UNIT-II

VIRTUALIZATION

Syllabus

- Introduction
- Characteristics of Virtualized Environment
- Taxonomy of Virtualization Techniques
- Virtualization and Cloud computing

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—to run applications.
- The term ***virtualization*** is often synonymous with ***hardware virtualization***, which plays a fundamental role in efficiently delivering ***Infrastructure-as-a-Service*** **(IaaS)** solutions for cloud computing.

Introduction

- Virtualization technologies available in many flavors by providing virtual environments at the operating system level, the programming language level, and the application level.
- Virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

Introduction

- Virtualization technologies have gained renewed interested recently due to the confluence of several phenomena:
  - ***Increased performance and computing capacity***
  - ***Underutilized hardware and software resources***
  - ***Lack of space***
  - ***Greening initiatives***
  - ***Rise of administrative costs***

Increased performance and computing capacity

- Nowadays, the average end-user desktop PC is powerful enough to meet almost all the needs of everyday computing, with extra capacity that is rarely used.
- Almost all these PCs have resources enough to host a virtual machine manager and execute a virtual machine with by far acceptable performance.
- The same consideration applies to the high-end side of the PC market, where supercomputers can provide immense compute power that can accommodate the execution of hundreds or thousands of virtual machines.

Underutilized hardware and software resources

- Hardware and software underutilization is occurring due to (1) increased performance and computing capacity, and (2) the effect of limited or sporadic use of resources.
- Computers today are so powerful that in most cases only a fraction of their capacity is used by an application or the system.
- Moreover, if we consider the IT infrastructure of an enterprise, many computers are only partially utilized whereas they could be used without interruption on a 24/7/365 basis.
- For example, desktop PCs mostly devoted to office automation tasks and used by administrative staff are only used during work hours, remaining completely unused overnight.
- Using these resources for other purposes after hours could improve the efficiency of the IT infrastructure.
- To transparently provide such a service, it would be necessary to deploy a completely separate environment, which can be achieved through virtualization.

Lack of space

- The continuous need for additional capacity, whether storage or compute power, makes data centers grow quickly.
- Companies such as Google and Microsoft expand their infrastructures by building data centers as large as football fields that are able to host thousands of nodes.
- Although this is viable for IT giants, in most cases enterprises cannot afford to build another data center to accommodate additional resource capacity.
- This condition, along with hardware underutilization, has led to the diffusion of a technique called ***server consolidation,*** for which virtualization technologies are fundamental.

Greening initiatives

- Recently, companies are increasingly looking for ways to reduce the amount of energy they consume and to reduce their carbon footprint.
- Data centers are one of the major power consumers.
- Maintaining a data center operation not only involves keeping servers on, but a great deal of energy is also consumed in keeping them cool.
- Infrastructures for cooling have a significant impact on the carbon footprint of a data center.
- Hence, reducing the number of servers through server consolidation will definitely reduce the impact of cooling and power consumption of a data center.
- Virtualization technologies can provide an efficient way of consolidating servers.

Rise of administrative costs

- The increased demand for additional capacity, which translates into more servers in a data center, is also responsible for a significant increment in administrative costs.
- Computers—in particular, servers—do not operate all on their own, but they require care and feeding from system administrators.
- Common system administration tasks include **hardware monitoring, defective hardware replacement, server setup and updates, server resources monitoring, and backups.**
- These are labor-intensive operations, and the higher the number of servers that have to be managed, the higher the administrative costs.
- Virtualization can help reduce the number of required servers for a given workload, thus reducing the cost of the administrative personnel.

Characteristics of Virtualized Environment

- Virtualization is a broad concept that refers to the creation of a virtual version of something, whether hardware, a software environment, storage, or a 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 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.

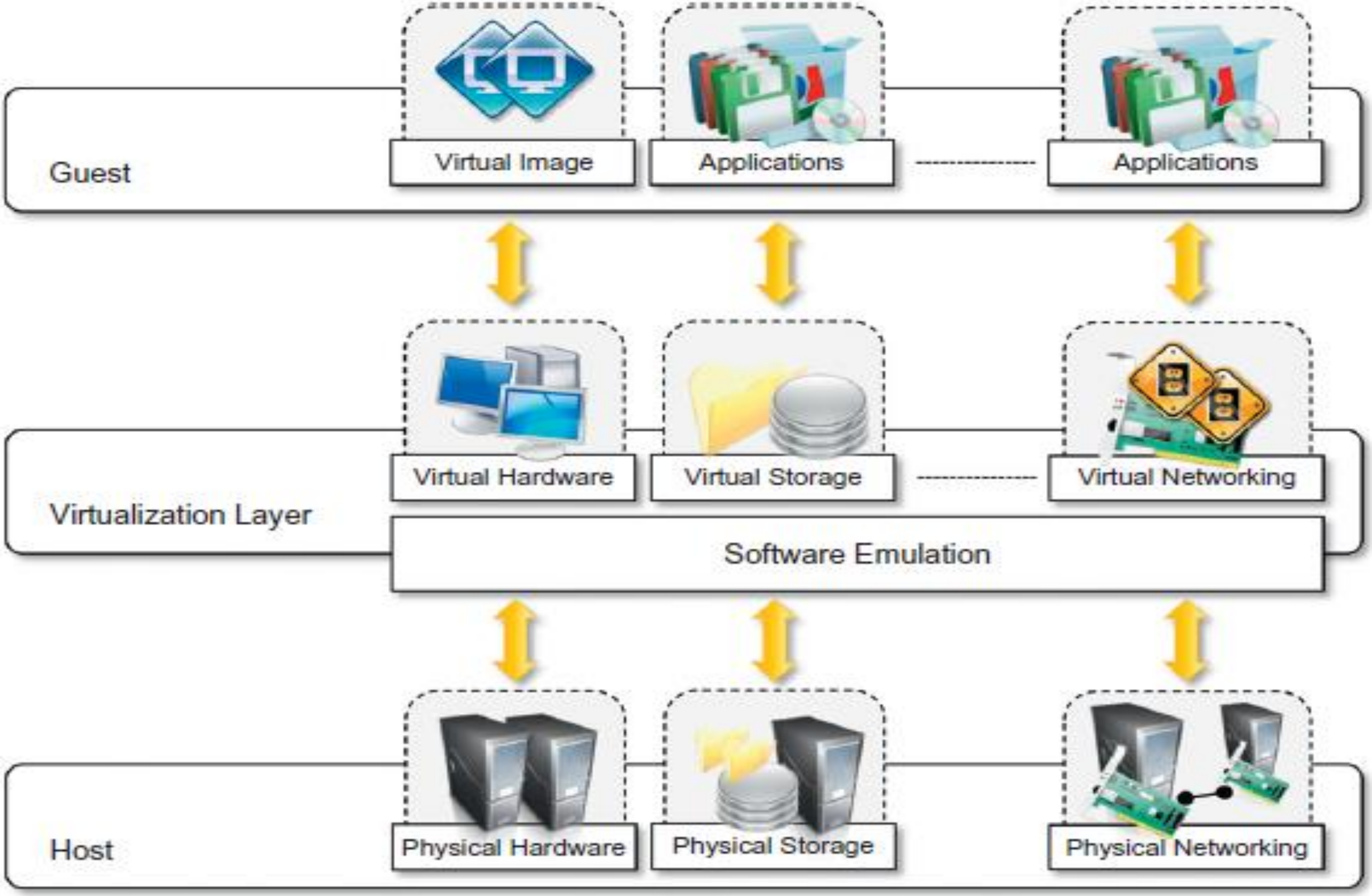


Figure: The virtualization reference model

Characteristics of Virtualized Environment

- The main common characteristic of all these different implementations is the fact that the virtual environment is created by means of a **software program**.
- The ability to use software to emulate such a wide variety of environments creates a lot of opportunities, previously less attractive because of excessive over head introduced by the virtualization layer.
- The technologies of today allow profitable use of virtualization and make it possible to fully exploit the advantages that come with it.
- Such advantages have always been characteristics of virtualized solutions.

- **Increased security**
- **Managed execution**
  - **sharing**
  - **Aggregation**
  - **Emulation**
  - **Isolation**
- **Portability**

Increased security

- The ability to control the execution of a guest in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment.
- The virtual machine represents an emulated environment in which the guest is executed.
- All the operations of the guest are generally performed against the virtual machine, which then translates and applies them to the host.
- Resources exposed by the host can then be hidden or simply protected from the guest.
- Sensitive information that is contained in the host can be naturally hidden without the need to install complex security policies.

Managed execution

- Virtualization of the execution environment not only allows increased security, but a wider range of features also can be implemented.
- In particular, **sharing, aggregation, emulation, and isolation** are the most relevant features.

Virtualization Features

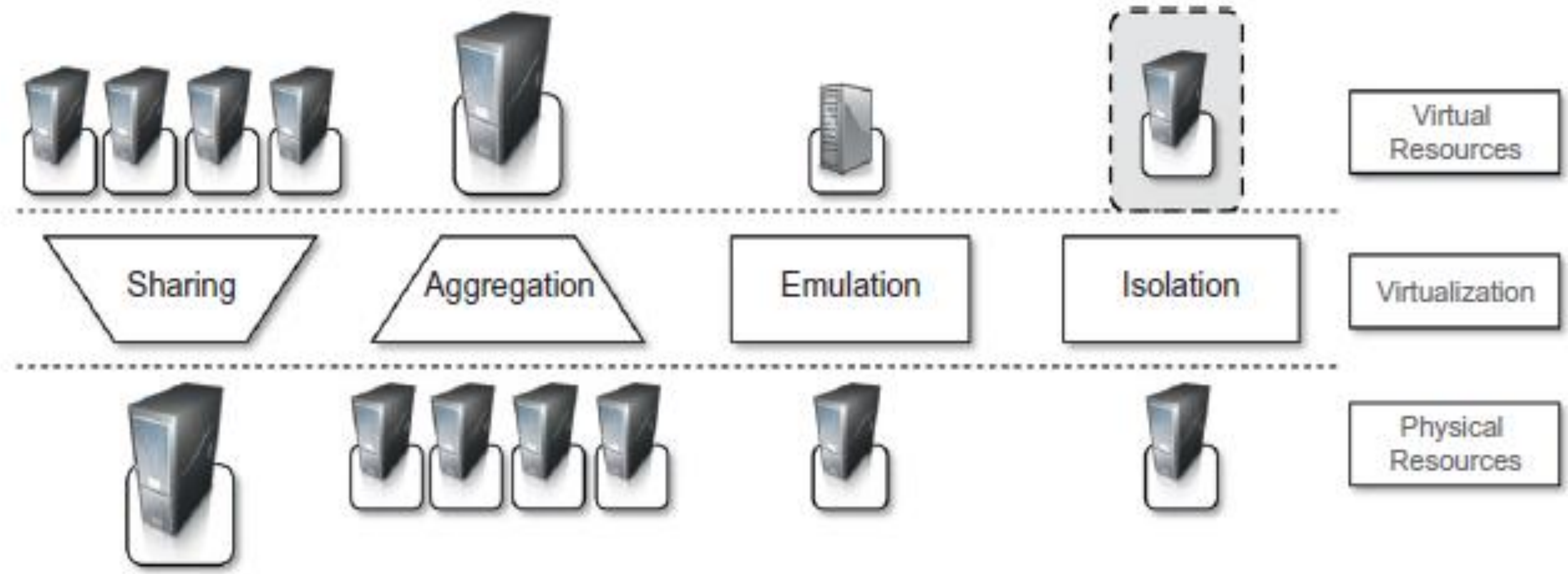


Figure: Functions enabled by managed execution

Sharing

- Virtualization allows the creation of a separate computing environments with in the same host.
- In this way it is possible to fully exploit the capabilities of a powerful guest, which would otherwise be underutilized.
- Sharing is a important feature in virtualized data centers, where this basic feature is used to reduce the number of active servers and limit power consumption.

Aggregation

- Not only is it possible to share physical resource among several guests, but virtualization also allows aggregation, which is the opposite process.
- A group of separate hosts can be tied together and represented to guests as a single virtual host.
- This function is naturally implemented in middleware for distributed computing, with a classical example represented by cluster management software, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.

Emulation

- Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program.
- This allows for controlling and tuning the environment that is exposed to guests.
- For instance, a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.
- Hardware virtualization solutions are able to provide virtual hardware and emulate a particular kind of device such as Small Computer System Interface (SCSI) devices for file I/O, without the hosting machine having such hardware installed.

Isolation

- Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a completely separate environment, in which they are executed.
- The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.
- The virtual machine can filter the activity of the guest and prevent harmful operations against the host.

Portability

- The concept of portability applies in different ways according to the specific type of virtualization considered.
- In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines.
- Except for the file size, this happens with the same simplicity with which we can display a picture image in different computers.
- Virtual images are generally proprietary formats that require a specific virtual machine manager to be executed.

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 us better understand their characteristics and use.

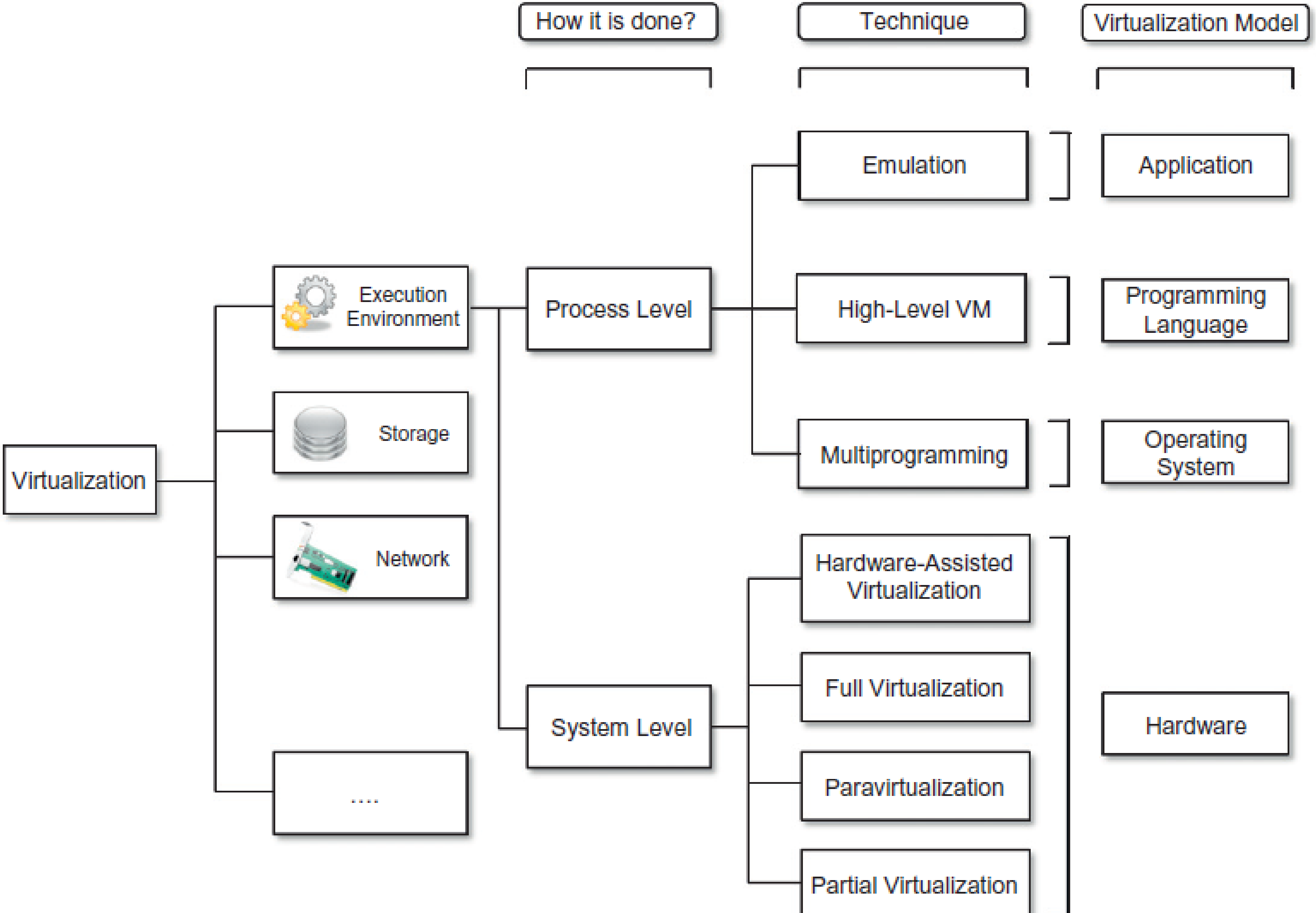


Figure: **A taxonomy of virtualization techniques**

- Execution virtualization
  - Machine reference model
  - Hardware-level Virtualization
    - Hypervisors
      - Type I
      - Type II
    - Hardware virtualization techniques
      - Hardware-assisted virtualization
      - Full virtualization
      - Paravirtualization
      - Partial virtualization
    - Operating system-level virtualization
  - Programming language-level virtualization
  - Application-level virtualization
    - Interpretation
    - Binary Translation
- Other types of virtualization
  - Storage virtualization
  - Network virtualization
  - Desktop virtualization
  - Application server virtualization

Taxonomy of Virtualization Techniques

- The first classification discriminates against the service or entity that is being emulated.
- 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.
- In particular we can divide these execution virtualization techniques into two major categories by considering the type of host they require.

Taxonomy of Virtualization Techniques

- ***Process-level*** techniques are implemented on top of an existing operating system, which has full control of the hardware.
- ***System-level*** techniques are implemented directly on hardware and do not require—or require a minimum of support from—an existing operating system.
- Within these two categories we can list various techniques that offer the guest a different type of virtual computation environment: ***bare hardware, operating system resources, low-level programming language***, and ***application libraries***.

Execution virtualization

- ***Execution virtualization*** includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer.
- All these techniques concentrate their interest on providing support for the execution of programs, whether these are the operating system, a binary specification of a program compiled against an abstract machine model, or an application.
- Execution virtualization can be implemented directly on top of the hardware by the operating system, an application, or libraries dynamically or statically linked to an application image.

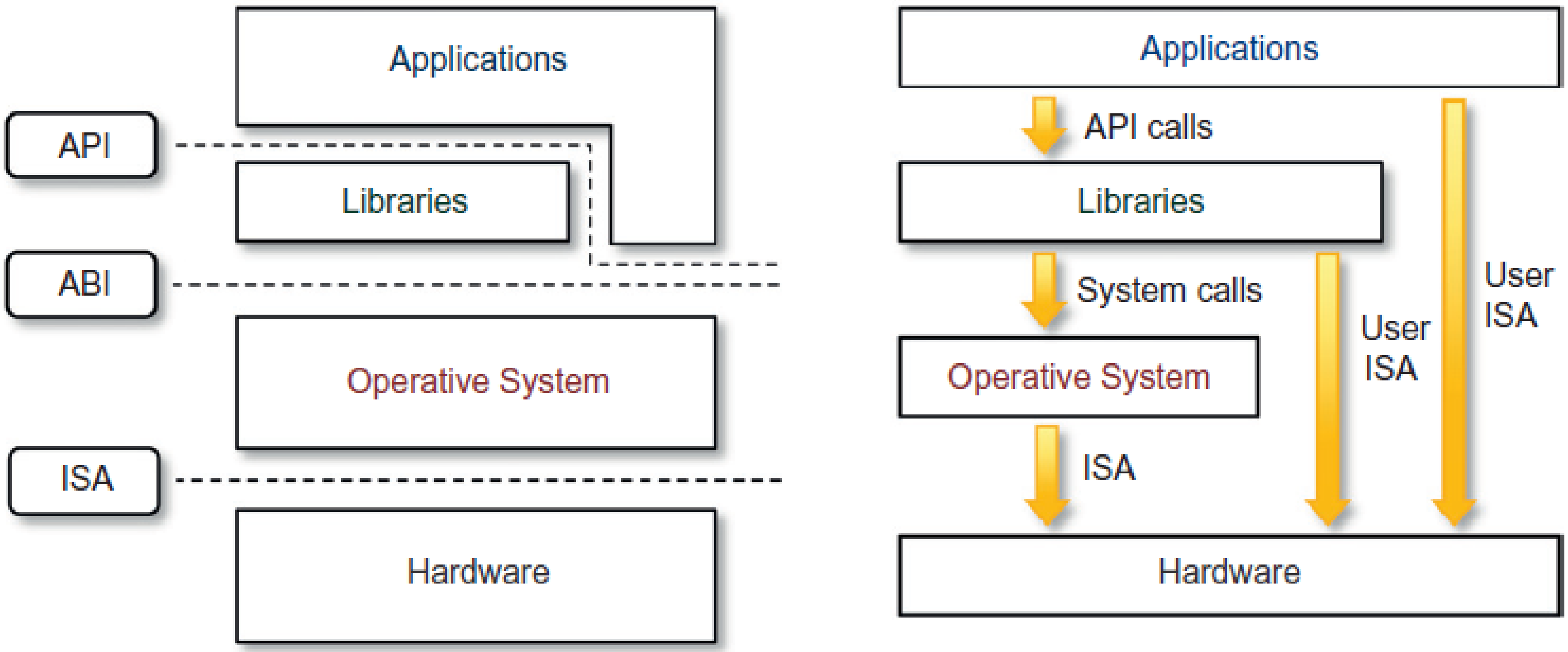
Machine reference model

- Virtualizing an execution environment at different levels of the computing stack requires a reference model that defines the interfaces between the levels of abstractions, which hide implementation details.
- From this perspective, virtualization techniques actually replace one of the layers and intercept the calls that are directed toward it.

Machine reference model

- Modern computing systems can be expressed in terms of the reference model described in below figure.
- At the bottom layer, the model for the hardware is expressed in terms of the **Instruction Set Architecture (ISA)**, which defines the instruction set for the processor, registers, memory, and interrupts management.
- ISA is the interface between hardware and software, and it is important to the operating system (OS) developer (System ISA) and developers of applications that directly manage the underlying hardware (User ISA).

Machine reference model



Machine reference model

- The **application binary interface** (ABI) separates the operating system layer from the applications and libraries, which are managed by the OS.
- ABI covers details such as low-level data types, alignment, and call conventions and defines a format for executable programs.
- System calls are defined at this level.
- This interface allows portability of applications and libraries across operating systems that implement the same ABI.
- The highest level of abstraction is represented by the **application programming interface (API)**, which interfaces applications to libraries and/or the underlying operating system.

Machine reference model

- The instruction set exposed by the hardware has been divided into different security classes that define who can operate with them.
- The first distinction can be made between ***privileged*** and ***nonprivileged*** instructions.
- ***Nonprivileged*** instructions are those instructions that can be used without interfering with other tasks because they do not access shared resources.
- This category contains, for example, all the floating, fixed-point, and arithmetic instructions.

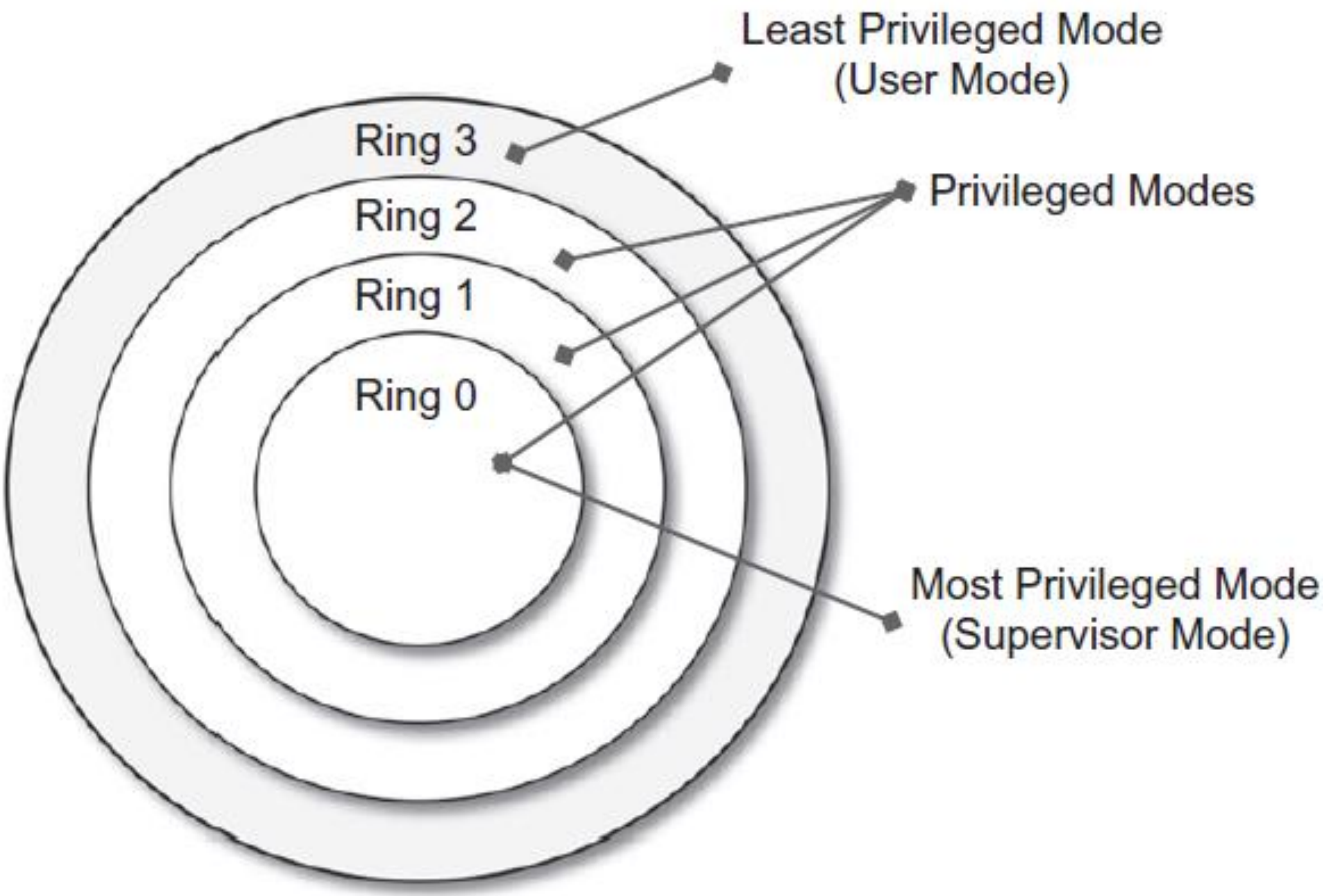
Machine reference model

- ***Privileged*** instructions are those that are executed under specific restrictions and are mostly used for sensitive operations, which expose (***behavior-sensitive***) or modify (***control-sensitive***) the privileged state.
- For instance, behavior-sensitive instructions are those that operate on the I/O, whereas control-sensitive instructions alter the state of the CPU registers.

Machine reference model

- A possible implementation features a hierarchy of privileges in the form of ring-based security: ***Ring 0, Ring 1, Ring 2,*** and ***Ring 3***; Ring 0 is in the most privileged level and Ring 3 in the least privileged level.
- Ring 0 is used by the kernel of the OS, rings 1 and 2 are used by the OS-level services, and Ring 3 is used by the user.
- Recent systems support only two levels, with Ring 0 for supervisor mode and Ring 3 for user mode.

Security rings and privilege modes



Machine reference model

- All the current systems support at least two different execution modes: ***supervisor mode*** and ***user mode***.
- The first mode denotes an execution mode in which all the instructions (privileged and nonprivileged) can be executed without any restriction.
- This mode, also called ***master mode*** or ***kernel mode***, is generally used by the operating system (or the hypervisor) to perform sensitive operations on hardware level resources.

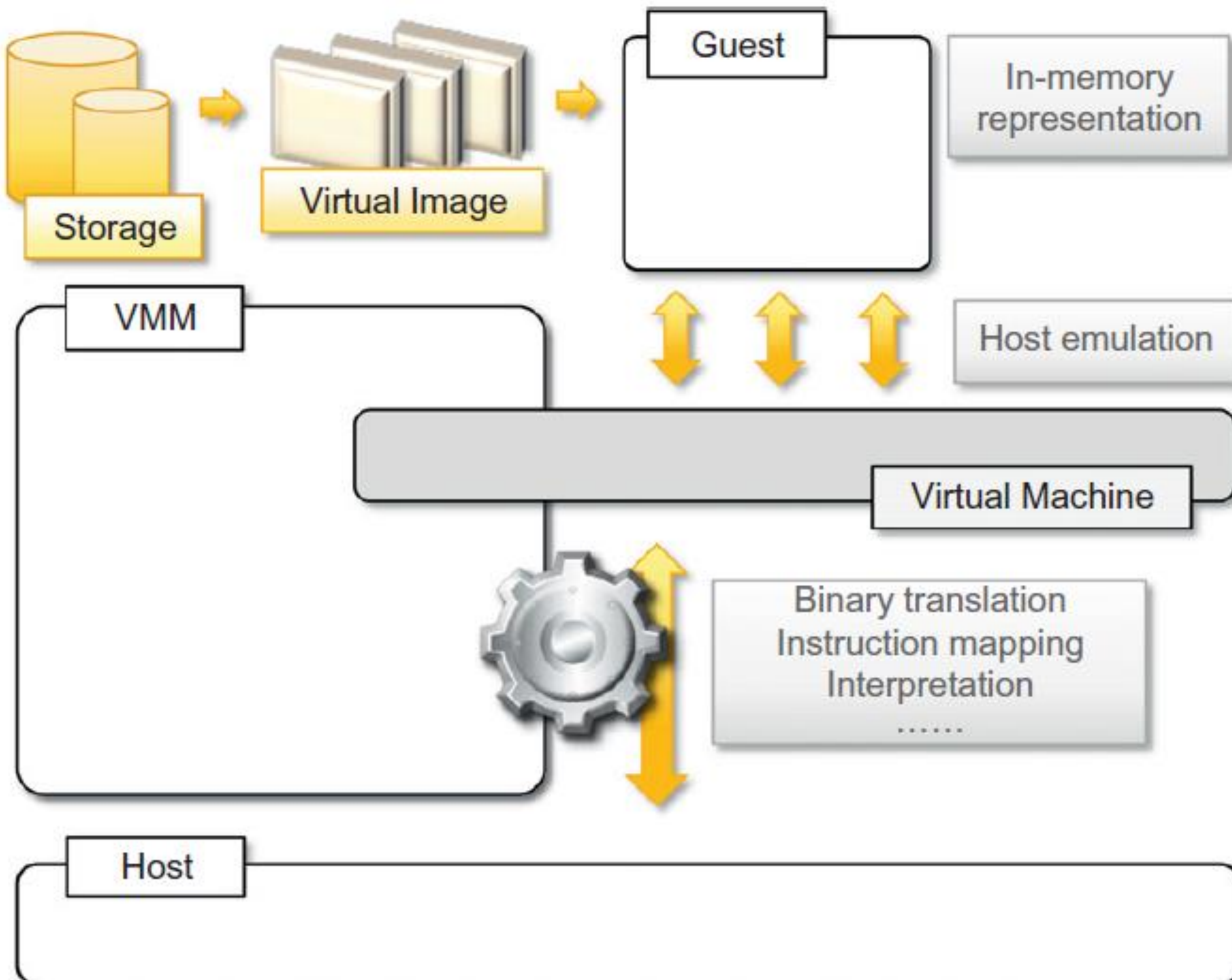
Machine reference model

- In **user mode**, there are restrictions to control the machine-level resources.
- If code running in user mode invokes the privileged instructions, hardware interrupts occur and trap the potentially harmful execution of the instruction.

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, the ***guest*** is represented by the operating system, the ***host*** by the physical computer hardware, the ***virtual machine*** by its emulation, and the ***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.

A hardware virtualization reference model



Hardware-level virtualization

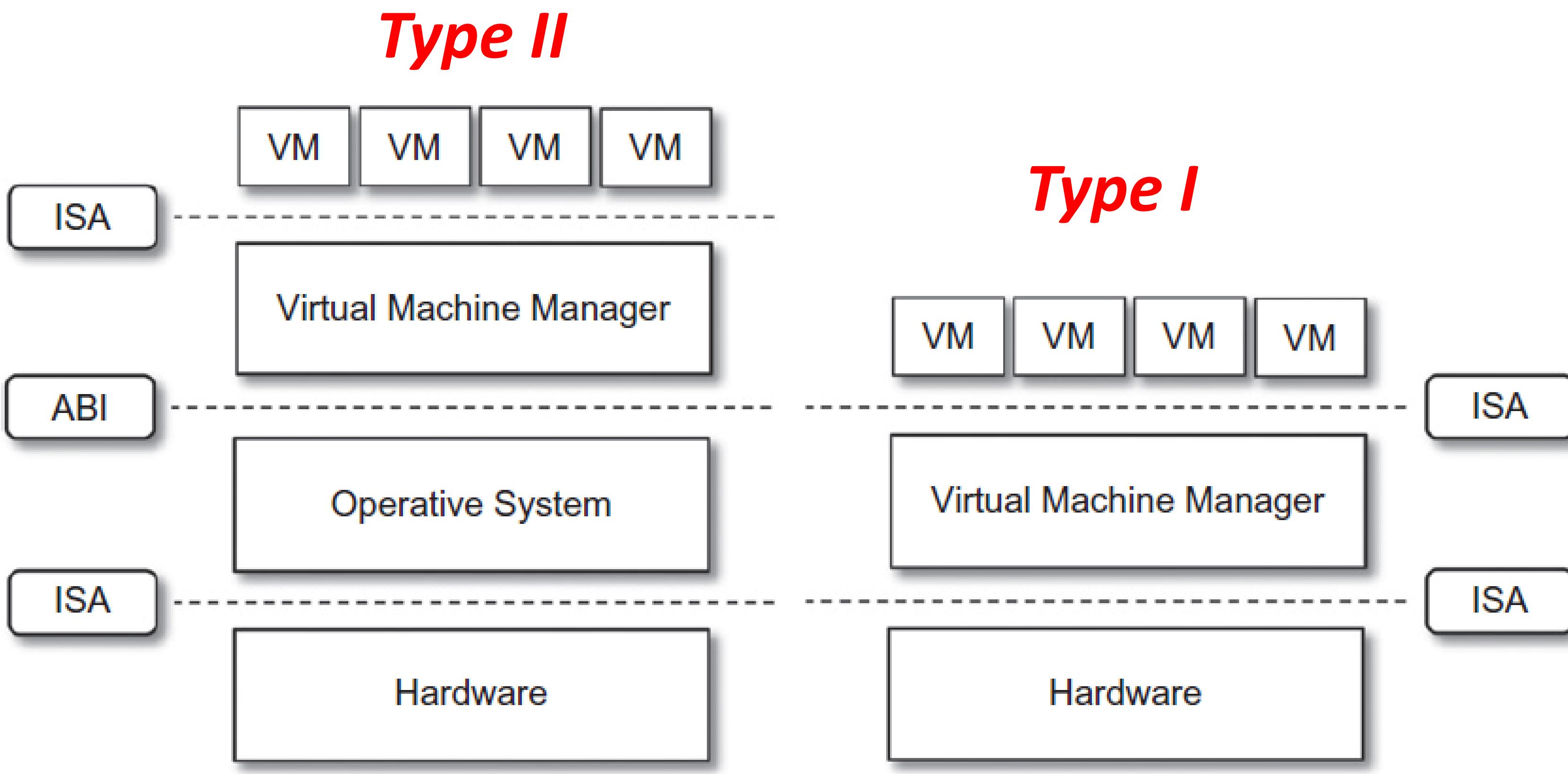
- Hardware-level virtualization is also called ***system virtualization***, since it provides ISA to virtual machines, which is the representation of the hardware interface of a system.
- This is to differentiate it from ***Process virtual machines***, which expose ABI to virtual machines.

Hypervisors

- A fundamental element of hardware virtualization is the hypervisor, or virtual machine manager (VMM).
- It recreates a hardware environment in which guest operating systems are installed.
- There are two major types of hypervisor:

  Type I and ***Type II***.

Hosted (left) and native (right) virtual machines



This figure provides a graphical representation of the two types of hypervisors

Hypervisors

- ***Type I*** hypervisors run directly on top of the hardware.
- They take the place of the operating systems and interact directly with the ISA interface exposed by the underlying hardware, and they emulate this interface in order to allow the management of guest operating systems.
- This type of hypervisor is also called a ***native virtual machine*** since it runs natively on hardware.

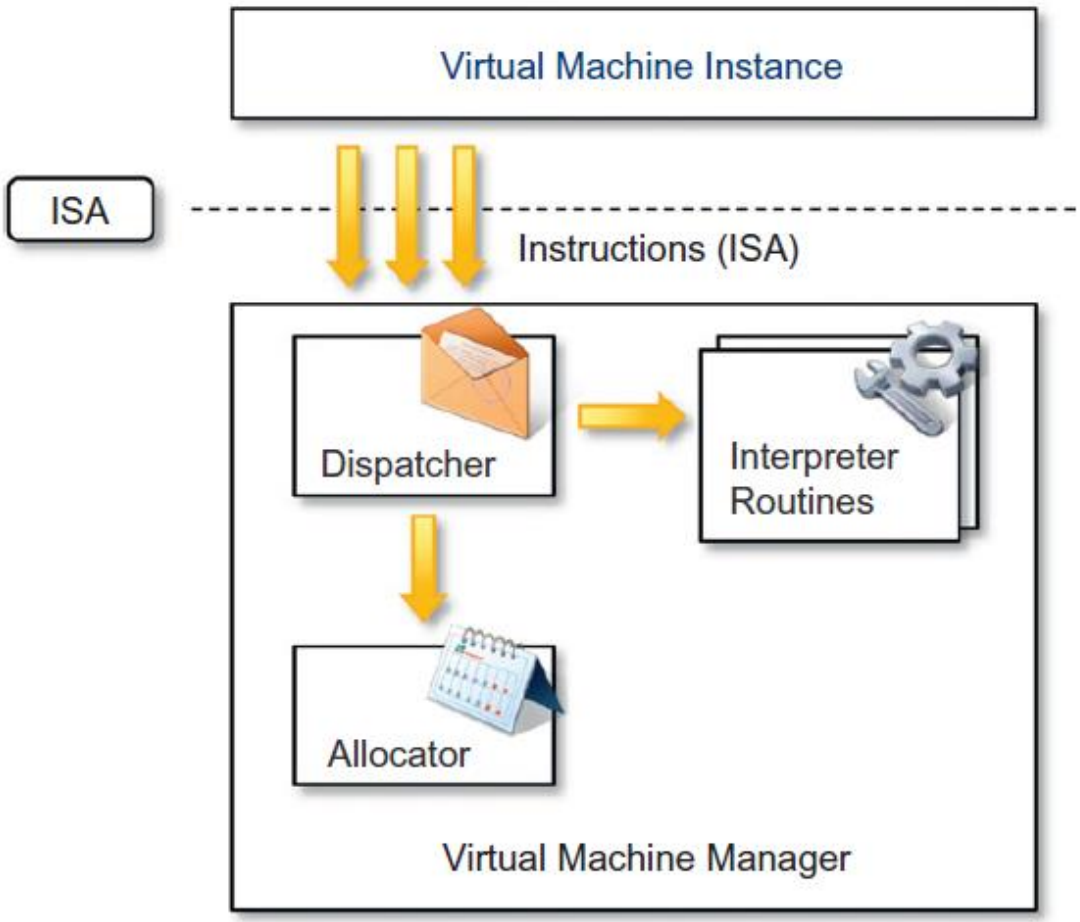
Hypervisors

- ***Type II*** hypervisors require the support of an operating system to provide virtualization services.
- This means that they are programs managed by the operating system, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems.
- This type of hypervisor is also called a ***hosted virtual machine*** since it is hosted within an operating system.

Hypervisors

- A virtual machine manager is internally organized as described in below figure.
- ***Three*** main modules, ***dispatcher, allocator, and interpreter,*** coordinate their activity in order to emulate the underlying hardware.

A hypervisor reference architecture



Hypervisors

- 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: whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with that VM, the allocator is invoked by the dispatcher.
- 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.

Hypervisors

- Three properties of Virtual Machine Manager that have to be satisfied:
  - **Equivalence -** A guest running under the control of a virtual machine manager should exhibit the same behavior as 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.

Hardware virtualization techniques

- Hardware-assisted virtualization
- Full virtualization
- Paravirtualization
- Partial virtualization

Hardware-assisted virtualization

- This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.
- This technique was originally introduced in the IBM System/370.
- At present, examples of hardware-assisted virtualization are the extensions to the x86-64 bit architecture introduced with ***Intel VT*** (formerly known as ***Vanderpool***) and ***AMD V*** (formerly known as ***Pacifica***).
- Products such as VMware Virtual Platform, introduced in 1999 by VMware, which pioneered the field of x86 virtualization, were based on this technique.
- After 2006, Intel and AMD introduced processor extensions, and a wide range of virtualization solutions took advantage of them: Kernel-based Virtual Machine (KVM), VirtualBox, Xen, VMware, Hyper-V, Sun xVM, Parallels, and others.

Full virtualization

- Full virtualization refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and 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.
- The principal advantage of full virtualization is complete isolation, which leads to enhanced security, ease of emulation of different architectures, and coexistence of different systems on the same platform.
- A simple solution to achieve full virtualization is to provide a virtual environment for all the instructions, thus posing some limits on performance.

Paravirtualization

- This is a not-transparent virtualization solution that allows implementing thin virtual machine managers.
- Paravirtualization techniques expose a software interface to the virtual machine that is slightly modified from the host and, as a consequence, guests need to be modified.
- The aim of paravirtualization is to provide the capability to demand the execution of performance-critical operations directly on the host, thus preventing performance losses that would otherwise be experienced in managed execution.
- This technique has been successfully used by Xen for providing virtualization solutions for Linux-based operating systems specifically ported to run on Xen hypervisors.

Partial virtualization

- Partial virtualization provides a partial emulation of the underlying hardware, thus not allowing the complete execution of the guest operating system in complete isolation.
- 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.
- An example of partial virtualization is address space virtualization used in time-sharing systems; this allows multiple applications and users to run concurrently in a separate memory space, but they still share the same hardware resources (disk, processor, and network).

Operating system-level virtualization

- Operating system-level virtualization offers the opportunity to create different and separated execution environments for applications that are managed concurrently.
- In Operating system-level virtualization, there is no virtual machine manager or hypervisor, and the virtualization is done within a single operating system, where the OS kernel allows for multiple isolated user space instances.
- The kernel is also responsible for sharing the system resources among instances and for limiting the impact of instances on each other.
- A user space instance in general contains a proper view of the file system, which is completely isolated, and separate IP addresses, software configurations, and access to devices.

Operating system-level virtualization

- Operating system-level virtualization aims to provide separated and multiple execution containers for running applications.
- Compared to hardware virtualization, this strategy imposes little or no overhead because applications directly use OS system calls and there is no need for emulation.
- Examples of operating system-level virtualizations are FreeBSD Jails, IBM Logical Partition (LPAR), SolarisZones and Containers, Parallels Virtuozzo Containers, OpenVZ, iCore Virtual Accounts, Free Virtual Private Server (FreeVPS) and others.

Programming language-level virtualization

- Programming language-level virtualization is mostly used to achieve 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 and used this technology to produce a binary format representing the machine code for an abstract architecture.
- The characteristics of this architecture vary from implementation to implementation.

Programming language-level virtualization

- Programming language-level virtualization has a long trail in computer science history and originally was used in 1966 for the implementation of Basic Combined Programming Language (BCPL), a language for writing compilers and one of the ancestors of the C programming language.
- Other important examples of the use of this technology have been the UCSD Pascal and Smalltalk.
- Virtual machine programming languages become popular again with Sun's introduction of the Java platform in 1996.

Programming language-level virtualization

- Currently, the Java platform and .NET Framework represent the most popular technologies for enterprise application development.
- The main advantage of programming-level virtual machines, also called process virtual machines, is the ability to provide 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.

Application-level virtualization

- Application-level virtualization is a technique allowing applications to be run in runtime environments that do not natively support all the features required by such applications.
- In this scenario, applications are not installed in the expected runtime environment but are run as though they were.
- In general, these techniques are mostly concerned with partial file systems, libraries, and operating system component emulation.

Application-level virtualization

- Such emulation is performed by a thin layer—a program or an operating system component—that is in charge of executing the application.
- Emulation can also be used to execute program binaries compiled for different hardware architectures.
- In this case, one of the following strategies can be implemented:

Interpretation

- In this technique every source instruction is interpreted by an emulator for executing native ISA instructions, leading to poor performance.
- Interpretation has a minimal startup cost but a huge overhead, since each instruction is emulated.

Binary translation

- In this technique every source instruction is converted to native instructions with equivalent functions.
- After a block of instructions is translated, it is cached and reused.
- Binary translation has a large initial overhead cost, but over time it is subject to better performance, since previously translated instruction blocks are directly executed.

Application-level virtualization

- Application virtualization is a good solution in the case of missing libraries in the host operating system.
- One of the most popular solutions 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

1. Storage virtualization
2. Network virtualization
3. Desktop virtualization
4. Application server virtualization

1. Storage virtualization

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.
- Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.
- There are different techniques for storage virtualization, one of the most popular being network-based virtualization by means of storage area networks (SANs).

2. 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 though they were located under the same broadcasting domain.
- There are several options for implementing internal network virtualization:
  - The guest can share the same network interface of the host and use Network Address Translation (NAT) to access the network.
  - The virtual machine manager can emulate, and install on the host, an additional network device, together with the driver.
  - The guest can have a private network only with the guest.

3. Desktop virtualization

- Desktop virtualization abstracts the desktop environment available on a personal computer in order to provide access to it using a client/server approach.
- Desktop virtualization provides the same out- come of hardware virtualization but serves a different purpose.
- Similarly to hardware virtualization, desktop virtualization makes accessible a different system as though it were 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.
- The advantages of desktop virtualization are high availability, persistence, accessibility, and ease of management.
- Infrastructures for desktop virtualization based on cloud computing solutions include Sun Virtual Desktop Infrastructure (VDI), Parallels Virtual Desktop Infrastructure (VDI), Citrix XenDesktop, and others.

4. 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 plays an important role in cloud computing since it allows for the appropriate degree of customization, security, isolation, and manageability that are fundamental for delivering IT services on demand.
- Virtualization technologies are primarily used to offer configurable computing environments and storage.
- Network virtualization is less popular and, in most cases, is a complementary feature, which is naturally needed in build virtual computing systems.

Virtualization and Cloud Computing

- 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.
- Hardware virtualization is an enabling factor for solutions in the Infrastructure-as-a-Service (IaaS) market segment, while programming language virtualization is a technology leveraged in Platform-as-a-Service (PaaS) offerings.

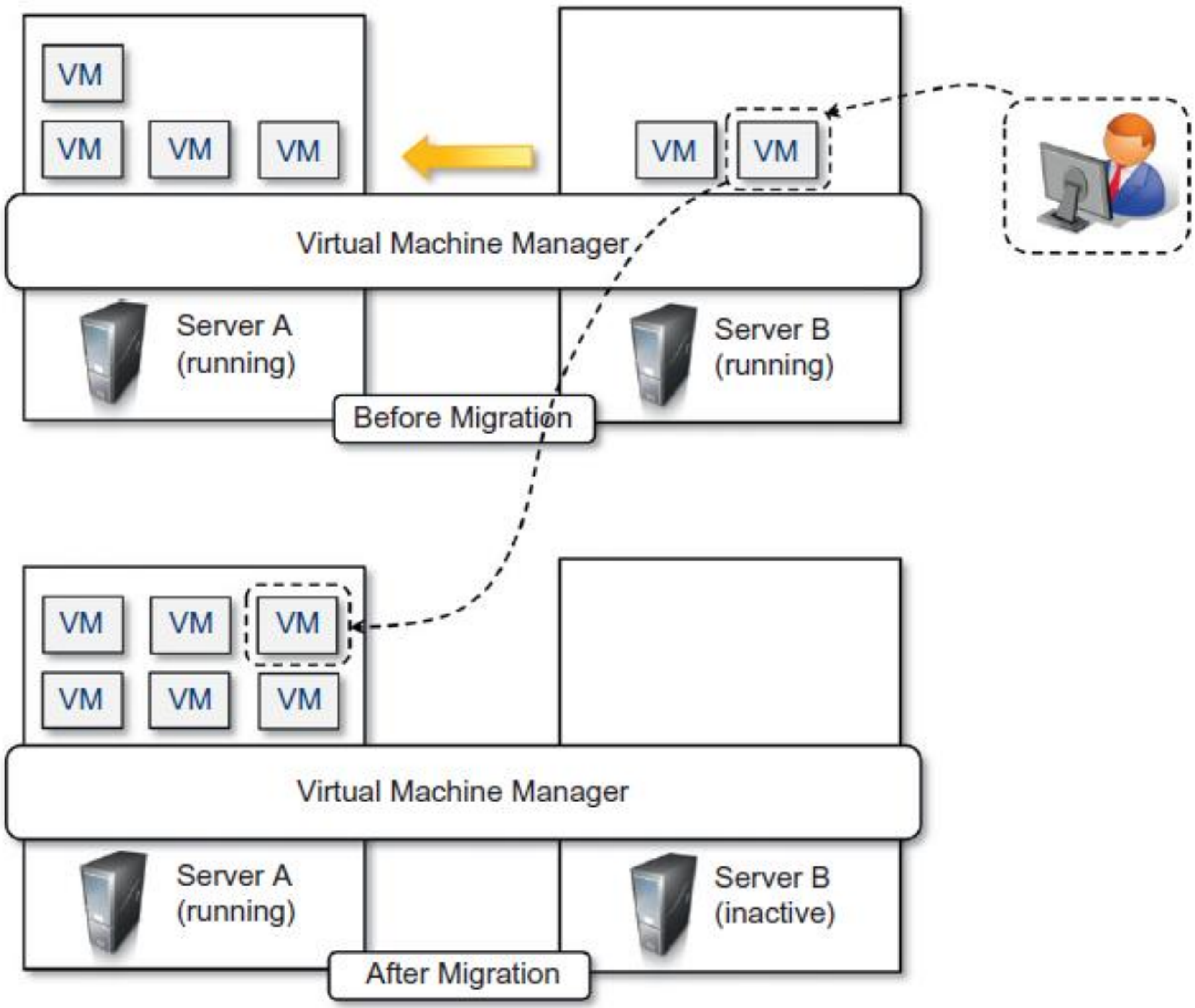
Virtualization and Cloud Computing

- In both cases, the capability of offering a customizable and sandboxed environment constituted an attractive business opportunity for companies featuring a large computing infrastructure that was able to sustain and process huge workloads.
- Moreover, virtualization also allows isolation and a finer control, thus simplifying the leasing of services and their accountability on the vendor side.

Virtualization and Cloud Computing

- Virtualization allows us to create isolated and controllable environments, it is possible to serve these environments with the same resource without them interfering with each other.
- If the underlying resources are capable enough, there will be no evidence of such sharing.
- It allows reducing the number of active resources by aggregating virtual machines over a smaller number of resources that become fully utilized.
- This practice is also known as server consolidation, while the movement of virtual machine instances is called virtual machine migration (see below Figure).

Virtualization and cloud computing



Virtualization and Cloud Computing

- Because virtual machine instances are controllable environments, consolidation can be applied with a minimum impact, either by temporarily stopping its execution and moving its data to the new resources or by performing a finer control and moving the instance while it is running.
- This second techniques is known as live migration and in general is more complex to implement but more efficient since there is no disruption of the activity of the virtual machine instance.

Virtualization and Cloud Computing

- Server consolidation and virtual machine migration are principally used in the case of hardware virtualization, even though they are also technically possible in the case of programming language virtualization (see below Figure).

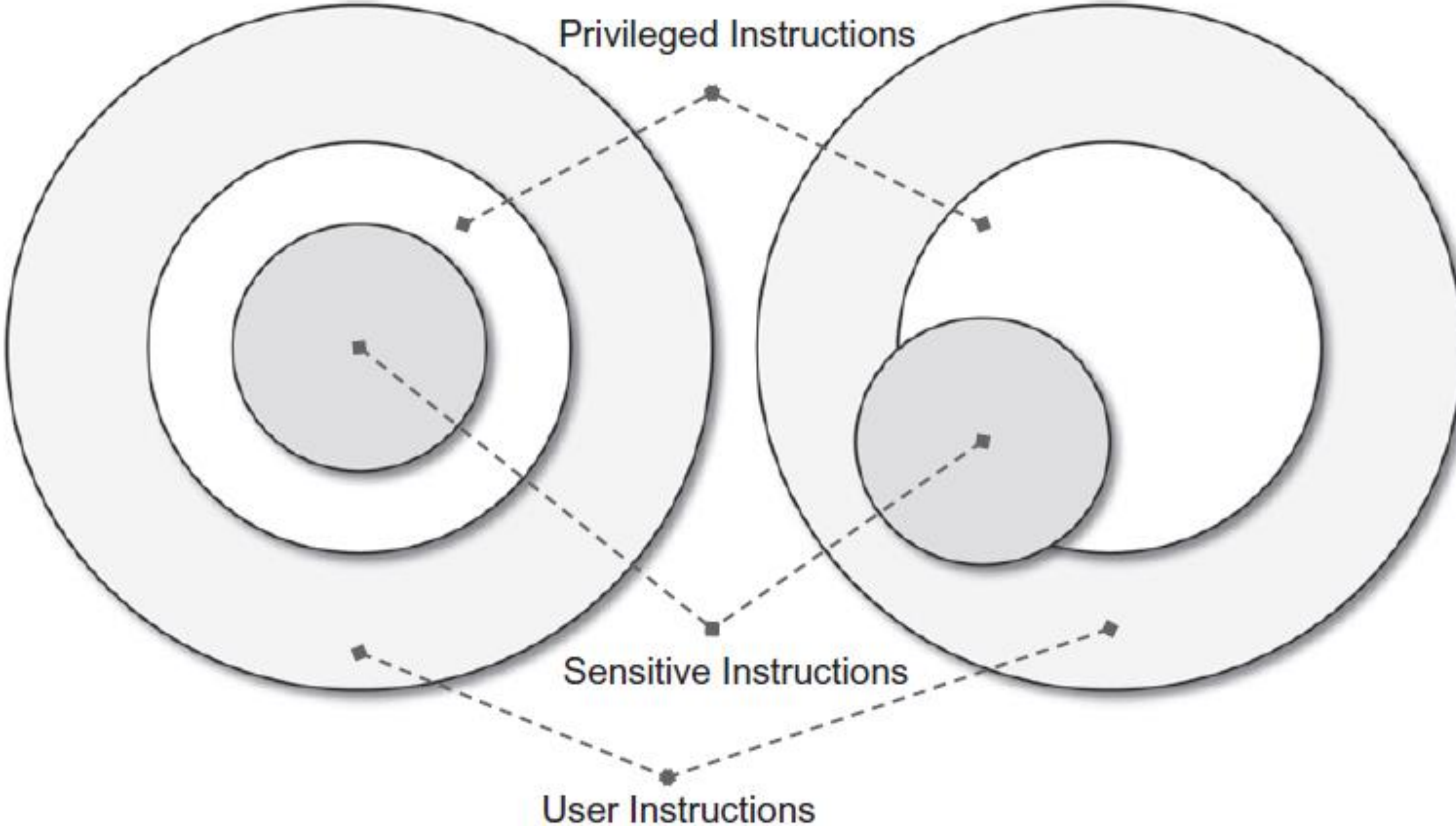


Figure: A virtualizable computer (left) and a nonvirtualizable computer (right)

Virtualization and Cloud Computing

- Storage virtualization constitutes an interesting opportunity given by virtualization technologies, often complementary to the execution of virtualization.
- Even in this case, vendors backed by large computing infrastructures featuring huge storage facilities can harness these facilities into a virtual storage service, easily partitionable into slices.
- These slices can be dynamic and offered as a service.
- Again, opportunities to secure and protect the hosting infrastructure are available, as are methods for easy accountability of such services.

Virtualization and Cloud Computing

- Finally, cloud computing revamps the concept of desktop virtualization, initially introduced in the mainframe era.
- The ability to recreate the entire computing stack—from infrastructure to application services—on demand opens the path to having a complete virtual computer hosted on the infrastructure of the provider and accessed by a thin client over a capable Internet connection.