Module 2- Virtualization

Basics of Virtualization - Types of Virtualizations, Taxonomy of Virtualization Techniques, Implementation Levels of Virtualization

Contents

3.1 Basics of Virtualization

3.1.1 Introduction

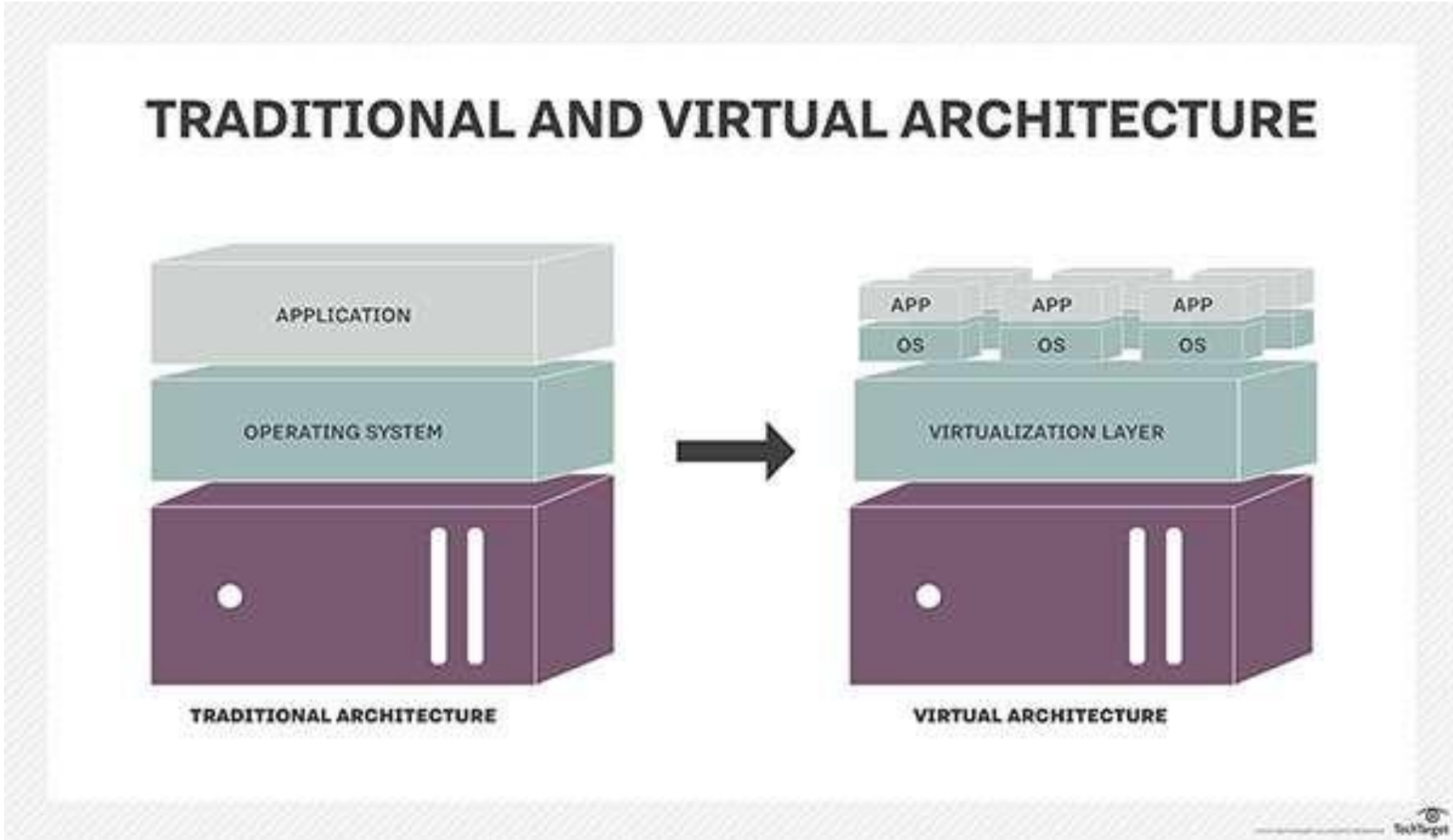
Virtualization

- **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

What is Virtualization?

Rockwell Automation

- 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

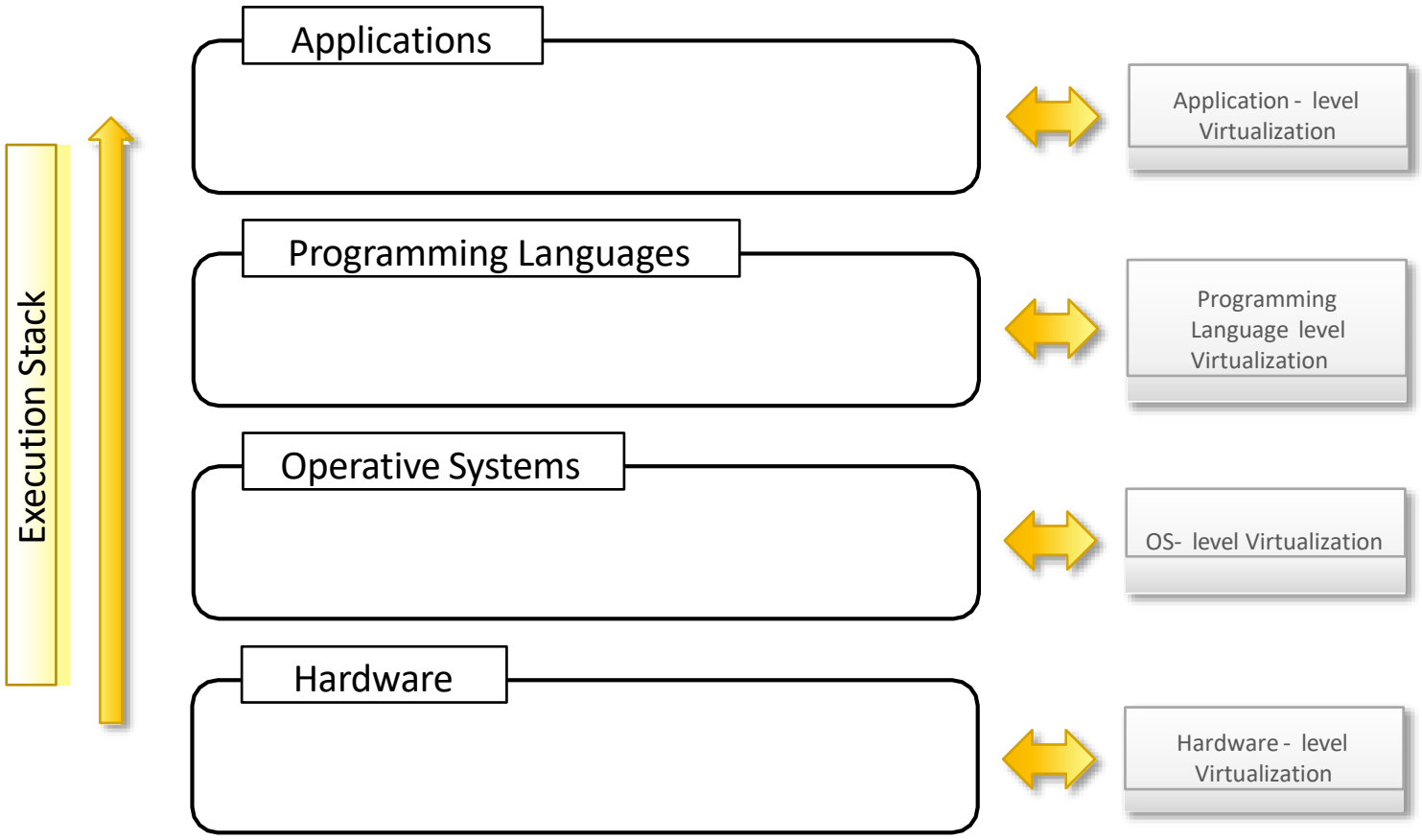

TRADITIONAL AND VIRTUAL ARCHITECTURE
APPLICATION
OPERATING SYSTEM
TRADITIONAL ARCHITECTURE
APP
APP
APP
OS
OS
OS
VIRTUALIZATION LAYER
VIRTUAL ARCHITECTURE

Why are virtualized environments so popular today?

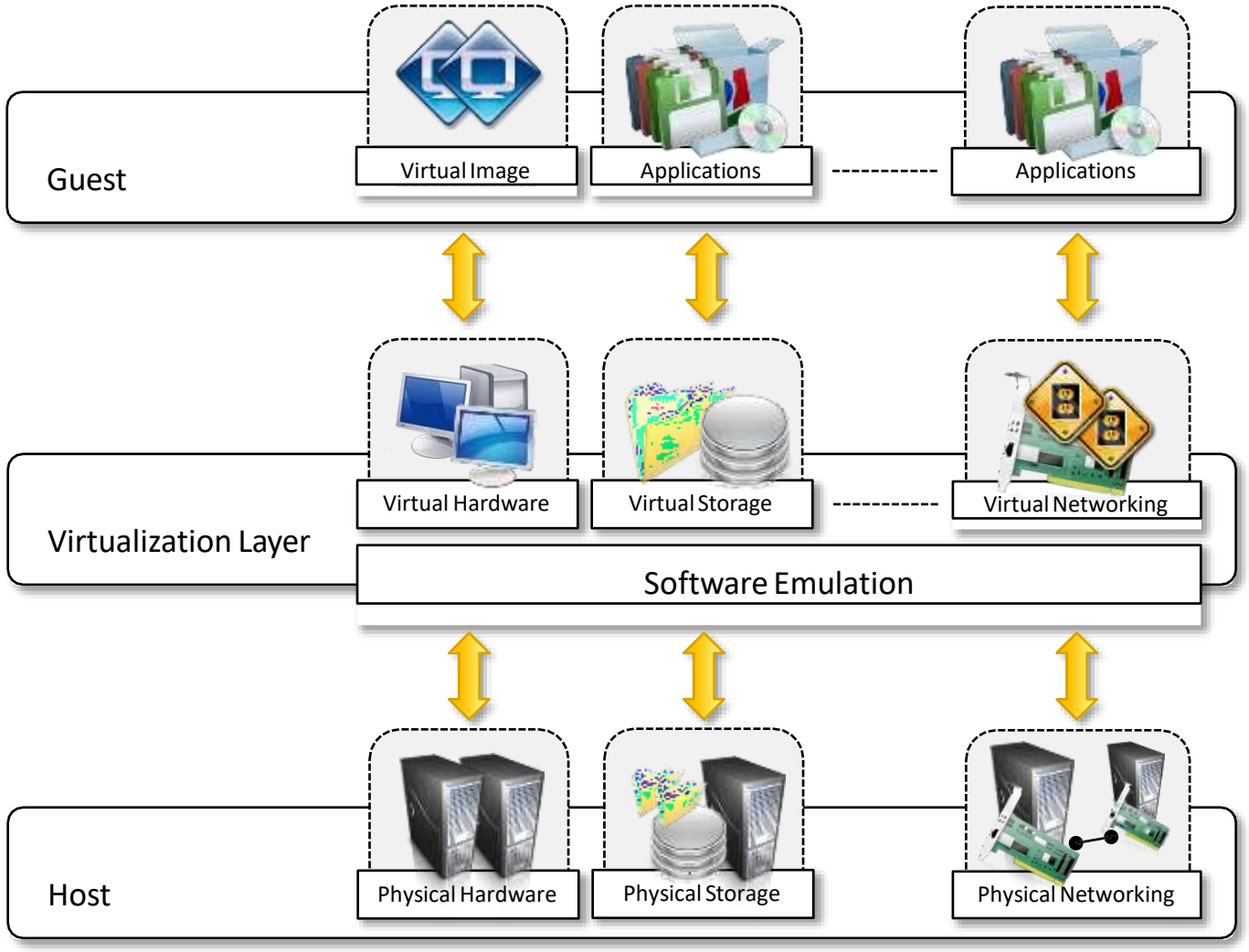
- **Increased performance and computing capacity**
  - PCs are having immense computing power.
- **Underutilized hardware and software resources**
  - Limited use of increased performance & computing capacity.
- **Lack of space**
  - Continuous need for additional capacity.
- **Greening initiatives**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

Virtualized Environments

- Virtualization is a method of logically dividing the system resources between different applications
- Application Virtualization
- Desktop Virtualization
- Server Virtualization
- Network Virtualization
- Storage Virtualization


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

- Three major components of Virtualized Environments
  - **Guest** – system component that interacts with Virtualization Layer.
  - **Host** – The host represents the original environment where the guest is supposed to be managed.
  - **Virtualization Layer** –The virtualization layer is responsible for recreating the same or a different environment where the guest will operate.



Virtualization Reference Model

3.1.2 Characteristics of VE

- Increased Security
- Managed Execution

✓ - Sharing

✓ - Aggregation

✓ - Emulation

✓ - Isolation

- Portability

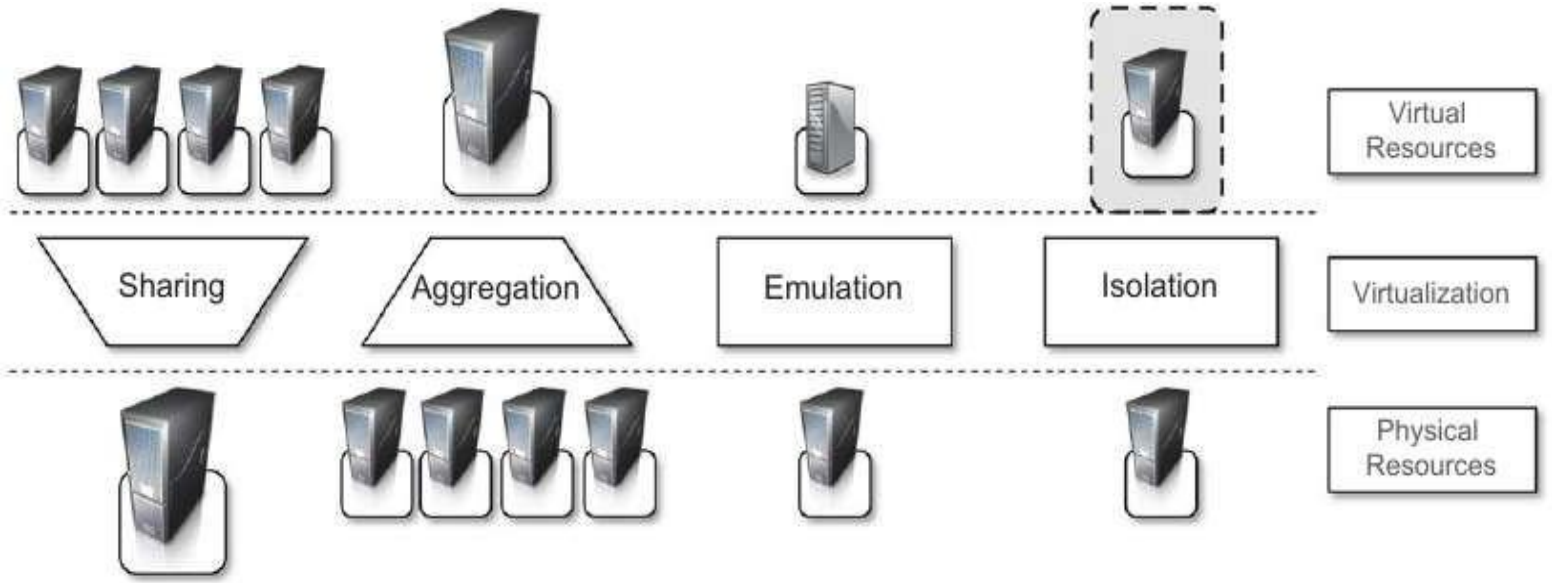
Increased Security

- 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

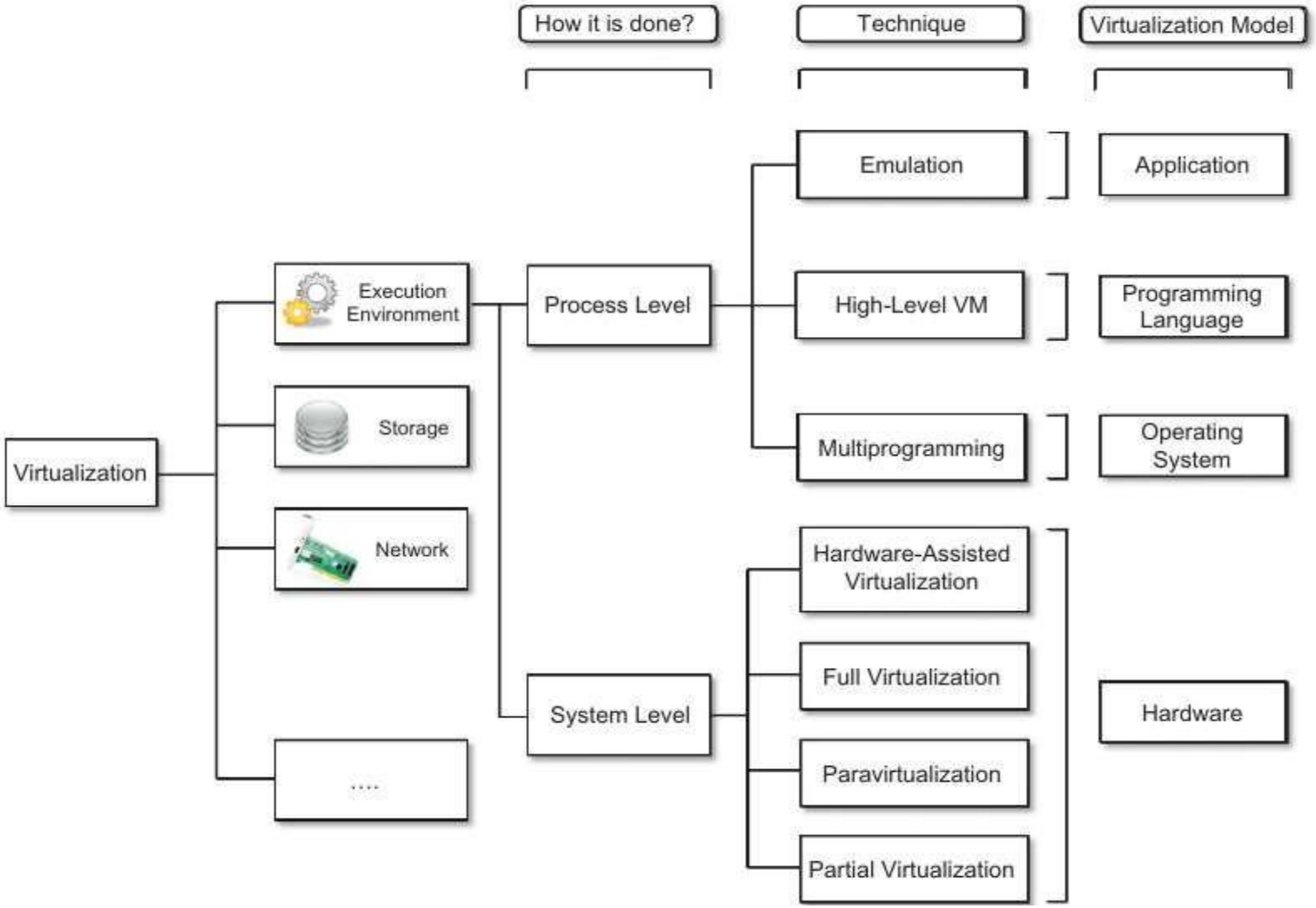
- safely moved and executed on top of different virtual machine.
- Application Development Cycle more flexible and application deployment very straight forward
- Availability of system is with you.

3.3 Taxonomy of Virtualization Techniques

Taxonomy of Virtualization Techniques

- 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 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

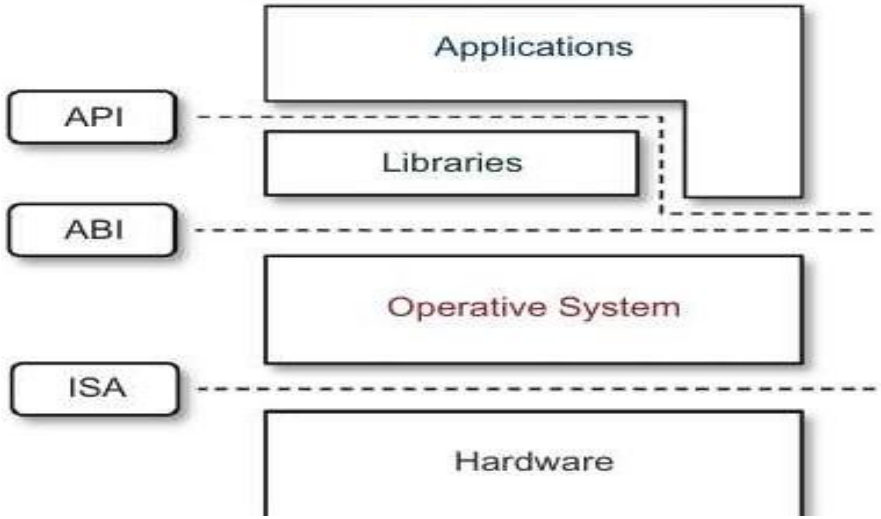
Taxonomy of virtualization

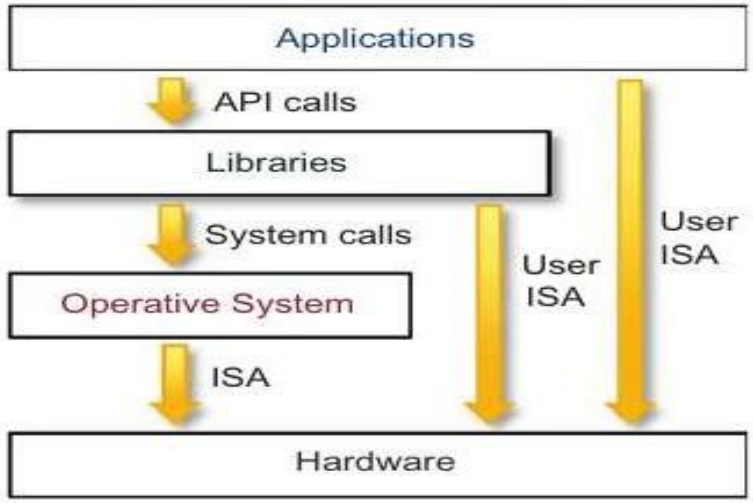


3.3.1 Execution Virtualization

- 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.

1. Machine Reference Model





- The model for 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.

Machine Reference Model [Cont.]

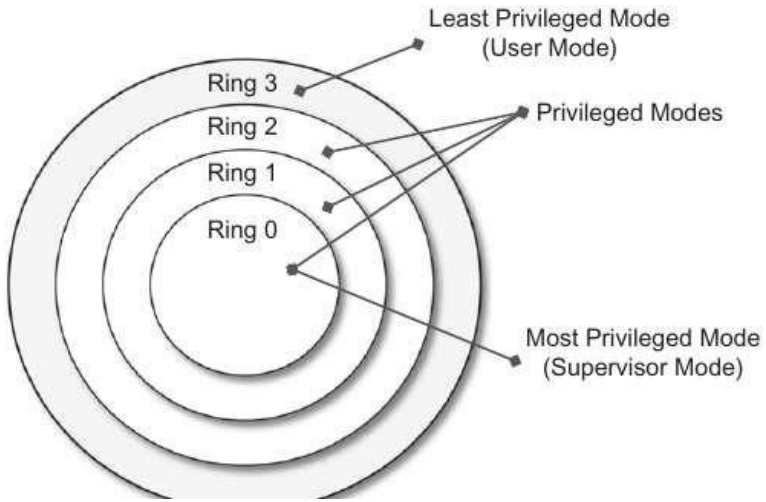
- API – it interfaces applications to libraries and/or the underlying OS.
- Layered approach simplifies the development and implementation of computing system.
- ISA has been divided into two security classes:-
  - ***Privileged Instructions***
  - ***Nonprivileged Instructions***

ISA: Security Classes

- **Nonprivileged instructions**
  - That can be used without interfering with other tasks because they ***do not access shared resources***. Ex. Arithmetic , floating & fixed point.
- **Privileged instructions**
  - That are executed under ***specific restrictions*** and are mostly used for ***sensitive operations***, which expose (*behavior-sensitive*) or modify (*control-sensitive*) the privileged state.
    - **Behavior-sensitive** – operate on the I/O
    - **Control-sensitive** – alter the state of the CPU register.

Privileged Hierarchy: Security Ring

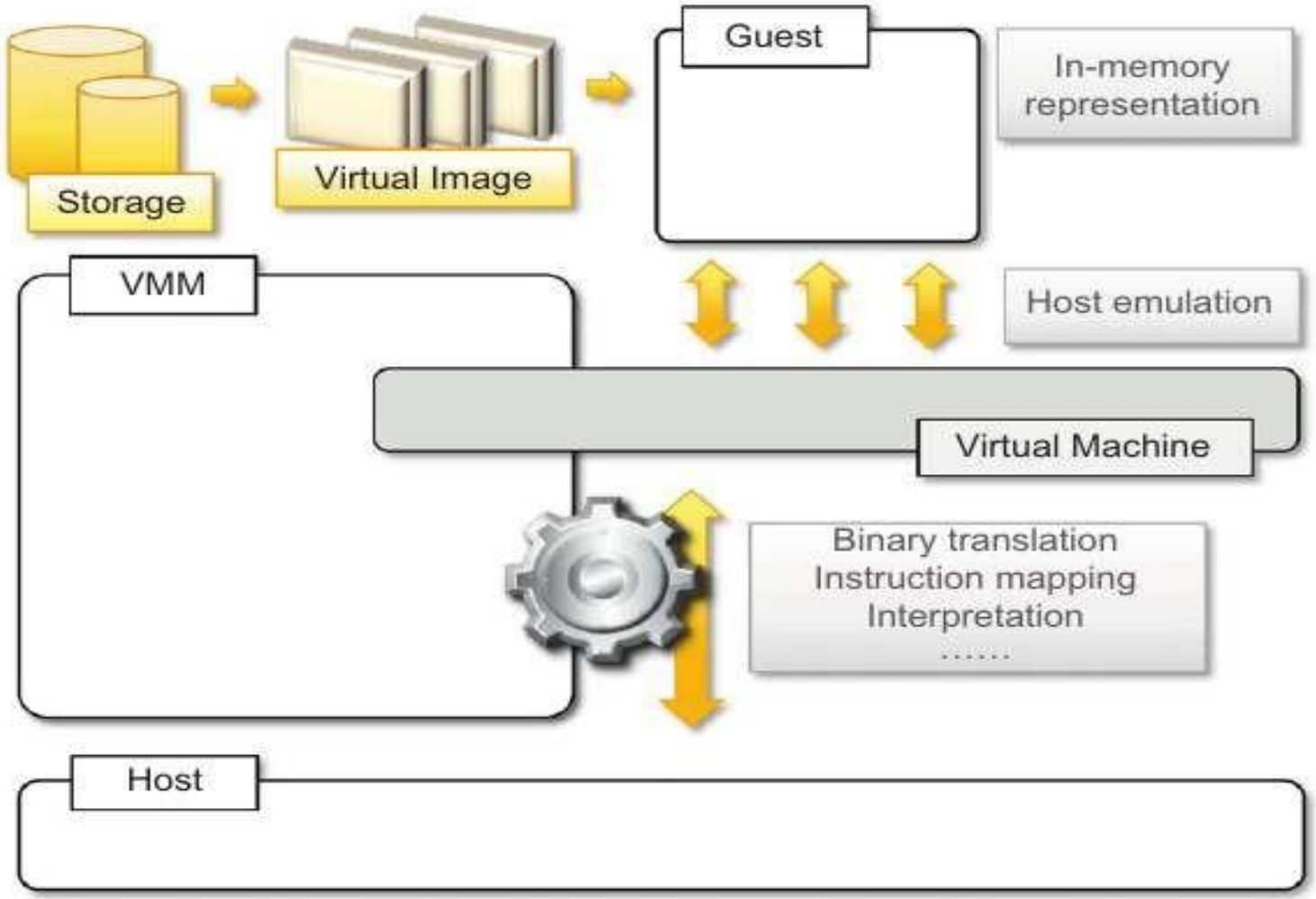
- *Ring-0* is in most privileged level , used by the kernel.
- *Ring-1 & 2* used by the OS-level services
- and , *R3* in the least privileged level , used by the user.
- Recent system support two levels :-
  - **Ring 0 – *supervisor mode***
  - **Ring 3 – *user mode***



2. 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.
- A fundamental element of hardware virtualization is the hypervisor, or Virtual Machine Manager (VMM).
- It recreates a hardware environment, where guest operating systems are installed.

Hardware-level virtualization

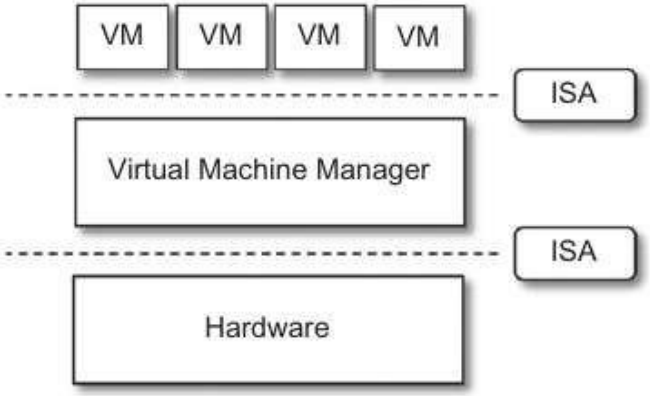


Hypervisor

- 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

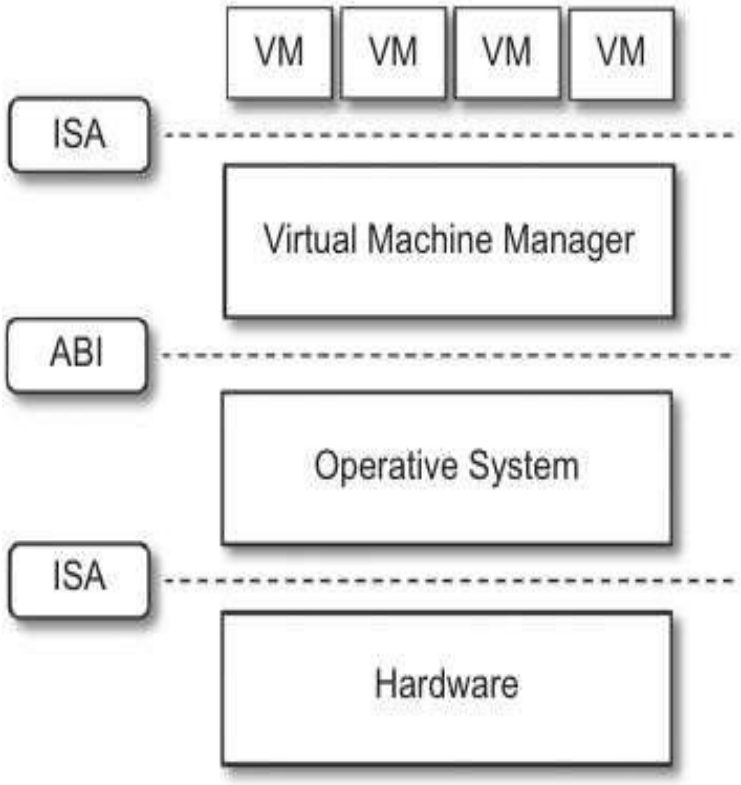
- It runs directly on top of the hardware.
- Takes place of OS.
- Directly interact with the ISA exposed by the underlying hardware.



- Also known as *native virtual machine*.

Type-II Hypervisor

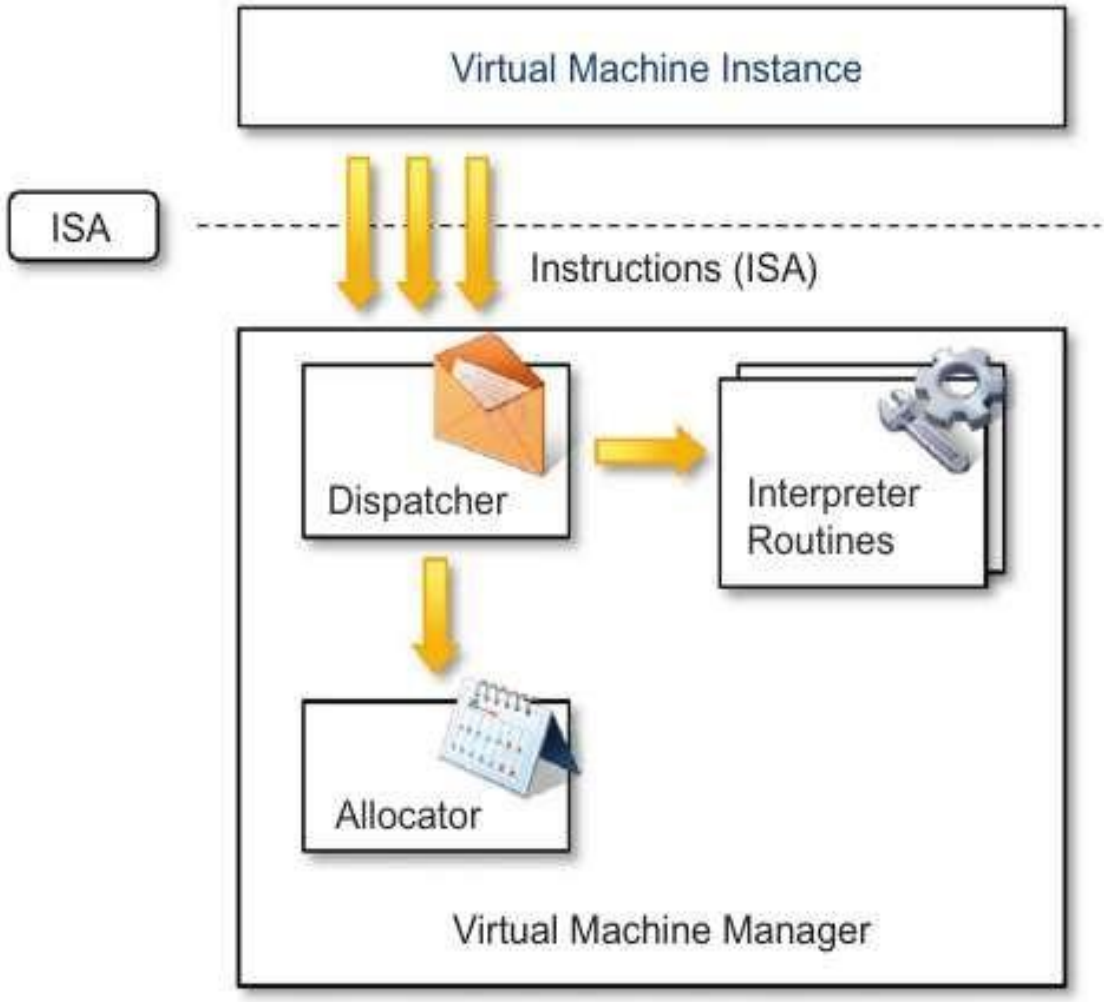
- It require the support of an operating system to provide virtualization services.
- Programs managed by the OS.
- Emulate the ISA of virtual h/w.
- Also called hosted virtual machine.



Virtual Machine Manager (VMM)

- Main Modules :-
  - **Dispatcher**
    - Entry Point of VMM
    - Reroutes the instructions issued by VM instance.
  - **Allocator**
    - Deciding the system resources to be provided to the VM.
    - Invoked by dispatcher
  - **Interpreter**
    - Consists of interpreter routines
    - Executed whenever a VM executes a privileged instruction.
    - Trap is triggered and the corresponding routine is executed.

Virtual Machine Manager (VMM)



Criteria of VMM

- 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 as when executed directly on the physical host.

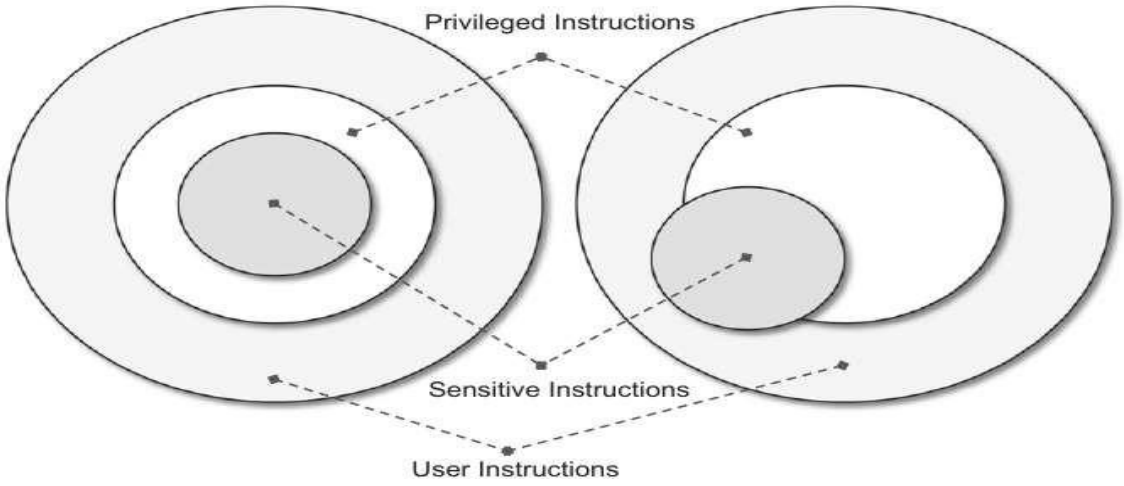
- **Resource control** – The virtual machine manager should be in complete control of virtualized resources.
- **Efficiency** – a statistically dominant fraction of the machine instructions should be ***executed without intervention*** from the VMM. All safe guest instructions are executed by the hardware directly.

Theorems

- 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.
- Classification of IS-
  - Privileged Instructions
    - Those that trap if the processor is in user mode and do not trap if it is in system mode (supervisor mode).
    - Control sensitive Instructions
    - Those that attempt to change the configuration of resources in the system.

Theorems-1

- Theorems 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.



Theorems

- Theorems 2
  - A conventional third-generation computers is recursively virtualizable if:
    - It is virtualizable and
    - A VMM without any timing dependencies can be constructed for it.

Theorems

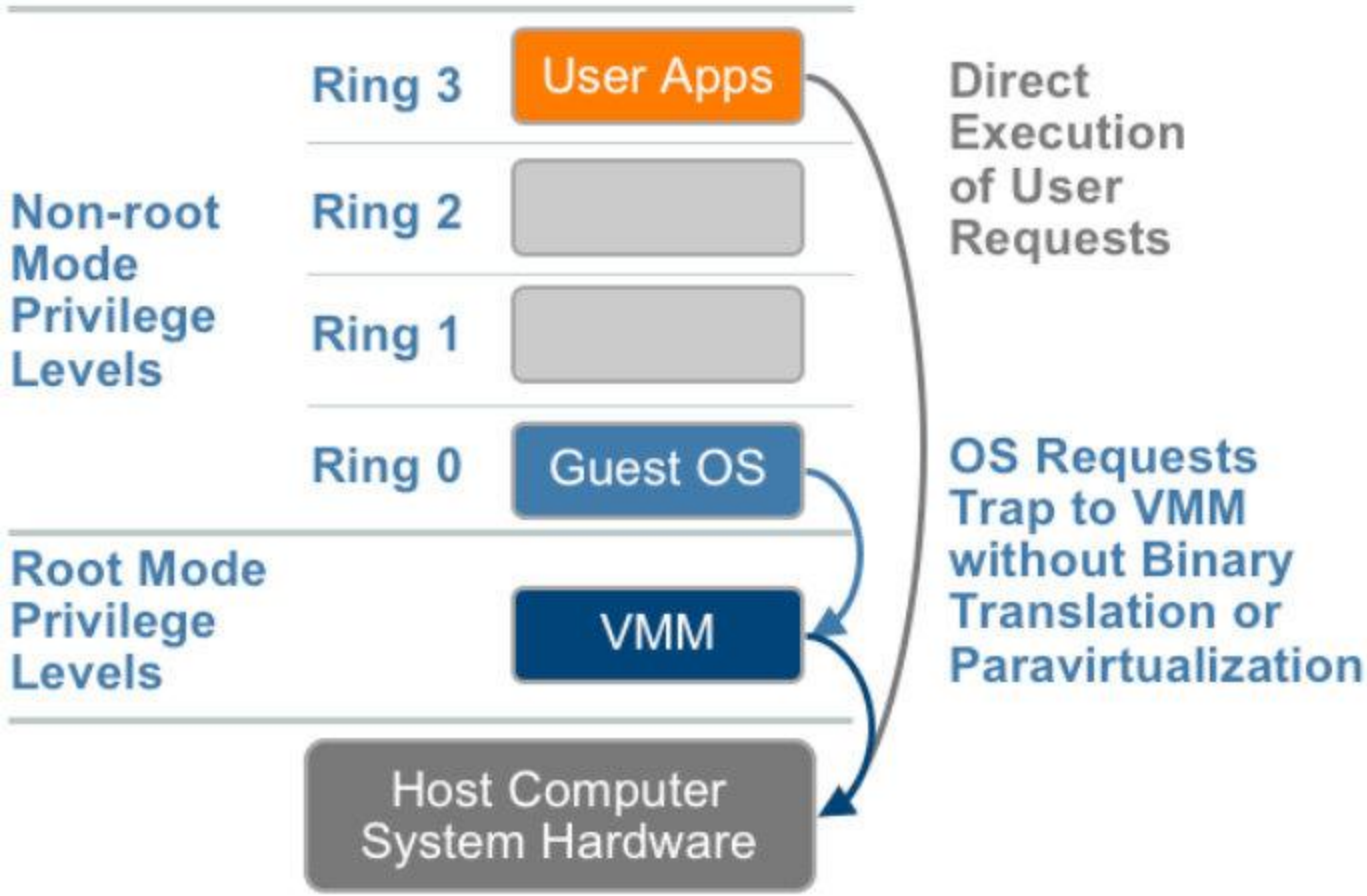
- Theorems 3
  - A hybrid VMM may be constructed third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.
  - *In HVM, more instructions are interpreted rather than being executed directly.*

3. 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**
  - In this hardware provides architectural support for building a VMM able to run a guest OS in complete isolation.
  - ***Intel VT*** and ***AMD V*** extensions.
  - Early products were using ***binary translation to trap some sensitive instructions*** and provide an emulated version

- Also known as native virtualization, in this technique, **underlying hardware provides special CPU instructions to aid virtualization**.
- This technique is also highly portable as the hypervisor can run an **unmodified guest OS.** This technique makes hypervisor implementation less complex and more maintainable.

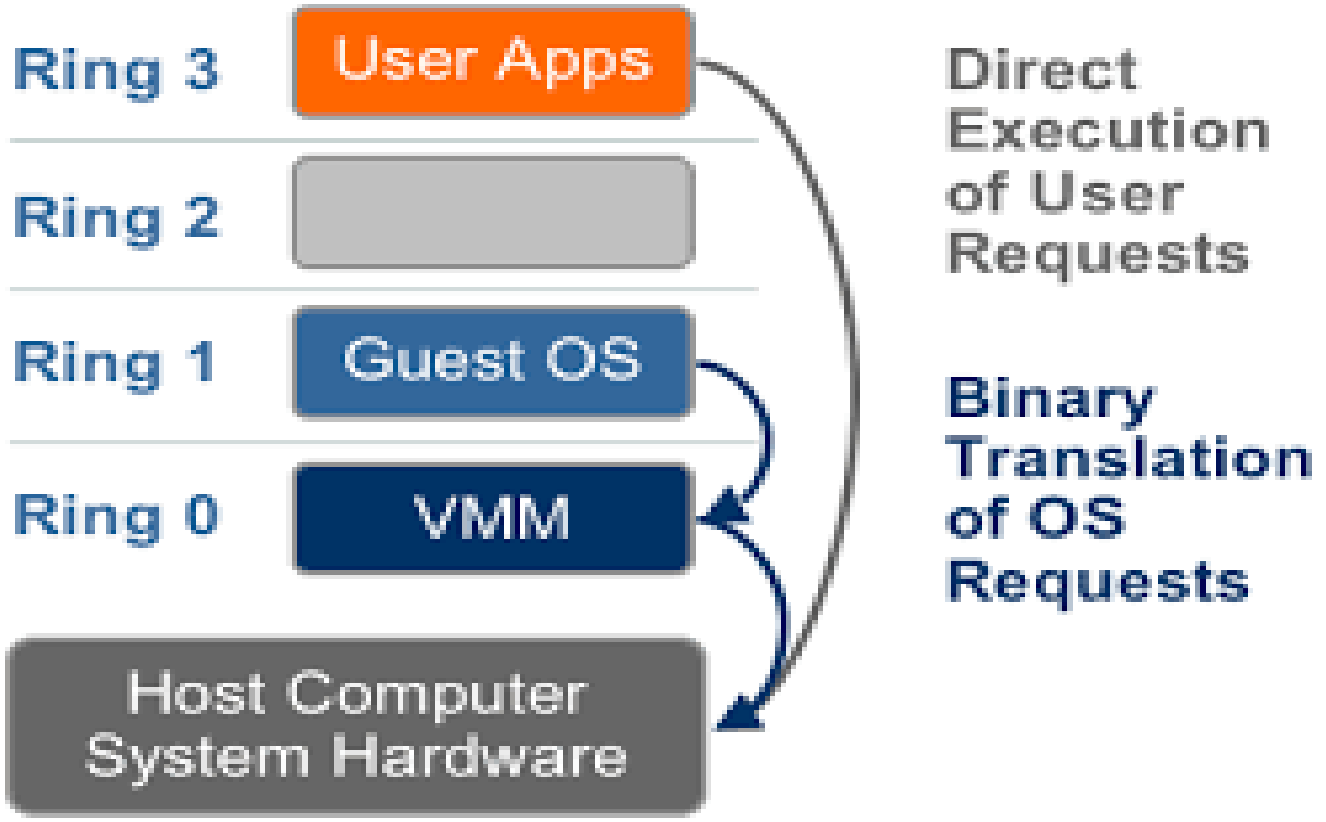

Ring 3
User Apps
Ring 2
Ring 1
Ring 0
Guest OS
Non-root Mode Privilege Levels
Root Mode Privilege Levels
VMM
Host Computer System Hardware
Direct Execution of User Requests
OS Requests Trap to VMM without Binary Translation or Paravirtualization

- **Full virtualization**
  - Ability to run program (OS) directly on ***top of a virtual machine*** and without any modification.
  - VMM ***require complete emulation*** of the entire underneath h/w
  - **Advantages**
    - Complete isolation
    - Enhanced security
    - Ease of emulation of different architectures and coexistence
  - ***Key challenge is interception of privileged instructions***

- This process was introduced by IBM in the year 1966. It is considered to be the first software solution for server virtualization. It uses binary translation and a direct approach method.

- In this, the guest OS is fully isolated using the virtual machine from the virtualization layer and hardware.

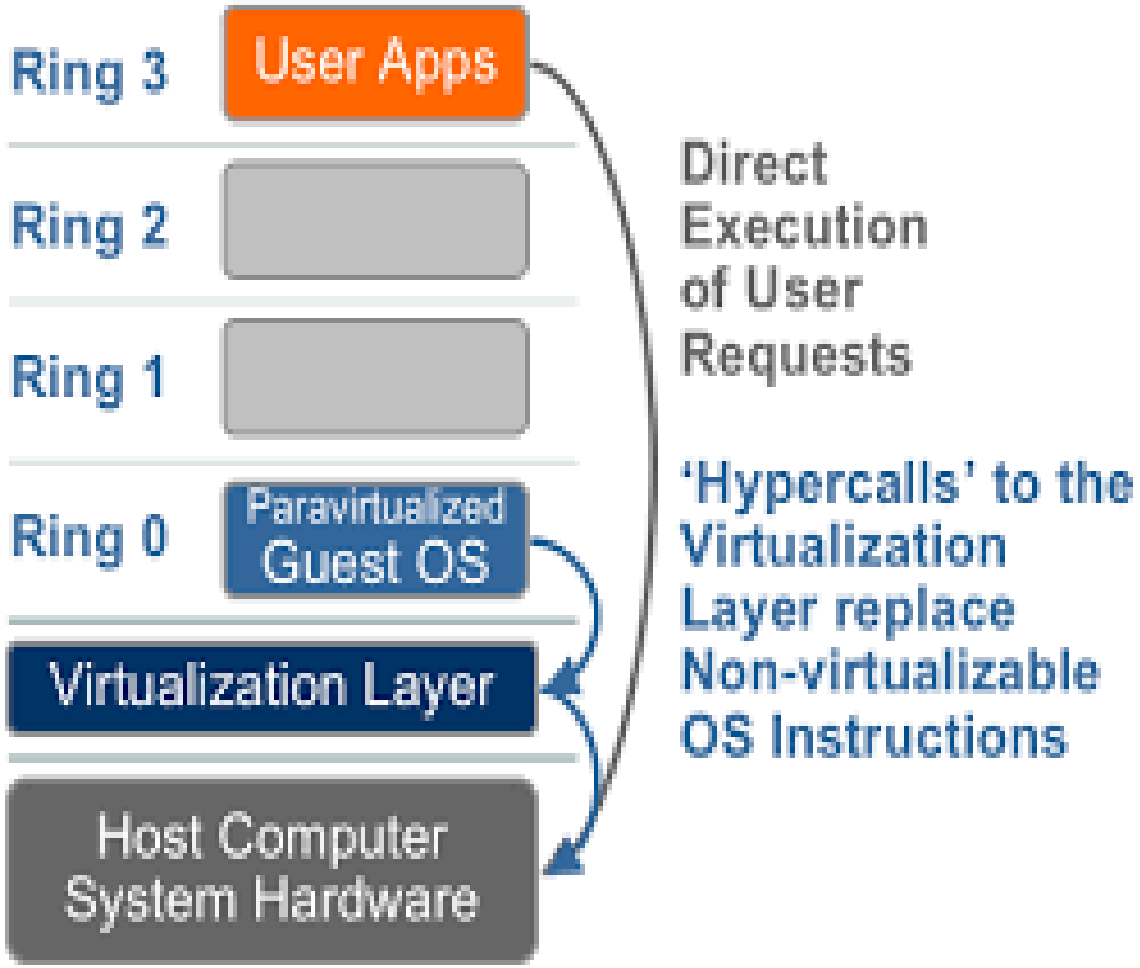
- Examples of full virtualization include Microsoft and Parallels systems.
- It is considered to be less secure in comparison to paravirtualization.


Ring 3
User Apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer
System Hardware
Direct
Execution
of User
Requests
Binary
Translation
of OS
Requests

- **Paravirtualization**
  - Not-transparent virtualization
  - Thin VMM
  - Expose software interface to the virtual machine that is slightly modified from the host.
  - Guest OS need to be modified.
  - Simply transfer the execution of instructions which were hard to virtualized, directly to the host.

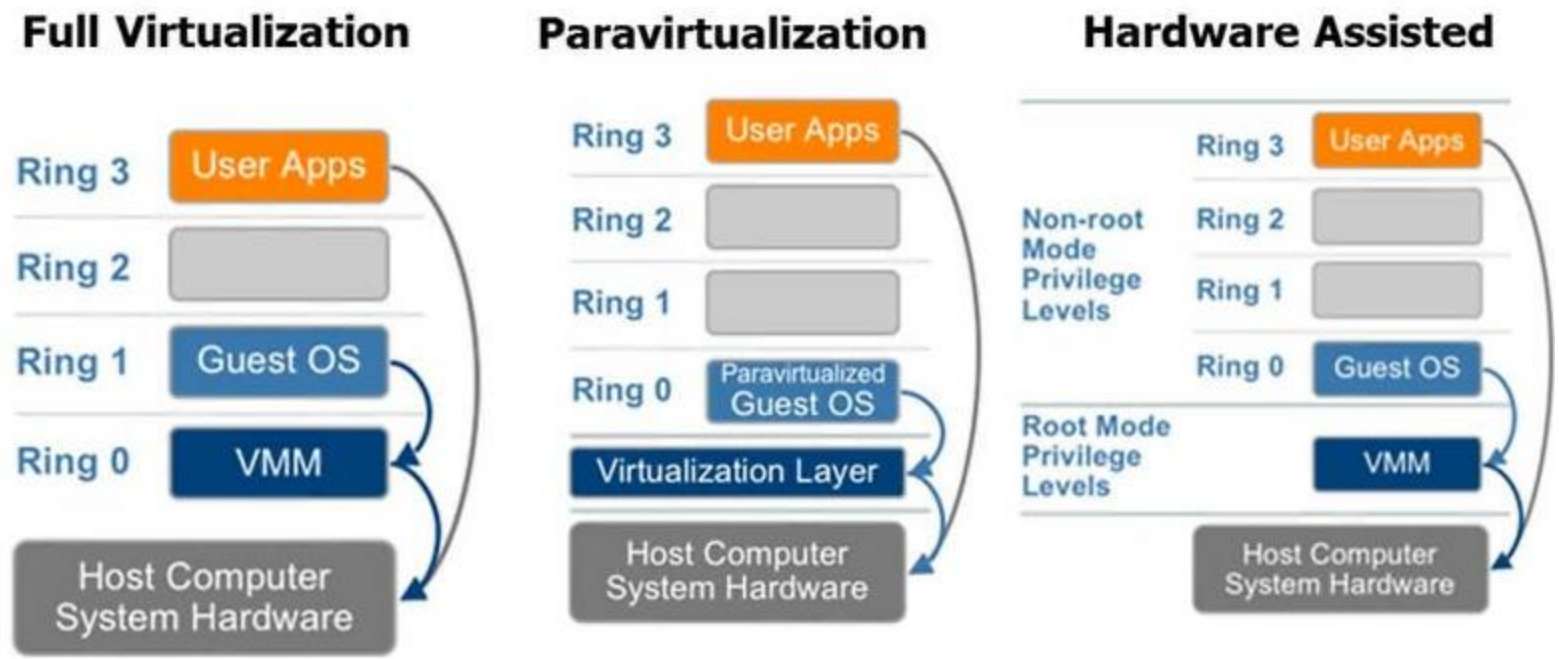
- Paravirtualization is the category of CPU virtualization which uses hyper calls for operations to handle instructions at compile time. **In paravirtualization, guest OS is not completely isolated but it is partially isolated by the virtual machine from the virtualization layer and hardware.**

- VMware and Xen are some examples of paravirtualization.


Ring 3
User Apps
Ring 2
Ring 1
Ring 0
Paravirtualized
Guest OS
Virtualization Layer
Host Computer
System Hardware
Direct
Execution
of User
Requests
'Hypercalls' to the
Virtualization
Layer replace
Non-virtualizable
OS Instructions

- **Partial virtualization**
  - Partial emulation of the underlying hardware
  - Not allow complete isolation to guest OS.
  - Address space virtualization is a common feature of comtemporary operating systems.
  - Address space virtualization used in time-sharing system.

Architectural Comparison



Comparison between the Full Virtualization and paravirtualization in Operating System

S.no	Full Virtualization	ParaVirtualization
1	In Full virtualization, virtual machines permit the execution of the instructions with the running of unmodified OS in an entirely isolated way.	In paravirtualization, a virtual machine does not implement full isolation of OS but rather provides a different API which is utilized when OS is subjected to alteration.
2.	Full Virtualization is less secure.	While the Paravirtualization is more secure than the Full Virtualization.
3.	Full Virtualization uses binary translation and a direct approach as a technique for operations.	While Paravirtualization uses hypercalls at compile time for operations.
4.	Full Virtualization is slow than paravirtualization in operation.	Paravirtualization is faster in operation as compared to full virtualization.

S.No	Full Virtualization	ParaVirtualization
5.	Full Virtualization is more portable and compatible.	Paravirtualization is less portable and compatible.
6.	Examples of full virtualization are Microsoft and Parallels systems.	Examples of paravirtualization are Microsoft Hyper-V, Citrix Xen, etc.
7.	It supports all guest operating systems without modification.	The guest operating system has to be modified and only a few operating systems support it.
8.	The guest operating system will issue hardware calls.	Using the drivers, the guest operating system will directly communicate with the hypervisor.
9.	It is less streamlined compared to para-virtualization.	It is more streamlined.
10.	It provides the best isolation.	It provides less isolation compared to full virtualization.

4. 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.*

5. 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

6. 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

3.3.2 Types of Virtualization

1. Application Server Virtualization

- Application virtualization helps a user to have remote access of an application from a server.
- The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet.
- Example of this would be a user who needs to run two different versions of the same software.
- Technologies that use application virtualization are hosted applications and packaged applications.

2. 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.
- The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.

2. Network Virtualization

- Network virtualization provides a facility to create and provision virtual networks—logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

Examples of Network Virtualization :

Virtual LAN (VLAN) –

- The performance and speed of busy networks can be improved by VLAN.
- VLAN can simplify additions or any changes to the network.

3. Desktop Virtualization

- Desktop virtualization allows the users' OS to be remotely stored on a server in the data centre. It allows the user to access their desktop virtually, from any location by a different machine.
- Users who want specific operating systems other than Windows Server will need to have a virtual desktop.
- Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates, and patches.

4. Storage Virtualization

- Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren’t aware of exactly where their data is stored, and instead function more like worker bees in a hive.
- It makes managing storage from multiple sources to be managed and utilized as a single repository.
- storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

5. Server Virtualization

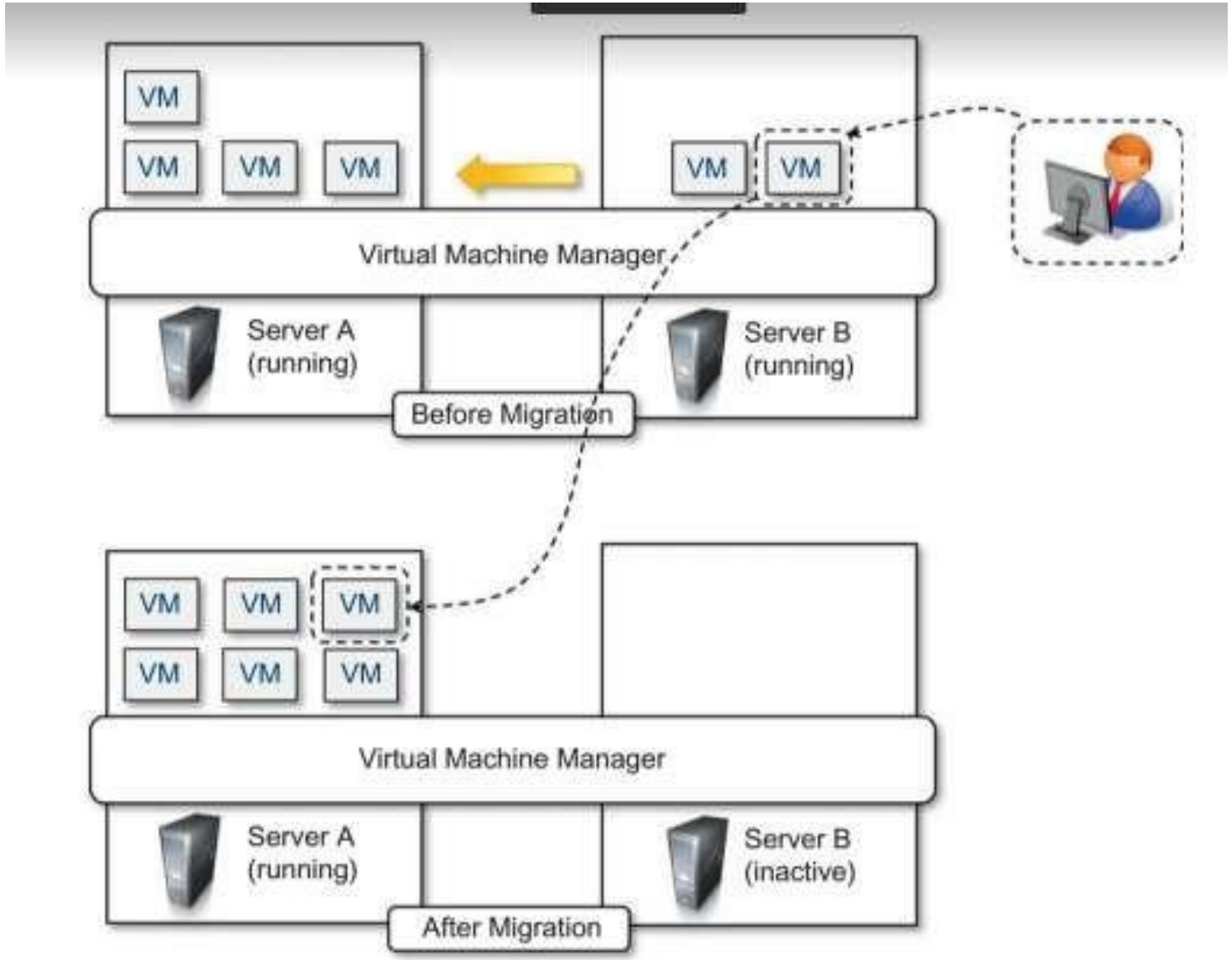
- The central-server(physical server) is divided into multiple different virtual servers by changing the identity number, processors. So, each system can operate its own operating systems in isolate manner. Where each sub-server knows the identity of the central server.
- It causes an increase in the performance and reduces the operating cost by the deployment of main server resources into a sub-server resource.
- It’s beneficial in virtual migration, reduce energy consumption, reduce infrastructural cost, etc.

6. Data Virtualization

- This can be defined as the type of Virtualization wherein data are sourced and collected from several sources and managed from a single location.
- No technical knowledge from where such data is sourced and collected, stored, or formatted for such data.
- The data is arranged logically, and the interested parties and stakeholders then access the virtual view of such data. These are reports are also accessed by end-users on a remote basis.

3.4 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

✓ 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**

✓ Software licensing considerations

✓ Possible learning curve

✓ Security holes and new threats

3.4 Implementation levels of virtualization

Levels of Virtualization

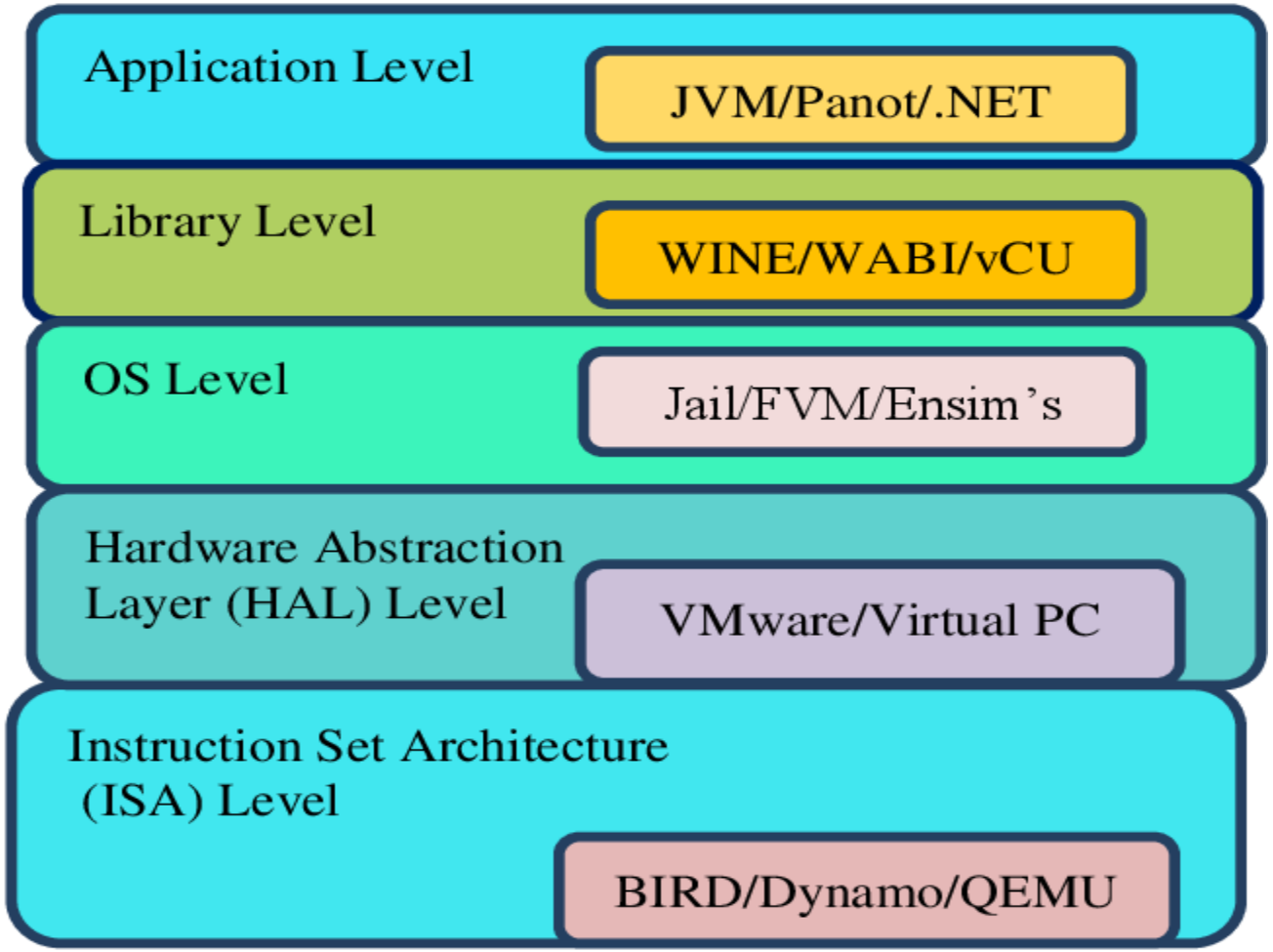


Fig. 2: Implementation Levels of Virtualization.

Levels of Virtualization

1) Instruction Set Architecture Level (ISA)

- ISA virtualization can work through ISA emulation. This is used to run many legacy codes that were written for a different configuration of hardware. These codes run on any virtual machine using the ISA.
- With this, a binary code that originally needed some additional layers to run is now capable of running on the x86 machines. It can also be tweaked to run on the x64 machine. With ISA, it is possible to make the virtual machine hardware agnostic.

- For the basic emulation, an interpreter is needed, which interprets the source code and then converts it into a hardware format that can be read. This then allows processing. This is one of the five implementation levels of virtualization in cloud computing.

Levels of Virtualization

2) Hardware Abstraction Level (HAL)

- HAL lets the virtualization perform at the level of the hardware. This makes use of a hypervisor which is used for functioning.
- At this level, the virtual machine is formed, and this manages the hardware using the process of virtualization.
- It allows the virtualization of each of the hardware components, which could be the input-output device, the memory, the processor, etc.
- Multiple users will not be able to use the same hardware and also use multiple virtualization instances at the very same time. This is mstly used in the cloud-based infrastructure.

Levels of Virtualization

3) Operating System Level

- At the level of the operating system, the virtualization model is capable of creating a layer that is abstract between the operating system and the application. This is an isolated container that is on the operating system and the physical server, which makes use of the software and hardware. Each of these then functions in the form of a server.
- When there are several users, and no one wants to share the hardware, then this is where the virtualization level is used. Every user will get his virtual environment using a virtual hardware resource that is dedicated. In this way, there is no question of any conflict.

Levels of Virtualization

4) Library Level

- The operating system is cumbersome, and this is when the applications make use of the API that is from the libraries at a user level. These APIs are documented well, and this is why the library virtualization level is preferred in these scenarios. API hooks make it possible as it controls the link of communication from the application to the system.

Levels of Virtualization

5) Application Level

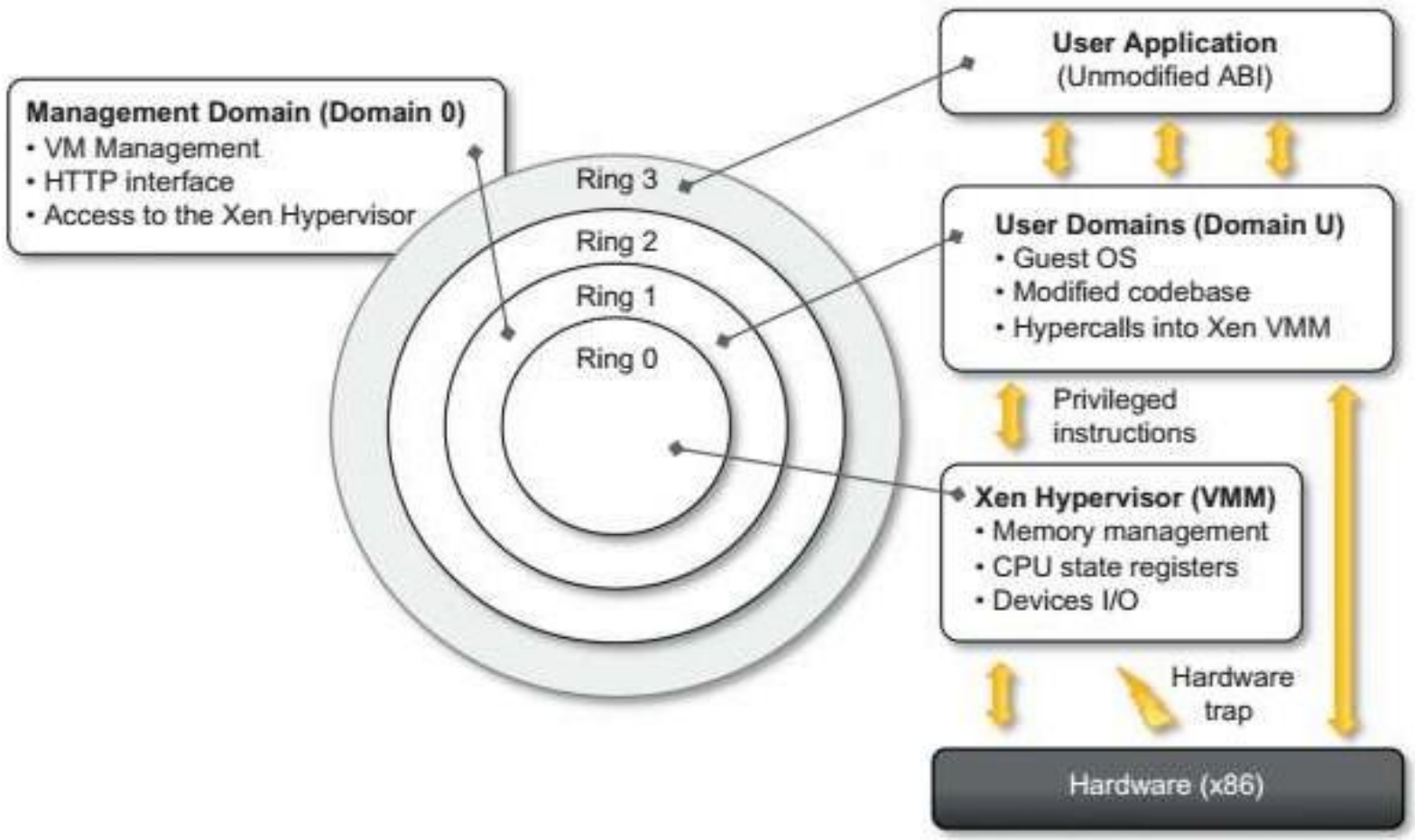
- The application-level virtualization is used when there is a desire to virtualize only one application and is the last of the implementation levels of virtualization in cloud computing. One does not need to virtualize the entire environment of the platform.
- This is generally used when you run virtual machines that use high-level languages. The application will sit above the virtualization layer, which in turn sits on the application program.
- It lets the high-level language programs compiled to be used in the application level of the virtual machine run seamlessly.

Technology examples

- Xen: paravirtualization
- VMware: full virtualization
- Microsoft Hyper-V

Xen: paravirtualization

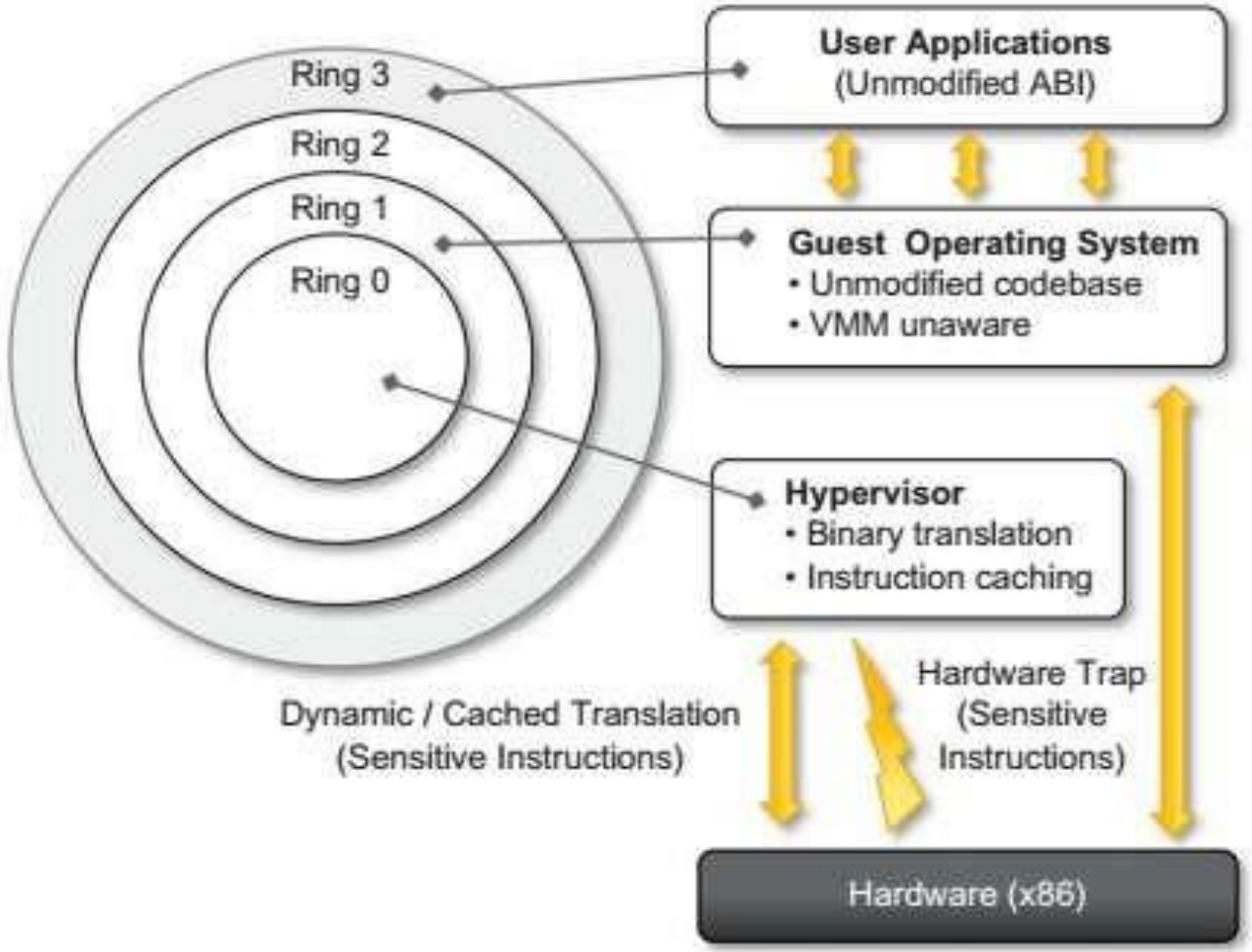
- Xen is an open-source initiative
- Developed by a group of researchers at the University of Cambridge
- XenSource.
- Desktop virtualization or server virtualization
- Xen Cloud Platform (XCP)
- https://www.xenproject.org/



Xen architecture and guest OS management.

VMWare: Full Virtualization

- Underlying hardware is replicated and made available to the guest operating system
- VMware implements full virtualization in the Desktop environments
- Type II hypervisor in Server Environment
- Type I hypervisor in Desktop and Server Environments
- Direct Execution
- Binary Translation

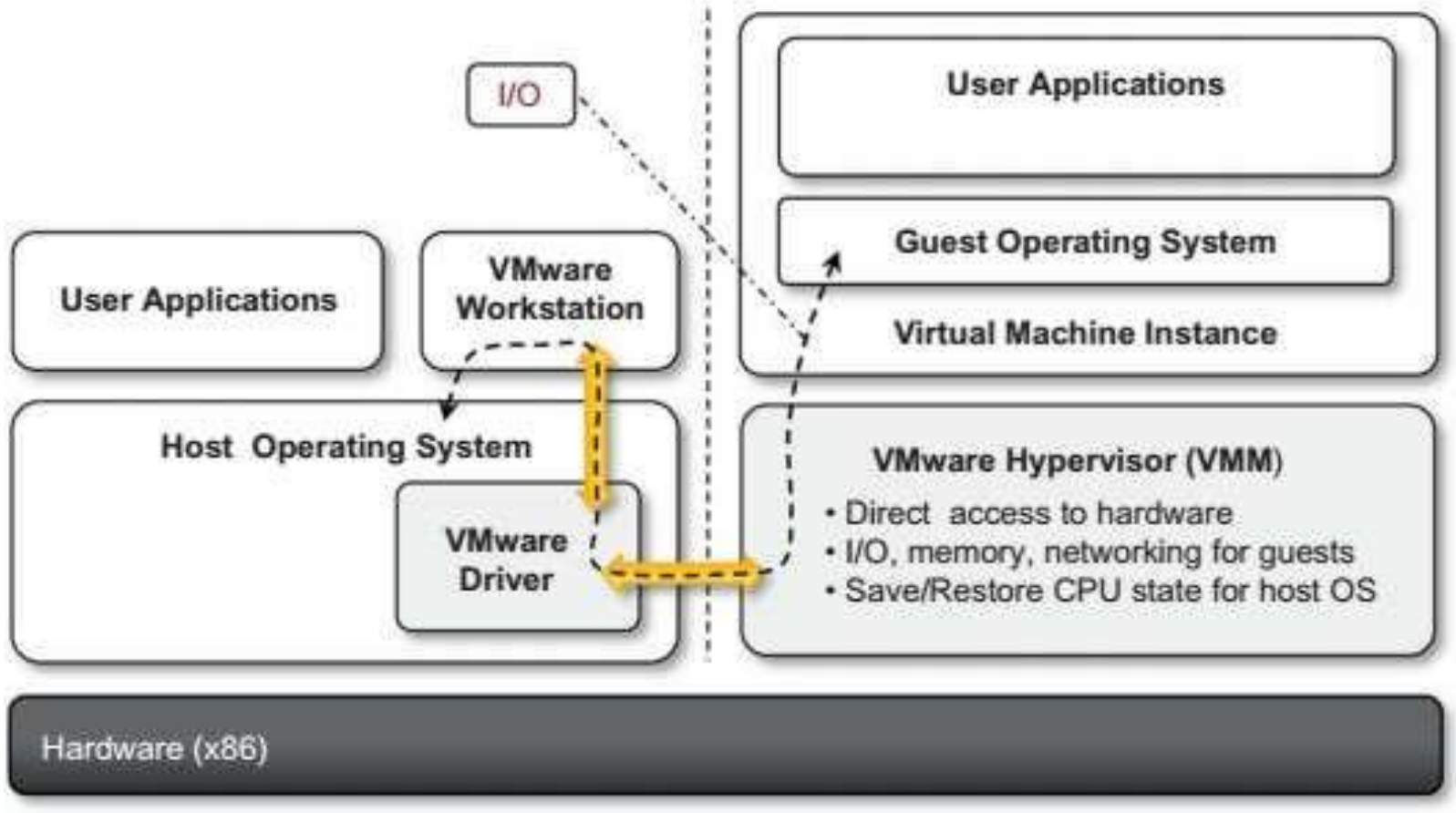


A full virtualization reference model.

Virtualization solutions by VMware

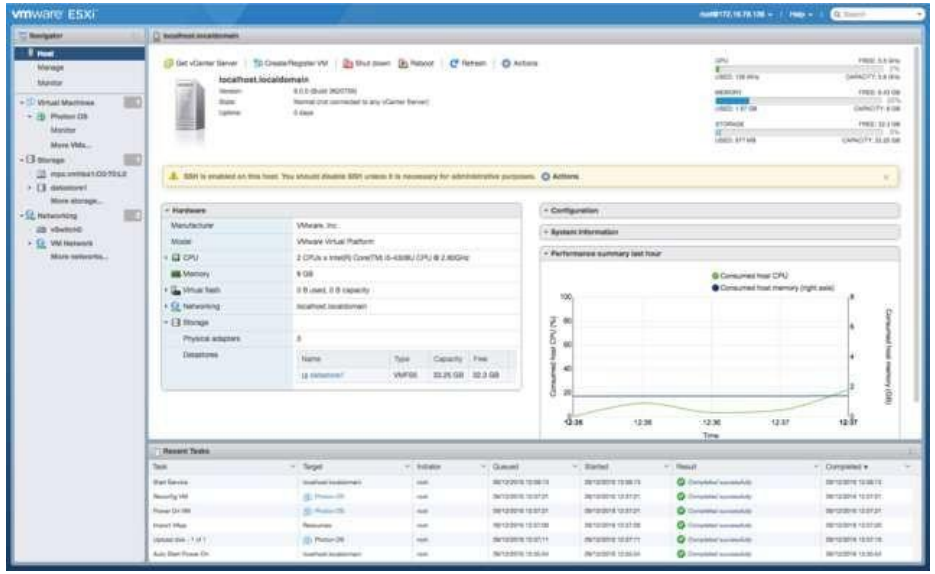
- End-user (desktop) virtualization

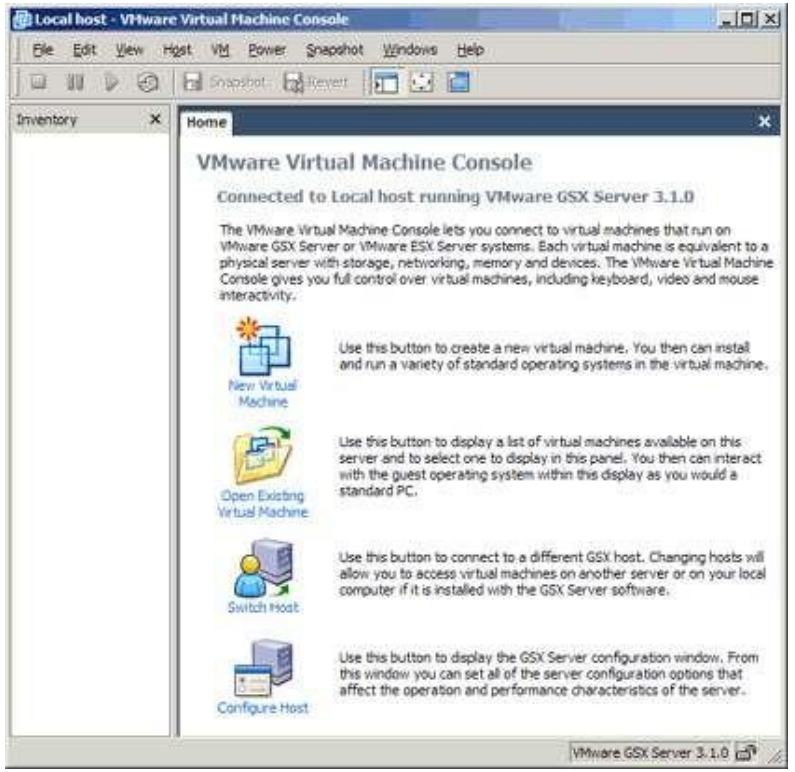
VMware workstation architecture.



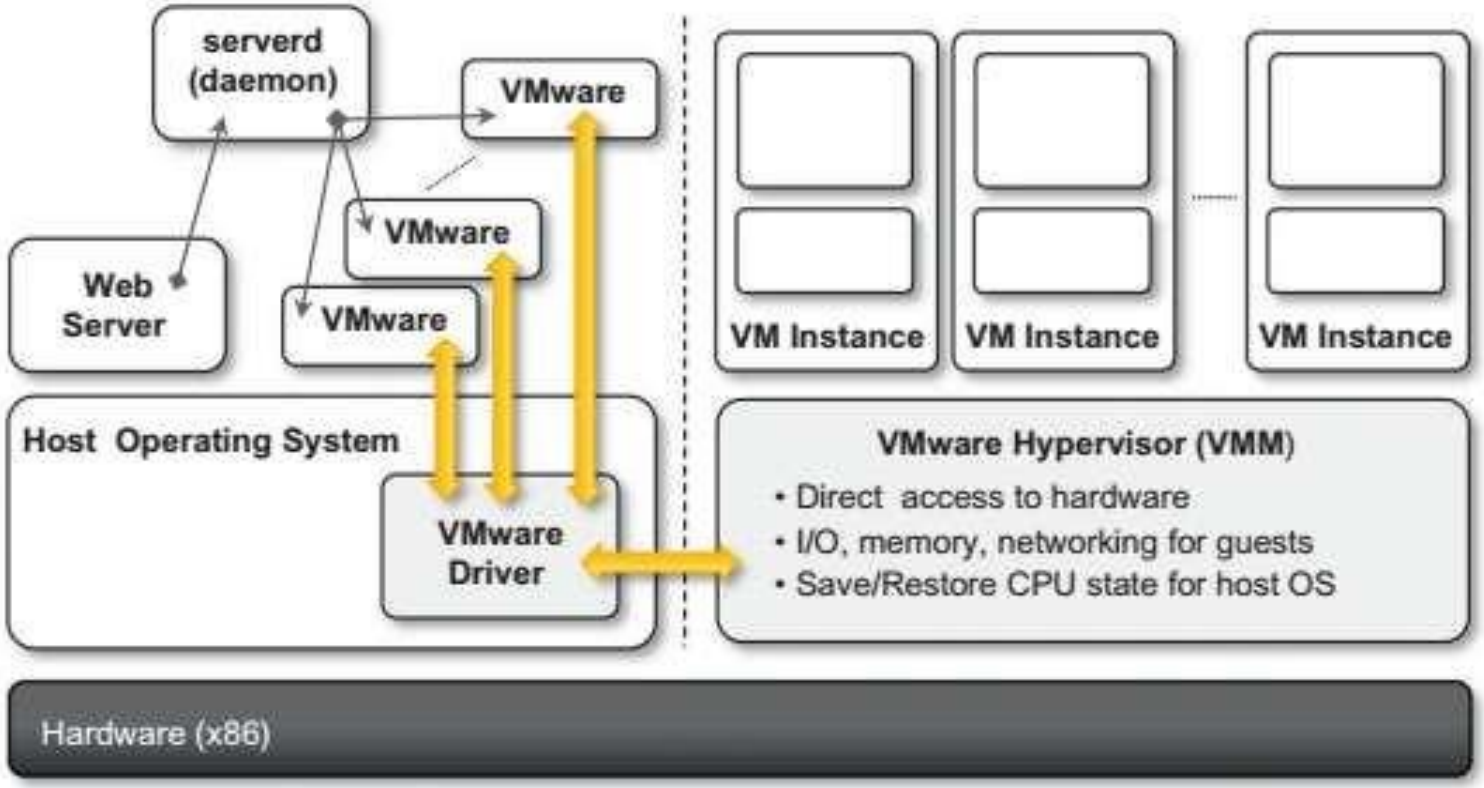
Virtualization solutions by VMware

- Server virtualization
- VMWare GSX
- VMWare ESXi

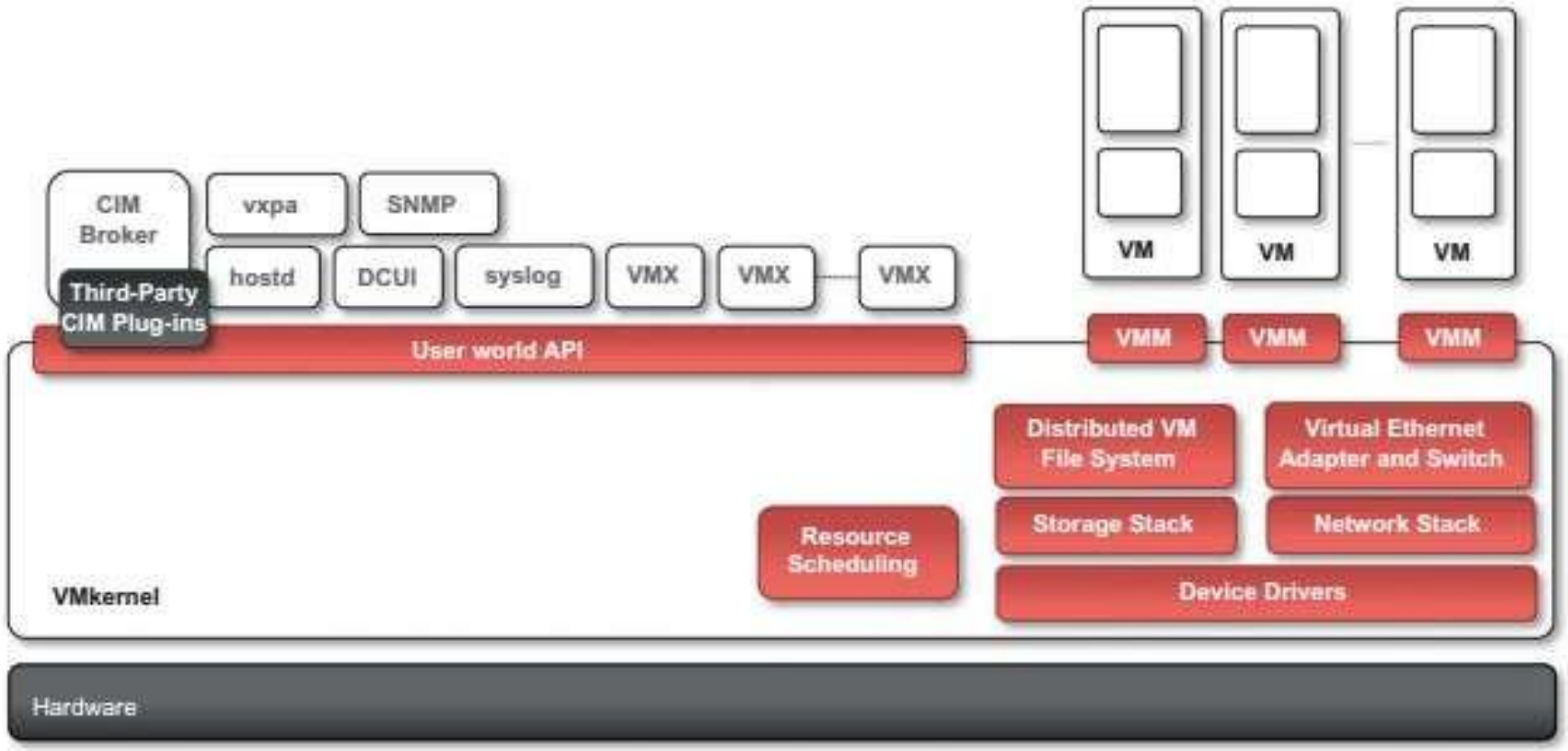




VMware GSX server architecture.



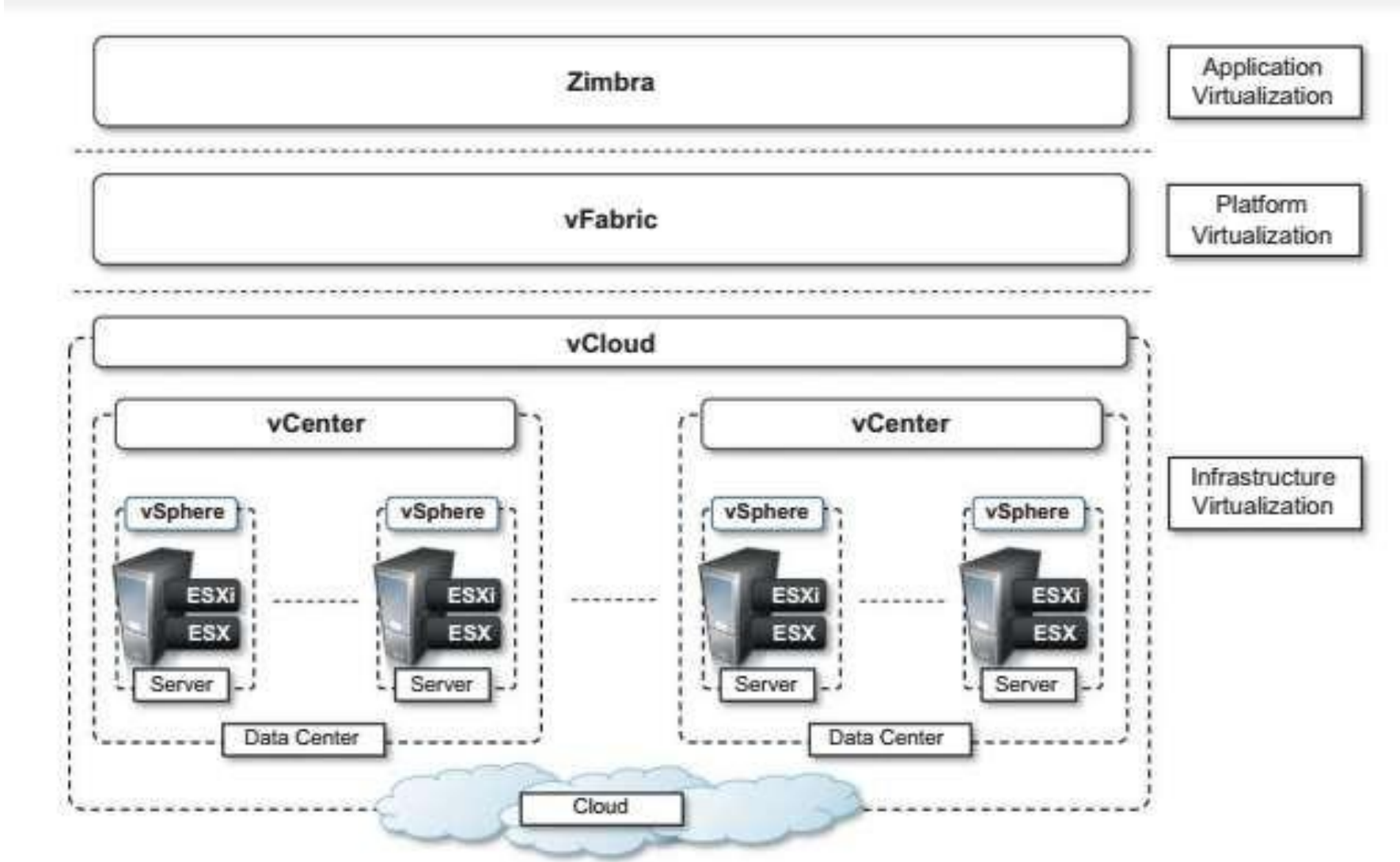
VMware ESXi server architecture.



Virtualization solutions by VMware

- Infrastructure virtualization and cloud computing solutions
- VMware provides a set of products covering the entire stack of cloud computing,

VMware Cloud Solution stack.



Microsoft Hyper-V: Server Virtualization

- formerly known as **Windows Server Virtualization**
- support a variety of guest operating systems.

Microsoft Hyper-V architecture.