

Virtualization

Introduction

Virtualization

- 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.

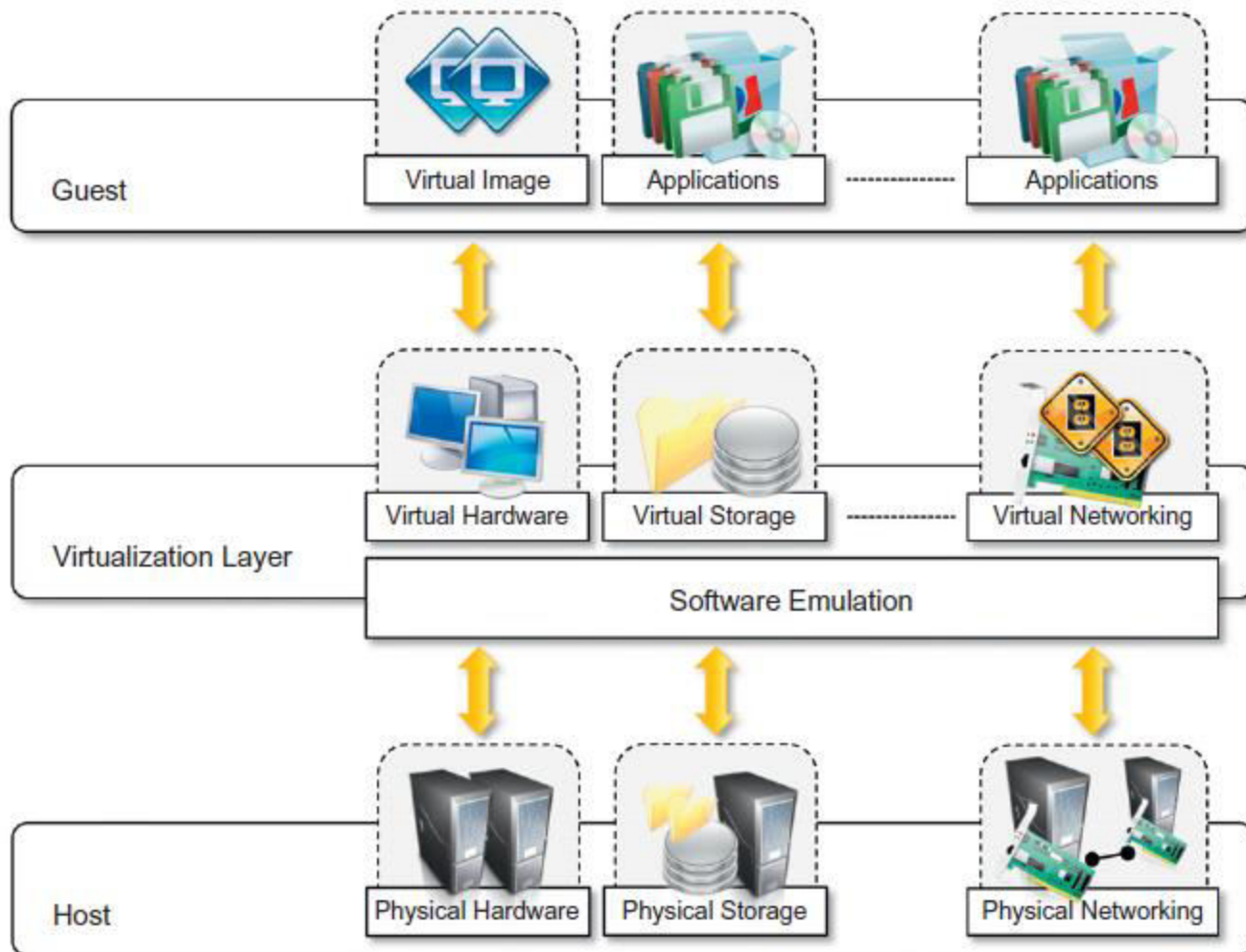
Virtualization

- Virtualization can be provided at
 - Operating system level
 - Programming language level
 - Application level
- Also can be provided as executing application, networking, storage and memory.

Motivation

- Increased performance and computing capacity
- Underutilized hardware and software resources.
- Lack of space
- Greening initiatives
- Rise of administrative costs

Virtualization Reference Model



Virtualization Reference Model

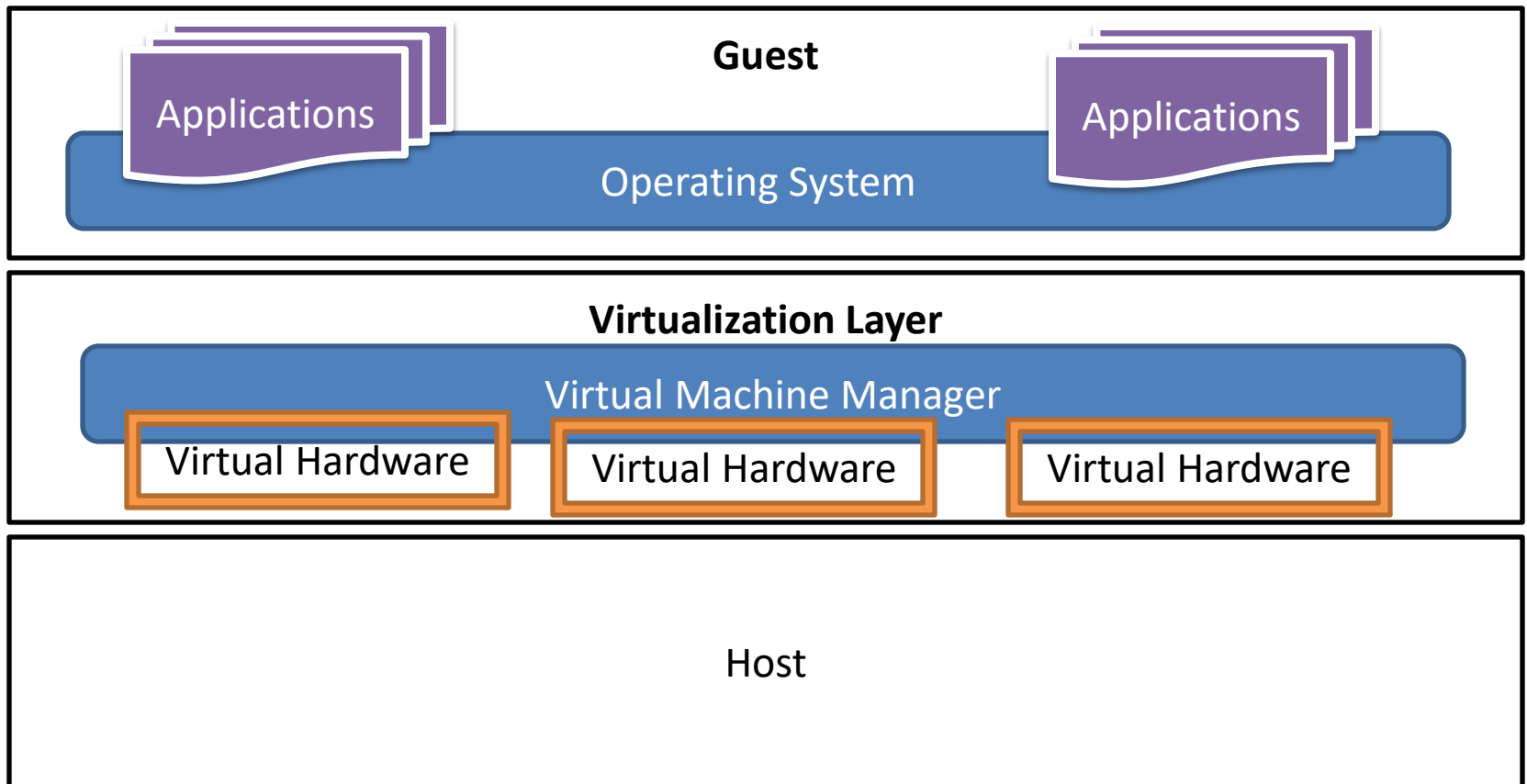
- Major component
 - Guest
 - The guest represents the system component that interacts with the virtualization layer rather than with the host, as would normally happen.
 - Host
 - The host represents the original environment where the guest is supposed to be managed.
 - Virtualization layer
 - The virtualization layer is responsible for recreating the same or a different environment where the guest will operate.

Virtualization Reference Model

- The most intuitive and popular is represented by hardware virtualization.
- Hardware virtualization constitutes the original realization of the virtualization concept.

Virtualization Reference Model

- Hardware virtualization



Virtualization Reference Model

- 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 overhead 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.

NEXT TOPIC:
CHARACTERISTICS OF VIRTUALIZED
ENVIRONMENT

Virtualization

Characteristics of Virtualized
Environment

Characteristics of Virtualized Environment

- Increased Security
- Managed Execution
- Portability

Increased Security

- The virtual machine represents an emulated environment in which the guest is executed.
- Secure and controlled execution environment.
- All the operations of the guest are generally performed against the virtual machine, which then translates and applies them to the host.
- This level of indirection allows the virtual machine manager to control and filter the activity of the guest.

Increased Security

- 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.
- Increased security is a requirement when dealing with untrusted code.
- i.e. JVM, .NET runtime

Increased Security

- Hardware virtualization solutions such as VMware Desktop, VirtualBox, and Parallels provide the ability to create a virtual computer with customized virtual hardware on top of which a new operating system can be installed.
- By default, the file system exposed by the virtual computer is completely separated from the one of the host machine.
- This becomes the perfect environment for running applications without affecting other users in the environment.

Managed Execution

- Virtualization of the execution environment leads to more features like sharing, aggregation, emulation, and isolation.

Managed Execution

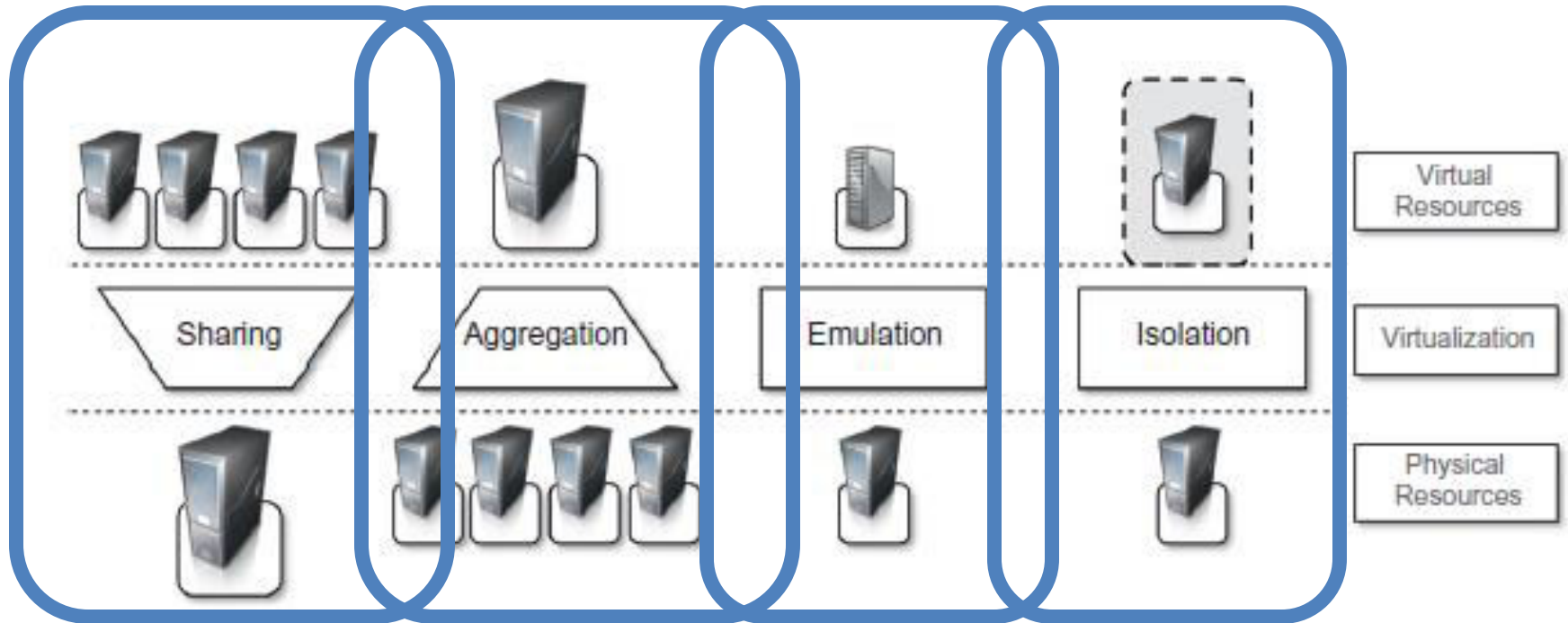


FIGURE 3.2

Functions enabled by managed execution.

Portability

- The guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines.
- One version of the application, in most cases, is able to run on different platforms with no changes.
- Portability allows having your own system always with you and ready to use as long as the required virtual machine manager is available.

NEXT TOPIC:
TAXONOMY OF VIRTUALIZATION
TECHNIQUES

Virtualization

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 us better understand their characteristics and use.

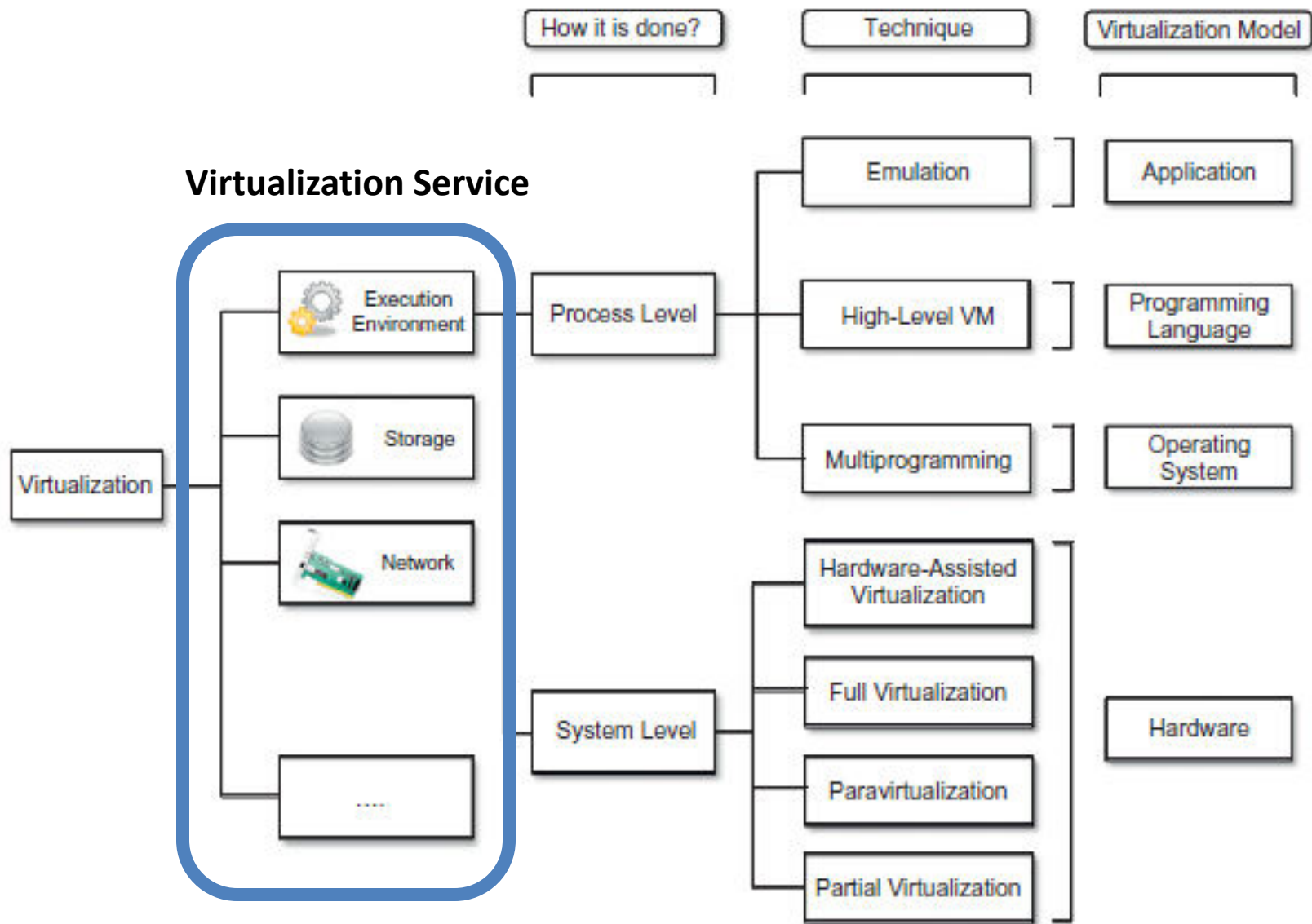


FIGURE 3.3

A taxonomy of virtualization techniques.

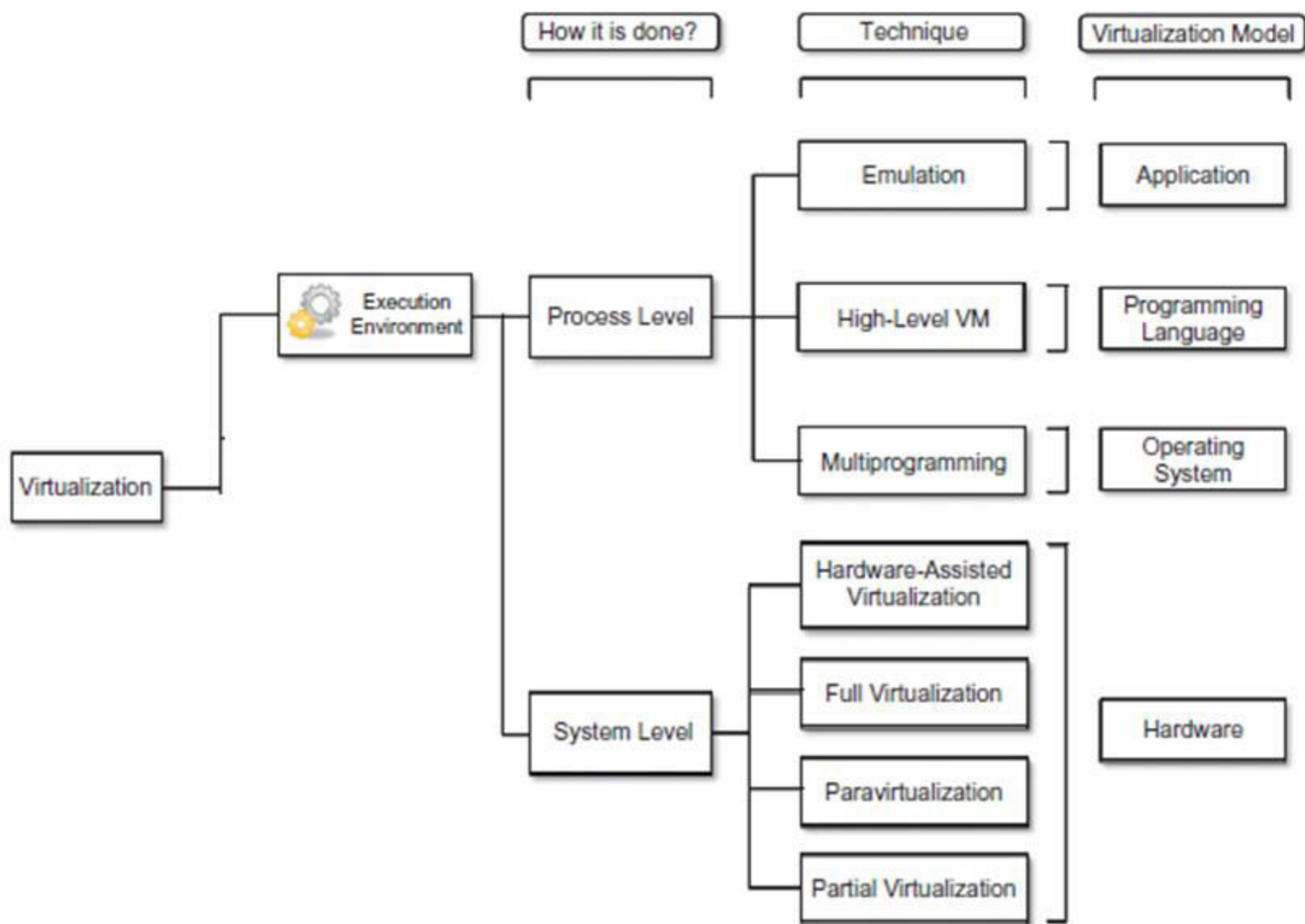


FIGURE 3.3

A taxonomy of virtualization techniques.

Taxonomy of Virtualization Techniques

- Execution virtualization
 - Execution virtualization includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer.
 - Providing support for the execution of programs.
 - 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

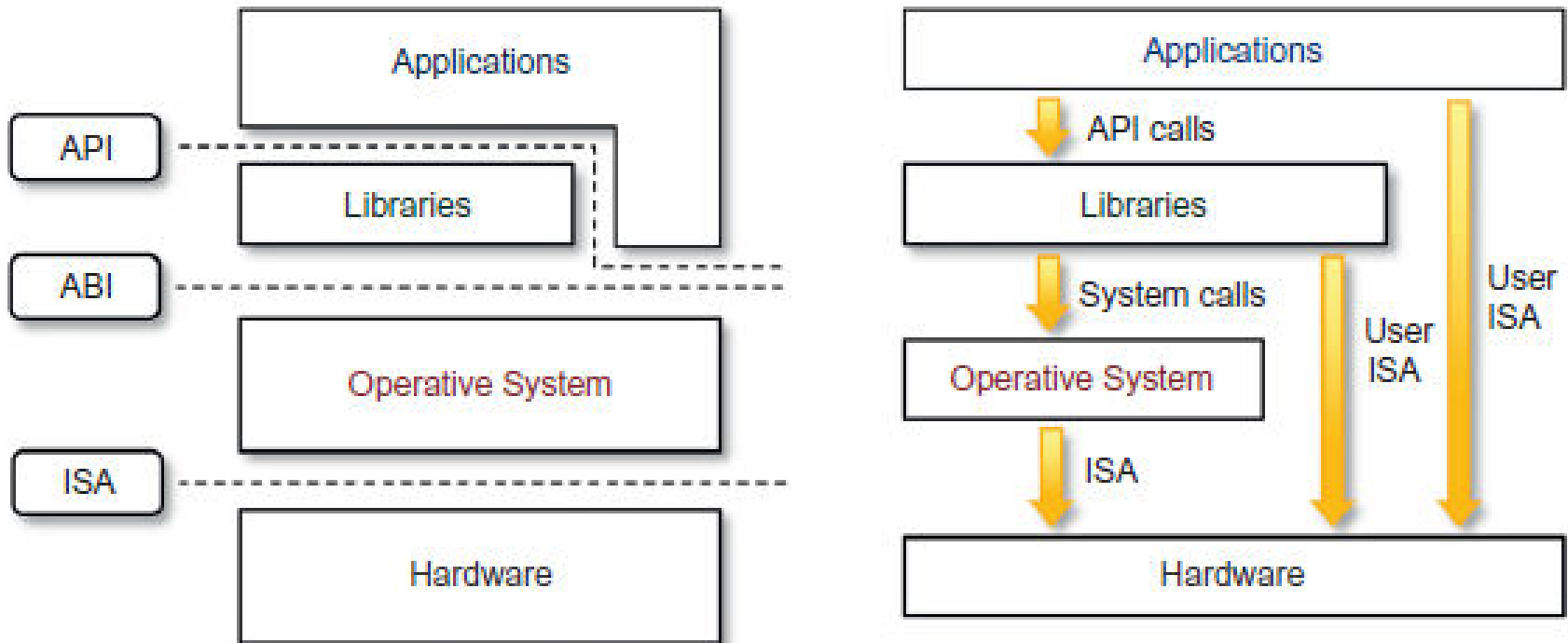


FIGURE 3.4

A machine reference model.

API – Application Programming Interface

ABI – Application Binary Interface

ISA – Instruction Set Architecture

This layered approach provides ways to implement a minimal security model for managing and accessing shared resources

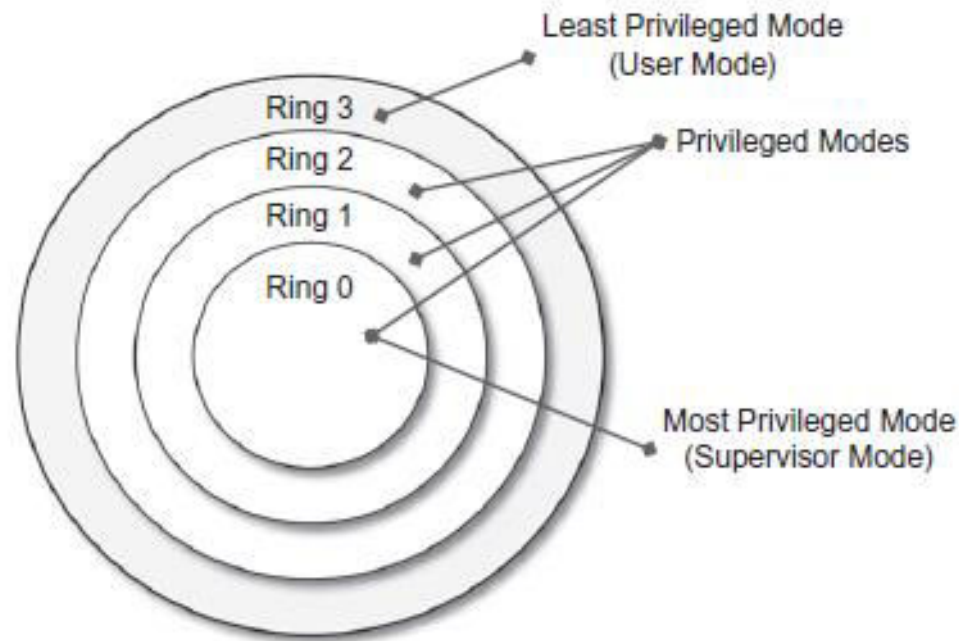


FIGURE 3.5

Security rings and privilege modes.

Hardware Level Virtualization

- Provides an abstract execution environment in terms of computer hardware on top of which a guest operating system can be run.
- 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**.

Hardware Level Virtualization

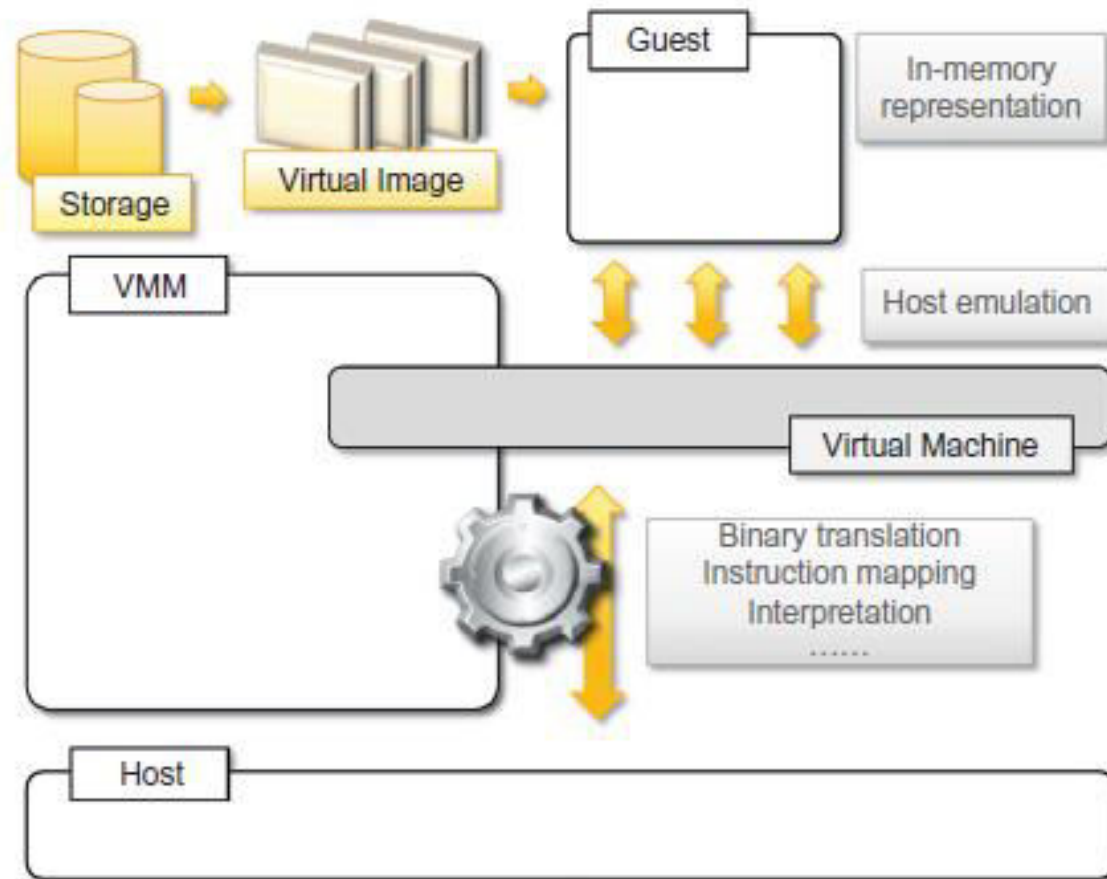


FIGURE 3.6

A hardware virtualization reference model.

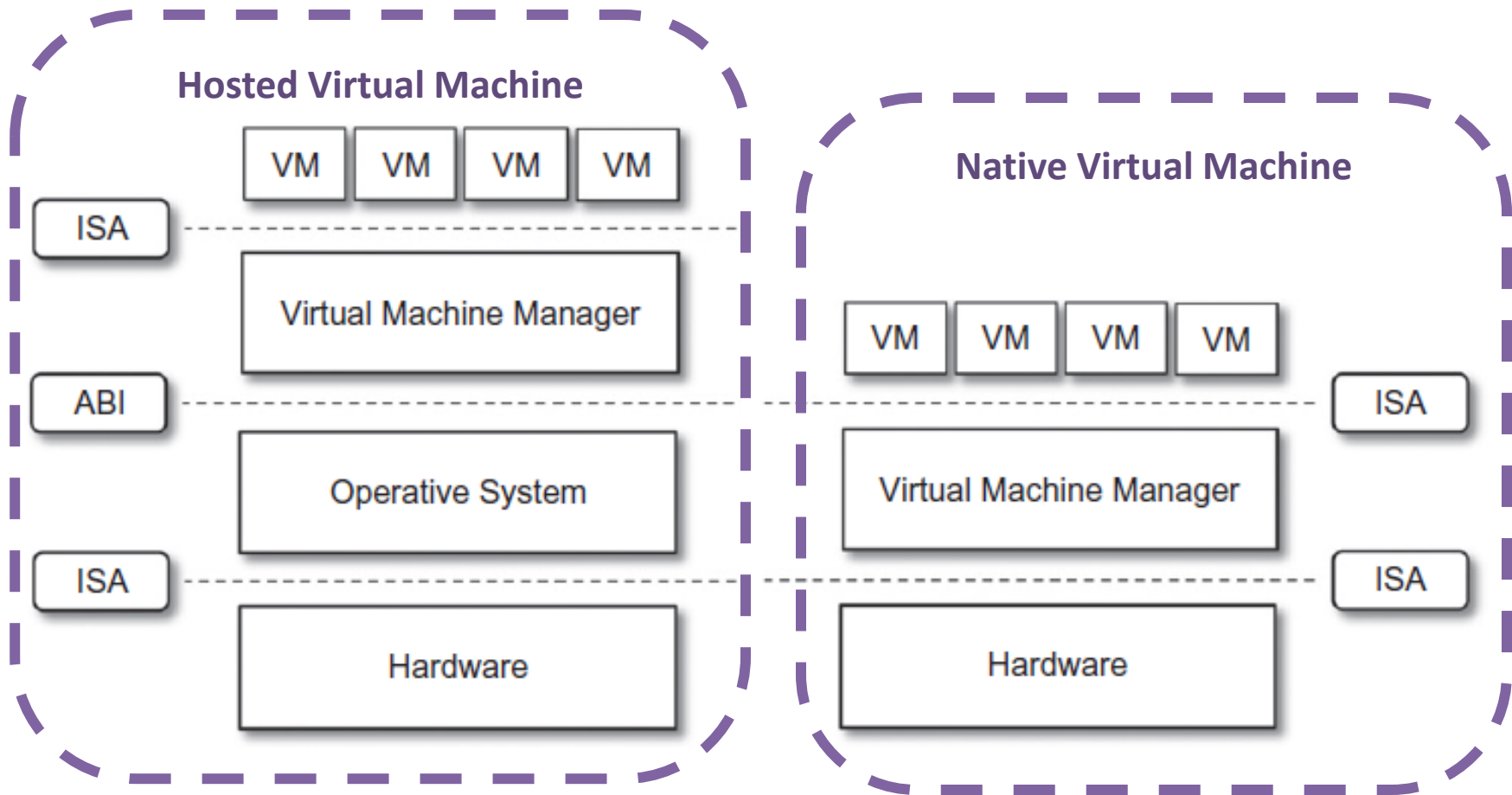
Hardware Level Virtualization

- The hypervisor is generally a program or a combination of software and hardware that allows the abstraction of the underlying physical hardware.
- Hardware level virtualization 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.

A Hardware Virtualization Reference Model

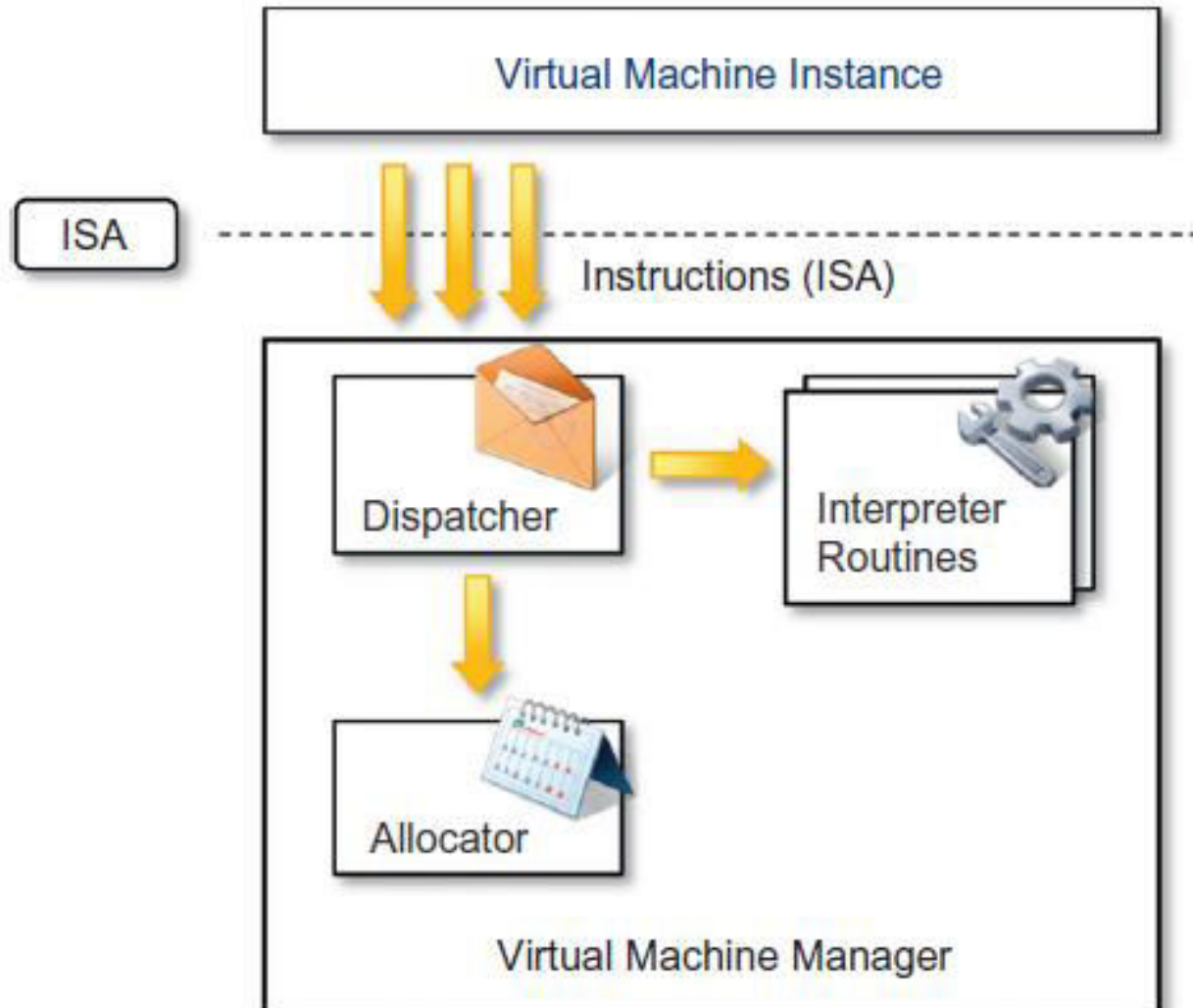
- Hypervisors - It recreates a hardware environment in which guest operating systems are installed
 - Type I
 - Run directly on top of the hardware
 - Interact directly with the ISA interface
 - Type II
 - Require the support of an operating system to provide virtualization services
 - Interact through ABI and emulate ISA of virtual hardware

Hypervisors



Hypervisor Reference Architecture

Internal architecture of virtual machine manager



Internal architecture of VMM

- Dispatcher
 - It is entry point of the monitor and reroutes the instructions issued by the virtual machine instance
- Allocator
 - Assigning system resources to VM
- Interpreter
 - Executed whenever a virtual machine executes a privileged instruction

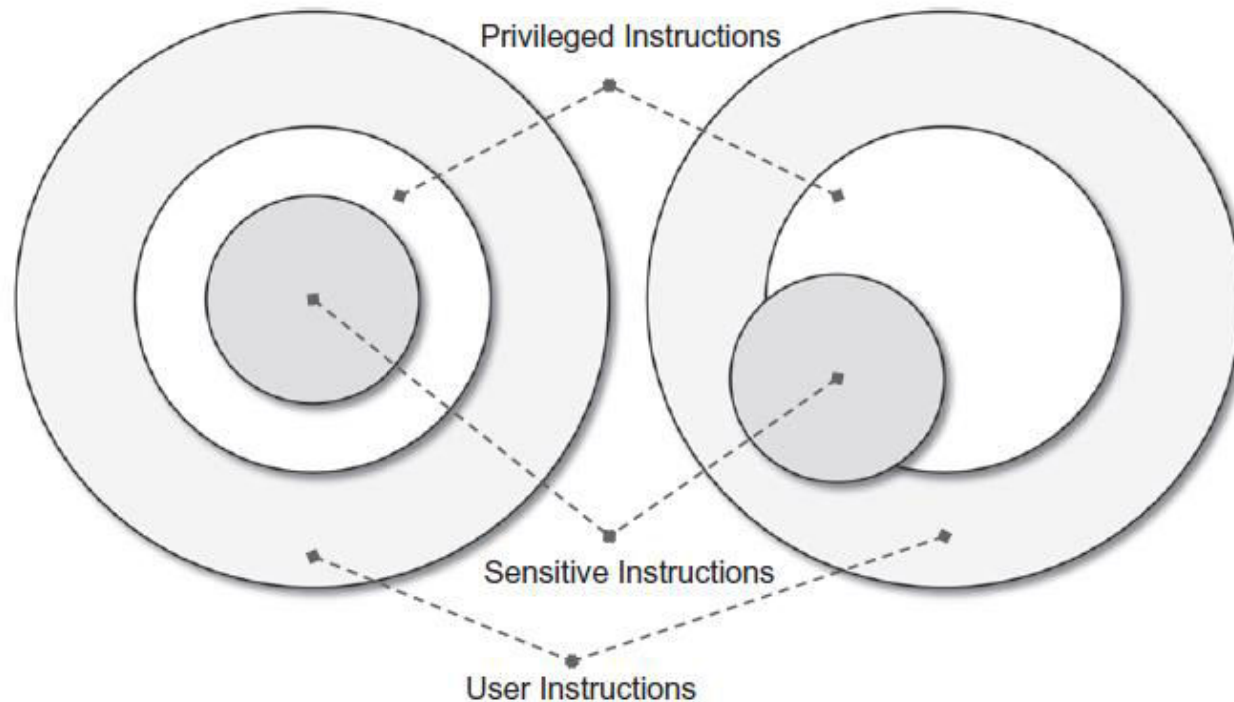
Criteria to efficiently support virtualization (Goldberg and Popek)

- **Equivalence:** A guest running under the control of a VMM should exhibit the same behavior as executed directly on the physical host.
- **Resource control:** The VMM 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.

Hardware-level Virtualization

- Popek and Goldberg provided properties that hardware instructions need to satisfy in order to efficiently support virtualization
- THEOREM3.1
 - For any conventional 3G computer, a VMM may be constructed if the set of **sensitive instructions** for that computer is a **subset** of the set of **privileged instructions**.

Hardware-level Virtualization



Virtualizable Computer

Non-Virtualizable Computer

Hardware-level Virtualization

- THEOREM3.2
 - A conventional third-generation computer is recursively virtualizable if:
 - It is virtualizable and
 - A VMM without any timing dependencies can be constructed for it.
- Recursive virtualization is the ability to run a VMM on top of another VMM.
- Virtualizable hardware is a prerequisite to recursive virtualization.

Hardware-level Virtualization

- THEOREM3.3
 - A hybrid VMM may be constructed for any conventional third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.
- hybrid virtual machine(HVM) considered less efficient than the virtual machine system.
- In HVM, more instructions are interpreted rather than being executed directly.
- All instructions in virtual supervisor mode are interpreted.

HARDWARE VIRTUALIZATION TECHNIQUES

Hardware-Assisted Virtualization

- Hardware provides architectural support for building a VMM able to run a guest OS in complete isolation.
- Example: x86-64 bit architecture introduced with Intel VT (Vanderpool) and AMD V (Pacifica)
- Before HAV, Software emulation of x86 hardware was significantly costly from the performance point of view. (as Popek and Goldberg requirement are not fulfilled)

Hardware-Assisted Virtualization

- Full Virtualization
 - 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.
 - VMM required to provide complete emulation of entire underlying hardware.
 - Advantage
 - Complete isolation lead to enhanced security
 - Ease of emulation of different architectures, and coexistence of different systems on the same platform.

Hardware-Assisted Virtualization

- Para Virtualization
 - Non-transparent virtualization solution that allows implementing thin VMM.
 - Expose software interface which is modified from the host.
 - Performance-critical operations directly on the host.
 - Guest operating systems need to be modified and explicitly ported by remapping the performance-critical operations through the virtual machine software interface.

Hardware-Assisted Virtualization

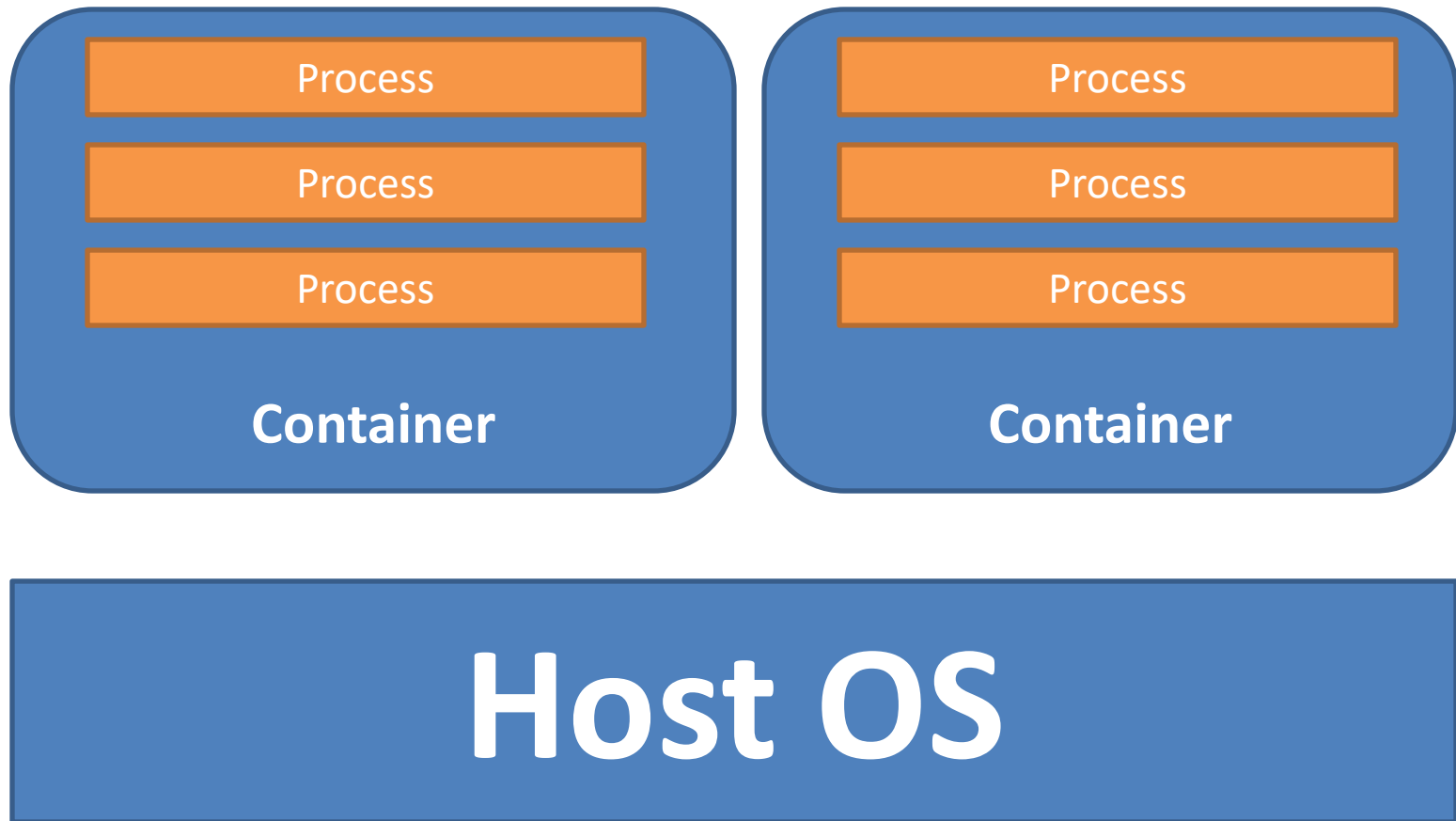
- Partial Virtualization
 - Provides a partial emulation of the underlying hardware.
 - Complete isolation not available.
 - Address space virtualization used in time-sharing systems allows multiple applications and users to run concurrently in a separate memory space, but they still share the same hardware resources (disk, processor, and network).

Operating System-level Virtualization

- Offers the opportunity to create different and separated execution environments for applications that are managed concurrently.
- No virtual machine manager or hypervisor.
- Virtualization is done within a single operating system.
- OS kernel allows for multiple isolated user space instances.

Operating System-level Virtualization

- Multiple isolated user-spaces
- Share one kernel
- Native performance



Operating System-level Virtualization

- 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

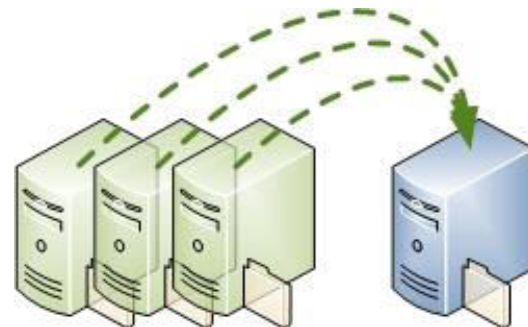
- Operating systems supporting this type of virtualization are general-purpose, time-shared operating systems with the capability to provide stronger namespace and resource isolation.
- Example: chroot

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.
- No need for emulation.
- No need to modify application to run them nor to modify any specific hardware.
- Do not expose the same flexibility of hardware virtualization.
- All the user space instances must share the same OS.

Operating System-level Virtualization

- This technique is an efficient solution for server consolidation scenarios in which multiple application servers share the same technology:
- When different servers are aggregated into one physical server, each server is run in a different user space, completely isolated from the others.



Operating System-level Virtualization

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

PROGRAMMING LANGUAGE-LEVEL VIRTUALIZATION

Programming Language-level Virtualization

- Also called process VM – provides uniform execution environment.
- Used to achieve ease of
 - deployment of applications,
 - managed execution, and
 - portability across different platforms and OS.
- It consists of a VM executing the byte code of a program, which is the result of the compilation process.
- Compilers produces a binary format representing the machine code for an abstract architecture.

Programming Language-level Virtualization

- The characteristics of this architecture vary from implementation to implementation.
- Generally these VM constitute a simplification of the underlying hardware instruction set and provide some high-level instructions that map some of the features of the languages compiled for them.
- At run time, the byte code can be either interpreted or compiled on the fly or jitted against the underlying hardware instruction set.

Programming Language-level Virtualization

- i.e. Basic Combined Programming Language (BCPL) - 1966, Java - 1996.
- Supports multiple programming languages using Common Language Infrastructure (CLI) in .NET Framework.

Programming Language-level Virtualization

- Two types in PL Virtualization
 - Stack based virtual machine
 - An execution stack used to perform operations.
 - i.e. Java, .NET
 - Register based virtual machine
 - Closer to underlying architecture used today.
 - i.e. Parrot – support execution of PERL, then generalized to execute dynamic languages

Programming Language-level Virtualization

- Advantages
 - Supports different platforms
 - Developer point of view simplifies the development and deployment efforts
 - Allows more control over execution of programs
 - Security – sandboxing of application
- Limitations
 - Advantages at the cost of performance

Programming Language-level Virtualization

- Implementations of this model are also called high-level virtual machines.
- As high-level programming languages are compiled to a conceptual ISA.
- Which is further interpreted or dynamically translated against the specific instruction of the hosting platform.

Application-level Virtualization

- Allows applications to be run in runtime environments that do not natively support all the features required by the application.
- Applications are not installed in the expected runtime environment but are run as though they were.
- These techniques are mostly concerned with partial file systems, libraries, and OS component emulation.
- Emulation is performed by a thin layer – a program/OS component – in charge of execution.

Application-level Virtualization

- Two strategies
- Interpretation
 - 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.

Application-level Virtualization

- Binary translation
 - 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 Level Virtualization

- Allows the execution of a program compiled against a different hardware.

Application Level Virtualization

- Works for a specific environment.

Hardware Level Virtualization

- Emulates a complete hardware environment where an entire operating system can be installed.

Programming Level Virtualization

- Works across all the applications developed for that virtual machine.

Application-level virtualization

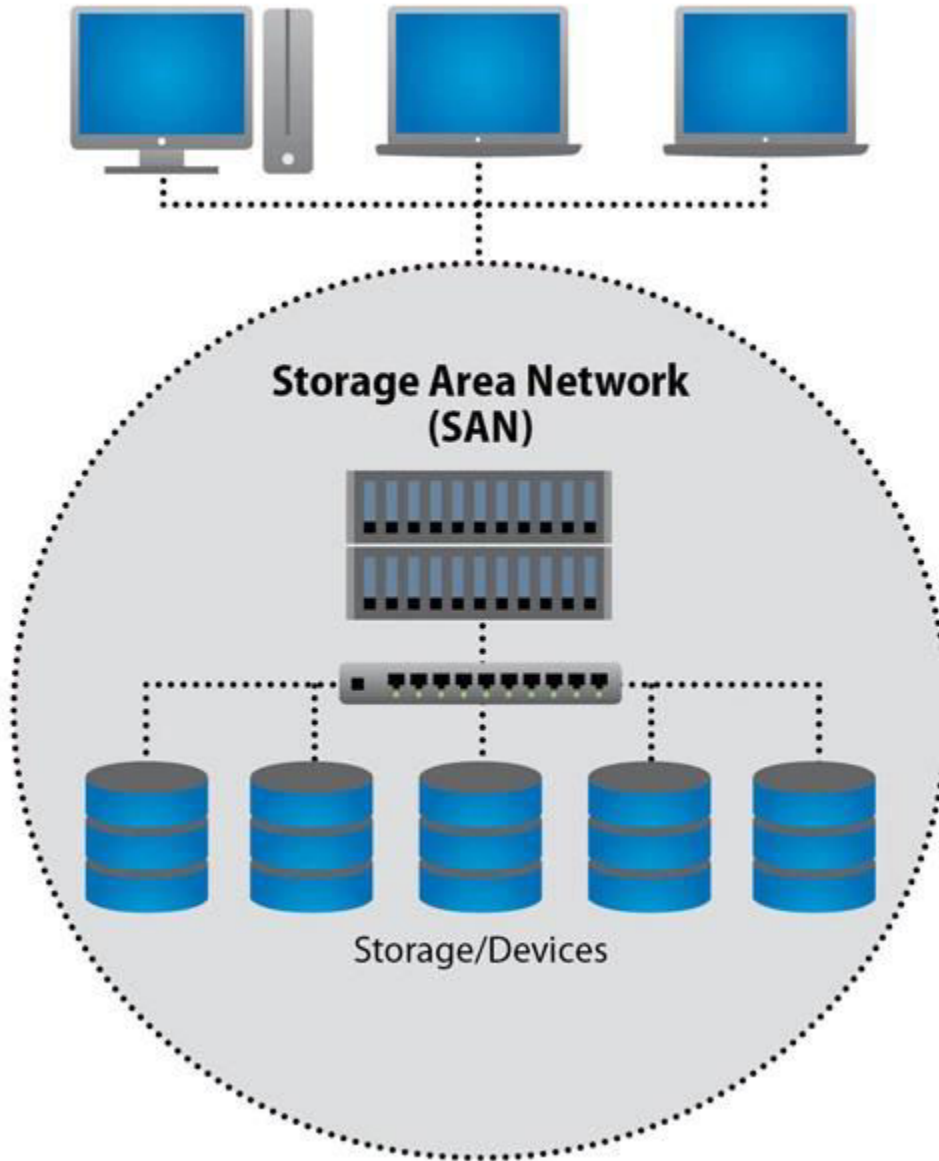
- Good solution in the case of missing libraries in the host operating system.
- Replacement library can be linked with the application, or library calls can be remapped to existing functions available in the host system.
- VMM is much lighter provides a partial emulation of the runtime environment compared to hardware virtualization.
- Allows incompatible applications to run together.

Application-level virtualization

- Examples:
- Wine - software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform.
- Windows Application Binary Interface(WABI) - which implements the Win 16 API specifications on Solaris.
- CrossOver – Similar solution for the Mac OS X
- VMware ThinApp – allow capturing the set up of installed application and package it to executable image isolated from hosting OS.

OTHER TYPES OF VIRTUALIZATION

- Allows c
the stor
- User de
specific
- i.e. Stor
 - SANs
large
faciliti



n

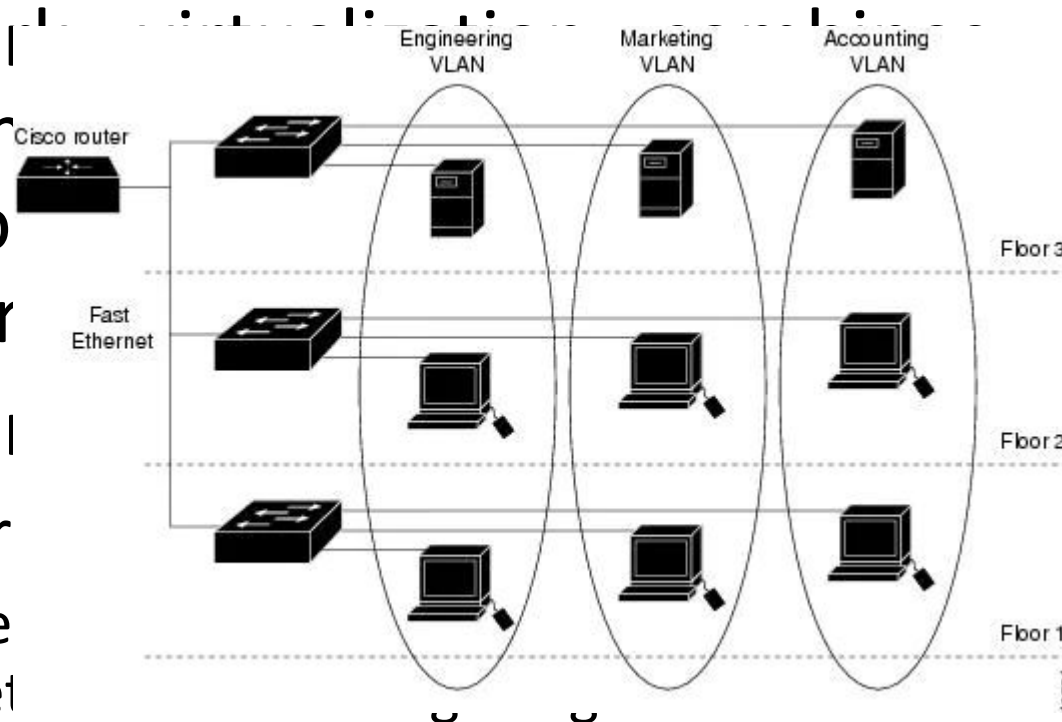
ization of
ation.

bout the

through a
de storage

Network Virtualization

- Network virtualization is the process of creating virtual networks over a physical network infrastructure.
- Network virtualization allows for the creation of multiple virtual networks on a single physical network.
- External network
- Network virtualization is used to create virtual networks that can be used to isolate different applications or users from each other.



hardware
for the
virtual

ent physical
LAN.

- A VLAN is an aggregation of hosts that communicate with each other as though they were located under the same broadcasting domain.

Network Virtualization

- Internal network virtualization
 - Provide network-like functionality to an operating system partition.
 - Applied together with hardware and operating system-level virtualization - guests obtain a virtual network interface to communicate with. Ways to implement:
 - The guest can share the same network interface of the host and use Network Address Translation (NAT) to access the network.
 - The VMM 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.

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.
- Remotely access a desktop environment.
- Same desktop environment accessible from everywhere.

Desktop Virtualization

- Generally the desktop environment is stored in a remote server or a data center that provides a high availability infrastructure and ensures the accessibility and persistence of the data.
- A specific desktop environment is stored in a virtual machine image that is loaded and started on demand when a client connects to the desktop environment.

Desktop Virtualization

- Advantages
 - high availability, persistence, accessibility, and ease of management
- 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 include Sun Virtual Desktop Infrastructure (VDI), Parallels Virtual Desktop Infrastructure (VDI), Citrix XenDesktop, and others.

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
- Same as storage virtualization: Provides a better quality of service rather than emulating a different environment.

**NEXT: VIRTUALIZATION AND CLOUD
COMPUTING**

Virtualization

Virtualization and Cloud Computing

Virtualization and Cloud Computing

- Virtualization plays an important role in cloud computing
 - it allows appropriate degree of Customization, Security, Isolation, Manageability that are fundamental for delivering IT service on demand.
- Virtualization technologies are primarily used to offer configurable computing environments and storage.

Virtualization and Cloud Computing

- Virtualization used in computing
 - Hardware virtualization is an enabling factor for solutions in the Infrastructure-as-a-Service (IaaS)
 - Programming language virtualization is a technology leveraged in Platform-as-a-Service (PaaS)

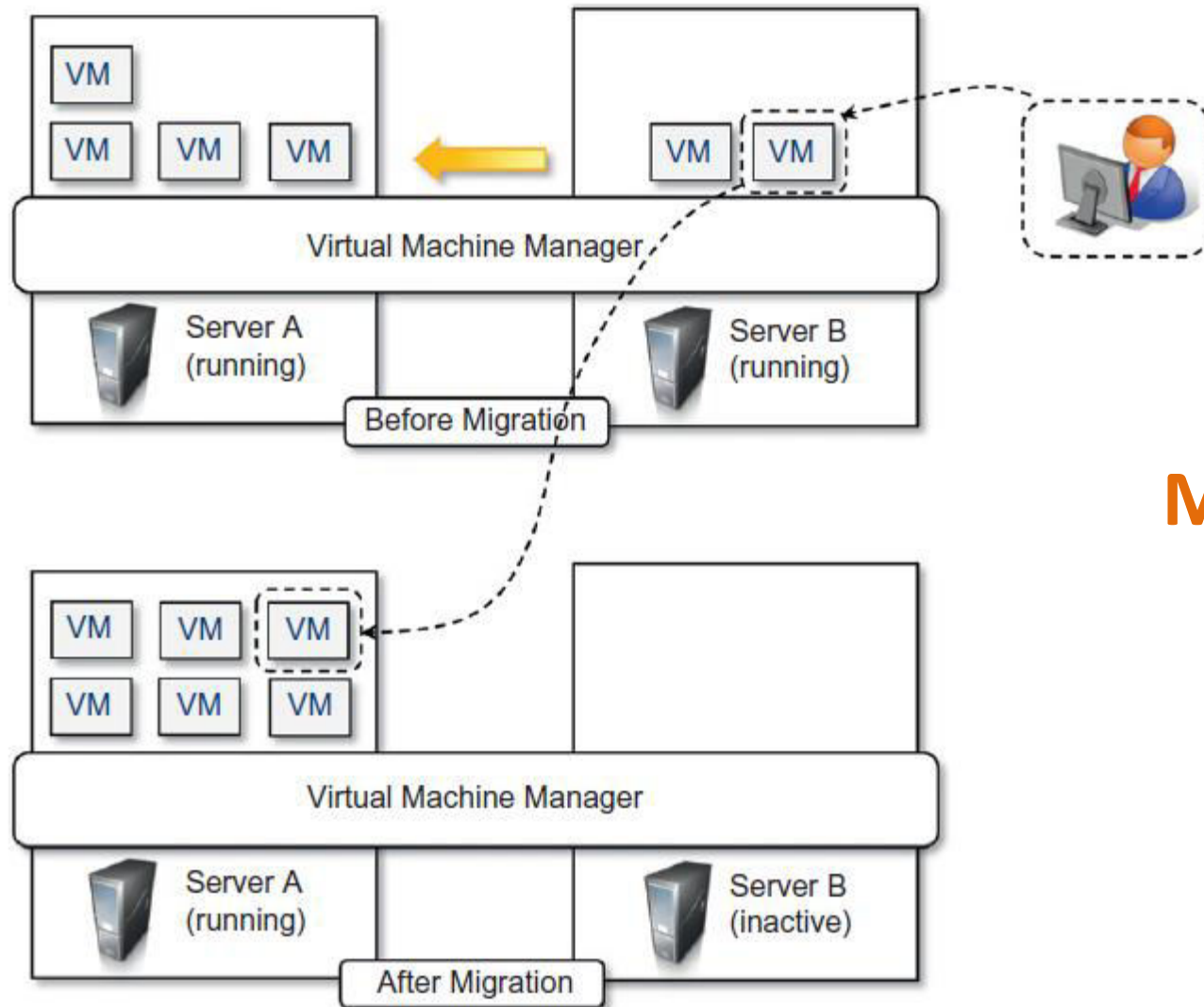
Virtualization and Cloud Computing

- Virtualization contribution in cloud computing
 - Customized and sandboxed environment
 - Isolation and finer control
- Supports service leasing and accountability

Virtualization and Cloud Computing

- Other contributions of Virtualization in cloud computing
 - Server Consolidation
 - Virtual machine migration
 - Offline migration
 - Live migration

Virtualization and Cloud Computing



**Live
Migration**

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.

Virtualization and Cloud Computing

- Storage virtualization constitutes an interesting opportunity
 - 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.
 - Other opportunities:
 - secure and protect the hosting infrastructure
 - easy accountability of services

Virtualization and Cloud Computing

- 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.

Pros and Cons

- Internet and the advancements in computing technology have made virtualization an interesting opportunity to deliver on-demand IT infrastructure and services

Pros

- Managed execution and isolation
 - 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.
 - Server consolidation
 - Allocation of resources and their partitioning among different guests is simplified, being the virtual host controlled by a program.
 - This enables fine-tuning of resources.
 - Effective quality of service.
 - Portability allows server consolidation.

Pros

- Portability
 - Virtual machine instances are normally represented by one or more files that can be easily transported with respect to physical systems.
 - They also tend to be self-contained since they do not have other dependencies besides the VMM for their use.
 - Portability and self-containment simplify their administration.

Pros

- Reduce cost of maintenance
 - Portability and self-containment also contribute to reducing the costs of maintenance, since the number of hosts $<$ number of virtual machine instances.
 - Very limited opportunity for the guest program to damage the underlying hardware.
 - Fewer VMM with respect to the number of virtual machine instances managed.

Pros

- More efficient use of resources
 - Multiple systems can securely coexist and share the resources of the underlying host, without interfering with each other.
 - This is a prerequisite for server consolidation.
 - Which allows adjusting the number of active physical resources dynamically according to the current load of the system.
 - Thus creating the opportunity to save in terms of energy consumption and to be less impacting on the environment

Cons

- Performance degradation
 - The guest can experience increased latencies due to virtualization layer.
 - In H/w virtualization the intermediate emulates a bare machine on top cause performance degradation.
 - Overhead introduced by the following activities:
 - Maintaining the status of virtual processors
 - Support of privileged instructions (trap and simulate privileged instructions)
 - Support of paging within VM
 - Console functions

Cons

- Degrade more when hardware virtualization is realized through a program.
- VMM is executed and scheduled together with other applications, thus sharing with them the resources of the host.
- Binary translation and interpretation slow down the execution of managed applications.
- **Solution: Paravirtualization - by offloading most of its execution to the host without any change**

Cons

- Inefficiency and degraded user experience
 - Inefficient use of the host.
 - some of the specific features of the host cannot be exposed by the abstraction layer and then become inaccessible.
 - i.e. Graphics support by Java – resolved by Swing and then support for OpenGL.

Cons

- Security holes and new threats
 - Virtualization opens the door to a new and unexpected form of phishing.
 - The capability of emulating a host in a completely transparent manner led the way to malicious programs that are designed to extract sensitive information from the guest.

Cons

-

SubVirt

SubVirt infects the guest OS, and when the virtual machine is rebooted, it gains control of the host.

Cons

- Security holes and new threats
 - Modified versions of the runtime environment can access sensitive information or monitor the memory locations utilized by guest applications while these are executed.
 - To make this possible, the original version of the runtime environment needs to be replaced by the modified one, which can generally happen if the malware is run within an administrative context or a security hole of the host operating system is exploited.

NEXT: TECHNOLOGY EXAMPLES

Cloud Reference Model

Definition

- Cloud computing is a **utility-oriented** and **Internet-centric** way of delivering **IT services on demand**.
- These services cover the **entire computing stack**: from the hardware infrastructure packaged as a set of virtual machines to software services such as development platforms and distributed applications.

Architecture

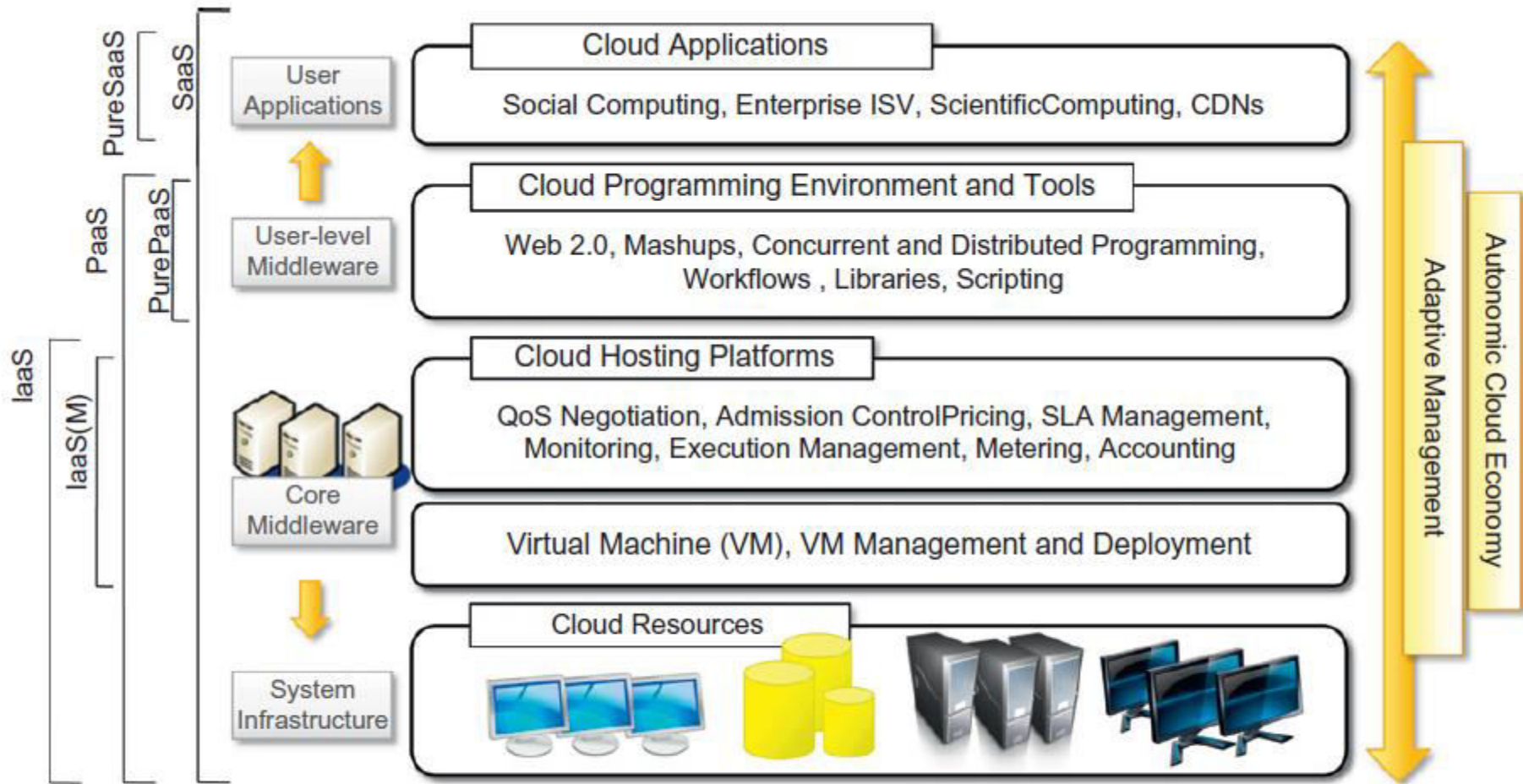


FIGURE 4.1

The cloud computing architecture.

Cloud resources

- Harnessed to offer “computing horsepower” required for providing services.
- Implemented using a datacenter .
- Heterogeneous in nature – cluster, n/w PC, database system, storage services

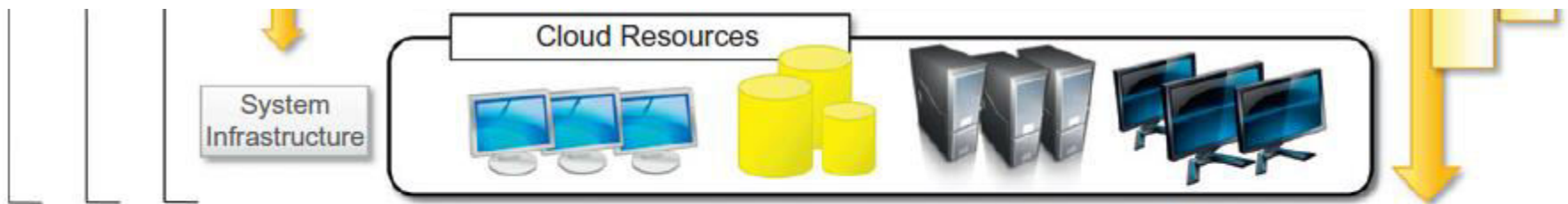


FIGURE 4.1

The cloud computing architecture.

Architecture

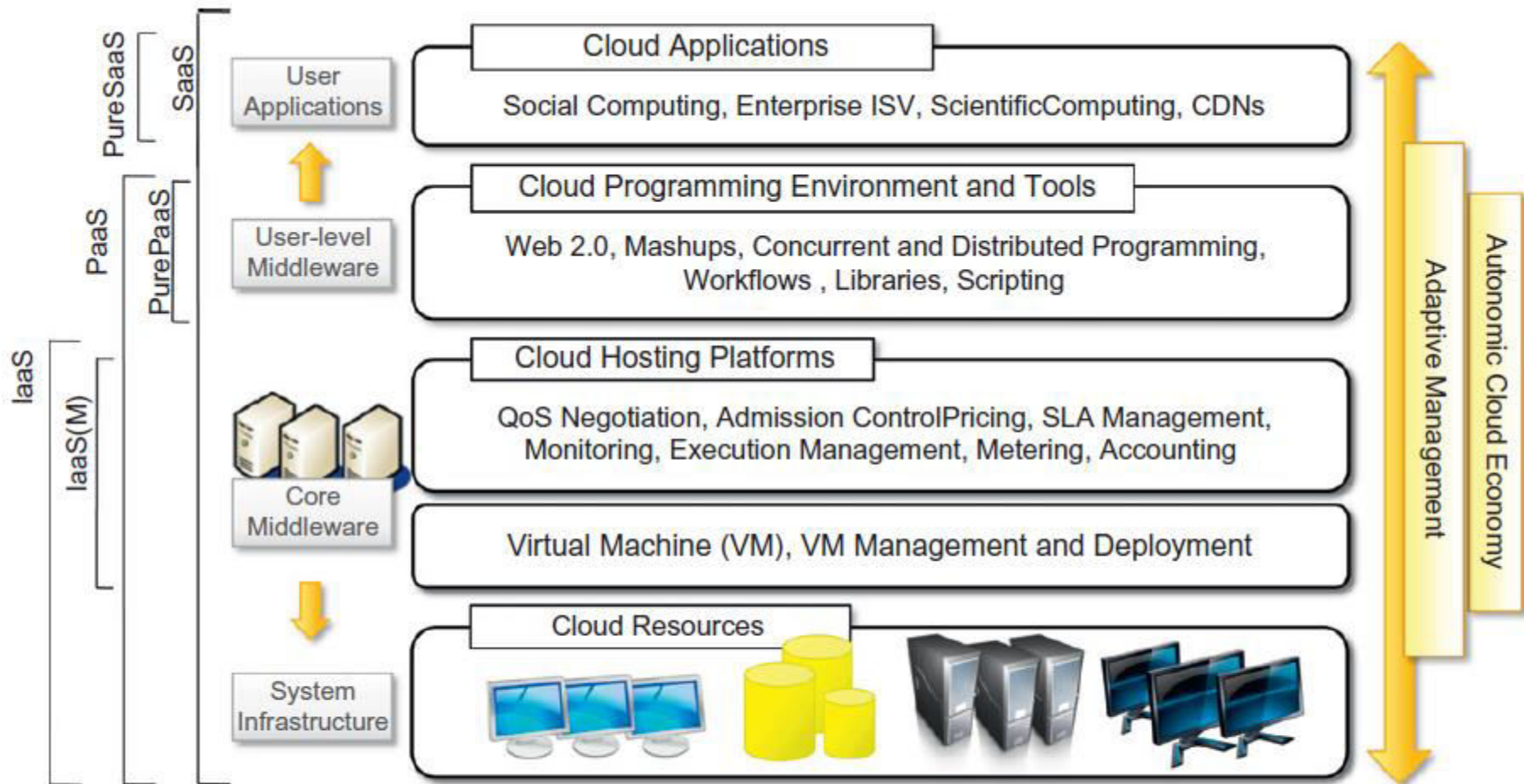
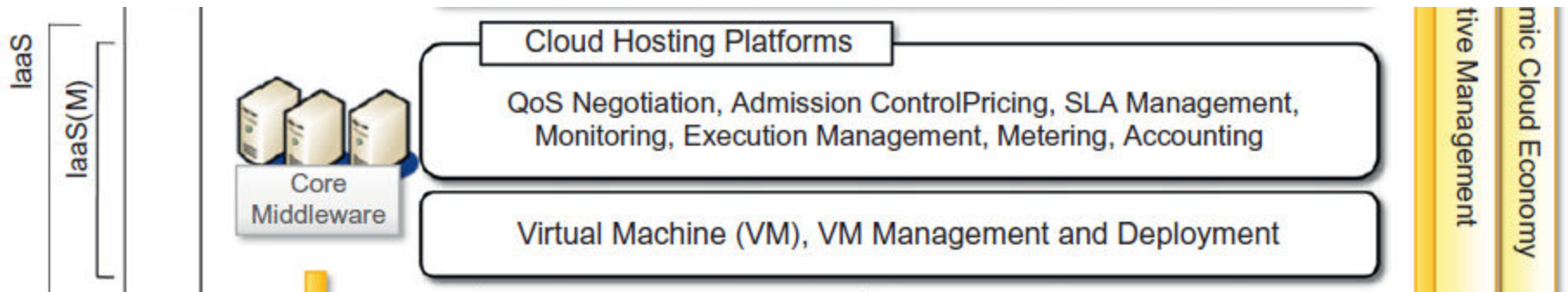


FIGURE 4.1

The cloud computing architecture.

Core Middleware

- The physical infrastructure is managed by the core middleware.



- To provide an appropriate runtime environment for applications and to best utilize resources.

Core Middleware

- Virtualization technologies are used to guarantee runtime environment customization, application isolation, sandboxing, and quality of service.
- Hardware virtualization is most commonly used at this level.
- Hypervisors manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines.

Core Middleware

- By using virtual machine technology it is possible to finely partition the hardware resources such as CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications.
- This solution is generally paired with storage and network virtualization strategies, which allow the infrastructure to be completely virtualized and controlled.

Core Middleware

- Infrastructure management is the key function of core middleware, which supports capabilities such as negotiation of the quality of service, admission control, execution management and monitoring, accounting, and billing.
- The combination of cloud hosting platforms and resources is generally classified as a Infrastructure-as-a-Service(IaaS) solution.
 - Some of them provide both the management layer and the physical infrastructure;
 - others provide only the management layer (IaaS (M)). The management layer is often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.

Architecture

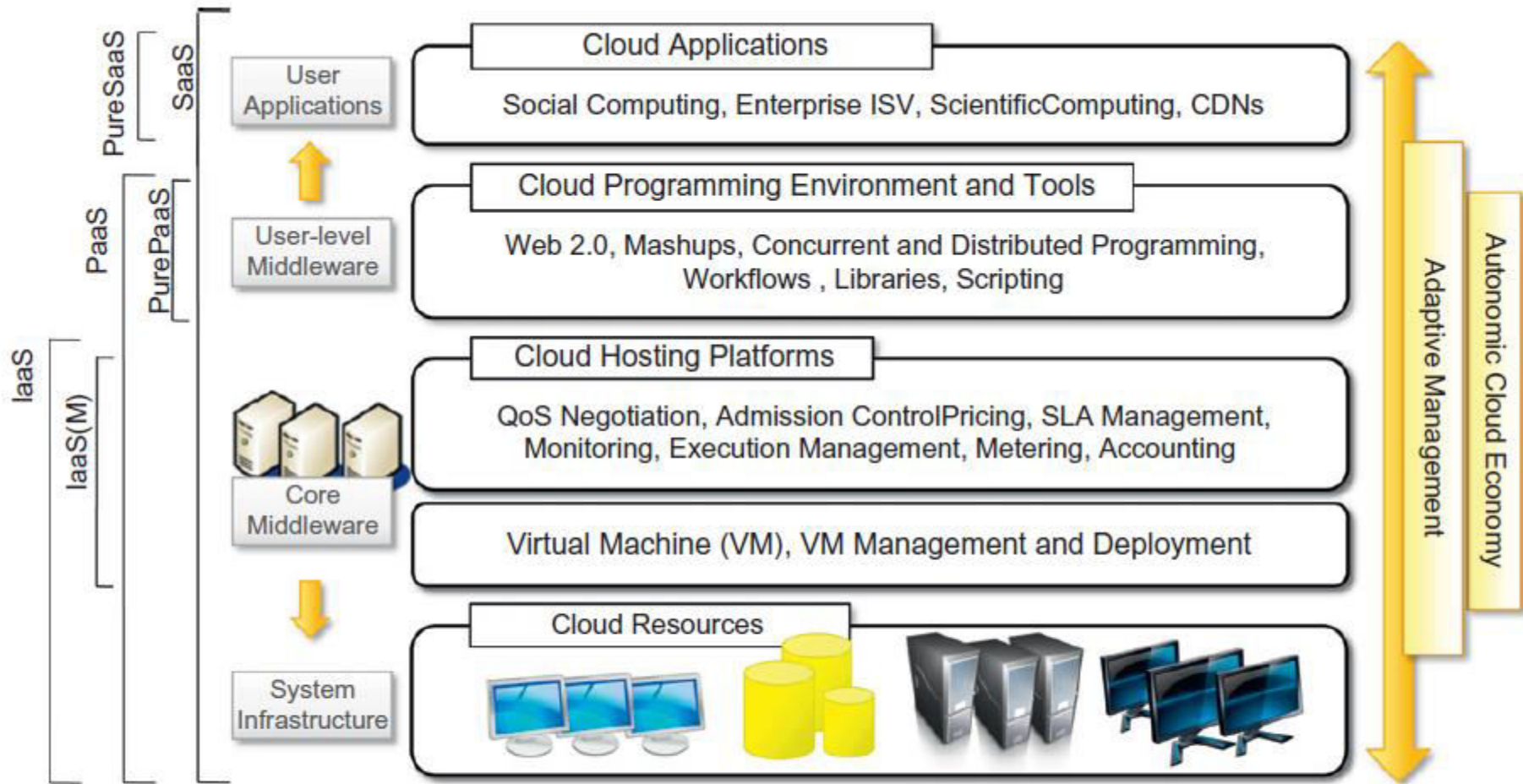


FIGURE 4.1

The cloud computing architecture.

User-level Middleware



- IaaS solutions are suitable for designing the system infrastructure.
 - But it provides limited services to build applications.
- Such service is provided by cloud programming environments and tools, which form a new layer (User-level Middleware) for offering users a development platform for applications.

User-level Middleware

- Users develop their applications specifically for the cloud by using the API exposed at the user-level middleware.
- The range of tools include
 - Web-based interfaces,
 - command-line tools, and
 - frameworks for concurrent and distributed programming.

User-level Middleware

- This approach is also known as Platform-as-a-Service(PaaS) because the service offered to the user is a development platform rather than an infrastructure.
- PaaS solutions generally include the infrastructure as well, which is bundled as part of the service provided to users.
- In the case of Pure PaaS, only the user-level middleware is offered, and it has to be complemented with a virtual or physical infrastructure.

Architecture

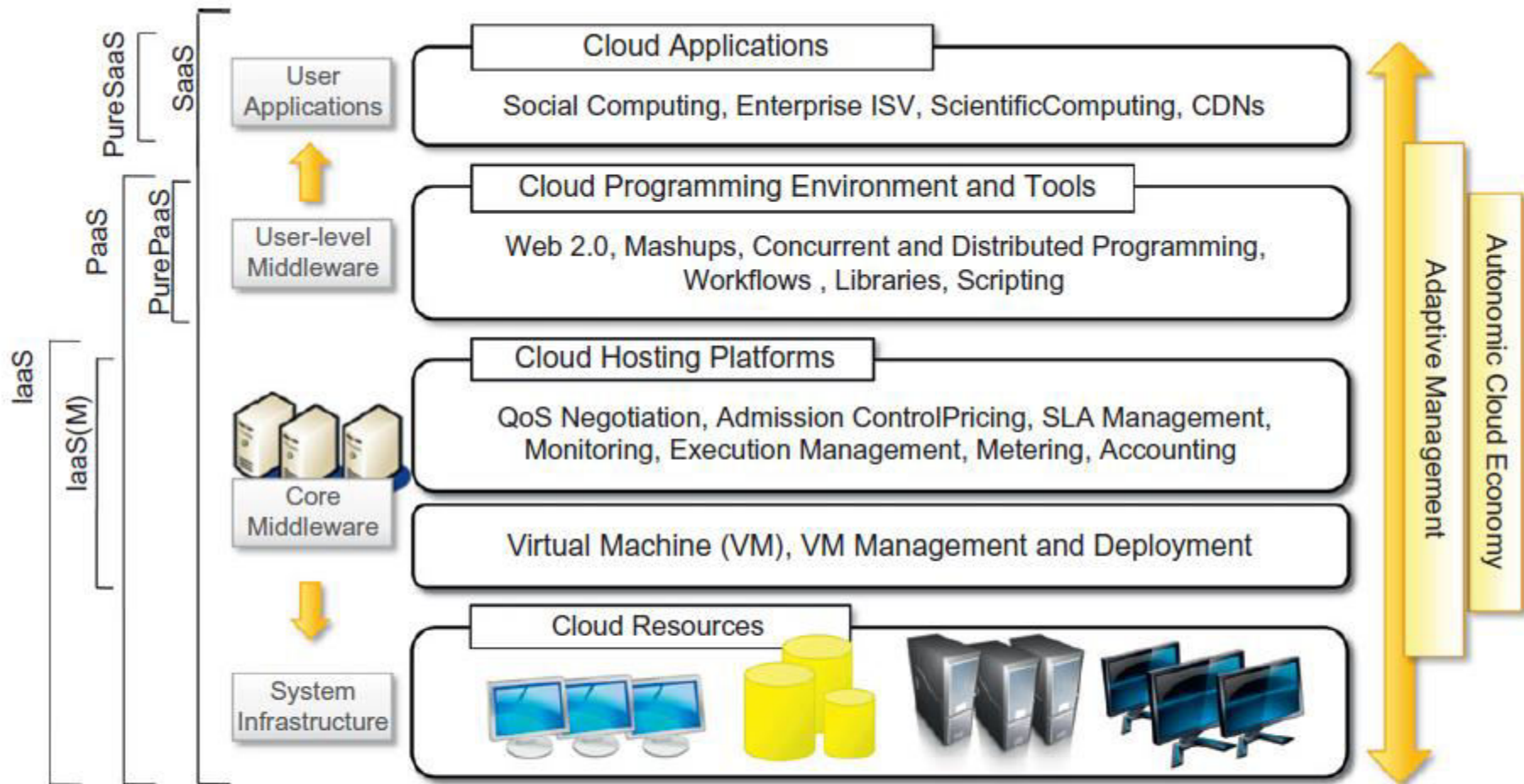


FIGURE 4.1

The cloud computing architecture.

User Applications



- The top layer of the reference model contains services delivered at the application level.
- These are mostly referred to as Software-as-a-Service(SaaS).
- In most cases these are Web-based applications that rely on the cloud to provide service to end users.

User Applications

- The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.
- Other applications belonging to this layer are those that **strongly leverage the Internet** for their core functionalities that rely on the cloud to sustain a **larger number of users**;
- Examples: gaming portals, social networking websites.


Reference Model

- Any service offered in the cloud computing style should be able to **adaptively change** and expose an **autonomic behavior**, in particular **for its availability and performance**.
- As a reference model, it is then expected to have an **adaptive management layer in charge of elastically scaling on demand**.
- SaaS implementations should feature such behavior automatically, whereas PaaS and IaaS generally provide this functionality as a part of the API exposed to users.

Reference Model

- The reference model also introduces the concept of everything as a Service (XaaS).
- This is one of the most important elements of cloud computing.
- Cloud services from different providers can be combined to provide a completely integrated solution covering all the computing stack of a system.
- IaaS providers can offer the bare metal in terms of virtual machines where PaaS solutions are deployed.

Architecture

- When there is no need for a PaaS layer, it is possible to directly customize the virtual infrastructure with the software stack needed to run applications.
- This is a system and low software cost solution. 

Web Farms
Hosting single web site on multiple web servers over load balancer is called web farm.
- This provides an interesting option for reducing startups' capital investment in IT, allowing them to quickly commercialize their ideas and grow their infrastructure according to their revenues.

**NEXT: CLOUD COMPUTING
SERVICES CLASSIFICATION**

Virtualization

Technology Examples

Xen: Paravirtualization

- Xen is an open-source initiative implementing a virtualization platform based on paravirtualization.
- Initially developed by a group of researchers at the University of Cambridge.
- Citrix also offers it as a commercial solution, XenSource.

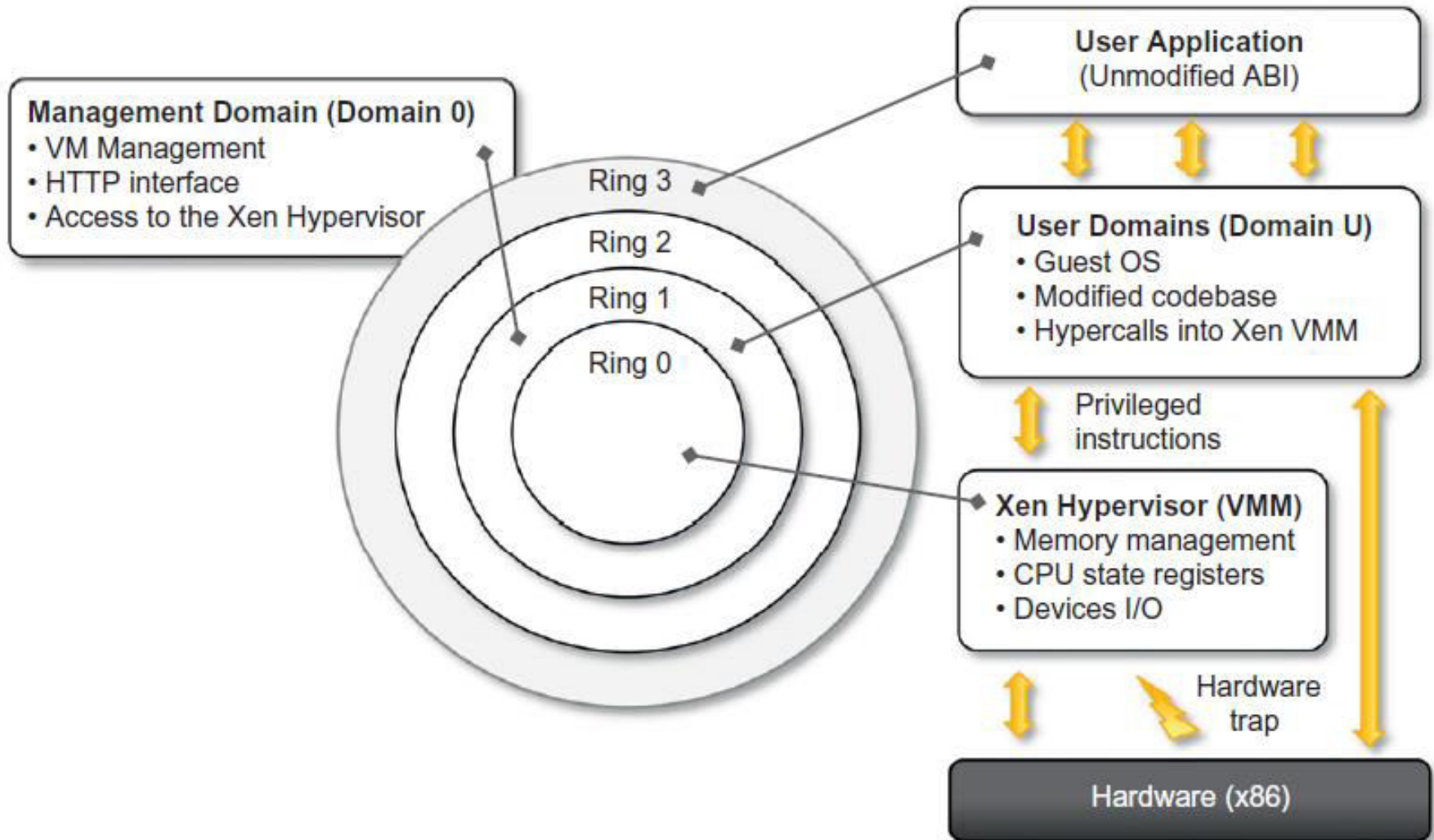
Xen: Paravirtualization

- Used for either desktop virtualization or server virtualization
- It has also been used to provide cloud computing solutions by means of Xen Cloud Platform (XCP)
- Xen is the most popular implementation of paravirtualization, which, in contrast with full virtualization, allows high-performance execution of guest operating systems

Xen: Paravirtualization

- Limitations:
 - Paravirtualization needs the operating system codebase to be modified, and hence not all operating systems can be used as guests in a Xen-based environment
 - Open-source operating systems such as Linux can be easily modified, since their code is publicly available and Xen provides full support for their virtualization, whereas components of the Windows family are generally not supported by Xen unless hardware-assisted virtualization is available.

Xen: Paravirtualization



VMware: Full Virtualization

- VMware implements full virtualization either
 - in the desktop environment, by means of Type II hypervisors, or
 - in the server environment, by means of Type I hypervisors
- Supports direct execution (for non-sensitive instructions) and binary translation (for sensitive instructions)

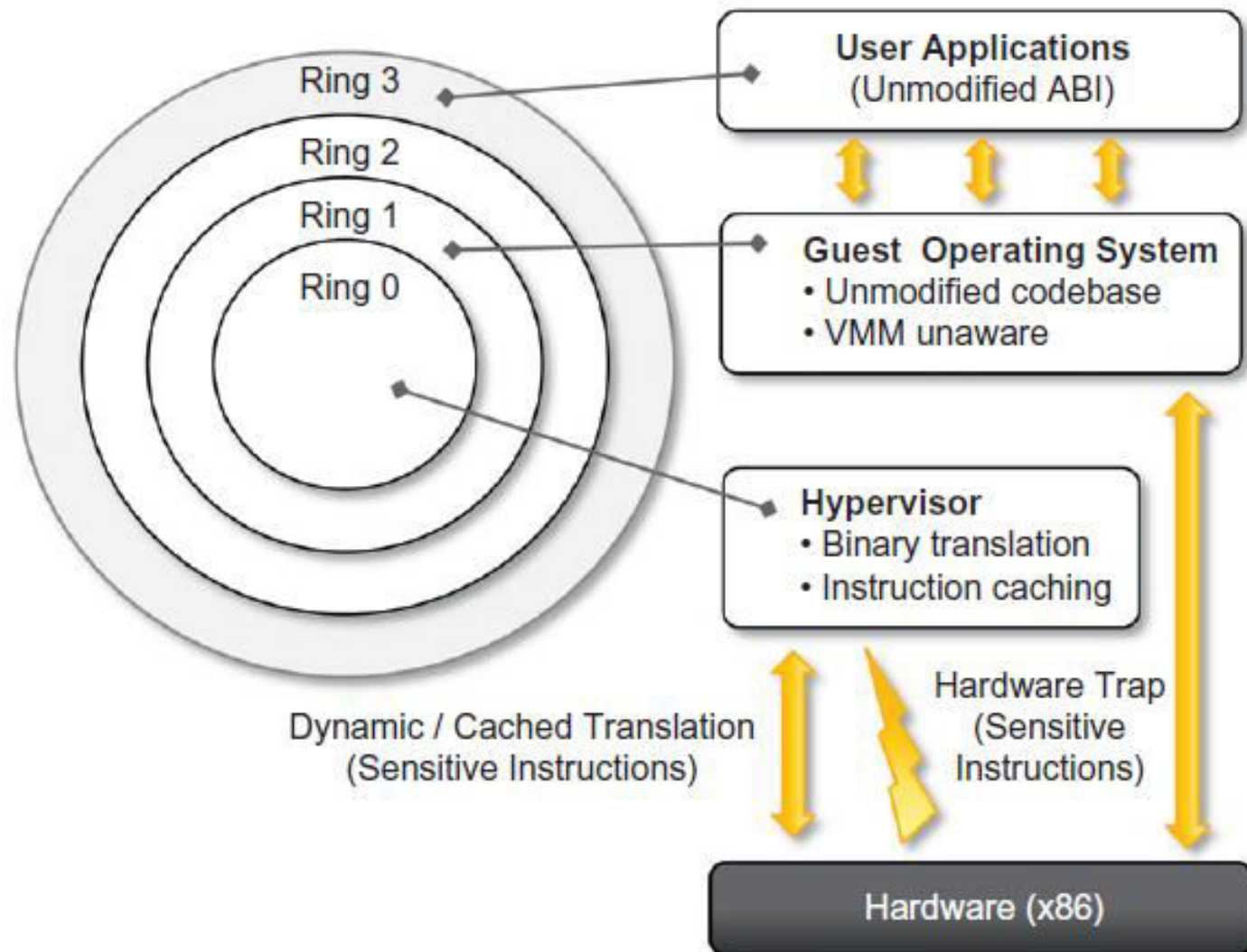


FIGURE 3.12

A full virtualization reference model.

VMware: Full Virtualization

- Full virtualization and binary translation
 - Dynamic binary translation allowed running x86 guest operating systems unmodified in a virtualized environment
 - x86 architecture design does not satisfy the first theorem of virtualization, since the set of sensitive instructions is not a subset of the privileged instructions
 - Sensitive instructions are not executed in Ring 0, which is the normal case in a virtualization scenario where the guest OS is run in Ring 1

VMware: Full Virtualization

- Generally, a trap is generated and managed
- In the case of dynamic binary translation, the trap triggers the translation of the offending instructions into an equivalent set of instructions that achieves the same goal without generating exceptions.
- To improve performance, the equivalent set of instruction is cached
- VMware achieves full virtualization by providing virtual representation of memory and I/O devices in addition to CPU virtualization

- Advantage
 - guests can run unmodified in a virtualized environment
- Limitation
 - translating instructions at runtime introduces an additional overhead

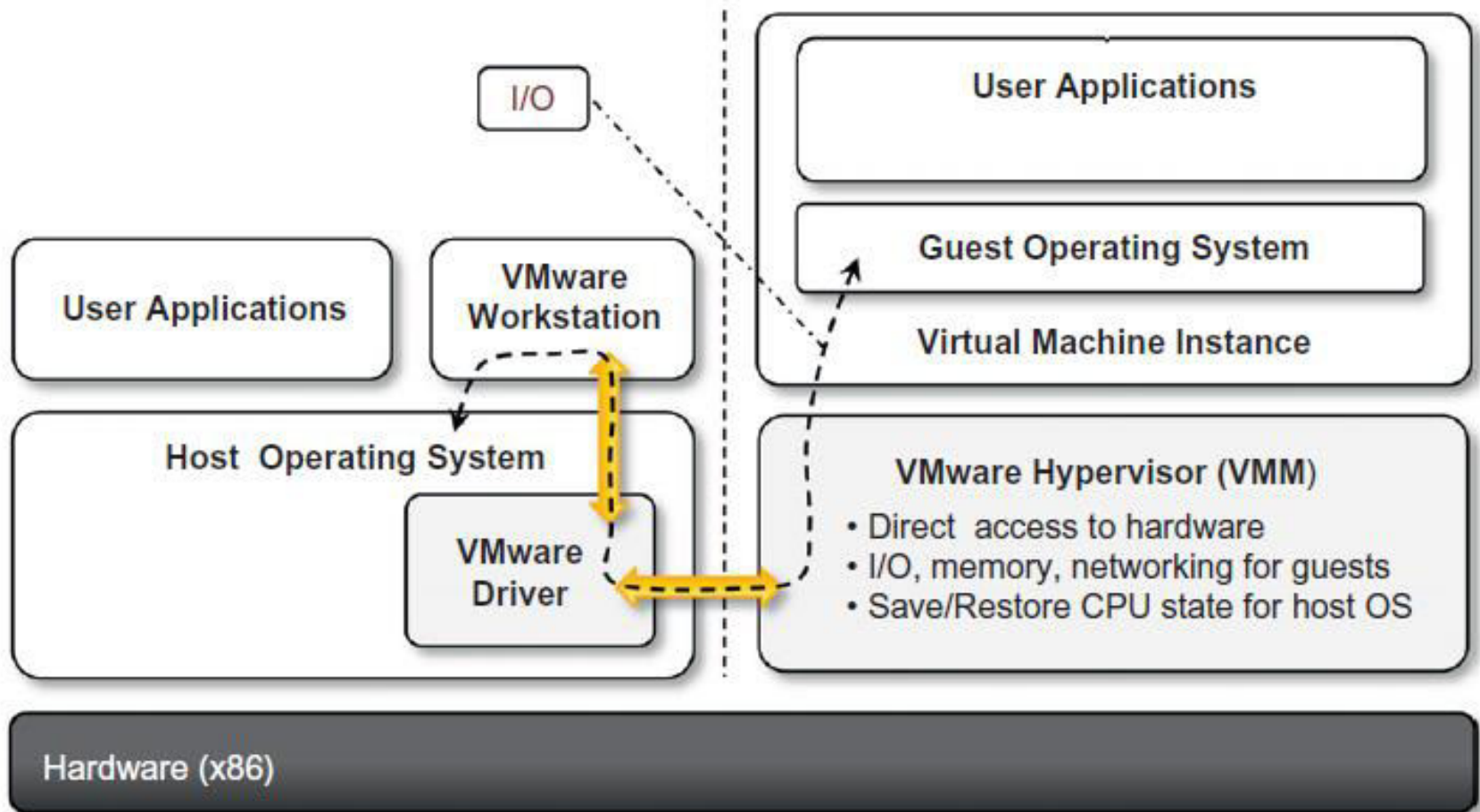


FIGURE 3.13

VMware workstation architecture.

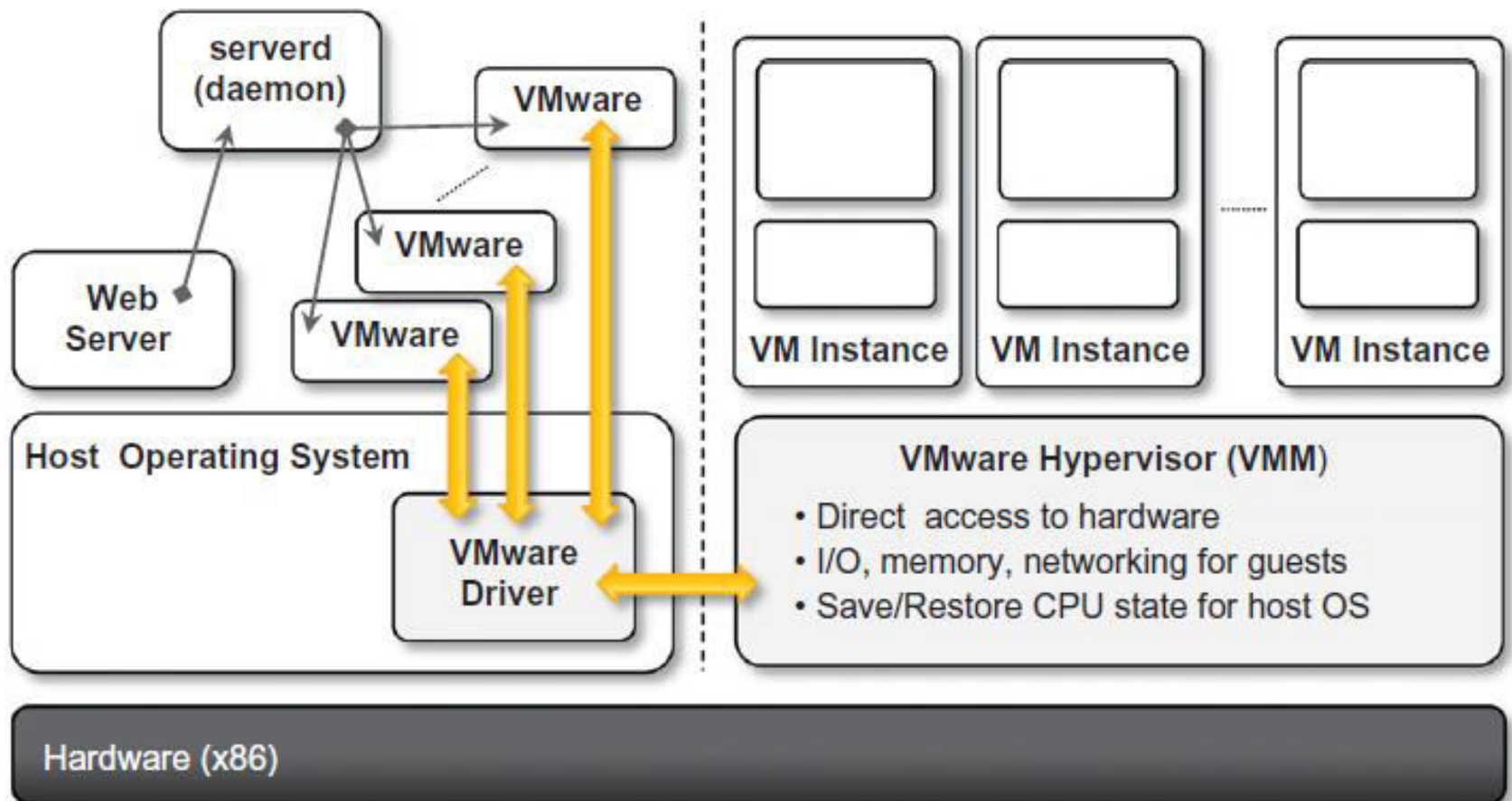


FIGURE 3.14

VMware GSX server architecture.

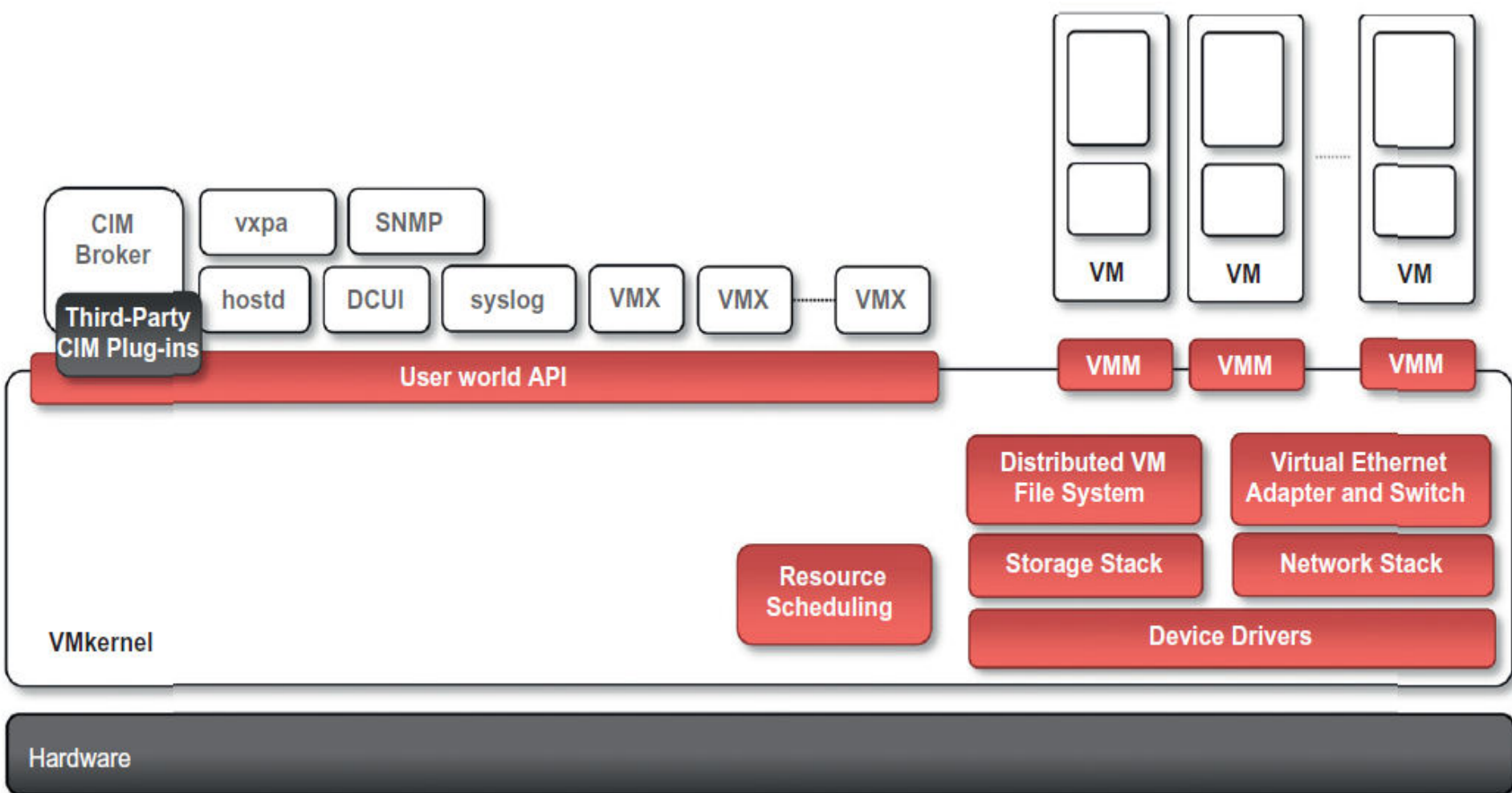


FIGURE 3.15
VMware ESXi server architecture.

