USN											

Internal Assessment Test 1 – Set 3 – April 2022

Sub:	Cloud Computing and its Applications				Sub Code:	18CS643	Branch	ISE	
Date:	25/04/2023	Duration:	90 min's	Max Marks:	50	Sem/Sec:	VI / A & B	OBE	
	Answer any FIVE questions						MARKS	CO	RBT
1	(a) Briefly explain the concept of Cloud Computing?						3 7	CO1	L1

Cloud computing allows renting infrastructure, runtime environments, and services on a pay-per-use basis. This principle finds several practical applications and then gives different images of cloud computing to different people. Chief information and technology officers of large enterprises see opportunities for scaling their infrastructure on demand and sizing it according to their business needs. End users leveraging cloud computing services can access their documents and data anytime, anywhere, and from any device connected to the Internet. Many other points of view exist.

A broad definition of the phenomenon could be as follows:
"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."

(b) Explain the Cloud Computing reference model with a neat diagram.

The Cloud Reference Model
Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. Such characterization includes quite different aspects: infrastructure, development platforms, application and services.

Architecture

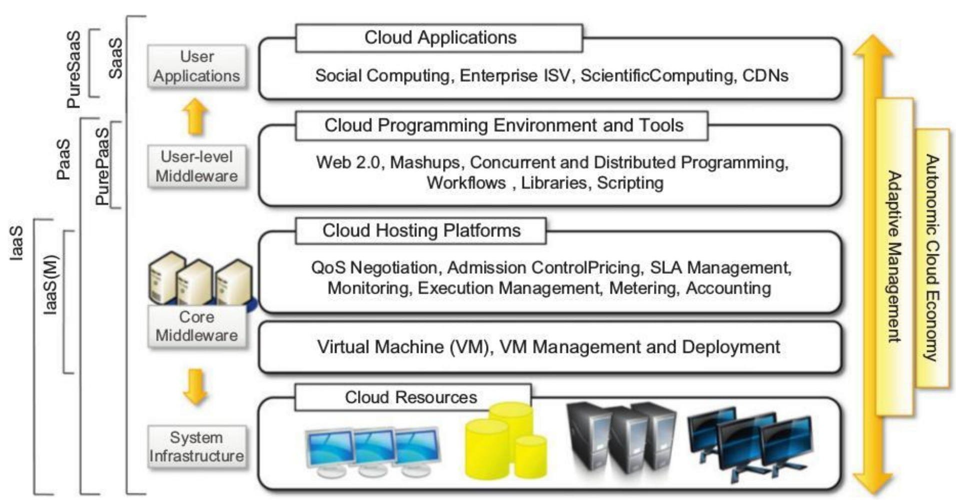


FIGURE 4.1

The cloud computing architecture.

It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack (see Figure 4.1), from hardware appliances to software systems. Cloud resources are harnessed to offer "computing horsepower" required for providing services. Cloud infrastructure can be heterogeneous in nature because a variety of resources, such as clusters and even networked PCs, can be used to build it.

The physical infrastructure is managed by the core middleware, the objectives of which are to provide an appropriate runtime environment for applications and to best utilize resources. At the bottom of the stack, 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. 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.

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 an Infrastructure-as-a-Service (IaaS) solution. We can organize the different examples of IaaS into two categories: Some of them provide both the management layer and the physical infrastructure; others provide only the management layer (IaaS (M)).

In this second case, the management layer is often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.

IaaS solutions are suitable for designing the system infrastructure but provide limited services to build applications. Such service is provided by cloud programming environments and tools, which form a new layer for offering users a development platform for applications.

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

The top layer of the reference model depicted in Figure 4.1 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. The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.

Table 4.1 summarizes the characteristics of the three major categories used to classify cloud computing solutions. In the following section, we briefly discuss these characteristics along with some references to practical implementations.

Table 4.1 Cloud Computing Services Classification

Category	Characteristics	Product Type	Vendors and Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizen.com (project management) Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

2	(a) What is Virtualization?	3 7	CO1	L1

Virtualization is another core technology for cloud computing. It encompasses a collection of solutions allowing the abstraction of some of the fundamental elements for computing, such as hardware, runtime environments, storage, and networking. Virtualization has been around for more than 40 years, but its application has always been limited by technologies that did not allow an efficient use of virtualization solutions.

Virtualization is essentially a technology that allows creation of different computing environments. These environments are called virtual because they simulate the interface that is expected by a guest. The most common example of virtualization is hardware virtualization.

Virtualization technologies are also used to replicate runtime environments for programs. Applications in the case of process virtual machines (which include the foundation of technologies such as Java or .NET), instead of being executed by the operating system, are run by a specific program called a virtual machine. This technique allows isolating the execution of applications and providing a finer control on the resource they access.

(b) Explain the 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 (see Figure given below).

The first classification discriminates against the service or entity that is being emulated.

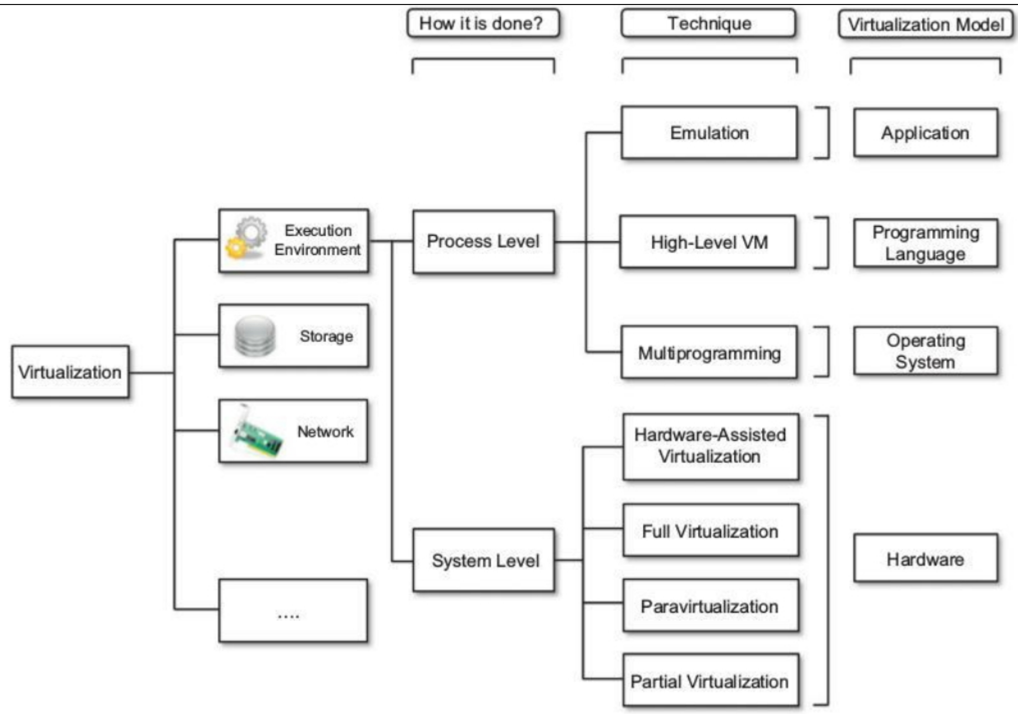


FIGURE 3.3

A taxonomy of virtualization techniques.

1. 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.

1. Machine reference model
2. Hardware-level virtualization
    a. Hypervisors
    b. Hardware virtualization techniques
    c. Operating system-level virtualization
3. Programming language-level virtualization
4. Application-level virtualization

1) Machine Reference Model

Modern computing systems can be expressed in terms of the reference model described in Figure 3.4. 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 interrupt 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). 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.

The highest level of abstraction is represented by the application programming interface (API), which interfaces applications to libraries and/or the underlying operating system.

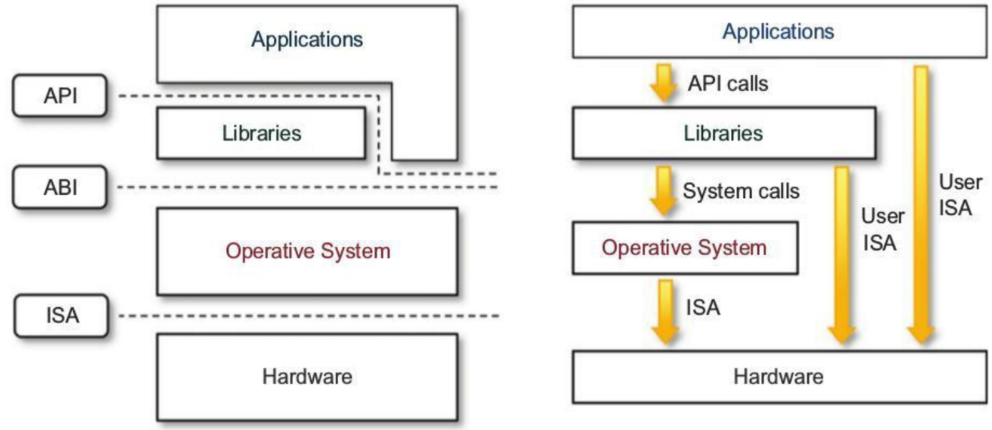


FIGURE 3.4

A machine reference model.

For this purpose, 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. 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, a possible implementation features a hierarchy of privileges (see Figure 3.5) 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.

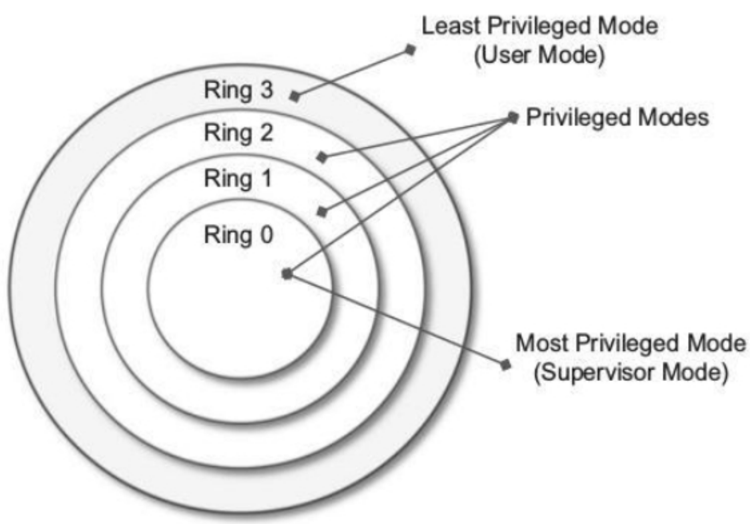
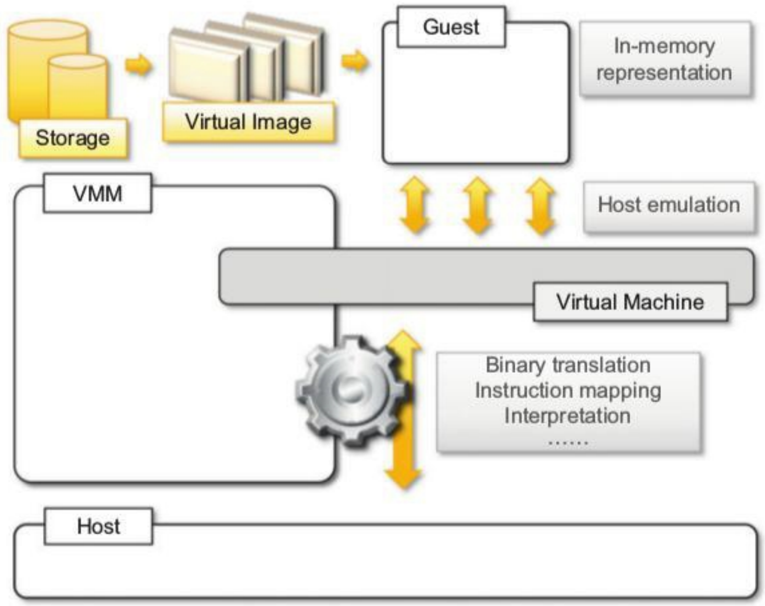


FIGURE 3.5

Security rings and privilege modes.

2. 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

	(see Figure 3.6). 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 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.  **FIGURE 3.6** A hardware virtualization reference model.			
3	(a) What are the major distributed computing technologies that led to cloud computing? 1. Distributed Systems a. Mainframes b. Clusters and c. Grid Computing	2 8	CO1	L2

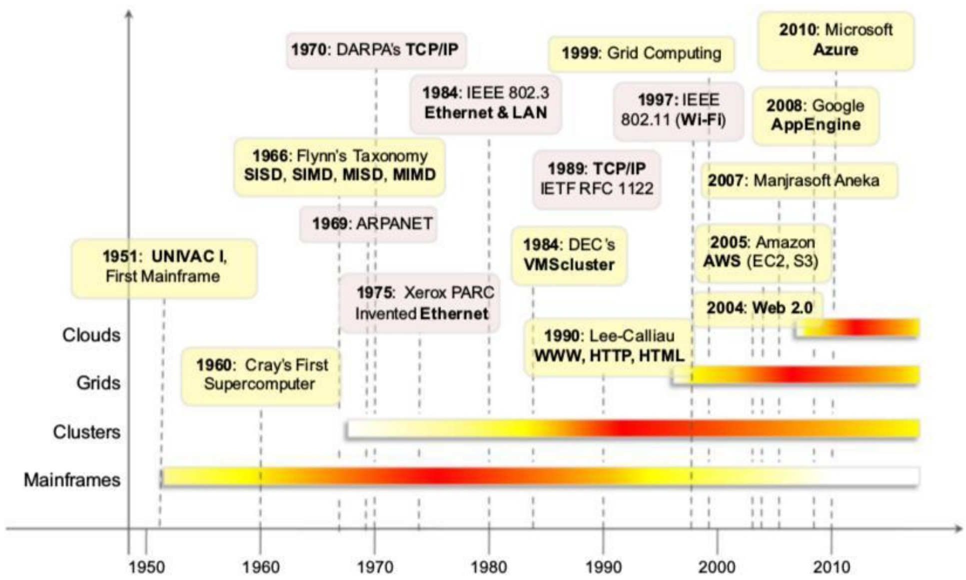


FIGURE 1.6

The evolution of distributed computing technologies, 1950s–2010s.

(b) Briefly explain each of these technologies with suitable diagrams.

Clouds are essentially large distributed computing facilities that make available their services to third parties on demand. As a reference, we consider the characterization of a distributed system proposed by Tanenbaum et al.:
"A distributed system is a collection of independent computers that appears to its users as a single coherent system."
Three major milestones have led to cloud computing: mainframe computing, cluster computing, and grid computing.

1. Mainframes: These were the first examples of large computational facilities leveraging multiple processing units. Mainframes were powerful, highly reliable computers specialized.
for large data movement and massive input/output (I/O) operations. They were mostly used by large organizations for bulk data processing tasks such as online transactions, enterprise resource planning, and other operations involving the processing of significant amounts of data.

2. Clusters: Cluster computing started as a low-cost alternative to the use of mainframes and supercomputers. The technology advancement that created faster and more powerful mainframes and supercomputers eventually generated an increased availability of cheap commodity machines as a side effect. These machines could then be connected by a high-bandwidth network and controlled by specific software tools that manage them as a single system. Starting in the 1980s.
Cluster technology contributed considerably to the evolution of tools and frameworks for distributed computing, including Condor, Parallel Virtual Machine (PVM), and Message Passing Interface (MPI).

3. Grid Computing: Grid computing appeared in the early 1990s as an evolution of cluster computing. In an analogy to the power grid, grid computing proposed a new approach to access large computational power, huge storage facilities, and a variety of services.

	A computing grid was a dynamic aggregation of heterogeneous computing nodes, and its scale was nationwide or even worldwide. Several developments made possible the diffusion of computing grids: a. clusters became quite common resources; b. they were often underutilized; c. new problems were requiring computational power that went beyond the capability of single clusters and d. the improvements in networking and the diffusion of the Internet made possible long-distance, high-bandwidth connectivity.			
4	(a) What is service orientation? Service orientation is the core reference model for cloud computing systems. This approach adopts the concept of services as the main building blocks of application and system development. Service-oriented computing (SOC) supports the development of rapid, low-cost, flexible, interoperable, and evolvable applications and systems. A service is an abstraction representing a self-describing and platform-agnostic component that can perform any function—anything from a simple function to a complex business process. (b) Describe the important characteristics of service orientation. A service is supposed to be loosely coupled, reusable, programming language independent, and location transparent. Loose coupling allows services to serve different scenarios more easily and makes them reusable. Independence from a specific platform increases services accessibility. Thus, a wider range of clients, which can look up services in global registries and consume them in a location-transparent manner, can be served. Service-oriented computing introduces and diffuses two important concepts, which are also fundamental to cloud computing: quality of service (QoS) and Software-as-a-Service (SaaS). •Quality of service (QoS) identifies a set of functional and nonfunctional attributes that can be used to evaluate the behavior of a service from different perspectives. These could be performance metrics such as response time, or security attributes, transactional integrity, reliability, scalability, and availability. •The concept of Software-as-a-Service introduces a new delivery model for applications. The term has been inherited from the world of application service providers (ASPs), which deliver software services-based solutions across the wide area network from a central datacenter and make them available on a subscription or rental basis.	4 6	CO1	L2
5	(a) What is virtualization and what are its benefits. 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. 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	4 6	CO1	L2

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.

(b) Discuss the reference model of full virtualization with a neat diagram.

3.6.2.1 Full virtualization and binary translation

VMware is well known for the capability to virtualize x86 architectures, which runs unmodified on top of their hypervisors. With the new generation of hardware architectures and the introduction of *hardware-assisted virtualization* (Intel VT-x and AMD V) in 2006, full virtualization is made possible with hardware support, but before that date, the use of *dynamic binary translation* was the only solution that allowed running x86 guest operating systems unmodified in a virtualized environment.

As discussed before, 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. This causes a different behavior when such 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. Generally, a trap is generated and the way it is managed differentiates the solutions in which virtualization is implemented for x86 hardware. 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. Moreover, to improve performance, the equivalent set of instruction is cached so that translation is no longer necessary for further occurrences of the same instructions. Figure 3.12 gives an idea of the process.

This approach has both advantages and disadvantages. The major advantage is that guests can run unmodified in a virtualized environment, which is a crucial feature for operating systems for which source code is not available. This is the case, for example, of operating systems in the Windows family. Binary translation is a more portable solution for full virtualization. On the other hand, translating instructions at runtime introduces an additional overhead that is not present in other approaches (paravirtualization or hardware-assisted virtualization). Even though such disadvantage exists, binary translation is applied to only a subset of the instruction set, whereas the others are managed through direct execution on the underlying hardware. This somehow reduces the impact on performance of binary translation.

CPU virtualization is only a component of a fully virtualized hardware environment. VMware achieves full virtualization by providing virtual representation of memory and I/O devices. Memory virtualization constitutes another challenge of virtualized environments and can deeply impact performance without the appropriate hardware support. The main reason is the presence of a *memory management unit (MMU)*, which needs to be emulated as part of the virtual hardware. Especially in the case of *hosted hypervisors* (Type II), where the virtual MMU and the host-OS MMU are traversed sequentially before getting to the physical memory page, the impact on performance can be significant. To avoid nested translation, the *translation look-aside buffer (TLB)* in the virtual MMU directly maps physical pages, and the performance slowdown only occurs in case of a TLB miss.

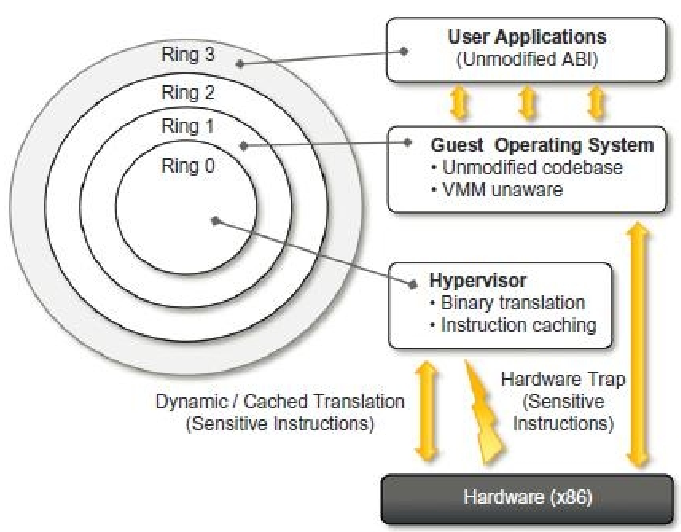


FIGURE 3.12

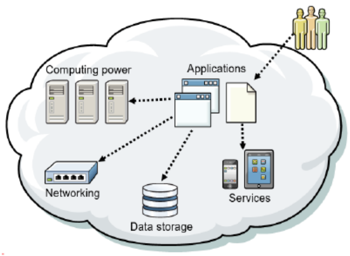
A full virtualization reference model.

<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>
A company is designing to move the in-premises hosted application to a Cloud environment. Briefly explain the components that needs to be virtualized with a suitable diagram.

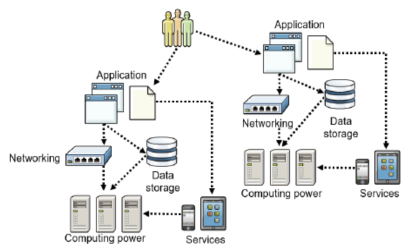
Components that need to be virtualized should be explained with suitable diagrams.

Typical components are: 1) SaaS, 2) PaaS and 3) HaaS.

- **Elastic resources**
- **Metered services**
- **Self-service**



Cloud computing model



Traditional on-premises computing model
</td><td>10</td><td>CO1</td><td>L3</td></tr>
</table>