. to Implement Desktop Virtualization

- Virtual Desktop Infrastructure
- Desktop as a service
- Remote desktop service

Different Tech. to Implement Desktop Virtualization

- Virtual Desktop Infrastructure:
  - VDI is a technology that creates a virtualized environment by segmenting the servers into different virtual machine.
  - These VM acts as virtual desktop instances with a dedicated set of applications and a client OS. The user can access these desktop on their personal devices.

Different Tech. to Implement Desktop Virtualization

- Desktop as a service:
  - DaaS is an offering in the VDI space offered by a third party where the virtual desktop are created on their cloud servers.
  - This will help in the reduction of on premise infra and cost.
  - Third party also handles all installation and updating of virtual desktops.

Different Tech. to Implement Desktop Virtualization

- Remote Desktop Service:
  - Unlike VDI, where each user gets a dedicated VM, machine running an OS, in RDS user works on desktop sessions on a shared virtual machine.
  - Users are not isolated and multiple users working on a shared VM.

Types of Desktop Virtualization

- Based on the location where desktop is, types of desktop virtualization are as follows:
  - Centralized Server Method
  - Shared Load Method
  - Client Hosted Method

Types of Desktop Virtualization

- Centralized Server Method:
  - In this method, desktop virtual machines are kept in a central server.
  - The end user equipment is just a thin client presenting a remote display to end user.
  - The hosted virtual desktops may be provisioned on demand, be maintained in current configuration for longer runs, or be saved and archived in a repository on the server.
  - Good network connection is major requirement for this kind of model.

Types of Desktop Virtualization

- Shared Load Method:
  - It is possible for end user client to take a copy of the hosted virtual desktop and run it locally.
  - This enables the end user to work even when network connection is not up to the mark or do heavy graphic processing on desktop requiring heavy graphics processing without any delay.
  - The downloaded virtual desktop can be uploaded back on server with all updates on periodic basis.
  - The end user machine must be capable enough to run the downloaded desktop instance.
  - Good network connection is required while moving desktop image either way.

Types of Desktop Virtualization

- Client hosted Method:
  - Virtual instances resides on end user equipment.
  - Access to the server is only required to manage desktop instances and keep the associated data on the server
  - End user equipment with capable hardware is pre-requisite of this method.

Benefits of Desktop Virtualization

- Better power management:
  - The desktop virtual instances located on the central server can be easily powered down when not in use by the end user. This kind of monitoring is generally not easily possible in case of physical desktops which are left running even when not in use.
- Easy to take backup:
  - Easy snapshots enable easier backup of desktop environments. These snapshots can also be used to spawn new desktop instances for new users without going for a long installation and configuration cycles.
- Application Licensing:
  - Since the desktops are now centralized and located on a remote server, it is easy to keep the desktops up to-date using advanced virtual machine monitoring and management software.

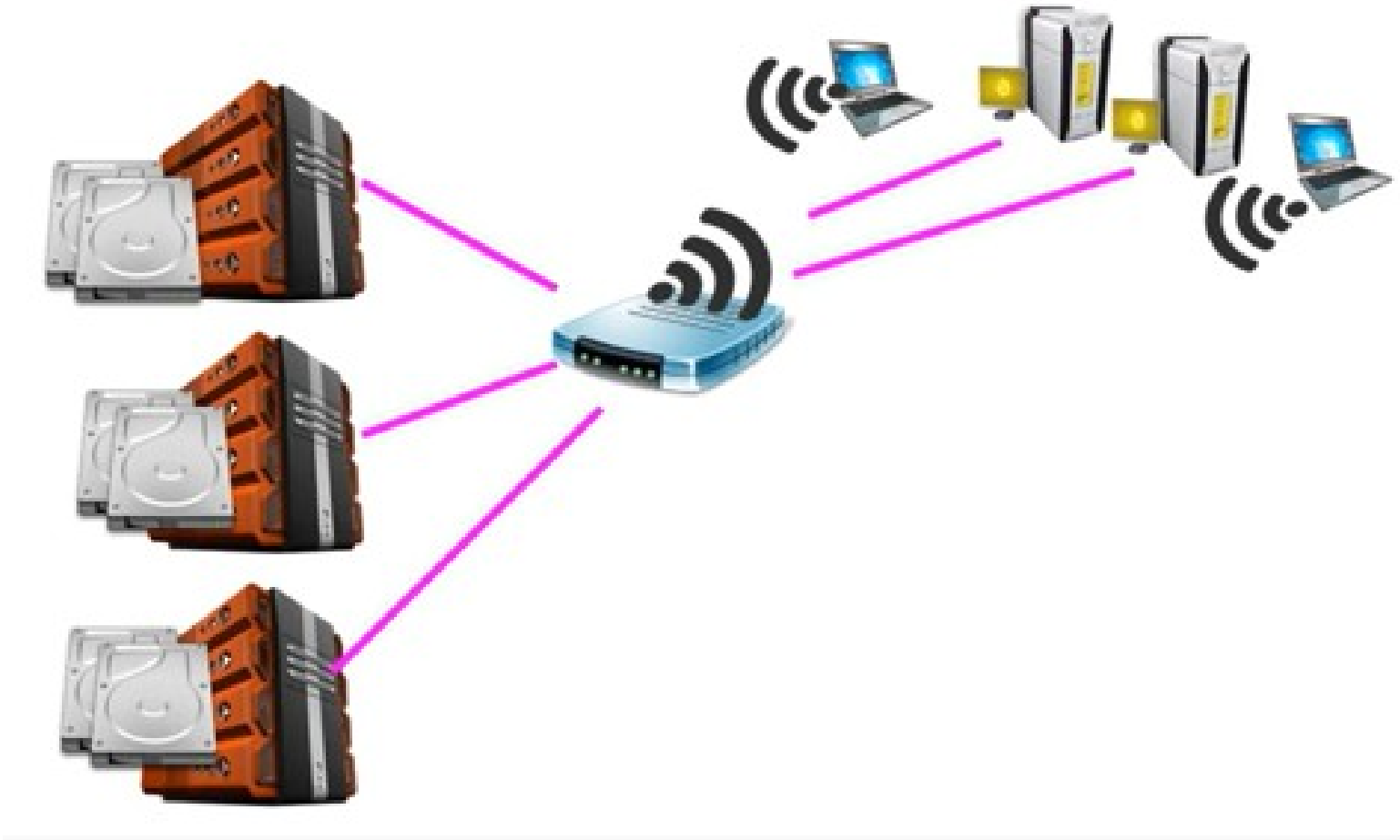
Benefits of Desktop Virtualization

- Ease of application access:
  - Hosted desktop instances can be accessed anywhere anytime from an end-user device like a smart phone or a thin client. This improves the mobility of the workforce.
- Sharing of performance-intensive applications:
  - Access to performance intensive applications which were traditionally hosted on high-end machines were constrained by their location and the number of simultaneous users. With the centralization of these desktop machines, the applications can be hosted on a central server, and can be accessed over a network anywhere anytime by any number of users.
- Desktop virtualization offers the same benefits as the Server virtualization from the cost, consolidation and utilization perspective.

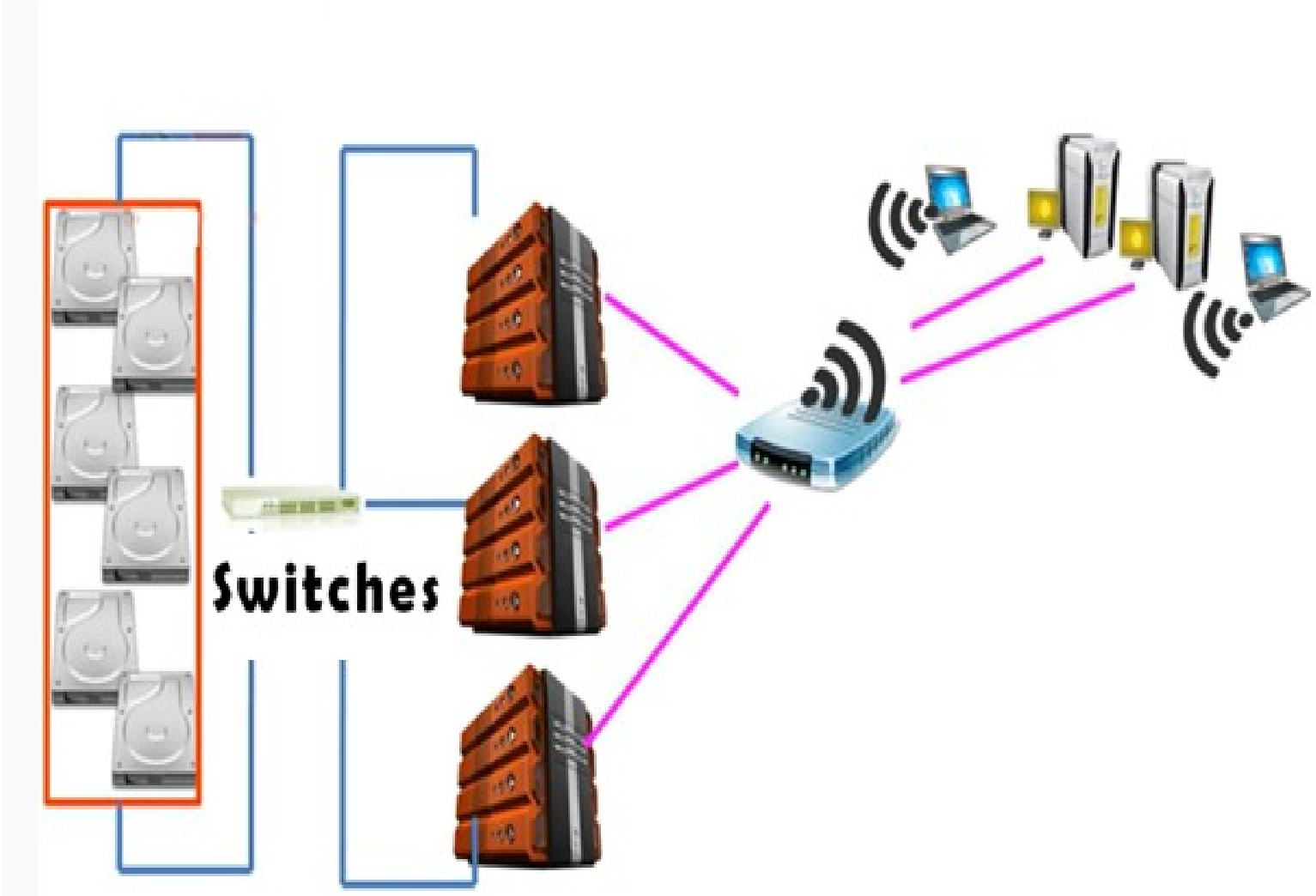
Constraints of Desktop Virtualization

- Network Bandwidth:
  - An important resource to enable desktop virtualization using VDI is the availability of a reasonably fast network. The network infrastructure has to be sized based on the number of simultaneous clients accessing the central desktop server. Any bottleneck in the network will have an impact on large number of users. This limits the widespread adoption of VDI especially when the end-users are geographically dispersed over many locations.
- Network security:
  - Security of the network connecting the end users with the desktop server is important as any kind of SPI (sensitive personal and business information) moving over the network can be easily compromised if required security infrastructure is not in place.
- Graphics intensive application:
  - Certain applications like graphic animations, movie editing, 3d graphics, games require quick response times when it comes to display and input response. These applications are not recommended for VDI.

Storage Area Network (SAN)



Storage Area Network (SAN)



Storage Area Network (SAN)

- A storage area network (SAN) is a dedicated, independent high-speed network that interconnects and delivers shared pools of storage devices to multiple servers.
- Each server can access shared storage as if it were a disc drive directly attached to the server.
- A SAN is typically assembled with cabling, host bus adapters, and SAN switches attached to storage arrays and servers.
- Each switch and storage system on the SAN must be interconnected.
- OS recognizes the SAN in a same manner as it recognize a local hard disc.

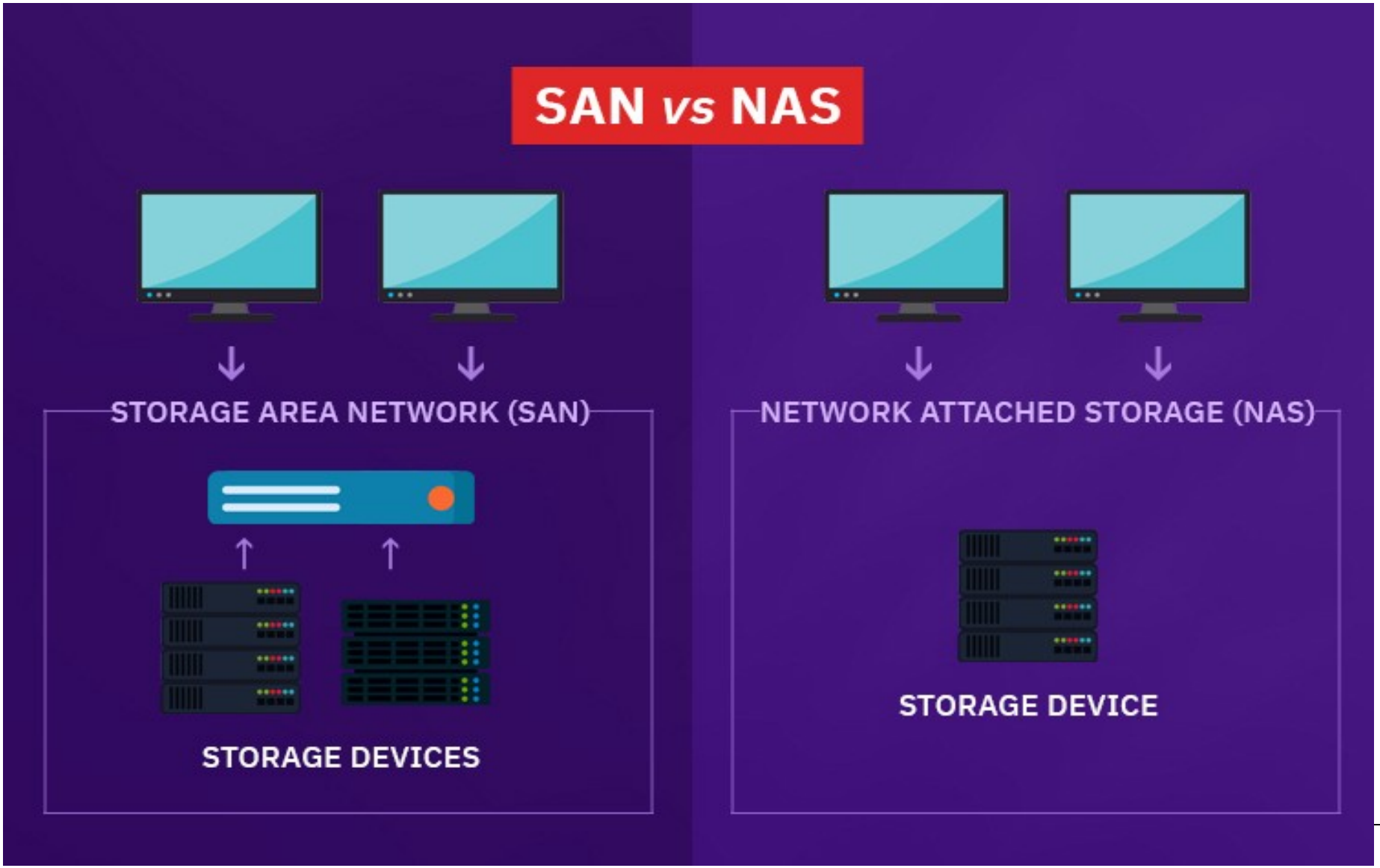
Network-attached storage (NAS)

- A NAS is a storage device that is used for storing data and it does not do anything else besides storing data.
- Typically, NAS is a box that have multiple hard drives in a RAID configuration for redundancy.
- It will also have a network interface card that will directly attach to a switch or router. This enables the access of data stored on the NAS by other devices on the same network.
- If we want to store data in a centralized location where it can be accessed from all devices on the same network, then NAS is used.
- Major issue with NAS is that it has a single point of failure. OS recognize the NAS as a shared drive.

SAN vs NAS

- A SAN and Network-Attached Storage (NAS) are two different types of shared networked storage solutions.
- NAS handles **unstructured data**, such as audio, video, websites, text files and Microsoft Office documents while SAN handles **structured data** like data of database
- While a SAN is a local network composed of **multiple devices**, NAS is a **single storage device** that connects to a local area network (LAN).
- Major issue with NAS is that it has a **single point of failure** which is not there in SAN due to usage of multiple devices.
- OS recognize the NAS as a **shared drive** while it recognizes SAN as a **local disc**.
- Implementation of SAN is much **costlier** than NAS as it uses fiber channels for connectivity.
- Transfer **speed is more** in SAN due to usage of fiber channels.

SAN vs NAS


SAN vs NAS
STORAGE AREA NETWORK (SAN)
STORAGE DEVICES
NETWORK ATTACHED STORAGE (NAS)
STORAGE DEVICE

Why Storage Virtualization

- In environment where multiple storage devices are there then storage management becomes difficult and time consuming.
- Storage virtualization addresses this problem.
- Storage virtualizing separates the storage management software from the underlying hardware infrastructure in order to provide more flexibility and scalable pool of storage resources.

Storage Virtualization

- Storage virtualization is a type of hardware virtualization which involves the pooling of multiple physical storage array from SAN and making them appear as a single virtual storage device.
- Basically, its consolidating all storage and putting a single front end on it for simplicity.
- Virtualization software creates a map to dynamically locate data on the fly. This single pool of storage can be used by both traditional architecture servers or virtual environments with VM.

Benefits of Storage Virtualization

- Easier Management:
  - Virtualization disguises the actual complexity of a storage system to help admins perform backups, archiving and recovery more quickly and easily.
- Better storage utilization:
  - Pooling storage capacity across multiple systems makes it easier to efficiently allocate and use the total storage.
- Extended life for older storage systems:
  - Including older storage gear in the virtualized storage pool extends its usefulness by handling archival or less critical data.
- Universal Additional Features:
  - Advanced storage features like tiering, caching and replication can be implemented at the virtualization level, helping to standardize these practices across all storage systems.

Types of Storage Virtualization

- **File storage and block storage** are two of the most common, and popular, ways to store and access data in on-premises, virtual, and cloud servers.
- Both storage types have their pros and cons and are therefore best suited for different requirements, projects, and use-cases.

Types of Storage Virtualization

- Block Level Storage Virtualization
  - Block level storage, or block storage, is storage used for structured data and is commonly deployed in Storage Area Network (SAN) systems.
  - It uses Internet Small Computer Systems Interface (iSCSI) and Fiber Channel (FC) protocols.
  - Block storage uses blocks, which are a set sequence of bytes, to store structured workloads. Each block is assigned a unique hash value which functions as an address.
  - In block storage, the data is stored without any metadata e.g., data format, type, ownership, etc.

Types of Storage Virtualization

- Block Based Storage Virtualization
  - In a block level storage device, raw storage volumes are created, and then the server-based operating system connects to these volumes and uses them as individual hard drives.
  - This makes block level storage usable for almost any kind of application, including file storage, database storage, virtual machine file system (VMFS) volumes, and more.
  - You can place any kind of file system on block level storage. So, if you're running Windows, your volumes will be formatted with NTFS; VMware servers will use VMFS..

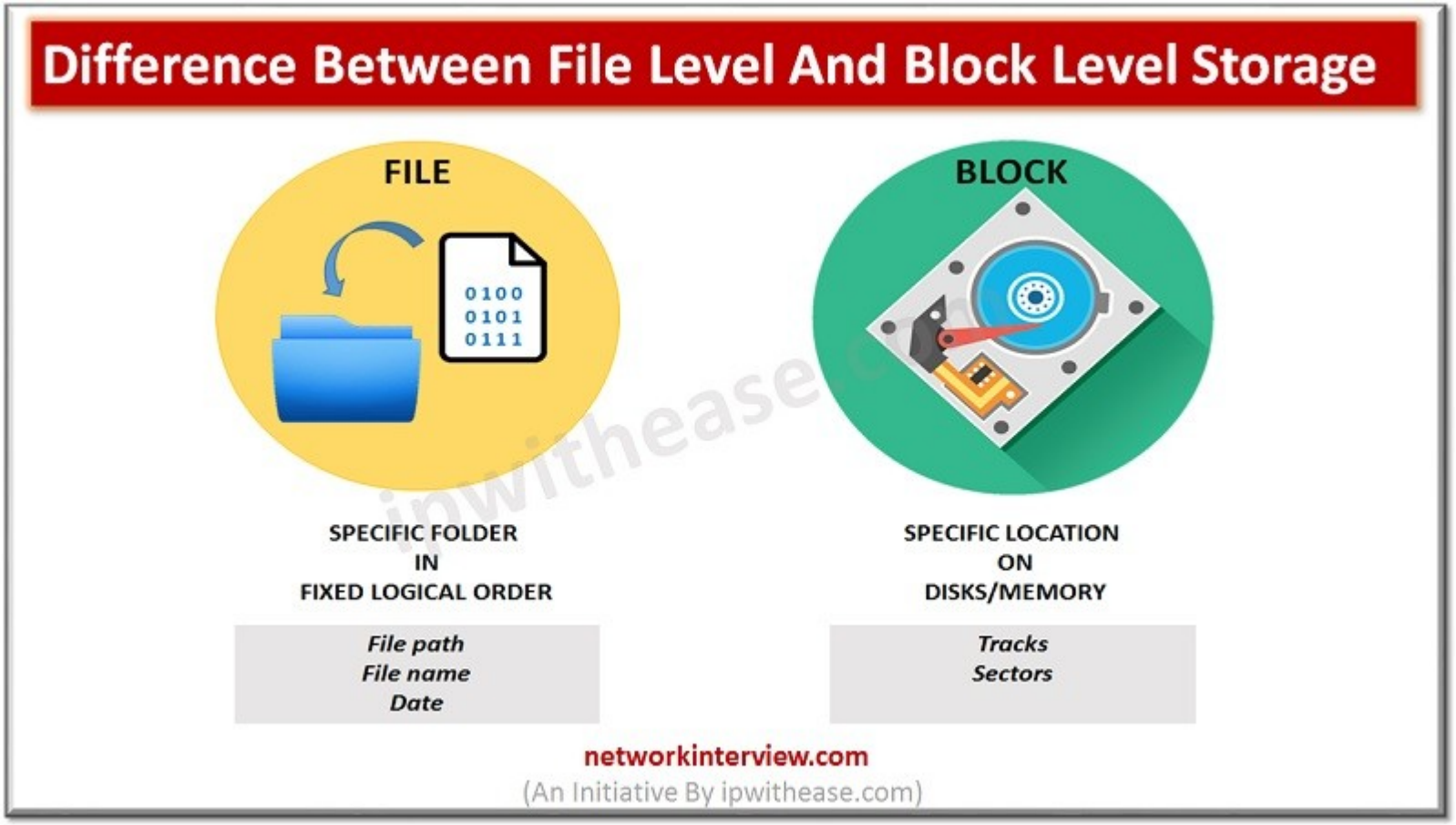
Types of Storage Virtualization

- File Based Storage Virtualization
  - File level storage is used for unstructured data and is commonly deployed in Network Attached Storage (NAS) systems.
  - It uses Network File System (NFS) for Linux, and Common Internet File System (CIFS) or Server Message Block (SMB) protocols for Windows.
  - File storage, as opposed to block storage, stores data in a hierarchical architecture; as such that the data and its metadata are stored as is – in the form of files and folders. Consequently, the stored data appears in a similar fashion to both systems; the one writing it and the one reading it.
  - In terms of cost, file storage systems are typically less costly than block storage.
  - Furthermore, file storage is dually scalable; which means it can scale up and scale out.

Types of Storage Virtualization

- File Based Storage Virtualization
  - File-level storage is a type of storage that has a file system installed directly onto it where the storage volumes appear as a hierarchy of files to the server, rather than blocks.
  - This is different from block type storage, which doesn't have a default file system and needs to have an administrator create one for non-administrator users to navigate and find data.
  - One benefit of using file storage is that it is easier to use.

Types of Storage Virtualization



Types of Storage Virtualization

BASIS	FILE LEVEL STORAGE	BLOCK LEVEL STORAGE
Deployed for	Network attached storage devices	Storage Area Networks
Storage Protocols	NFS, SMB or CIFS	Fibre Channel, iSCSI and FCoE (Fibre Channel over Ethernet)
Kind of storage system	Centralized file storage	Blocks are created to store files which are utilized as hard drives by the server operating system
File and Folder Access	Can be accessed and managed by this storage system	Not able to manage or control smaller blocks of storage
Implementation	More simple to implement	Complex in usage
Flexibility	Less flexible	More flexible
File sharing	Access permissions can be given to the users for sharing the files	For sharing files, an operating system is to be separately installed and attached to the block
Cost	Less expensive to manage	More expensive to manage
Application	A suitable option when only an empty space is required to dump the files	Larger enterprises use it to store data in the form of blocks and their flexibility
Uses	Used for mass storage of files and VMware	It has more uses as it can store files, databases, VMware.
Server boot	Cannot be done	Server reboots can be done if right kind of devices are used

Server Virtualization

- Server virtualization is the process of dividing a physical server into multiple unique and isolated virtual servers by means of a software application. Each virtual server can run its own operating systems independently.
- Server virtualization refers to a logical partitioning of server components to enable multiple logical servers to run on a single physical server.

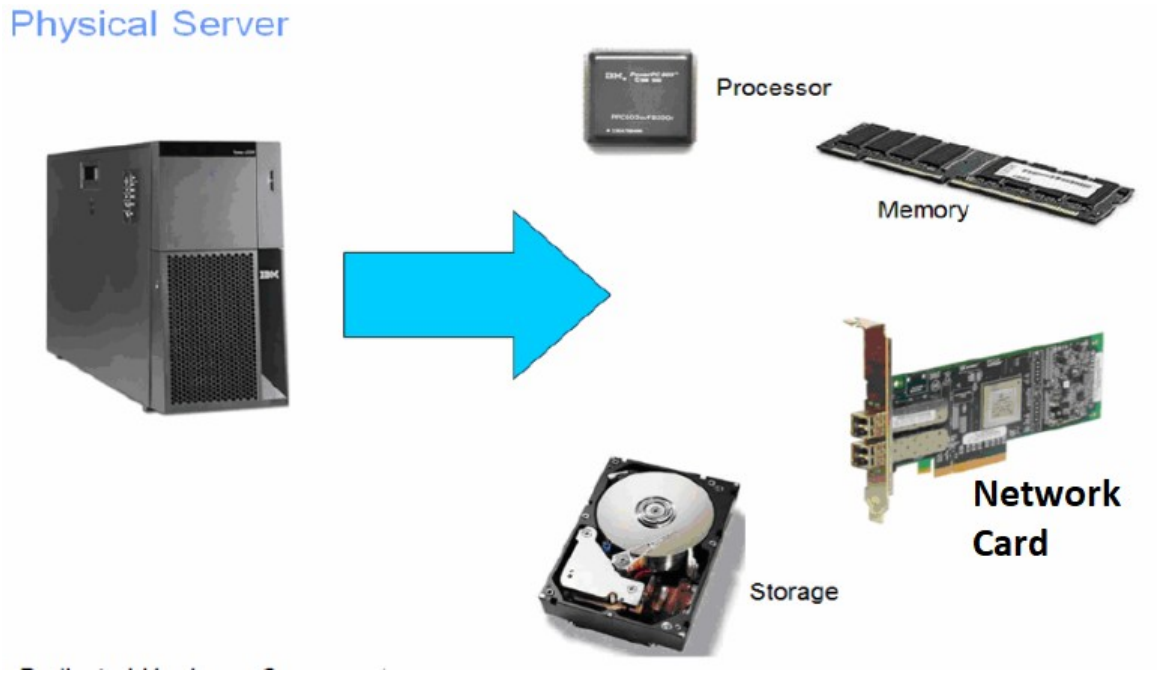
- Logical server may be of different architecture than the underlying hardware. E. g. when an application requires to be run on the ARM architecture OS on a X86 physical server, the ARM platform is emulated in software.

Advantages of Server Virtualization

- A logical server runs in complete isolation from another logical server.
- Strict isolation also ensures that the failure of a logical server will not impact the operation of another logical server. This increases the availability and average utilization of the system.
- Manageability of the system also becomes easy.
- It is possible to save and restore logical servers in seconds using snapshots, thus reducing the provisioning time involved and increases the availability.
- Migration of logical server also helps in load balancing and ensure the satisfaction of SLA.
- Based on the demand at run time, new logical servers can be added or removed (scalability).
- As number of hardware deployed are less, so it requires less power, cooling facility and space.

Server Virtualization

- A physical server is typically composed of four major physical components

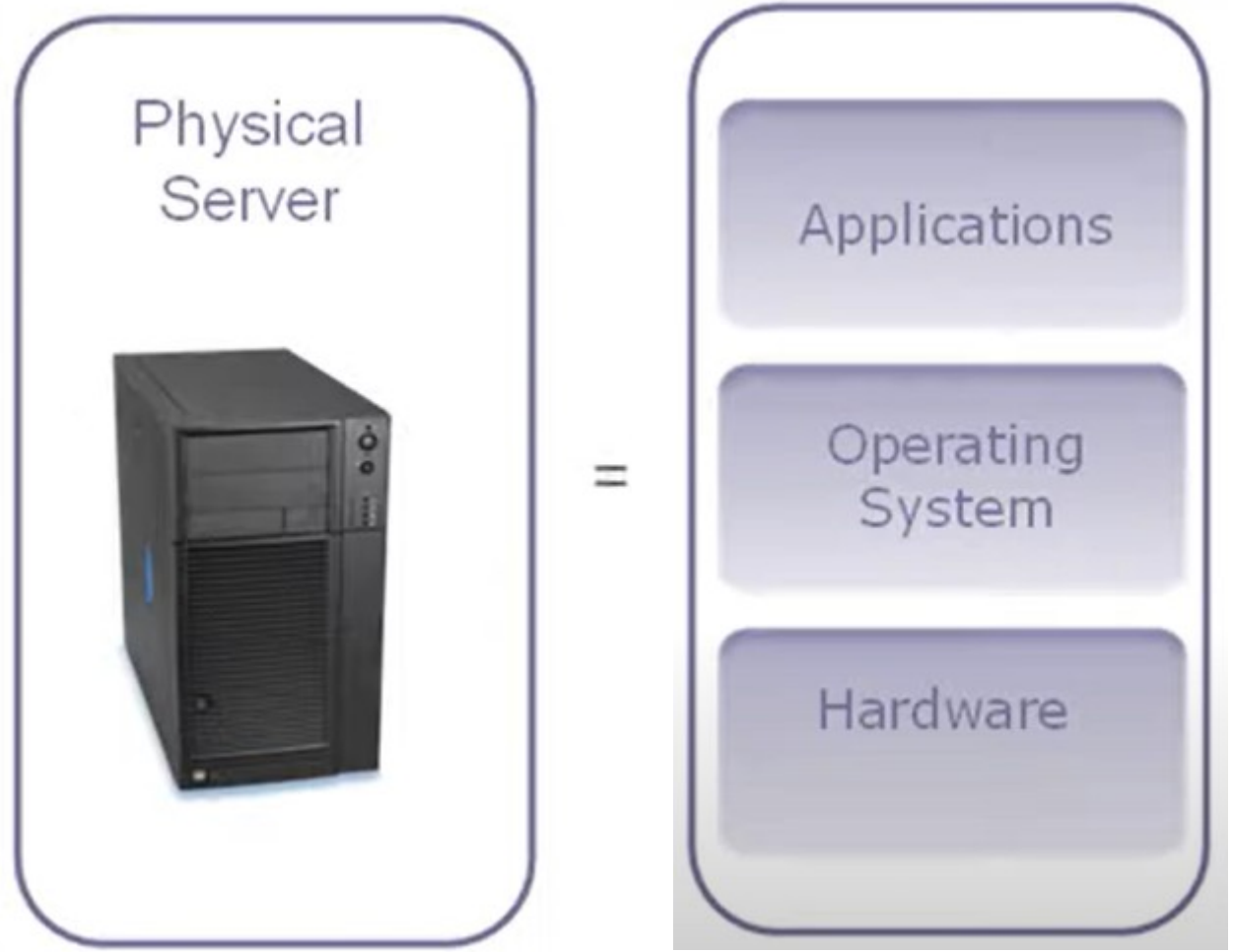


Server Virtualization

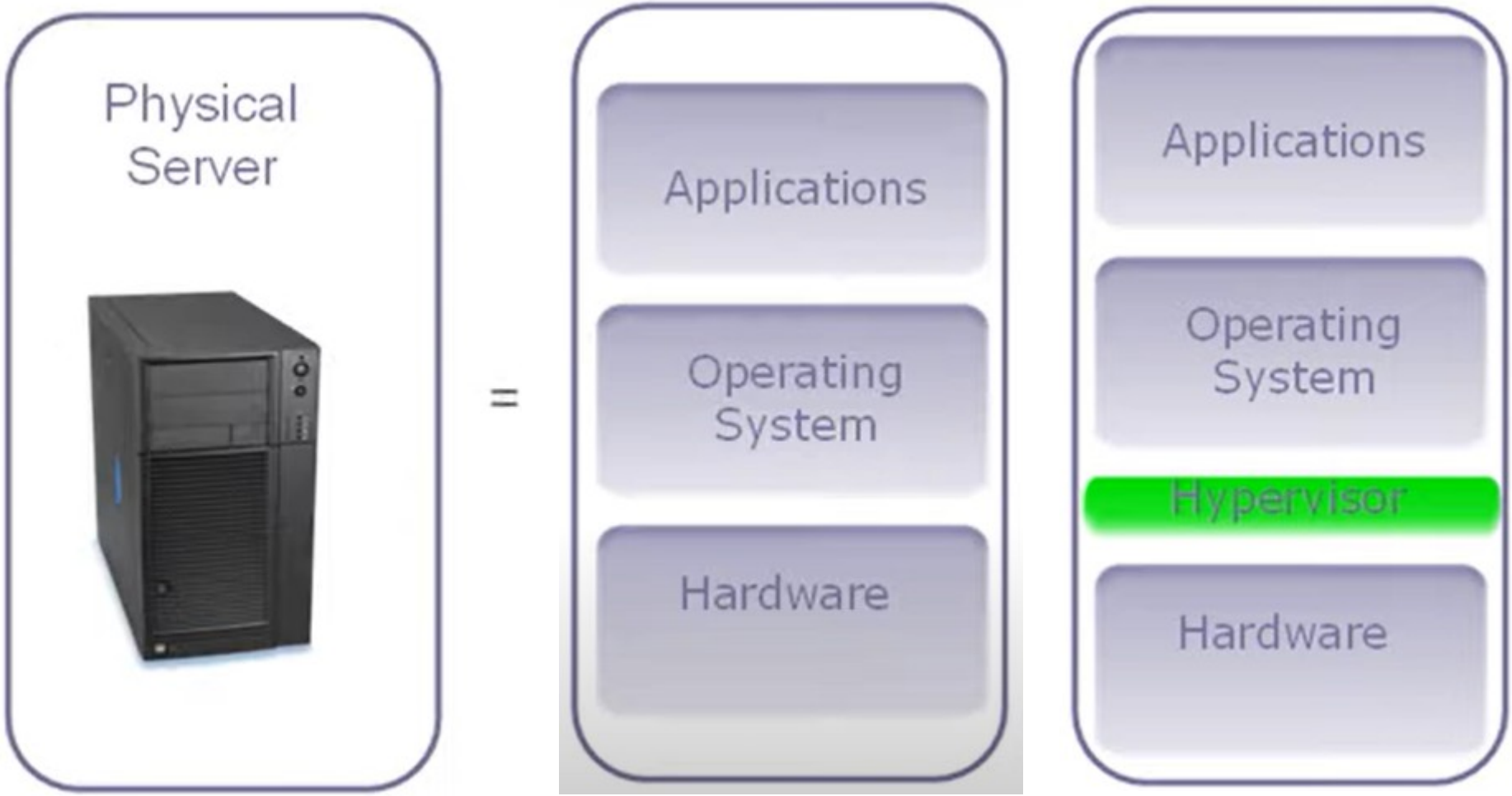
Physical
Server

=

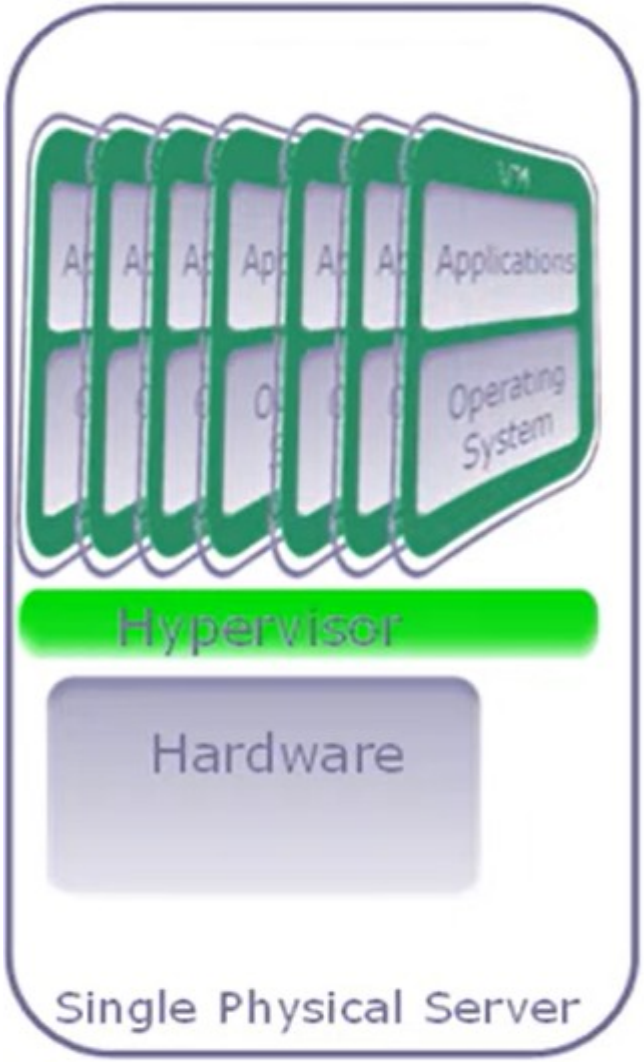
Server Virtualization



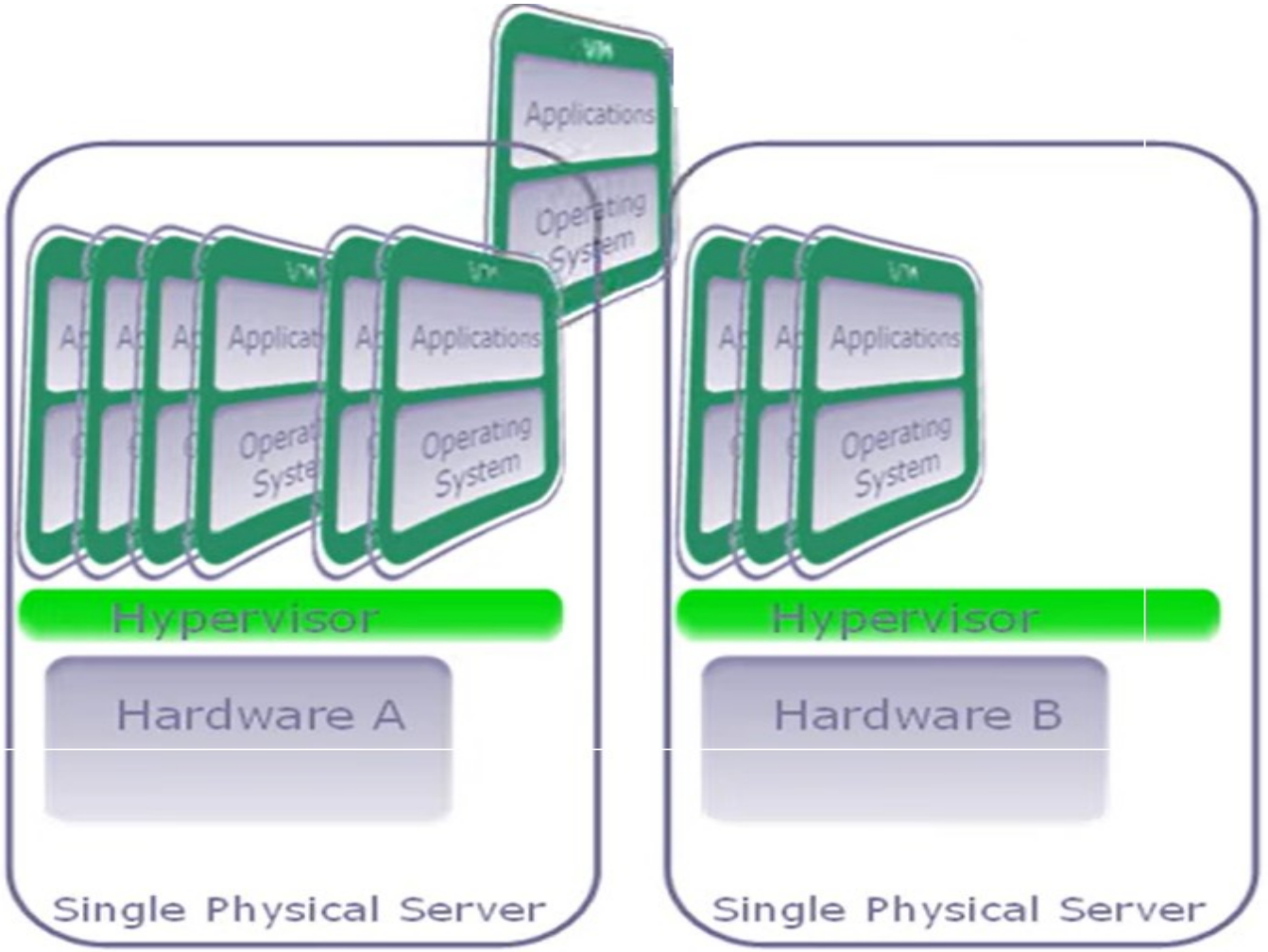
Server Virtualization



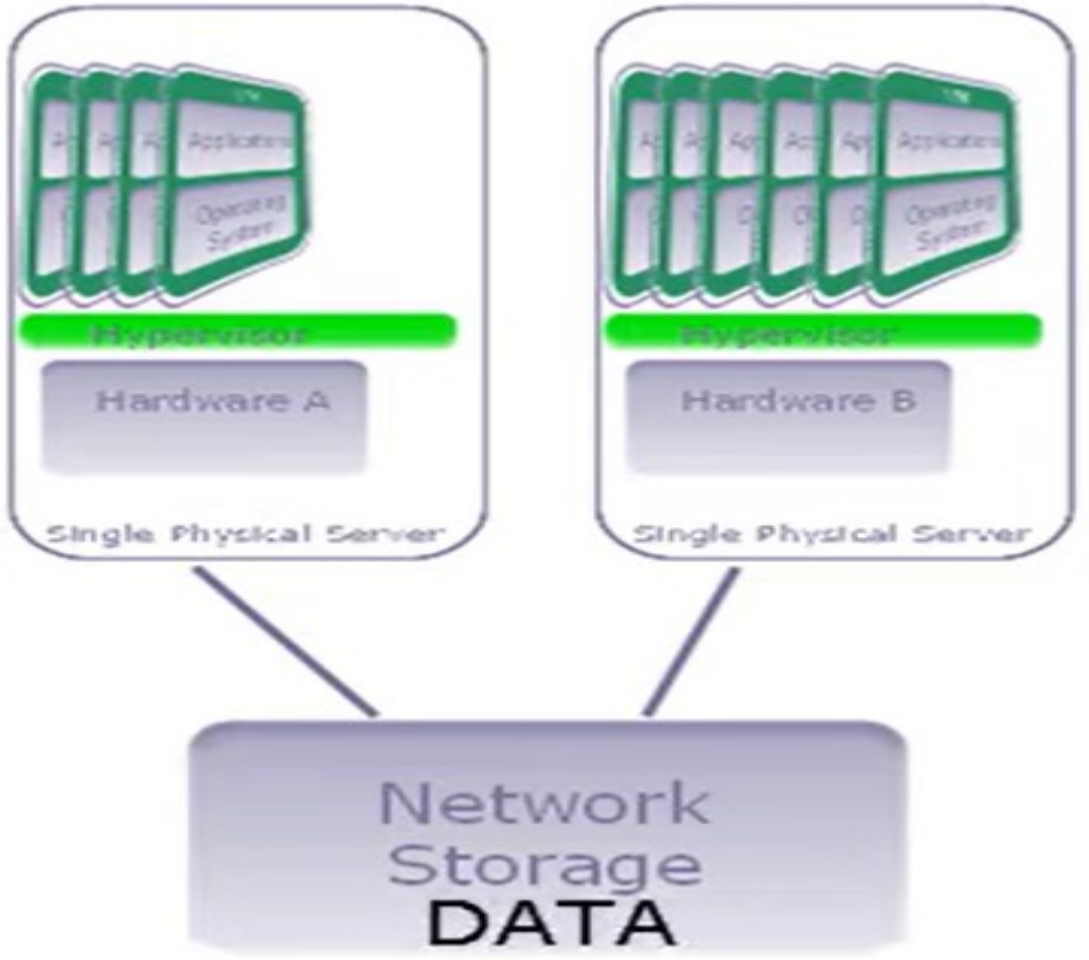
Server Virtualization



Server Virtualization



Server Virtualization

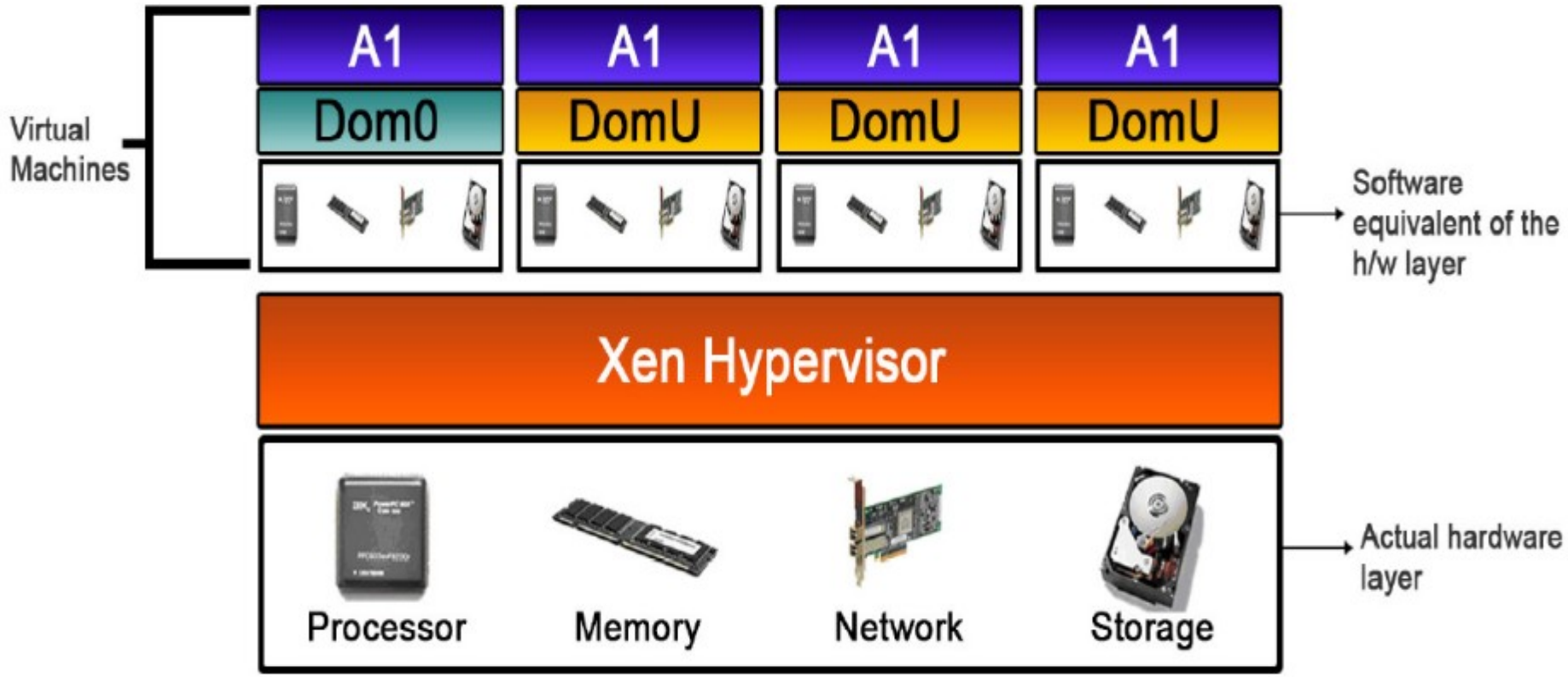


Anatomy of Server Virtualization: Xen Server Virtualization

- It is an open-source virtualization product.
- It was started as a project at university of Cambridge and later acquired by company Citrix.
- XEN provides bare metal server virtualization which is Type 1.
- It allows multiple guest OS to run on virtual machines managed by hypervisor.
- Hypervisor has complete control of CPU, memory, network and storage and multiplexes theses resources among virtual machines.

Anatomy of Server Virtualization: Xen Server Virtualization

- There are three major layers in a Xen Server:
  - Hardware Layers
  - Hypervisor
  - Domains Layer



Anatomy of Server Virtualization: Xen Server Virtualization

- Hardware Layers:
  - The hardware architectures supported by Xen are x86, x86_64, ia64 and ARM.
  - Xen will generally make use of the virtualization capabilities provided by the hardware.
  - While running on x86 and variants, it typically uses either Intel-VTx or AMD-V virtualization features.

Anatomy of Server Virtualization: Xen Server Virtualization

- Hypervisor Layer:
  - The Xen hypervisor provides the required virtual platform for defining and running the virtual machines.
  - The hypervisor supports two kinds of virtualization – Para and Full Virtualization.
    - In case of Para Virtualization,
      - the Xen hypervisor provides the Hyper-call APIs which the guest machines use to request services from the hypervisor.
      - Para-Virtualization is not a pure-virtualization; rather it is implemented as a set of APIs.
      - The virtual machine OS hosted on Para-Virtualized VM requires modification to use the Hyper-call APIs.
      - Only OSes with explicit support for Para-Virtualization can be run in this mode.
    - Full Virtualization
      - provides full emulation of the underlying platform with help from the hardware.
      - No OS modification is necessary to run on a fully virtualized Xen machine.

Anatomy of Server Virtualization: Xen Server Virtualization

- Domain Layer:
  - A differentiating factor of any virtualization solution is the platform, and the manageability tools it provides to the system administrator.
  - Dom 0
    - Xen Hypervisor boots into a control domain (Dom0), a default OS, typically a version of Linux.
    - The control domain can be seen as the first virtual machine on the physical system.
    - This control domain provides the administrator with a platform to create, manage, store and delete and track virtual machines.
    - The control domain has complete access to the hardware on the physical machine.
  - Dom U
    - However, starting with the second virtual machine, the virtual machines are termed as DomU.
    - The DomU are the actual user defined virtual machines.
    - Using the control domain, the administrator can design the configuration of DomU.
    - A DomU could be a para-virtualized machine or a fully virtualized machine.