Virtualization Technology

Dr. Shreeya Swagatika Sahoo

Introduction

- Virtualization is a large umbrella of technologies and concepts that are meant to provide an abstract environment-whether virtual hardware or an operating system.
- This technology allows to create multiple simulated environments or dedicated resources from a single, physical hardware system.
- Virtualization enables multiple operating systems to run on the same physical platform (running Windows OS on top of virtual machine, which itself is running on Linux OS).
- Virtualization provides a great opportunity to build elastically scalable systems, which are capable of provisioning additional capability with minimum costs

Virtualization: Reasons for renewed interest

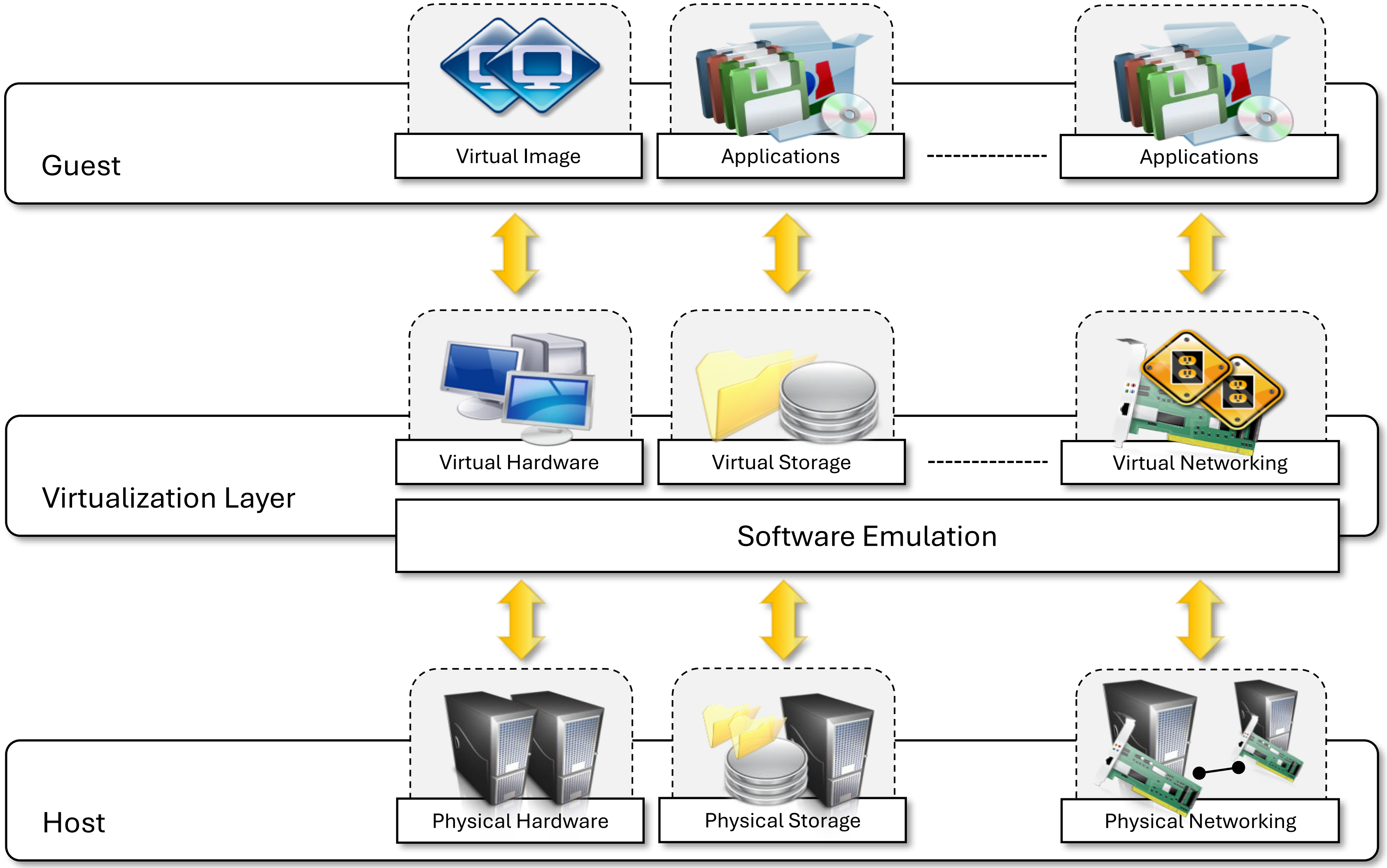
Virtualization technologies have recently gained renewed interest due to several factors:

- ➢**Increased performance** of modern PCs allows them to run virtual machines efficiently.
- ➢**Underutilized hardware/software** can be better used through virtualization, improving IT infrastructure efficiency.
- ➢**Lack of space** in growing data centers led to **server consolidation**, where virtualization plays a key role.
- ➢**Greening initiatives** push companies to reduce energy use and carbon footprint, achieved by consolidating servers.
- ➢**Rising administrative costs** (power, cooling, and management) are reduced by minimizing the number of servers via virtualization.

Virtualization reference model

- Virtualization is a broad concept and it refers to the creation of a virtual version of something, whether this is hardware, software environment, storage, or network.
- In a virtualized environment there are three major components: *guest*, *host*, and *virtualization layer*.
- The *guest* represents the system component that interacts with the virtualization layer rather than with the host as it would normally happen.
- The *host* represents the original environment where the guest is supposed to be managed.
- The *virtualization layer* is responsible for recreating the same or a different environment where the guest will operate.

Virtualization reference model



Characteristics of virtualized environments

➢Increased Security

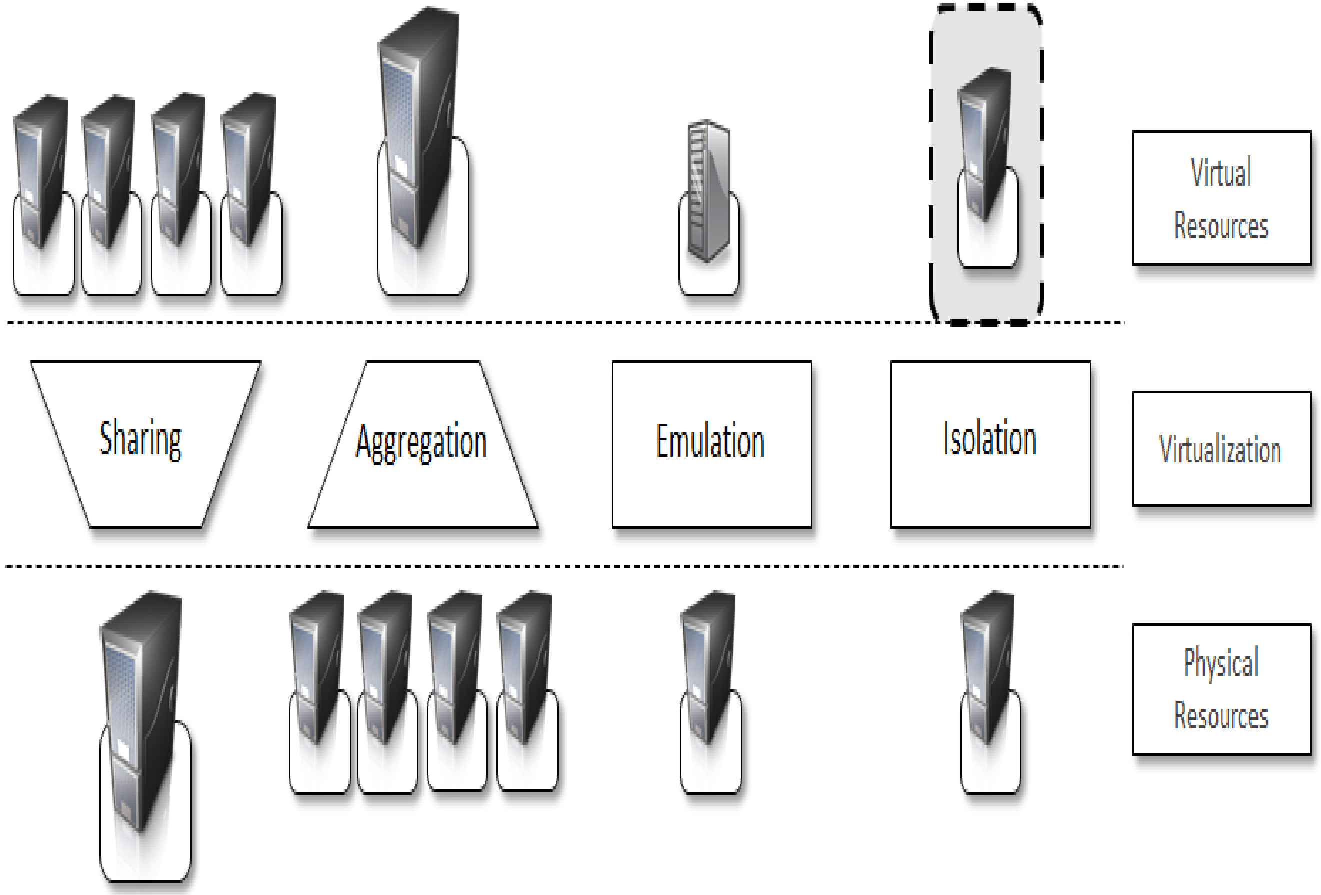
- Ability to control the execution of the guest.
- Guest is executed in an emulated environment.
- VMM controls and filters the activity of the guest.
- VM instances are isolated from each other.

➢Portability

- For **hardware virtualization**, the guest is packaged into a **virtual image** (often one or more files) that can generally be moved and executed on different virtual machines with ease. These images are self-contained, simplifying administration.
- For **programming language-level virtualization** (e.g., Java Virtual Machine, .NET runtime), binary code (like JARs or assemblies) can run on any platform where the corresponding virtual machine is installed, without recompilation.

- **Managed Execution**

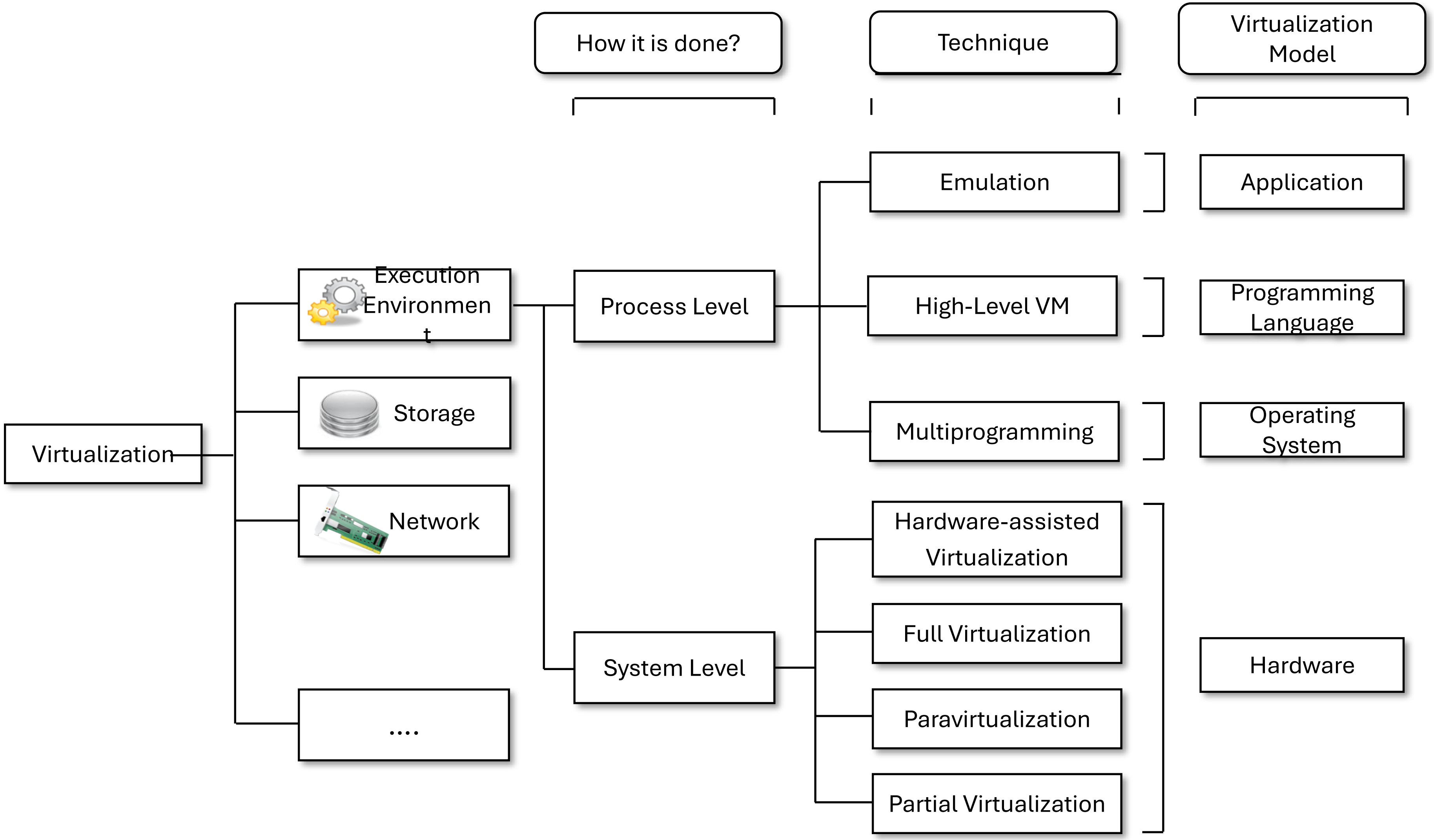
Sharing -Creating separate environment within the same host.
Aggregation - A group of separate host can be tied together and represented as single virtual host.
Emulation -Controlling and tuning the environment exposed to guest.
Isolation - Complete separate environment for guests.



Taxonomy of Virtualization Techniques

- Execution Virtualization
- Storage Virtualization
- Network Virtualization
- Desktop Virtualization
- Application Server Virtualization

A taxonomy of virtualization techniques

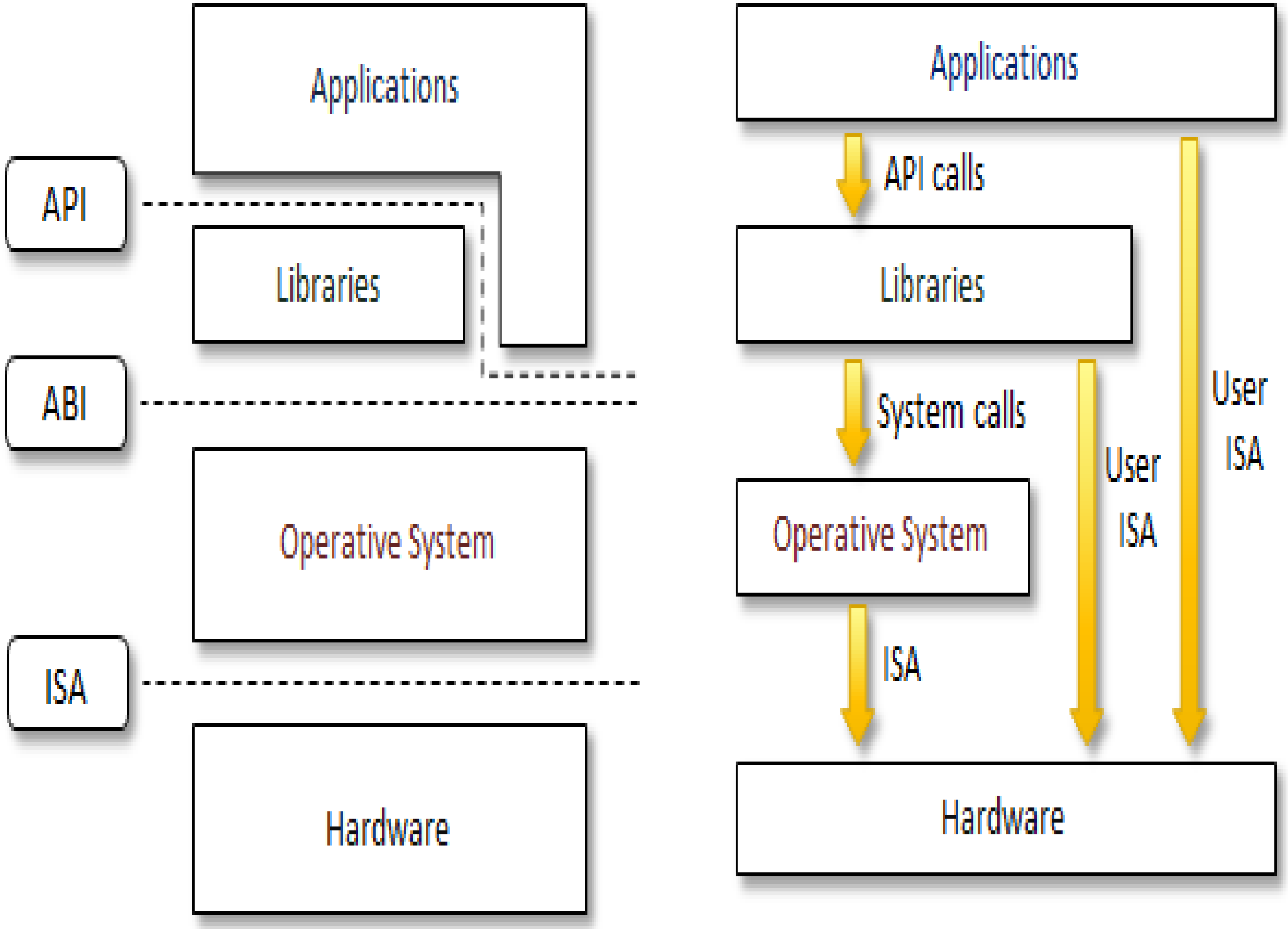


Execution Virtualization

- Execution virtualization includes all those techniques whose aim is to emulate an execution environment that is separate from the one hosting the virtualization layer.
- Execution virtualization can be implemented directly on top of the hardware, by the operating system, an application, or libraries dynamically or statically linked against an application image.
- Virtualizing an execution environment at different levels of the computing stack requires a reference model (**Machine Reference Model**) that defines the interfaces between the levels of abstractions, which hide implementation details.

Machine Reference Model

Applications: Programs written by developers.
Libraries: Provide common functionality (math libraries, GUI libraries, etc.) that applications reuse.
Operative System (OS): Manages hardware and provides system calls for applications.
Hardware: Physical machine (CPU, memory, devices).



Machine Reference Model

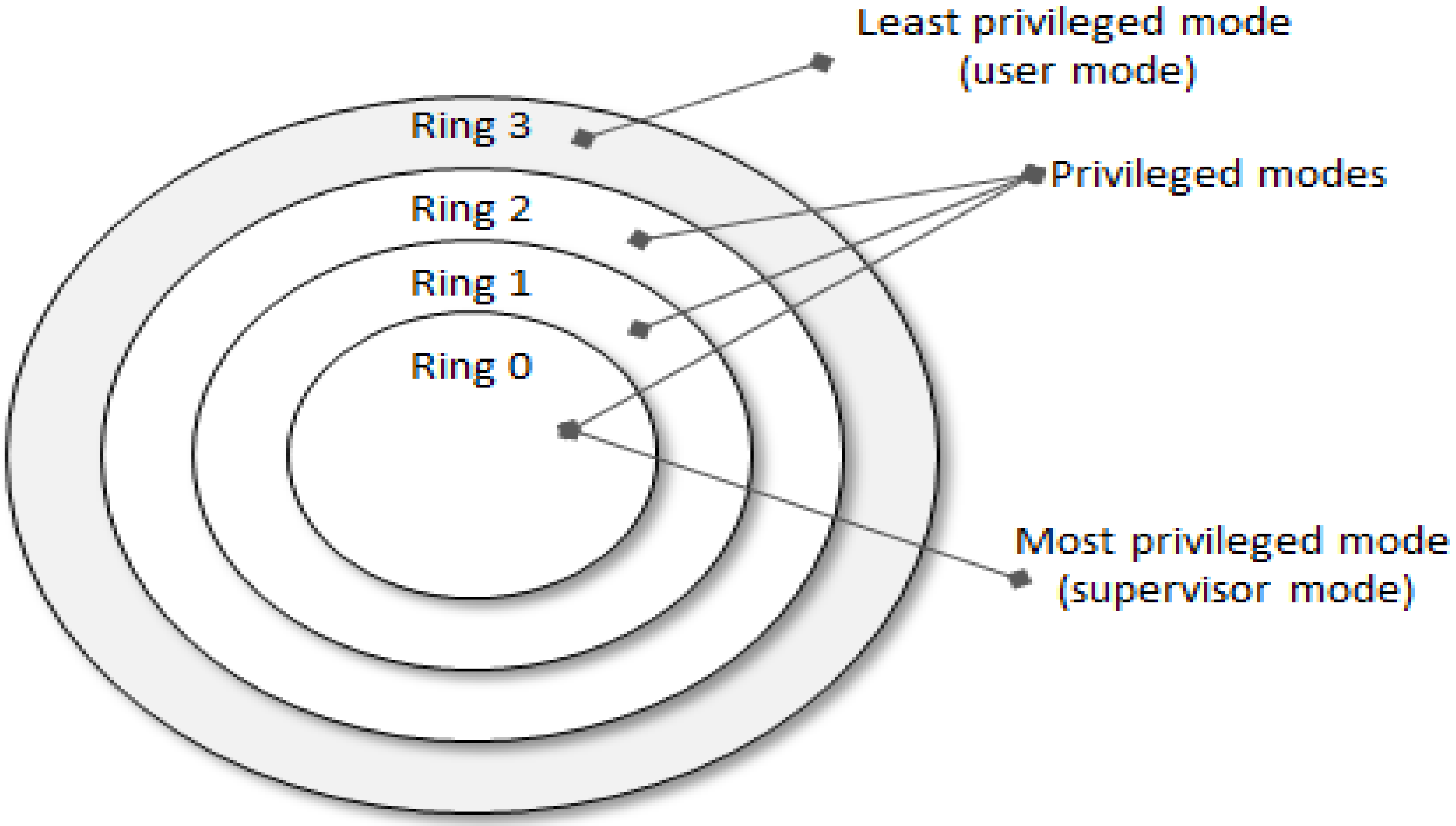
- **API (Application Programming Interface):**
  - Interface between applications and libraries.
  - Example: printf() in C, Java’s standard library functions.

- **ABI (Application Binary Interface):**
  - Interface between compiled applications and the operating system.
  - Defines system call conventions, data types, binary formats, calling conventions.
  - Example: Linux system calls (open, read, write).

- **ISA (Instruction Set Architecture):**
  - Interface between the OS (or compiler) and hardware.
  - Defines the machine instructions the processor understands (e.g., x86, ARM).

Security Rings and Privileged Modes

- Machine reference model also provides ways for implementing a minimal security model for managing and accessing shared resources.



Security Rings and Privileged Modes

Instruction Classes

- Non-privileged instructions:
  - Do not affect shared resources.
  - Safe for all tasks. For Examples: Arithmetic, floating point operations.
- Privileged instructions:
  - Restricted, sensitive operations.
  - Can expose/modify system state.
  - There are two Types: Behavior-sensitive (expose privileged state) and Control-sensitive (modify privileged state).
- Privilege Hierarchy (Rings):
  - Ring 0: Highest privilege → OS kernel.
  - Ring 1 & 2: OS-level services.
  - Ring 3: Lowest privilege → User applications.

Forms of Execution Virtualization

1. Hardware-level Virtualization (System Virtualization)
2. OS-level Virtualization (Containers)
3. Programming Language-level Virtualization
4. Application-level Virtualization

1. Hardware-level Virtualization (System Virtualization)

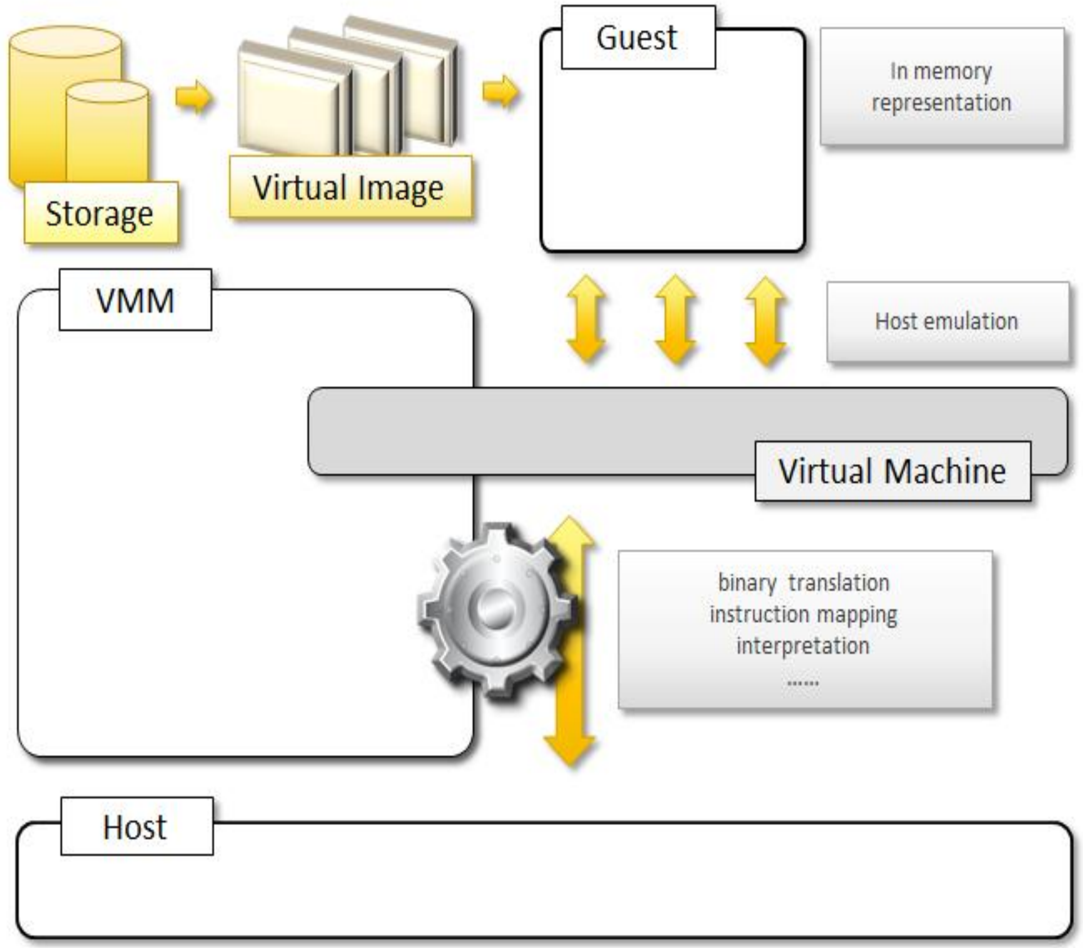
Hardware-level virtualization, also called **system virtualization**, provides an abstract execution environment at the hardware level so that a **guest operating system** can run as if on real hardware.

Guest → Operating system (with applications).

Host → Physical computer hardware.

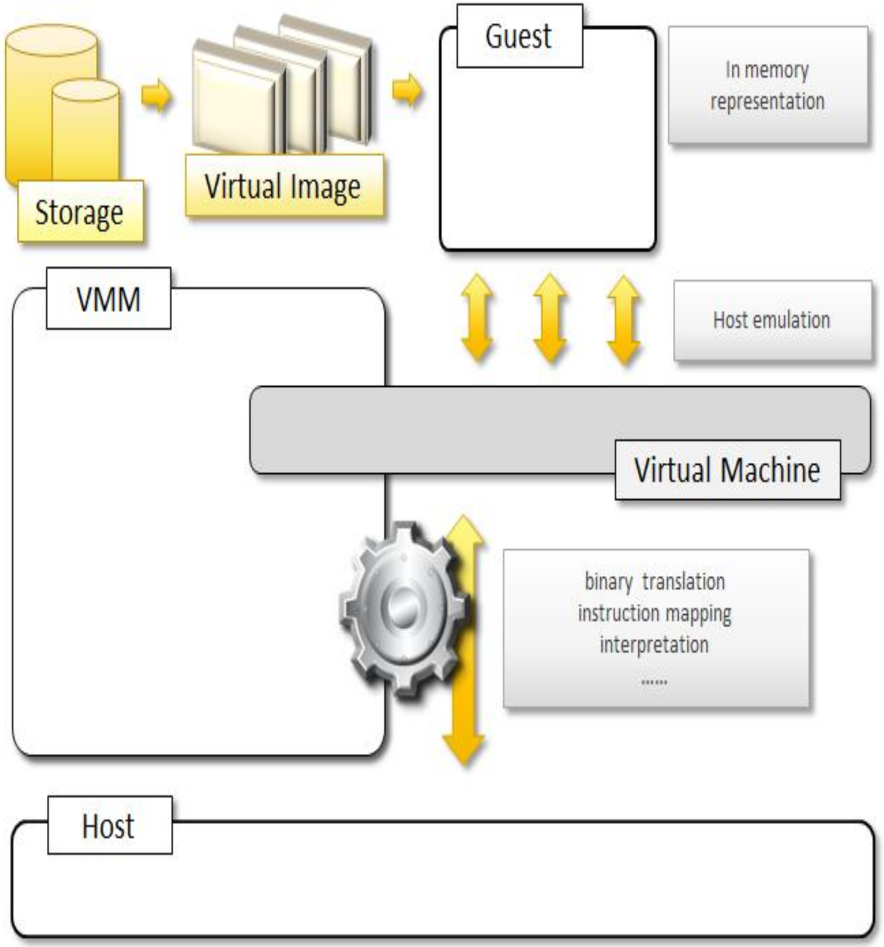
Virtual Machine (VM) → Emulated hardware environment.

Virtual Machine Manager (VMM)/Hypervisor → Software or software-hardware combination that abstracts hardware and manages VMs.



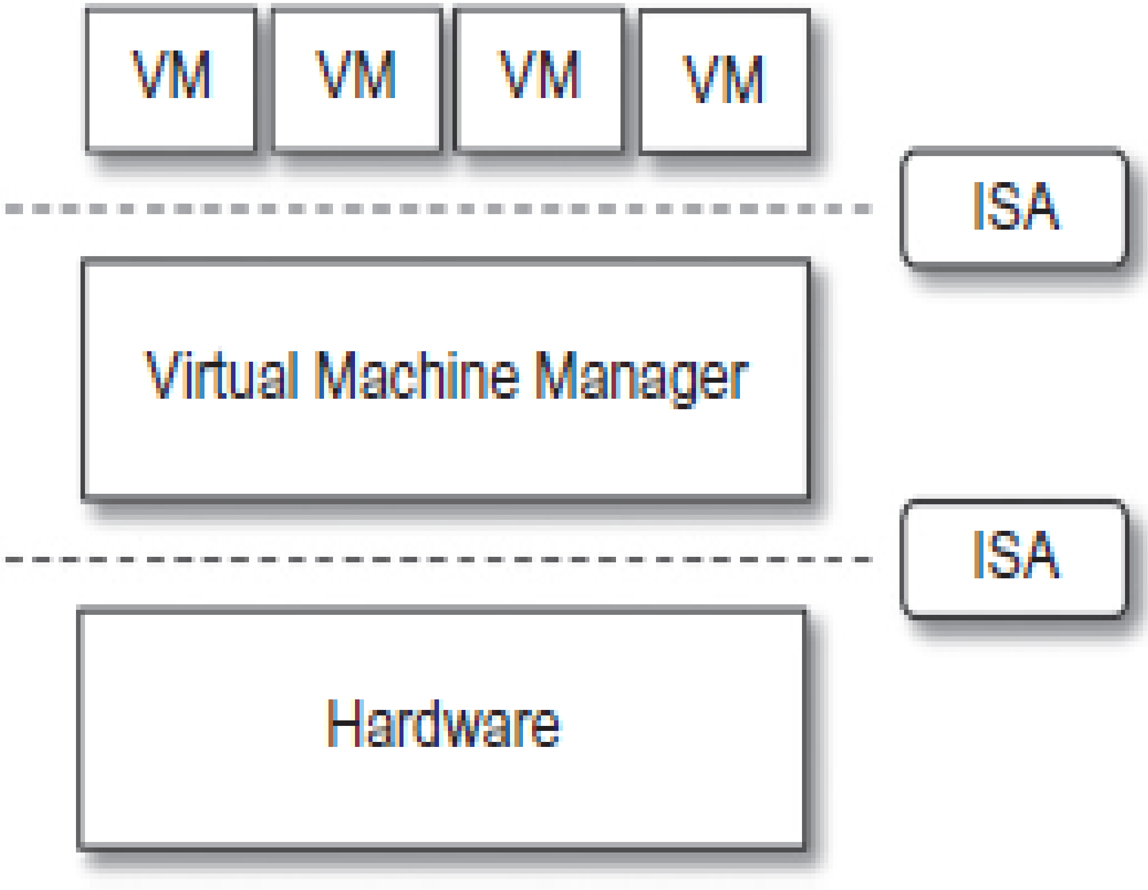
Execution Flow

- **Virtual Image** (disk file) is loaded from **Storage**.
- The **VMM** takes the image and creates the **Guest** (OS + apps) inside a **Virtual Machine**.
- The **Guest OS** issues instructions as if it were running on real hardware.
- The **VMM** intercepts and translates these instructions:
  - **Binary translation**: Converts guest instructions into host instructions.
  - **Instruction mapping**: Maps guest ISA (Instruction Set Architecture) to host ISA.
  - **Interpretation/emulation**: Provides hardware functions that don't exist on the host.
- The **Host** executes the final mapped instructions on physical hardware.



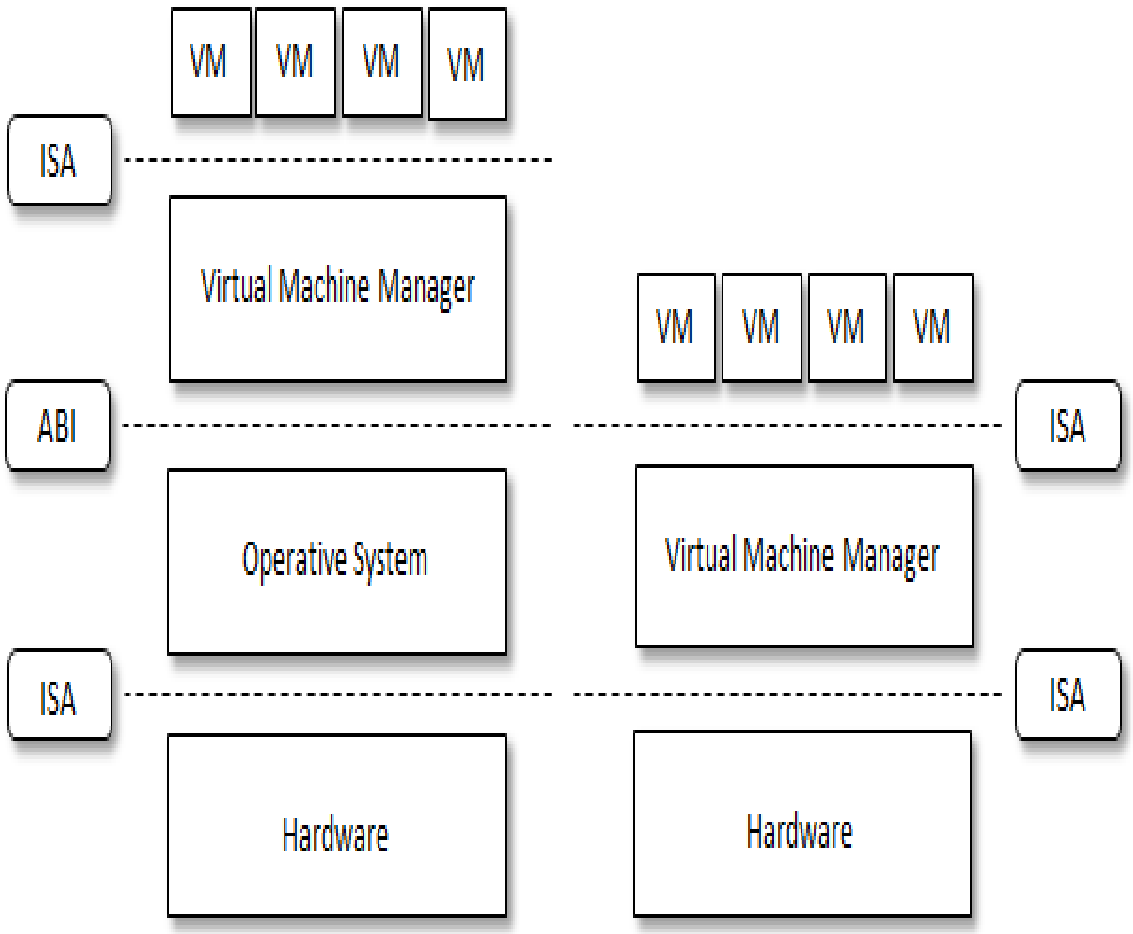
Type I Hypervisor

- It runs directly on top of the hardware and takes the place of OS.
- Directly interact with the ISA interface exposed by the underlying hardware.
- Examples: Vmware ESXi, Microsoft Hyper-V (bare-metal), Xen, KVM etc.

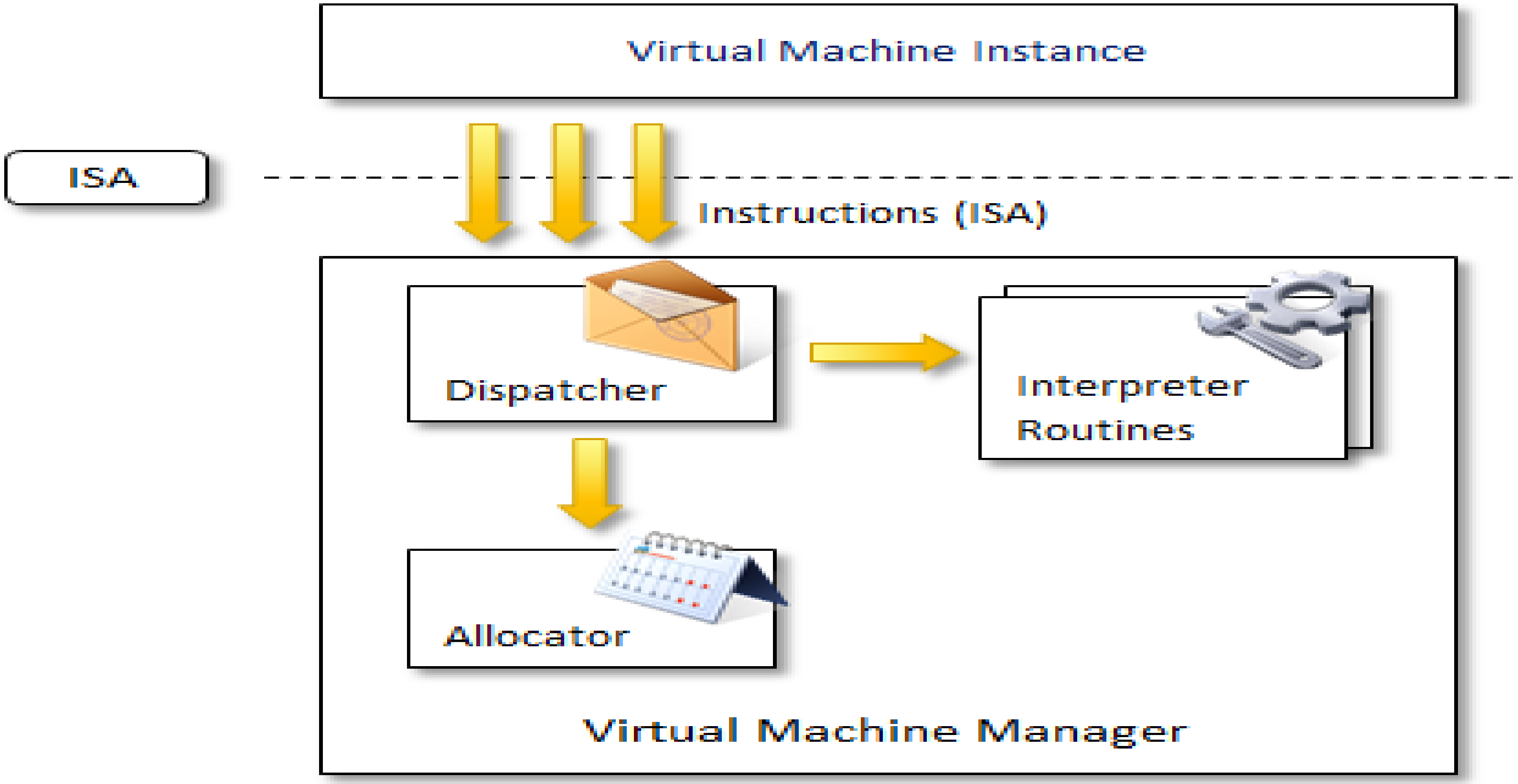


Type II Hypervisor

- This hypervisor requires the support of an OS to provide virtualization services.
- It consists of programs which are managed by the OS.
- The OS interact with the hypervisor through the ABI.
- The hypervisor emulate the ISA of virtual hardware for the guest OS.
- **Examples:** VMware Workstation, Oracle VirtualBox, Parallels Desktop.



Hypervisor Reference Architecture



Hypervisor Reference Architecture

- Three main modules coordinate their activity in order to emulate the underlying hardware: ***dispatcher*, *allocator***, and ***interpreter***.
- The **dispatcher** constitutes the entry point of the monitor and reroutes the instructions issued by the virtual machine instance to one of the two other modules.
- The **allocator** is responsible for deciding the system resources to be provided to the VM. The **allocator** manages resources (CPU time, memory, I/O devices) among multiple VMs. It ensures fair sharing, isolation, and efficiency.
- The interpreter module consists of interpreter routines. These are executed whenever a virtual machine executes a privileged instruction: a trap is triggered and the corresponding routine is executed. Interpreter routines performs Binary translation, Instruction mapping, Emulation etc.

- The criteria that need to be met by a virtual machine manager to efficiently support virtualization were established by Goldberg and Popek in 1974. Three properties have to be satisfied:
  - *Equivalence:* a guest running under the control of a virtual machine manager should exhibit the same behavior that when it is executed directly on the physical host.
  - *Resource control:* the virtual machine manager should be in complete control of virtualized resources.
  - *Efficiency:* a statistically dominant fraction of the machine instructions should be executed without intervention from the virtual machine manager.
- The major factor that determines whether these properties are satisfied is represented by the layout of the ISA of the host running a virtual machine manager. Popek and Goldberg provided a classification of the instruction set and proposed **three theorems** that define the properties that hardware instructions need to satisfy in order to efficiently support virtualization.

Popek and Goldberg theorems

- Theorem-1*: For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.*
- Theorem 2: *A conventional third-generation computer is recursively virtualizable if*
  - *It is virtualizable and.*
  - *A VMM without any timing dependencies can be constructed for it.*
- Theorem 3: *A hybrid VMM may be constructed for any conventional third generation machine, in which the set of user sensitive instructions are a subset of the set of privileged instructions.*

Hardware Virtualization Techniques

- CPU installed on the host is only one set, but each VM that runs on the host requires there one CPU.
- It means CPU needs to be virtualized, and is performed by a hypervisor.

Types of Hardware Level Virtualization

- **Hardware assisted virtualization**: In this, H/W provides architectural support for building a VMM able to run a guest OS in complete isolation.
- **Full virtualization** :
  - Ability to run program (OS) on top of a virtual machine and without any modification.
  - VMM requires complete emulation of the entire underneath hardware.
- **Para virtualization**:
  - This is not a transparent virtualization solution that allows implementing thin virtual machine manager.
  - Expose software interface to the virtual machine that is slightly modified by the host.
  - Guest OS need to be modified.
- **Partial virtualization** :
  - Partial emulation of the underlying hardware.
  - Not allow the complete execution of the guest OS in complete isolation.

2. OS-level Virtualization (Containers)

- It offers the opportunity to create different and separate execution environments for applications that are managed concurrently.
- No VMM or hypervisor is required.
- Virtualization is done within a single OS.
- OS kernel allows for multiple isolated user space instances .

3.Programming Language-level Virtualization

- It mostly used for achieving ease of deployment, managed execution, and portability across different platforms 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.
- Provide uniform execution environment across different platforms.

4. Application-level Virtualization

- Its a technique allowing applications to run in runtime environments that do not natively support all the features required by such applications.
- Applications are not installed in the expected runtime environment, but run as if they are.
- 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.

Other Types of Virtualization

- Storage Virtualization
- Network Virtualization
- Desktop Virtualization
- Application Server Virtualization

Storage Virtualization

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.
- By using this technique users do not have to be worried about the specific location of their data, which can be identified by using a logical path.
- Storage virtualization allows harnessing a wide range of storage facilities and representing them under a single logical file system.
- There are different techniques for storage virtualization one of the most popular includes network based virtualization by means of *Storage Area Networks (SANs)*.
- Storage Area Networks use a network accessible device through a large bandwidth connection to provide storage facilities.

Network Virtualization

- Network virtualization combines hardware appliances and specific software for the creation and management of a virtual network.
- Network virtualization can aggregate different physical networks into a single logical network (*external* network virtualization) or provide network like functionality to an operating system partition (*internal* network virtualization).
- The result of external network virtualization is generally a *Virtual LAN (VLAN)*. A *VLAN* is an aggregation of hosts that communicate with each other as if they were located under the same broadcasting domain.
- Internal network virtualization is generally applied together with hardware and operating system level virtualization in which the guests obtain a virtual network interface to communicate with.
- There are several options for implementing internal network virtualization: the guest can share the same network interface of the host and use NAT to access the network; the virtual machine manager can emulate, and install on the host, an additional network device together with the driver; or the guest can have a private network only with the guest.

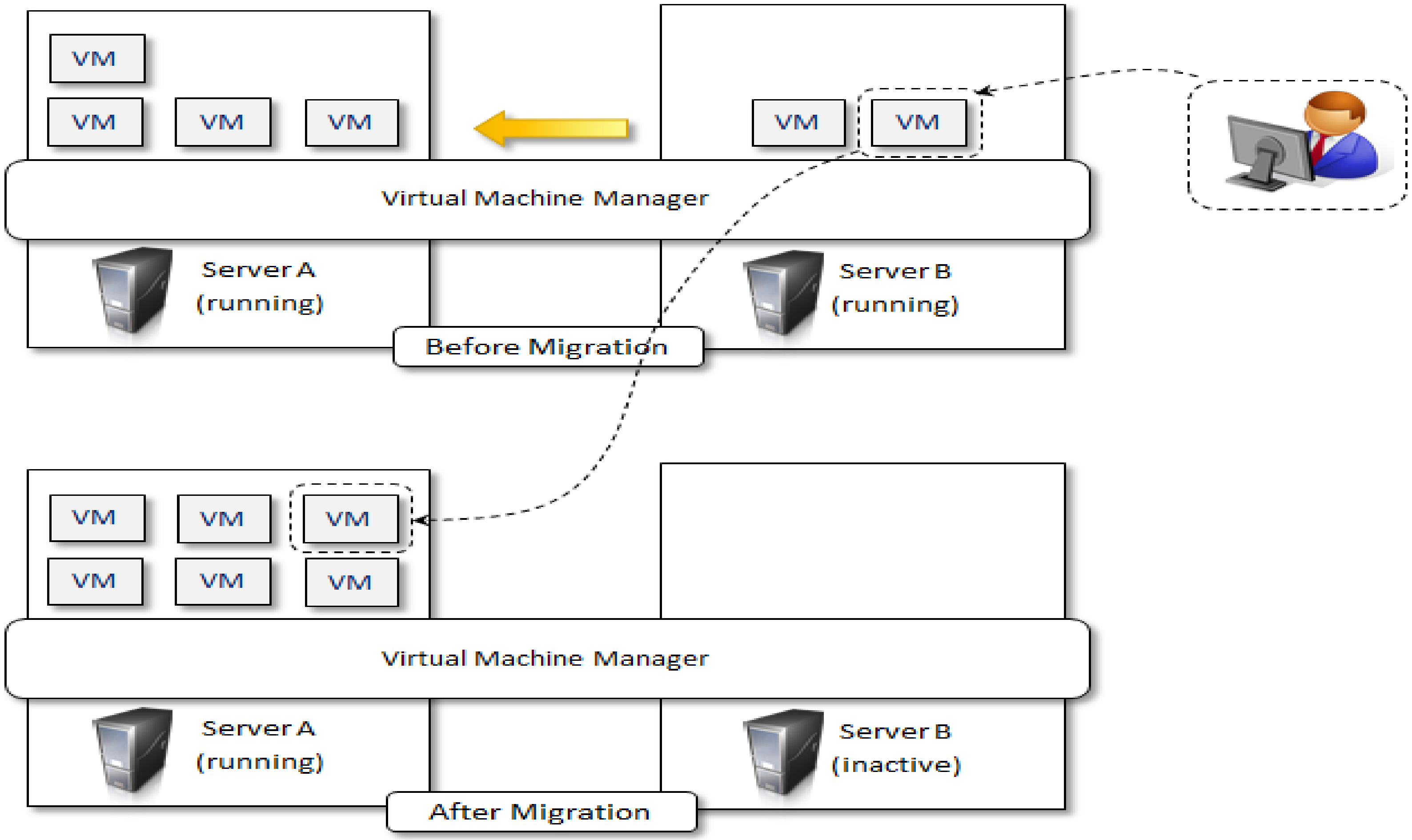
Desktop Virtualization

- Desktop virtualization abstracts the desktop environment available on a personal computer in order to provide access to it by using a client server approach.
- Desktop virtualization provides the same outcome of hardware virtualization but serves a different purpose.
- Similarly to hardware virtualization it makes accessible a different system as if it was natively installed on the host, but this system is remotely stored on a different host and accessed through a network connection.
- Moreover, desktop virtualization addresses the problem of making the same desktop environment accessible from everywhere.
- While the term desktop virtualization strictly refers to the ability to remotely access a desktop environment, generally, the desktop environment is stored in a remote server or a data center which provides a high availability infrastructure and ensures the accessibility and the persistence of the data.
- The basic services for remotely accessing a desktop environment are implemented in software components such as: *Windows Remote Services*, *VNC*, and *X Server*.
- Infrastructures for desktop virtualization based on Cloud computing solutions are: *Sun Virtual Desktop Infrastructure (VDI)*, *Parallels Virtual Desktop Infrastructure (VDI)*, *Citrix XenDesktop* and others.

Application Server Virtualization

- Application server virtualization abstracts a collection of application servers that provide the same services as a single virtual application server by using load balancing strategies and providing a high availability infrastructure for the services hosted in the application server.
- This is a particular form of virtualization and serves the same purpose of storage virtualization: providing a better quality of service rather than emulating a different environment.

Virtualization and Cloud Computing



Virtualization and Cloud Computing

- Virtualization plays an important role in Cloud computing, since it allows for the appropriate degree of customization, security, isolation.
- Virtualization technologies are primarily used to offer configurable computing environments and storage.
- Particularly important is the role of virtual computing environment and execution virtualization techniques. Among these, hardware and programming language virtualization are the techniques adopted in Cloud computing systems.
- virtualization also gives the opportunity of designing more efficient computing systems by means of consolidation
- Server consolidation and virtual machine migration are principally used in case of hardware virtualization even though technically possible also in case of programming language virtualization.
- Storage virtualization constitutes an interesting opportunity given by virtualization technologies, often complimentary to the execution virtualization.
- Finally, Cloud computing revamps the concept of desktop virtualization, initially introduced in the mainframe era.

Advantages of Virtualization

- Managed execution and isolation are the most important advantages of virtualization.
- These two characteristics allow building secure and controllable computing environments. A virtual execution environment can be configured as a sandbox, thus preventing any harmful operation to cross the borders of the virtual host.
- Moreover, allocation of resources and their partitioning among different guests is simplified, being the virtual host controlled by a program.

- Portability is another advantage of virtualization, especially for execution virtualization techniques.
- Portability and self-containment also contribute to reduce the costs for maintenance, since the number of hosts is expected to be lower than the number of virtual machine instances.
- it is possible to achieve a more efficient use of resources. Multiple systems can securely coexist and share the resources of the underlying host, without interfering with each other.

Disadvantages of Virtualization

- Performance degradation
- Inefficiency and degraded user experience
- Security holes and new threats
- Migration Issues

Thank You