Virtualization

Session 5

Virtualization

- **Virtualization** is the creation of a virtual rather than actual version of something, such as an operating system, a server, a storage device or network resources
- One of the fundamental Concepts of Cloud Computing

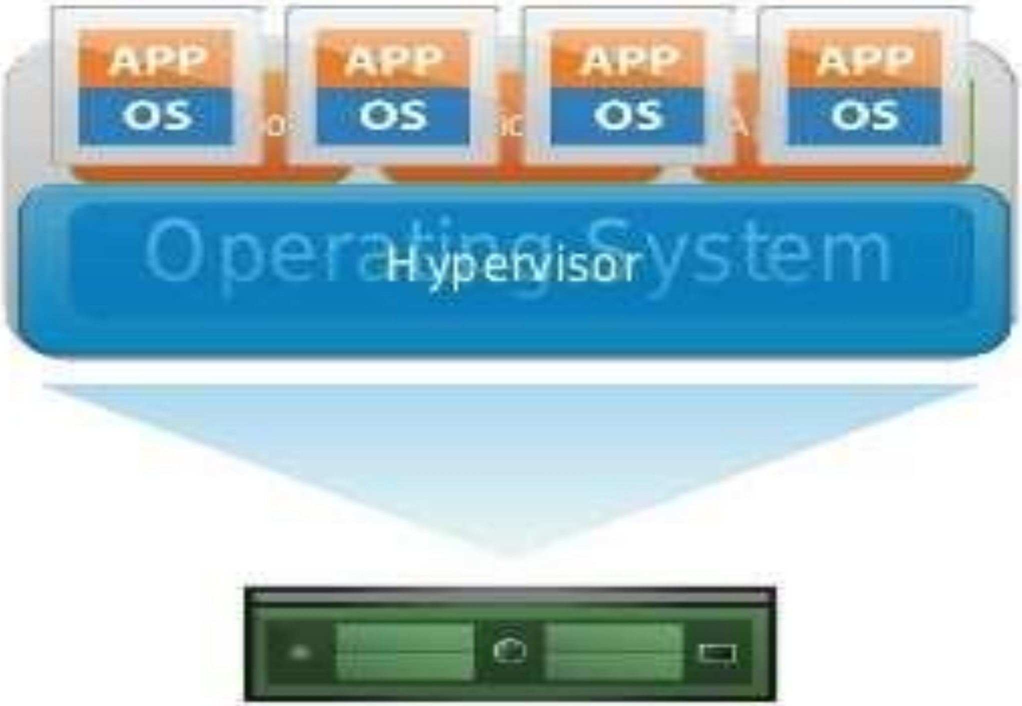
vmware®

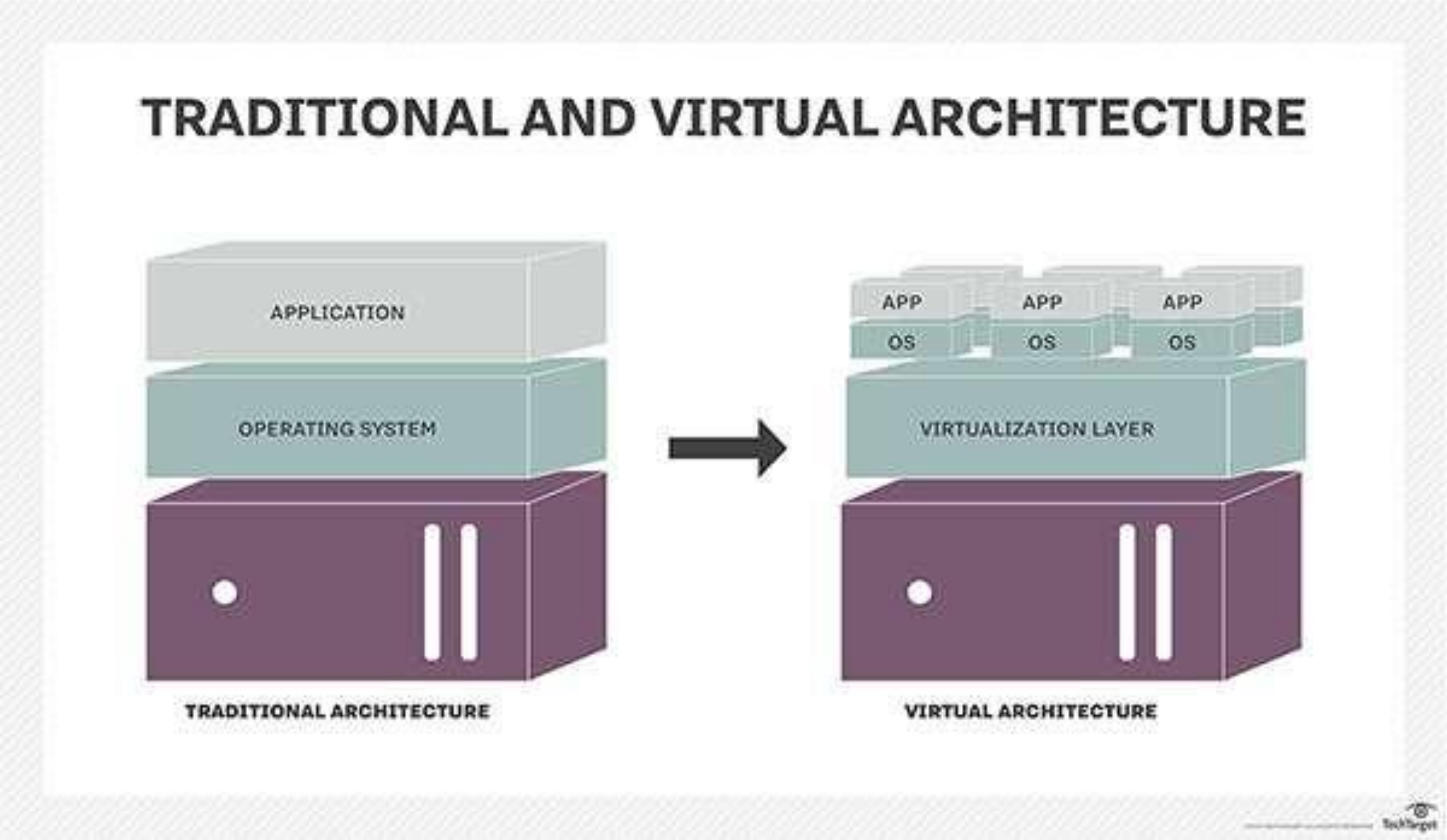
Microsoft Virtual PC

CITRIX® XenServer

What is Virtualization?

- Traditionally the OS and its applications were tightly coupled to the hardware they were installed on
- Virtualization decouples the operating system from physical hardware
- This allows the ability to change hardware without replacing the OS or applications
- Additionally, multiple instances of an OS with independent applications can now run on the same hardware

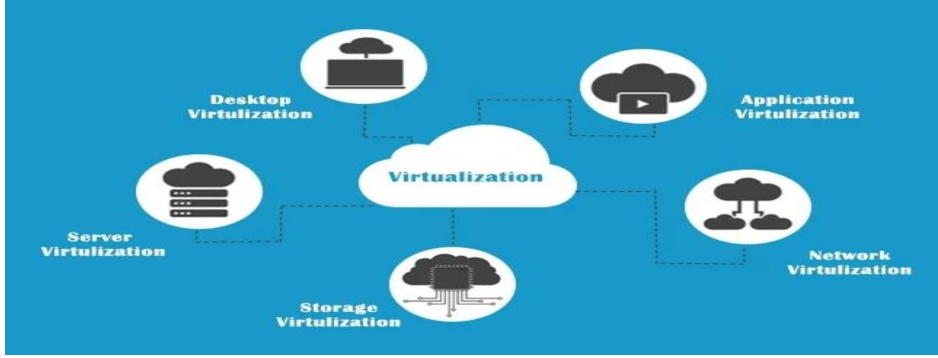



TRADITIONAL AND VIRTUAL ARCHITECTURE
APPLICATION
OPERATING SYSTEM
TRADITIONAL ARCHITECTURE
APP
APP
APP
OS
OS
OS
VIRTUALIZATION LAYER
VIRTUAL ARCHITECTURE
TechTarget

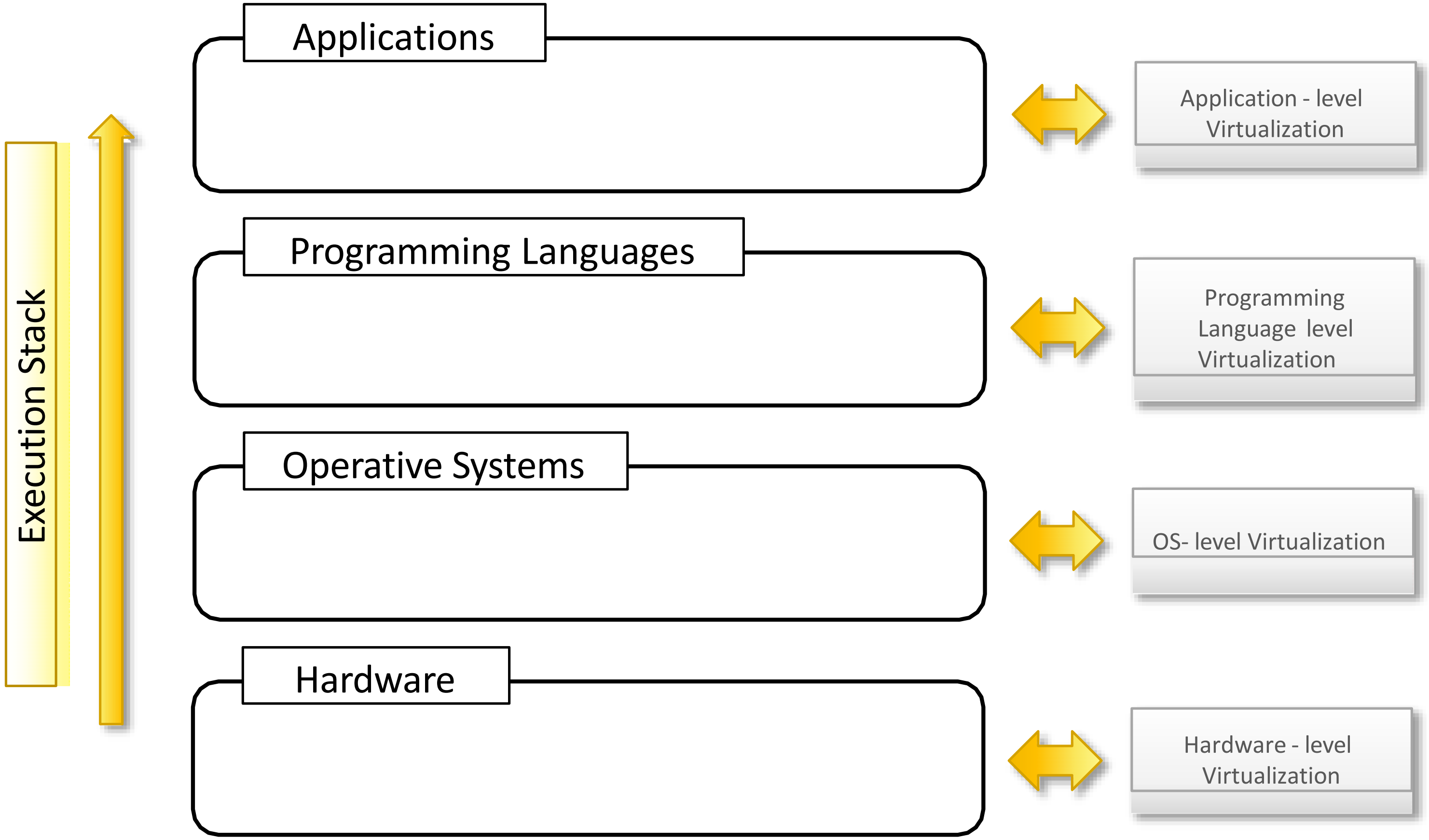
Why are virtualized environments so popular today?

- **Increased performance and computing capacity**
  - PCs are having immense computing power.
- **Underutilized hardware and software resources**
  - Limited use of increased performance & computing capacity.
- **Lack of space**
  - Continuous need for additional capacity.
- **Greening initiatives**
  - Reduce carbon footprints
  - Reducing the number of servers, reduce power consumption.
- **Rise of administrative costs**
  - Power and cooling costs are higher then IT equipments.

Types of virtualization

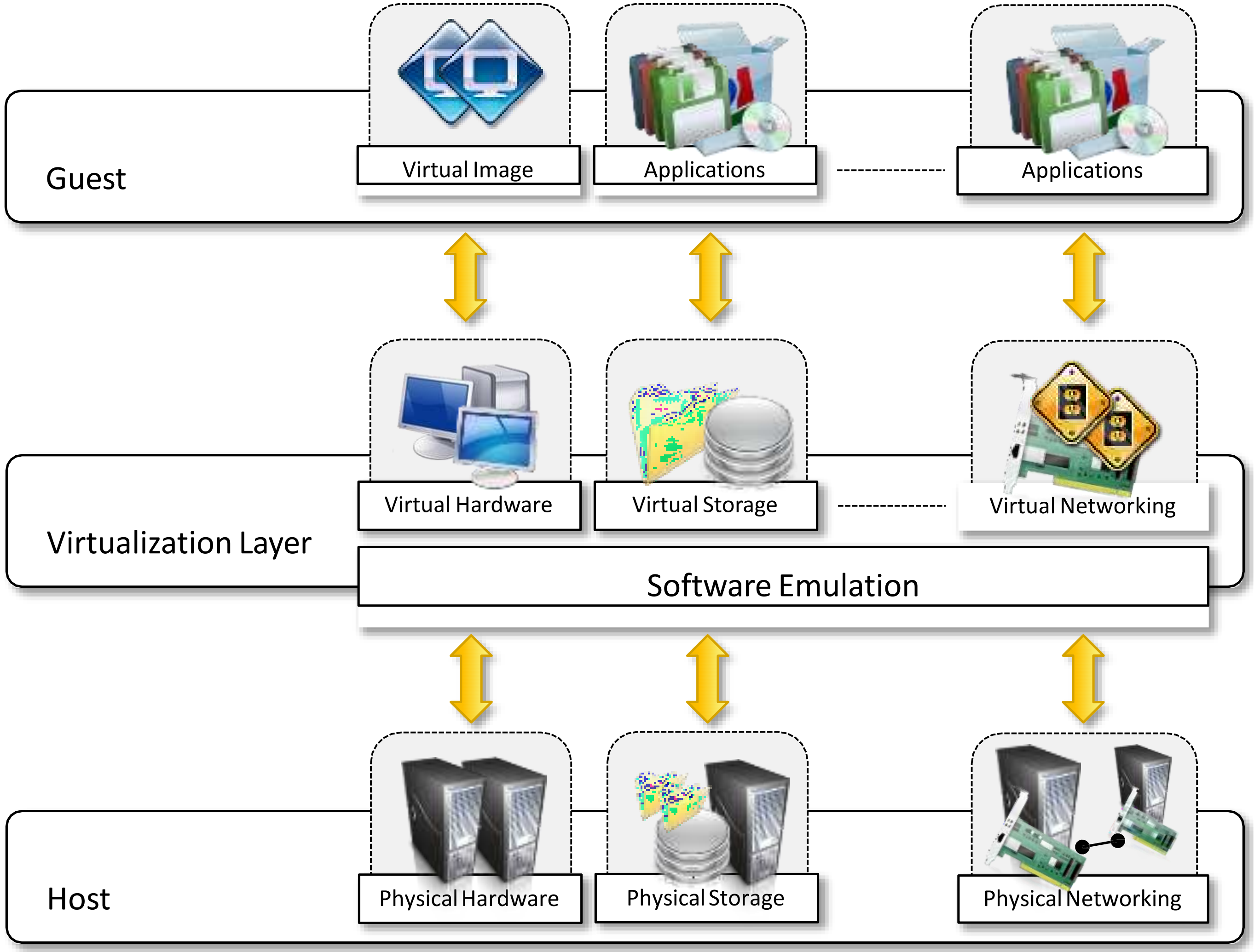


- **Server virtualization:** This involves partitioning a physical server into multiple virtual servers, each running its own operating system and applications.
- **Desktop virtualization:** This involves running multiple virtual desktops on a single physical machine, allowing users to access their desktops remotely from any device.
- **Application virtualization:** This involves isolating an application from the underlying operating system and other applications, allowing it to run independently in its own virtual environment.
- **Network virtualization:** This involves creating a virtual network that abstracts the physical network hardware, allowing multiple virtual networks to coexist on the same physical network.
- **Storage virtualization:** This involves abstracting physical storage resources and presenting them as a single, virtualized storage pool, which can be allocated to applications as needed.
- **Operating system virtualization:** This involves running multiple instances of the same operating system on a single physical machine, each isolated from the others and running its own applications.


Execution Stack
Applications
Programming Languages
Operative Systems
Hardware
Application - level Virtualization
Programming Language level Virtualization
OS- level Virtualization
Hardware - level Virtualization

Components of Virtualized Environments

- Three major components of Virtualized Environments
  - **Guest** – system component that interacts with Virtualization Layer.
  - **Host** – original environment where guest runs.
  - **Virtualization Layer** – recreate the same or different environment where guest will run.



Virtualization Reference Model

Characteristics of VE

- Increased Security
- Managed Execution

✓ - Sharing

✓ - Aggregation

✓ - Emulation

✓ - Isolation

- Portability

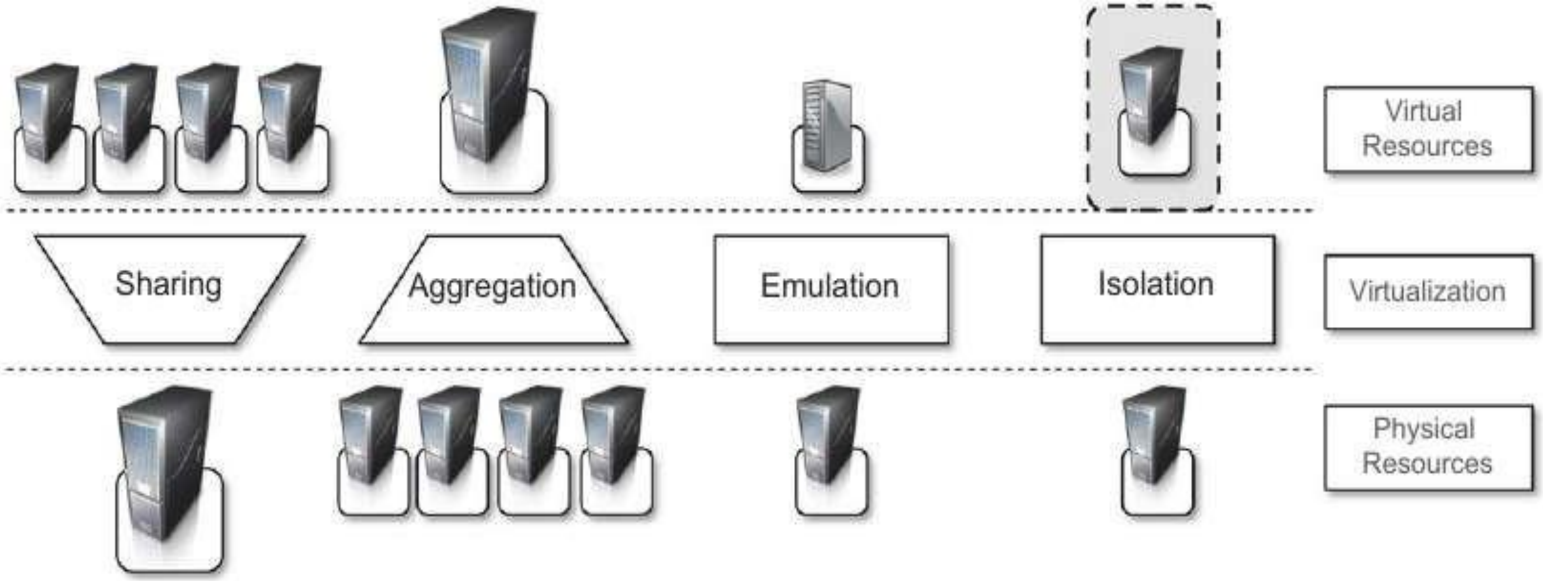
Increased Security

- Ability to control the execution of a guest
- Guest is executed in emulated environment.
- Virtual Machine Manager control and filter the activity of the guest.
- Hiding of resources.
- Having no effect on other users/guest environment.

Managed Execution types

- **Sharing**
  - Creating separate computing environment within the same host.
  - Underline host is fully utilized.
- **Aggregation**
  - A group of separate hosts can be tied together and represented as single virtual host.
- **Emulation**
  - Controlling & Tuning the environment exposed to guest.
- **Isolation**
  - Complete separate environment for guests.

Managed Execution



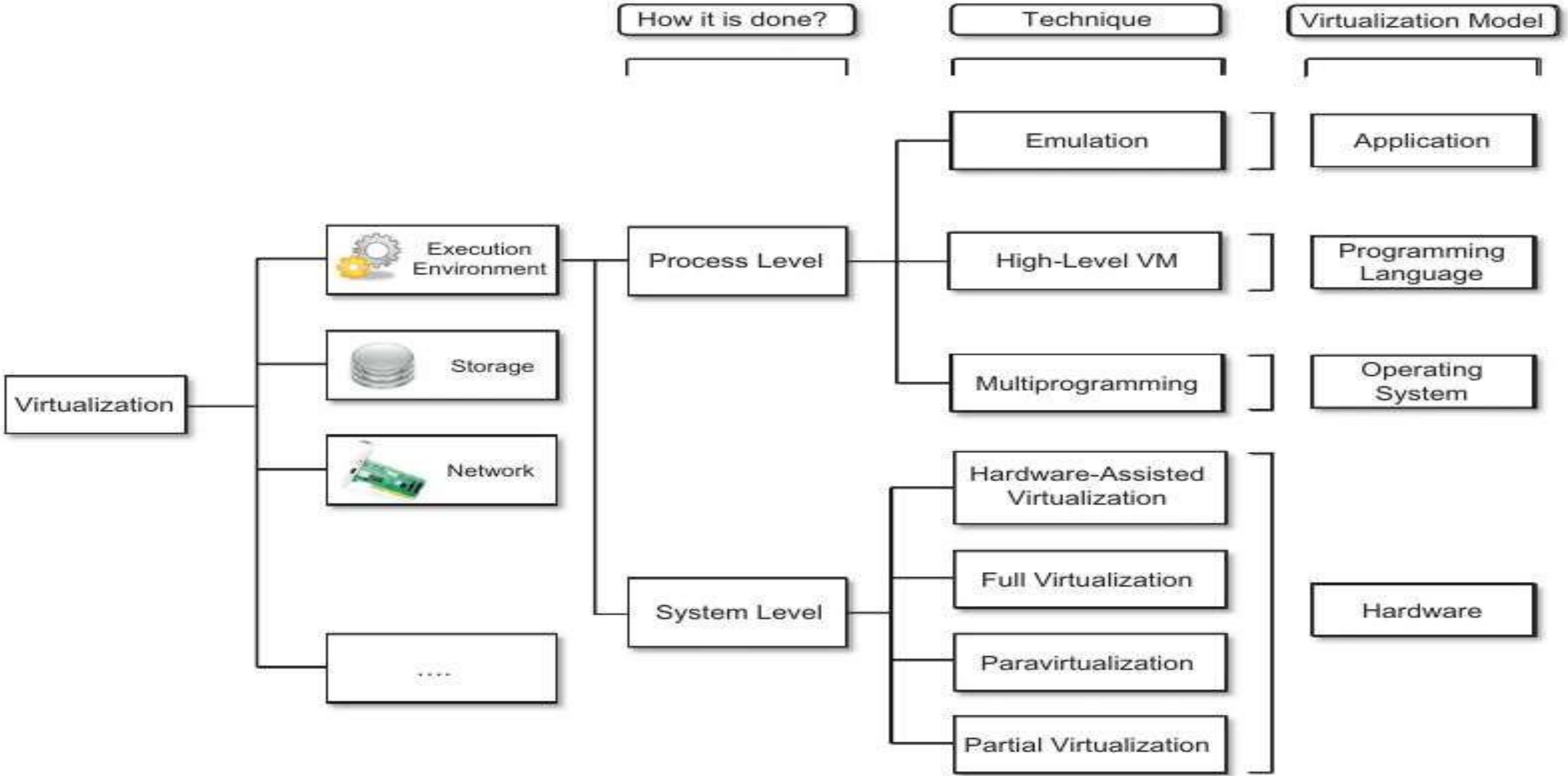
Portability

- Safely moved and executed on top of different virtual machine.
- Application Development Cycle more flexible and application deployment very straight forward
- Availability of system is with you.

Taxonomy of Virtualization Techniques

- Virtualization is mainly used to emulate ***execution environment*** , ***storage*** and ***networks***.
- Execution Environment classified into two :-
  - **Process-level** – implemented on top of an existing operating system.
  - **System-level** – implemented directly on hardware and do not or minimum requirement of existing operating system

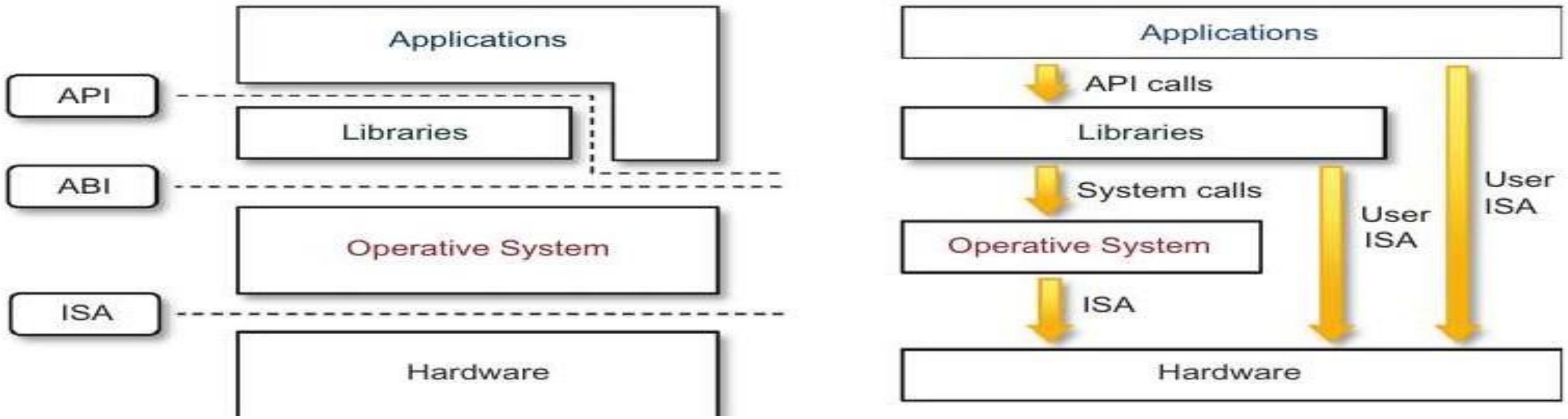
Taxonomy of virtualization



Execution Virtualization

- It defines the ***interfaces between the levels*** of abstractions, which ***hide implementation details***.
- Virtualization techniques actually ***replace one of the layers*** and intercept the calls that are directed towards it.

Machine Reference Model



- Hardware is expressed in terms of the ***Instruction Set Architecture (ISA).***
  - *ISA for processor, registers, memory and the interrupt management.*
- ***Application Binary Interface (ABI***) separates the OS layer from the application and libraries which are managed by the OS.
  - System Calls defined
  - Allows probabilities of applications and libraries across OS.

ISA and ABI

- Instruction Set Architecture (ISA) is part of the abstract model of a computer that defines how the CPU is controlled by the software. ISA is a set of instructions that define the interface between the software and hardware of a computer system. The ISA defines the operations that a processor can perform, the format of the instructions that the processor can execute, and the way in which the processor interacts with memory and other system resources.
- Application binary interface (ABI) is an interface between two binary program modules. Often, one of these modules is a library or operating system facility, and the other is a program that is being run by a user. ABI defines how data structures or computational routines are accessed in machine code, which is a low-level, hardware-dependent format.
- API defines this access(data structures) in source code, which is a relatively high-level, hardware-independent, often human-readable format.

ISA

- The ISA serves as an interface between the hardware and the software running on a computer system.
- The ISA defines the machine language instructions that a processor can execute.
- These instructions are typically represented as binary codes and are designed to be understood and executed directly by the processor.
- Examples of instructions that are commonly found in Instruction Set Architectures (ISAs) are Arithmetic Instructions, Data Transfer Instructions, Control Flow Instructions, Logical Instructions.

Machine Reference Model [Cont.]

- API – it interfaces applications to libraries and/or the underlying OS.
- Layered approach simplifies the development and implementation of computing system.
- ISA has been divided into two security classes:-
  - ***Privileged Instructions***
  - ***Nonprivileged Instructions***

ISA: Security Classes

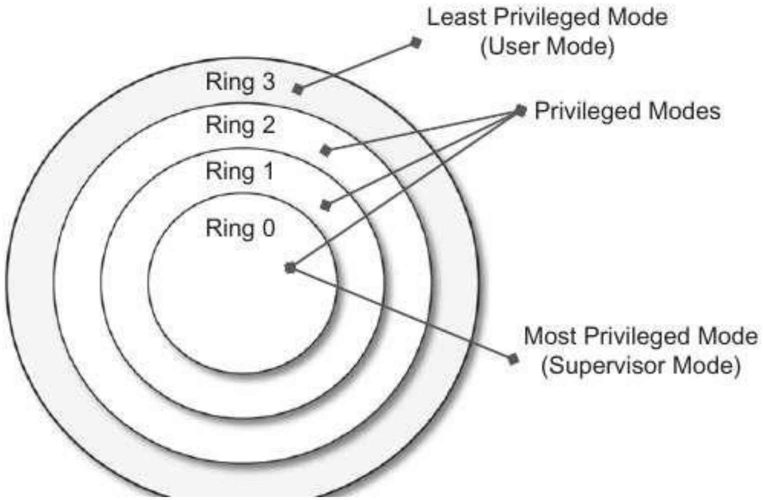
- **Privileged ISA** is designed to provide instructions and features that are **accessible only to privileged software components**, such as the **operating system kernel or hypervisor**.
- These instructions allow privileged software to **perform low-level operations** and access system resources that are typically restricted from user-level applications.
- Privileged ISAs often include instructions for **managing memory protection, controlling interrupts and exceptions, modifying processor state, and accessing privileged system registers**.

ISA: Security Classes **[Cont.]**

- **Non-Privileged ISA**, also known as a user ISA, is designed to provide instructions and features that are accessible to user-level applications and software running in a less privileged mode.
- These instructions are typically limited to high-level operations and do not have direct control over critical system resources.

Privileged Hierarchy: Security Ring

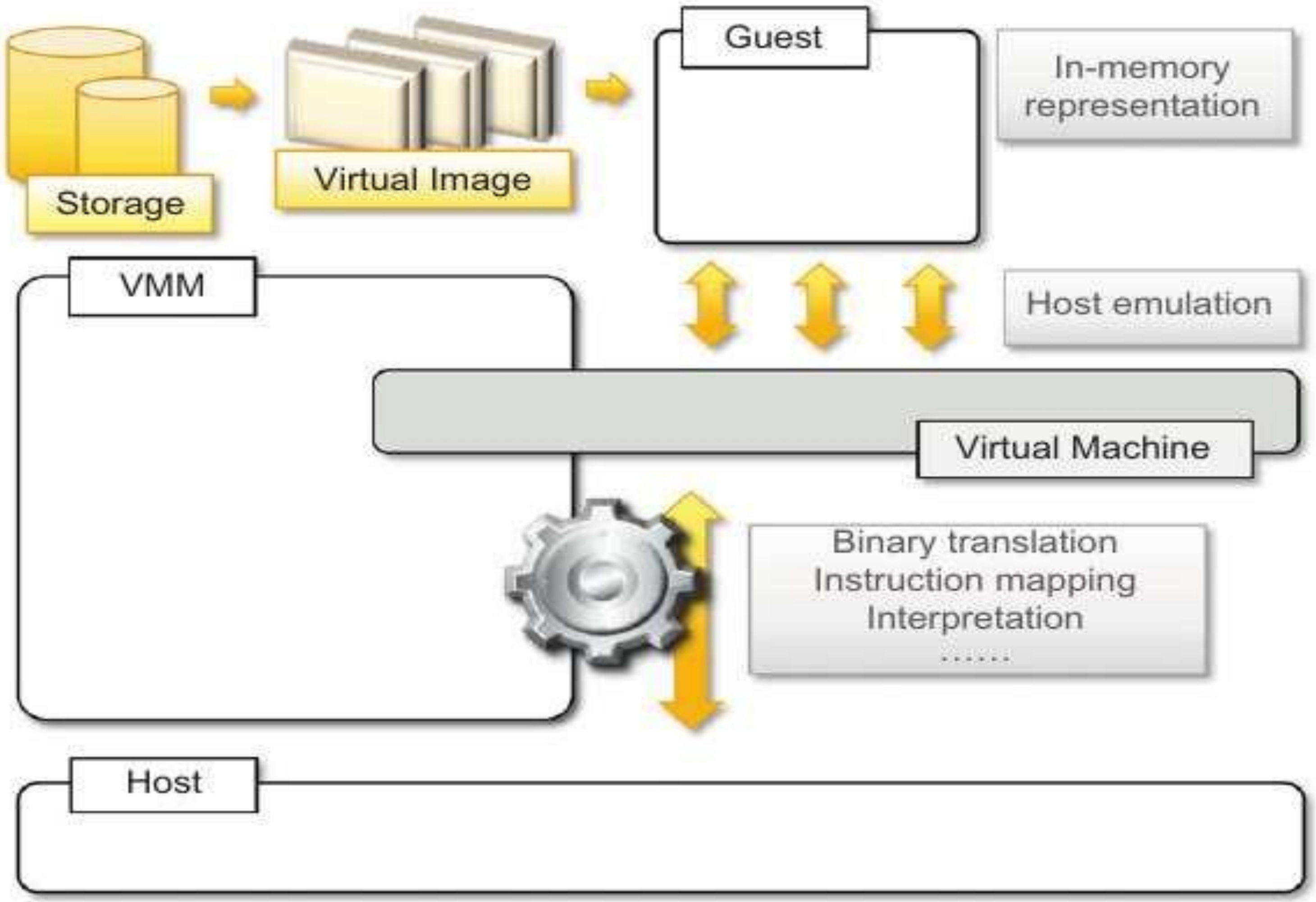
- *Ring-0* is in most privileged level, used by the **kernel**.
- *Ring-1 & 2* used by the **OS-level services**
- and , *Ring-3* in the least privileged level , used by the **user**.
- Recent system support two levels :-
  - **Ring 0 – *supervisor mode***
  - **Ring 3 – *user mode***



Hardware-level virtualization

- It is a virtualization technique that provides an ***abstract execution environment*** in terms of ***computer hardware*** on top of which a ***guest OS*** can be run.
- It is also called as system virtualization.

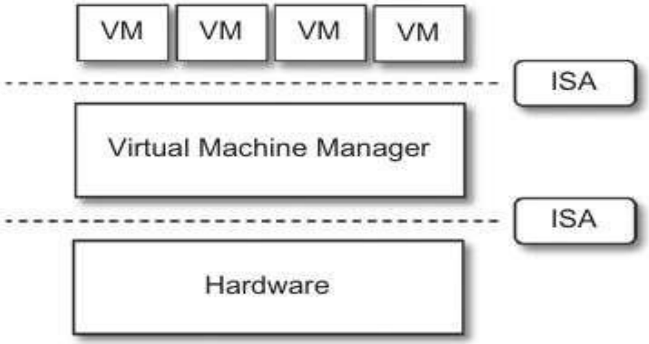
Hardware-level virtualization



Hypervisor

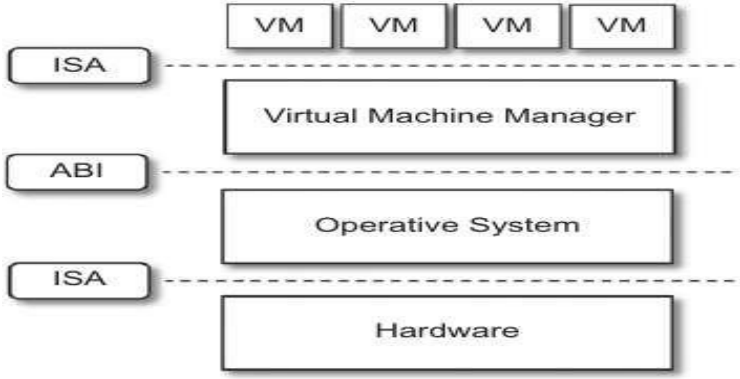
- Hypervisor runs above the supervisor mode.
- It runs in supervisor mode.
- It recreates a h/w environment.
- It is a piece of s/w that enables us to run one or more VMs on a physical server(host).
- Two major types of hypervisor
  - ***Type -I***
  - ***Type-II***

Type-I Hypervisor



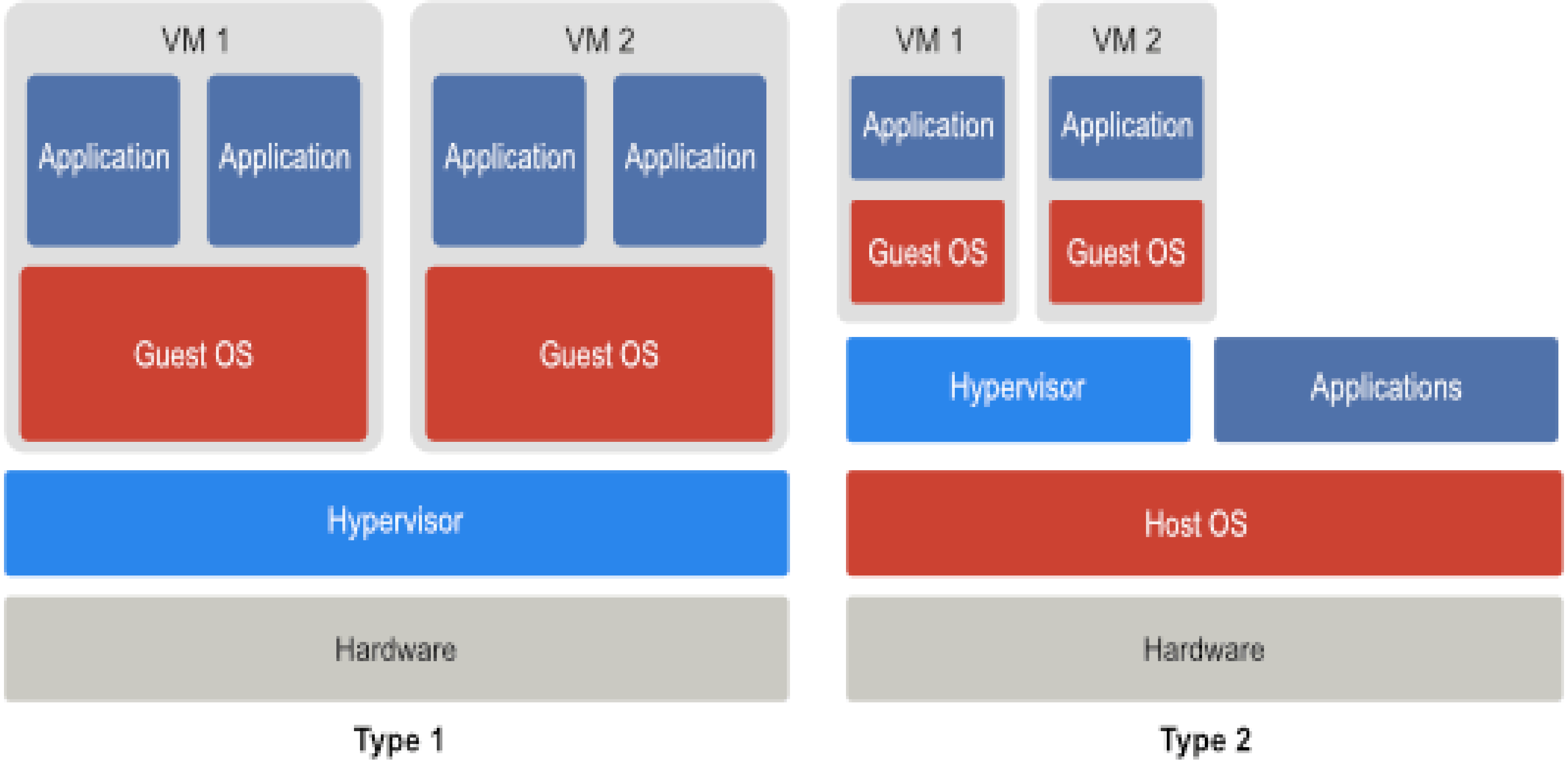
- Type 1 hypervisor, also known as a native or bare-metal hypervisor. It runs directly on top of the hardware.
- Takes place of OS.
- Directly interact with the ISA exposed by the underlying hardware.
- Type 1 hypervisors are commonly used in enterprise data centers and cloud computing environments to consolidate multiple physical servers into a single physical host, maximizing hardware utilization and reducing costs.

Type-II Hypervisor



- It require the support of an operating system to provide virtualization services.
- Programs managed by the OS.
- Type 2 hypervisors rely on the underlying operating system to manage hardware resources and provide device drivers.
- Type 2 hypervisors are often used for desktop virtualization and testing environments, as they are generally easier to install and manage than Type 1 hypervisors. Examples of Type 2 hypervisors include Oracle VirtualBox, VMware Workstation.
- Also called hosted virtual machine.
- Type 2 hypervisor, the host operating system must be installed first, and then the hypervisor is installed as an application within the operating system.

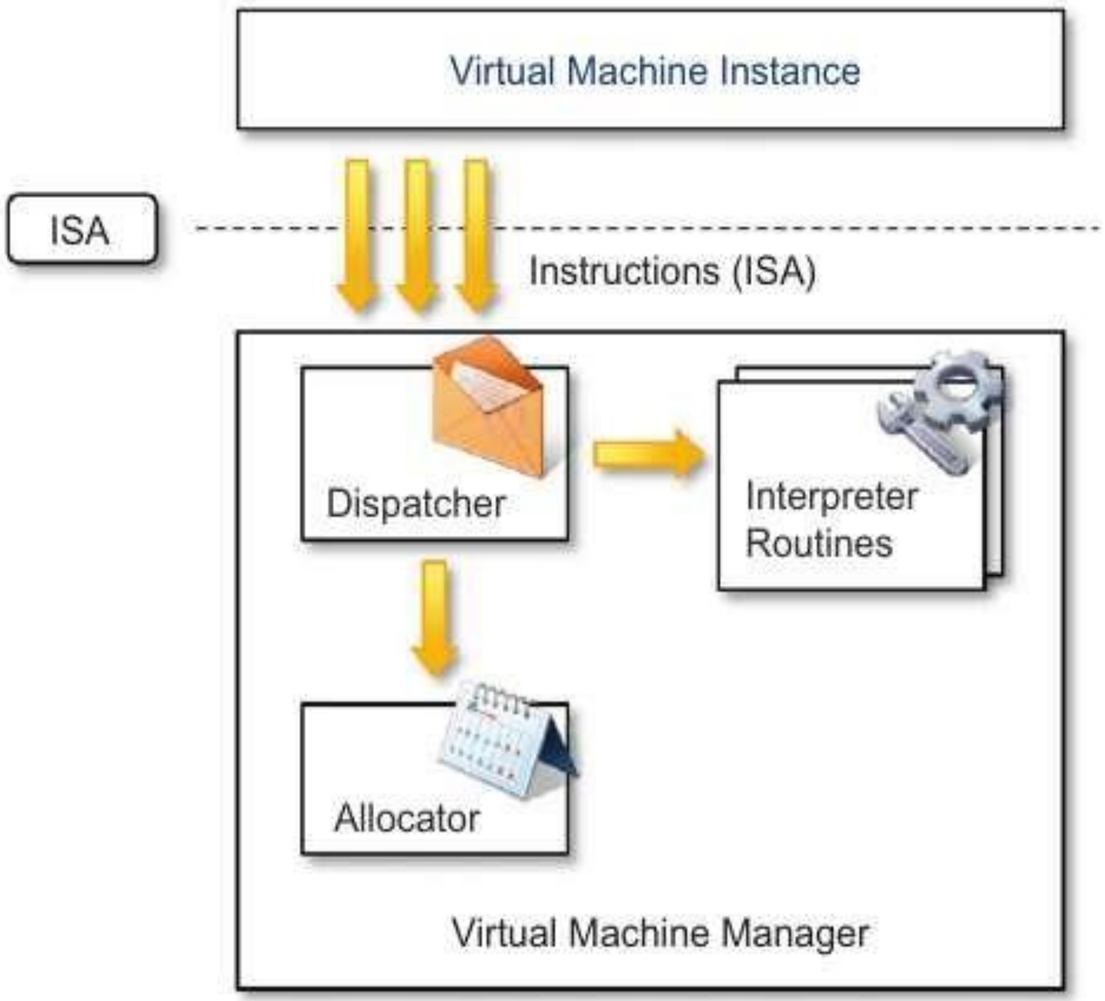
Hypervisor Types



Virtual Machine Manager (VMM)

- Main Modules :-
  - **Dispatcher**
    - Entry Point of VMM. Reroutes the instructions issued by VM instance to the appropriate virtual processor or resource manager.
  - **Allocator**
    - Deciding the system resources to be provided to the VM. Invoked by dispatcher
    - It uses various algorithms to determine the optimal distribution of resources based on factors such as workload, priority, and resource availability.
  - **Interpreter**
    - Consists of interpreter routines which are executed whenever a VM executes a privileged instruction.
    - Trap is triggered and the corresponding routine is executed.

Virtual Machine Manager (VMM)



Hardware virtualization Techniques

- CPU installed on the host is only one set, but each VM that runs on the host requires their own CPU.
- It means CPU needs to virtualized, done by hypervisor.

Hardware-assisted virtualization

- It uses hardware features built into modern processors to provide better performance and security for virtual machines.
- Hardware-assisted virtualization provides a way for the hypervisor to directly access and control the underlying hardware resources, such as CPU, memory, and I/O devices, without going through the host operating system. This reduces the overhead of virtualization and improves the performance of the virtual machines.
- ***Intel VT*** and ***AMD V*** extensions.

Full virtualization

- Full virtualization is a virtualization technique that allows multiple operating systems (referred to as guest operating systems) to run concurrently on a single physical machine (referred to as the host machine) without any modifications to the guest operating systems.
- In full virtualization, the guest operating systems are unaware that they are running inside a virtualized environment.
- Popular examples of full virtualization solutions include VMware ESXi, Microsoft Hyper-V, and KVM (Kernel-based Virtual Machine).

Para-virtualization

- Para-virtualization is a virtualization technique that allows multiple guest operating systems to run on a single physical machine by modifying the guest operating systems to be aware of the virtualized environment. Unlike full virtualization, which emulates the underlying hardware, para-virtualization requires modifications to the guest operating systems to communicate and cooperate with the hypervisor or virtual machine monitor (VMM).
- Xen, a popular open-source hypervisor, is an example of Para-virtualization.

Partial virtualization

- When entire operating systems cannot run in the virtual machine, but some or many applications can, it is known as Partial Virtualization.
- Basically, it partially simulates the physical hardware of a system.
- This type of virtualization is far easier to execute than full virtualization.

Operating system-level virtualization

- It offers the opportunity to create different and ***separated execution environments*** for applications that are managed concurrently.
- No VMM or hypervisor
- Virtualization is in single OS
- OS kernel allows for multiple isolated user space instances
- Good for server consolidation.
- Ex. *chroot , Jails, OpenVZ etc.*

Programming language-level virtualization

- It is mostly used to achieve ***ease of deployment*** of application, ***managed execution*** and ***portability across*** different platform and OS.
- It consists of a virtual machine ***executing the byte code of a program***, which is the result of the ***compilation process***.
- Produce a binary format representing the machine code for an abstract architecture.
- Example
  - Java platform – Java virtual machine (JVM)
  - .NET provides Common Language Infrastructure (CLI)
- They are stack-based virtual machines

Advantage of programming/process - level VM

- Provide ***uniform execution environment*** across different platforms.
- This ***simplifies*** the development and deployment efforts.
- Allow more ***control over the execution*** of programs.
- Security; by filtering the I/O operations
- Easy support for sandboxing

Application-level virtualization

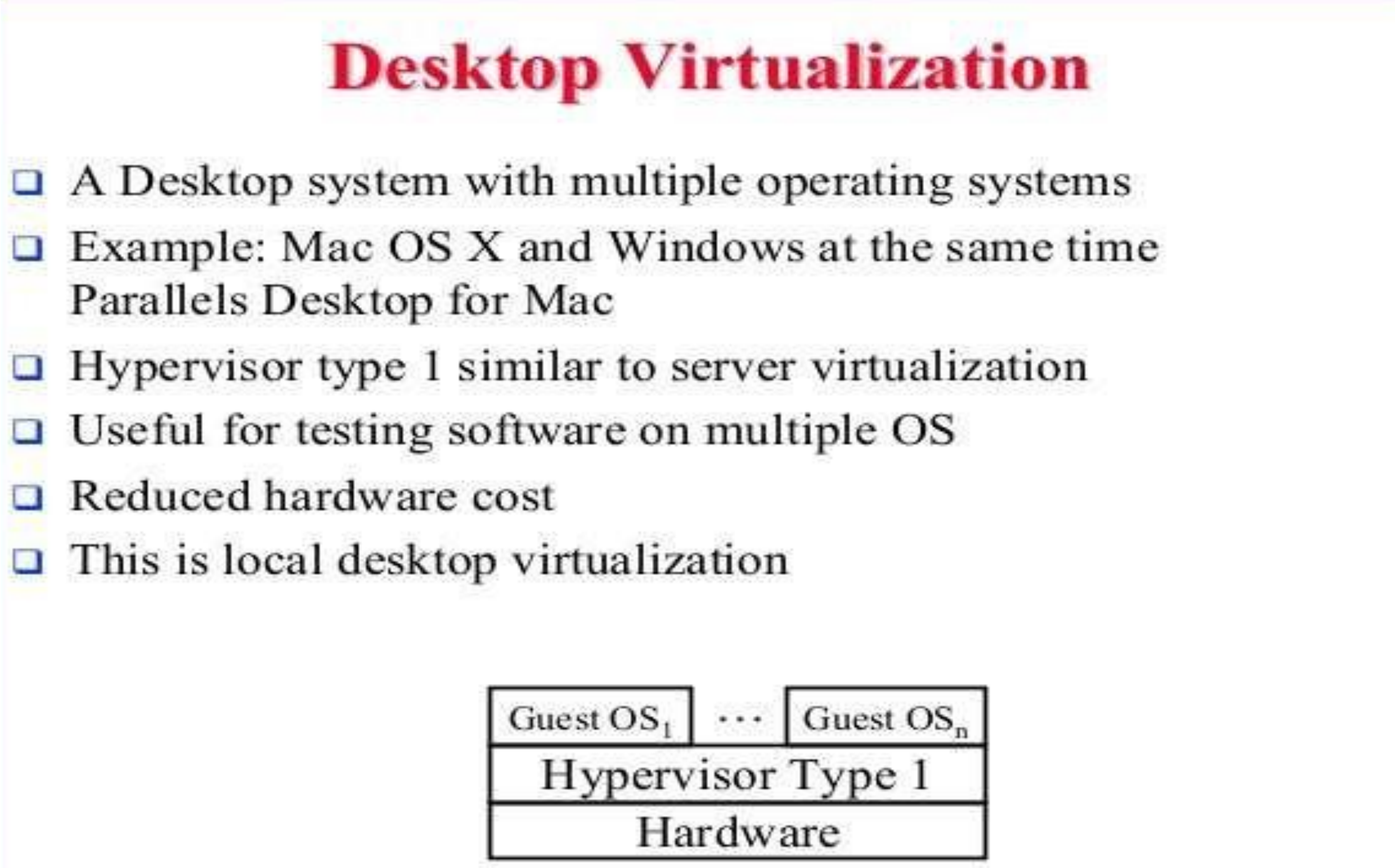
- It is a technique allowing applications to run in ***runtime environments*** that do not ***natively support*** all the features required by such applications.
- In this, applications are not installed in the ***expected runtime environment.***
- This technique is most concerned with :-
  - Partial file system
  - Libraries
  - Operating System component emulation

Types: Storage Virtualization

- It allows decoupling the physical organization of the h/w from its logical representation.
- Using Network based virtualization known as ***storage area network*** (SAN).

Network Virtualization

- It combines h/w appliances and specific software for the creation and management of a virtual n/w.
- It can aggregate ***different physical networks*** into a single logical network.



Desktop Virtualization

- A Desktop system with multiple operating systems
- Example: Mac OS X and Windows at the same time Parallels Desktop for Mac
- Hypervisor type 1 similar to server virtualization
- Useful for testing software on multiple OS
- Reduced hardware cost
- This is local desktop virtualization

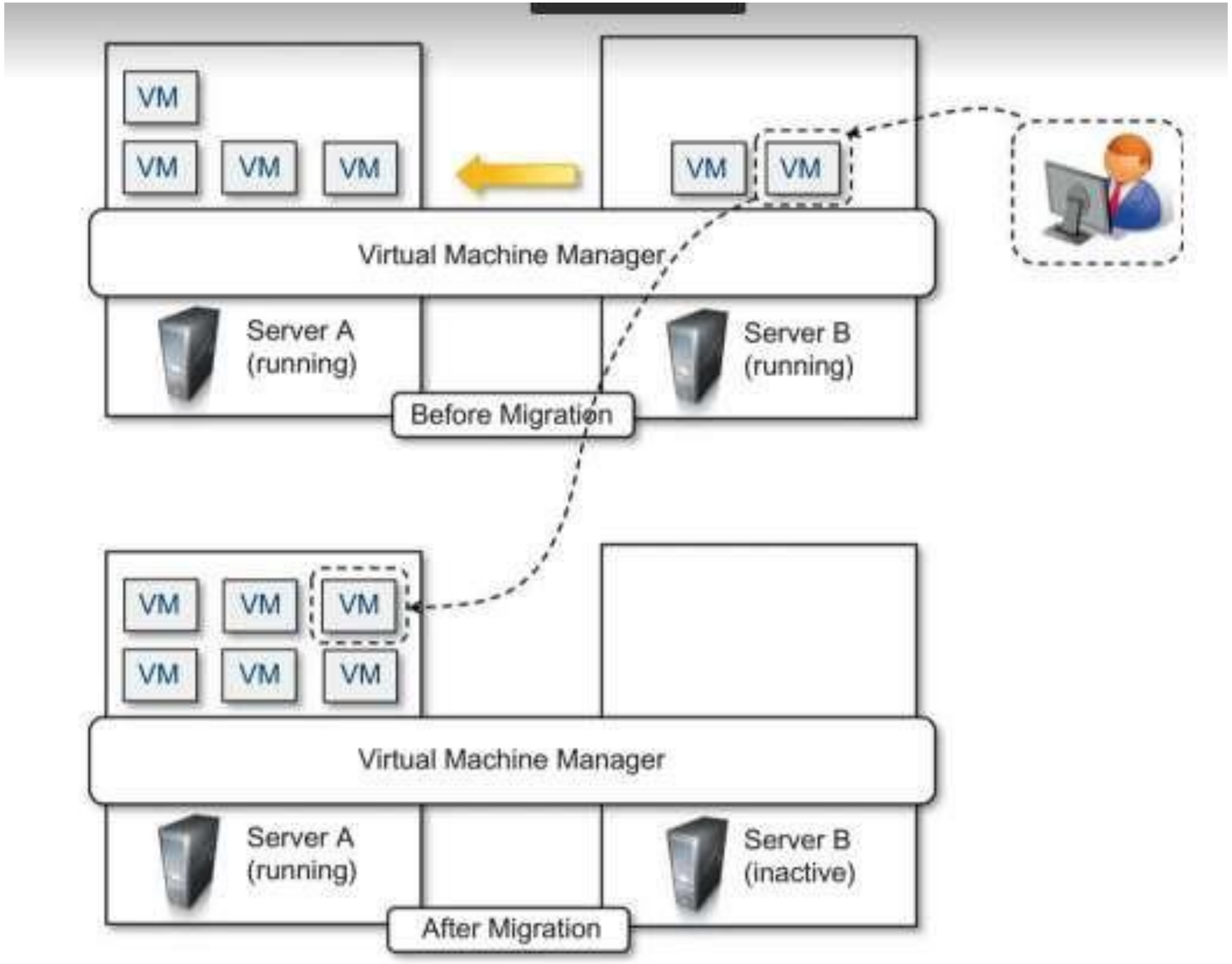
Guest OS$_1$
· · ·
Guest OS$_n$
Hypervisor Type 1
Hardware

Application Server Virtualization

- Application server virtualization abstracts a collection of application servers that provide the same service as a single virtual application server
- Providing better quality of service rather than emulating a different environment

Virtualization and cloud computing

- Virtualization plays an important role in cloud computing
- Virtualization technologies are primarily used to offer configurable computing environments and storage.
- Hardware virtualization is an enabling factor for solutions in the (IaaS) market segment
- programming language virtualization is a technology leveraged in (PaaS) offerings.



Server consolidation and virtual machine migration

Pros and cons of virtualization

- **Advantages of Virtualization**

✓ Reduced spending

✓ Sandbox

✓ Portability

✓ Efficient use of resources.

✓ Easier backup and disaster recovery

✓ Better business continuity

✓ More efficient IT operations

Pros and cons of virtualization

- **Disadvantages of Virtualization**

✓ Upfront costs.

✓ Software licensing considerations

✓ Possible learning curve

✓ Performance degradation

✓ Inefficiency and degraded user experience

✓ Security holes and new threats