# CBCS SCHEME

USN | | | | | | | | | | |

**22MCA332**

## Third Semester MCA Degree Examination, Dec.2024/Jan.2025

## Cloud Computing

Time: 3 hrs. Max. Marks: 100

*Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.*
*2. M : Marks , L: Bloom's level , C: Course outcomes.*

| | | Module – 1 | M | L | C |
|---|---|---|---|---|---|
| Q.1 | a. | What is the innovative aspect of cloud computing? | 10 | L2 | CO1 |
| | b. | Write short notes on:<br>(i) Web 2.0 Applications<br>(ii) Key features of service orientation | 10 | L2 | CO1 |
| | | OR | | | |
| Q.2 | a. | Write about the primary distributed computing technologies in cloud computing. | 10 | L2 | CO1 |
| | b. | What is utility computing? Explain in detail. | 05 | L1 | CO1 |
| | c. | Explain about the virtualization and its key aspect. | 05 | L2 | CO1 |
| | | Module – 2 | | | |
| Q.3 | a. | What is parallel and distributed computing and distinction between them? | 10 | L2 | CO2 |
| | b. | Write detail notes on:<br>(i) The major categories of parallel computing system<br>(ii) What is service-oriented computing? | 10 | L2 | CO2 |
| | | OR | | | |
| Q.4 | a. | Explain about the parallel processing and its appealing computing alternatives. | 10 | L2 | CO2 |
| | b. | Discuss bout the technologies that enable service computing. | 10 | L2 | CO2 |
| | | Module – 3 | | | |
| Q.5 | a. | What are the characteristics of virtual environments? | 10 | L2 | CO2 |
| | b. | Discuss about the classification of taxonomy of virtualization at various levels. | 10 | L2 | CO2 |
| | | OR | | | |
| Q.6 | a. | Discuss about the Machine Reference Model for execution in virtualization. | 05 | L2 | CO2 |
| | b. | What is Xen? Discuss the elements of virtualization. | 05 | L2 | CO2 |
| | c. | Discuss Hyper-V's architecture. Illustrate its application in cloud computing. | 10 | L2 | CO2 |
| | | Module – 4 | | | |
| Q.7 | a. | What is "XaaS" abbreviation stand for? Explain in detail. | 10 | L2 | CO3 |
| | b. | Explain briefly about cloud reference model and essential components. | 10 | L2 | CO3 |
| | | OR | | | |
| Q.8 | a. | Explain about the different forms of cloud and its classification. | 10 | L2 | CO3 |
| | b. | Discuss about the challenges of cloud computing. List out the application development concept. | 10 | L2 | CO3 |
| | | Module – 5 | | | |
| Q.9 | a. | What is AWS? What are types of services does it provide? | 10 | L3 | CO4 |
| | b. | Discuss the storage services provided by windows AZURE in detail. | 10 | L3 | CO4 |
| | | OR | | | |
| Q.10 | a. | Describe how cloud computing technologies can be used to facilitate remote ECG monitoring. | 10 | L3 | CO4 |
| | b. | What are the most significant benefits of cloud technologies for social networking applications? | 10 | L3 | CO4 |

* * * * *

**Third Semester MCA Degree Examination, Dec2024/ Jan 2025**

**Cloud Computing (22MCA332)**

**VTU QP Solution**

**Q1 a)What is the innovative aspect of cloud computing?**

Cloud computing allows anyone with a credit card to provision virtual hardware, runtime environments, and services. These are used for as long as needed, with no up-front commitments required. The entire stack of a computing system is transformed into a collection of utilities, which can be provisioned and composed together to deploy systems in hours rather than days and with virtually no maintenance costs. Previously, the lack of effective standardization efforts made it difficult to move hosted services from one vendor to another. The long-term vision of cloud computing is that IT services are traded as utilities in an open market, without technological and legal barriers. In this cloud marketplace, cloud service providers and consumers, trading cloud services as utilities, play a central role.


Many of the technological elements contributing to this vision already exist. Different stakeholders leverage clouds for a variety of services.

The need for ubiquitous storage and compute power on demand is the most common reason to consider cloud computing. A scalable runtime for applications is an attractive option for application and system developers that do not have infrastructure or cannot afford any further expansion of existing infrastructure. The capability for Webbased access to documents and their processing using sophisticated applications is one of the appealing factors for end users. Vision of cloud computing is that in the near future it will be possible to find the solution that matches our needs by simply entering our request in a global digital market that trades cloud computing services. The existence of such a market will enable the automation of the discovery process and its integration into existing software systems, thus allowing users to transparently leverage cloud resources in their applications and systems. The existence of a global platform for trading cloud services will also help service providers become more visible and therefore potentially increase their revenue. A global cloud market also reduces the barriers between service consumers and providers.

**Q1 b) Write Short notes on : i) Web 2.0 ii) key features of service orientation**

i) Web 2.0

The Web is the primary interface through which cloud computing delivers its services. At present, the Web encompasses a set of technologies and services that facilitate interactive information sharing, collaboration, user-centered design, and application composition. This evolution has transformed the Web into a rich platform for application development and is known as Web 2.0. This term captures a new way in which developers architect applications and deliver services through the Internet and provides new experience for users of these applications and services. Web 2.0 brings interactivity and flexibility into Web pages, providing enhanced user experience by gaining Web based access to all the functions that are normally found in desktop applications. These capabilities are obtained by integrating a collection of standards and technologies such as XML, Asynchronous JavaScript and XML (AJAX), Web Services, and others. These technologies allow us to build applications leveraging the contribution of users, who now become providers of content. Web 2.0 applications are extremely dynamic: they improve continuously, and new updates and features are integrated at a constant rate by following the usage trend of the community. There is no need to deploy new software releases on the installed base at the client side.

Web 2.0 applications aim to leverage the “long tail” of Internet users by making themselves available to everyone in terms of either media accessibility or affordability. Examples of Web 2.0 applications are Google Documents, Google Maps, Flickr, Facebook, Twitter, YouTube, delicious, Blogger, and Wikipedia. In particular, social networking Websites take the biggest advantage of Web 2.0. The level of interaction in Websites such as Facebook or Flicker would not have been possible without the support of AJAX, Really Simple Syndication

(RSS), and other tools that make the user experience incredibly interactive. This idea of the Web as a transport that enables and enhances interaction was introduced in 1999 by Darcy DiNucci and started to become fully realized in 2004. Today it is a mature platform for supporting the needs of cloud computing, which strongly leverages Web 2.0. Applications and frameworks for delivering rich Internet applications (RIAs) are fundamental for making cloud services accessible to the wider public.

ii) Service oriented computing

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. 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.

**Q2a) Write about the primary distributed computing technologies in cloud computing**

Three major milestones have led to cloud computing: mainframe computing cluster computing, and grid computing.

• Mainframe computing

• Cluster computing and

• Grid computing.

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.

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).

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.

**Q2b) What is utility computing? Explain in detail**

Utility computing is a vision of computing that defines a service-provisioning model for compute services in which resources such as storage, compute power, applications, and infrastructure are packaged and offered on a pay-per-use basis. The idea of providing computing as a utility like natural gas, water, power, and telephone connection has a long history but has become a reality today with the advent of cloud computing. The American scientist John McCarthy, who, in a speech for the Massachusetts Institute of Technology (MIT) centennial in 1961, observed:

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility, just as the telephone system is a public utility . . . The computer utility could become the basis of a new and important industry."

The first traces of this service-provisioning model can be found in the mainframe era. IBM and other mainframe providers offered mainframe power to organizations such as banks and government agencies throughout their datacenters. From an application and system development perspective, service-oriented computing and service- oriented architectures (SOAs) introduced the idea of leveraging external services for performing a specific task within a software system.

**Q2c) Explain about the virtualization and its key aspect**

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.

**Q3 a) What is parallel and distributed computing and distinction between them?**

The term parallel implies a tightly coupled system, whereas distributed refers to a wider class of system, including those that are tightly coupled. The architecture of a parallel computing system is often characterized by the homogeneity of components: each processor is of the same type and it has the same capability as the others.

The shared memory has a single address space, which is accessible to all the processors. Parallel programs are then broken down into several units of execution that can be allocated to different processors and can communicate with each other by means of the shared memory.

The term Distributed refers to a wider class of system, including those that are tightly coupled. The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor. Therefore, distributed computing includes a wider range of systems and applications than parallel computing.

| Parallel Computing | Distributed Computing |
|---|---|
| Many operations are performed simultaneously. | System components are located at different locations. |
| Single computer is required. | Uses multiple computers. |
| Multiple processors perform multiple operations. | Multiple computers perform multiple operations. |
| It may have shared or distributed memory. | It have only distributed memory. |
| Processors communicate with each other through bus. | Computer communicate with each other through message passing. |
| Improves the system performance. | Improves system scalability, fault tolerance and resource sharing capabilities. |

**Q3 b) Write detail notes on:**

**i) The major categories of parallel computing system**

- Single-instruction, single-data (SISD) systems
- Single-instruction, multiple-data (SIMD) systems
- Multiple-instruction, single-data (MISD) systems
- Multiple-instruction, multiple-data (MIMD) systems

# SINGLE-INSTRUCTION, SINGLE-DATA (SISD) SYSTEMS

An SISD computing system is a uniprocessor machine capable of executing a single instruction, which operates on a single data stream

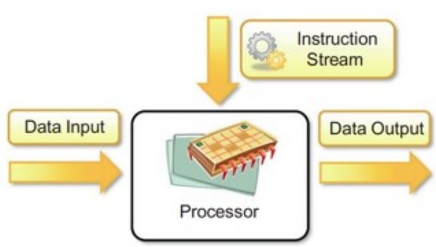



**FIGURE 2.2**

Single-instruction, single-data (SISD) architecture.

# SINGLE-INSTRUCTION, MULTIPLE-DATA (SIMD) SYSTEMS

An SIMD computing system is a multiprocessor machine capable of executing the same instruction on all the CPUs but operating on different data streams

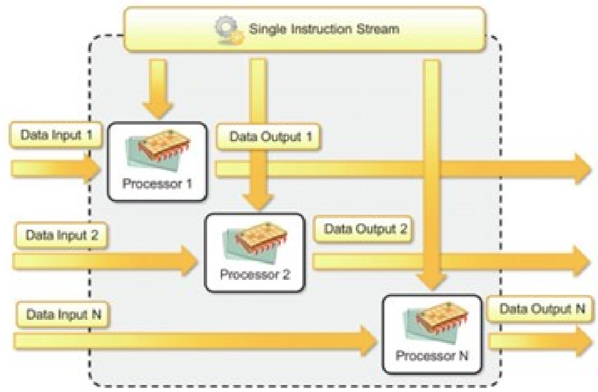



**FIGURE 2.3**

Single-instruction, multiple-data (SIMD) architecture.

# MULTIPLE-INSTRUCTION, SINGLE-DATA (MISD) SYSTEMS

An MISD computing system is a multiprocessor machine capable of executing different instructions on different PEs but all of them operating on the same data set

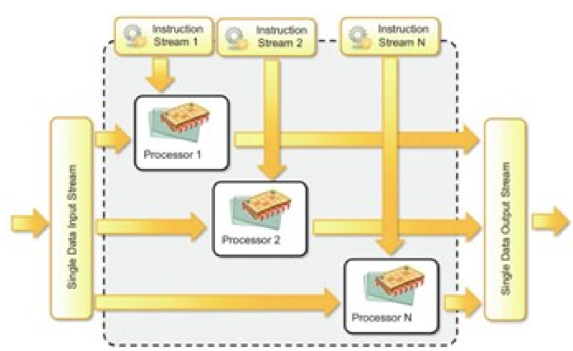



**FIGURE 2.4**

Multiple-instruction, single-data (MISD) architecture.

# MULTIPLE-INSTRUCTION, MULTIPLE-DATA (MIMD) SYSTEMS

An MIMD computing system is a multiprocessor machine capable of executing multiple instructions on multiple data sets

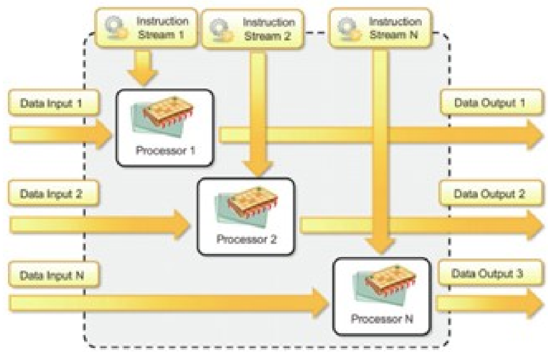



FIGURE 2.5
Multiple-instructions, multiple-data (MIMD) architecture.


# MULTIPLE-INSTRUCTION, MULTIPLE-DATA (MIMD) SYSTEMS


MIMD machines are broadly categorized into shared-memory MIMD and distributed-memory MIMD based on the way PEs are coupled to the main memory.

Shared memory

MIMD machines In the shared memory MIMD model, all the PEs are connected to a single global memory and they all have access to it

Systems based on this model are also called tightly coupled multiprocessor systems. The communication between PEs in this model takes place through the shared memory; modification of the data stored in the global memory by one PE is visible to all other PEs.

Distributed memory

MIMD machines In the distributed memory MIMD model, all PEs have a local memory. Systems based on this model are also called loosely coupled multiprocessor systems. The communication between PEs in this model takes place through the interconnection network (the interprocess communication channel, or IPC).

The network connecting PEs can be configured to tree, mesh, cube, and so on. Each PE operates asynchronously, and if communication/synchronization among tasks is necessary, they can do so by exchanging messages between them.

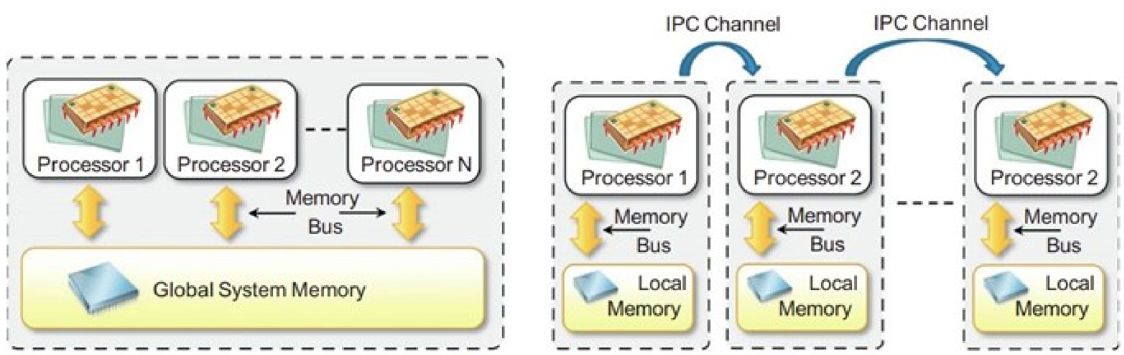



**FIGURE 2.6**

Shared (left) and distributed (right) memory MIMD architecture.

The shared-memory MIMD architecture is easier to program but is less tolerant to failures and harder to extend with respect to the distributed memory MIMD model.

Failures in a shared-memory MIMD affect the entire system, whereas this is not the case of the distributed model, in which each of the PEs can be easily isolated. Moreover, shared memory MIMD architectures are less likely to scale because the addition of more PEs leads to memory contention.

This is a situation that does not happen in the case of distributed memory, in which each PE has its own memory. As a result, distributed memory MIMD architectures are most popular today.

**ii) What is service-oriented computing?**

Service-oriented computing organizes distributed systems in terms of services, which represent the major abstraction for building systems. Service orientation expresses applications and software systems as aggregations of services that are coordinated within a service-oriented architecture (SOA).

Even though there is no designed technology for the development of service-oriented software systems, Web services are the de facto approach for developing SOA. Web services, the fundamental component enabling cloud computing systems, leverage the Internet as the main interaction channel between users and the system

**What is a service?**

A service encapsulates a software component that provides a set of coherent and related functionalities that can be reused and integrated into bigger and more complex applications.

Don Box [107] identifies four major characteristics that identify a service:

- Boundaries are explicit. A service-oriented application is generally composed of services that are spread across different domains, trust authorities, and execution environments. Generally, crossing such boundaries is costly; therefore, service invocation is explicit by design and often leverages message passing. With respect to distributed object programming, whereby remote method invocation is transparent, in a service-oriented computing environment the interaction with a service is explicit and the interface of a service is kept minimal to foster its reuse and simplify the interaction.
- Services are autonomous. Services are components that exist to offer functionality and are aggregated and coordinated to build more complex system. They are not designed to be part of a specific system, but they can be integrated in several software systems, even at the same time. The notion of autonomy also affects the way services handle failures. Services operate in an unknown environment and interact with third-party applications. Therefore, minimal assumptions can be made concerning such environments: applications may fail without notice, messages can be malformed, and clients can be unauthorized. Service-oriented design addresses these issues by using transactions, durable queues, redundant deployment and failover, and administratively managed trust relationships among different domains.
- Services share schema and contracts, not class or interface definitions. Services are not expressed in terms of classes or interfaces, as happens in object-oriented systems, but they define themselves in terms of schemas and contracts. A service advertises a contract describing the structure of messages it can send and/or receive and additional constraint—if any—on their ordering. Because they are not expressed in terms of types and classes, services are more easily consumable in wider and heterogeneous environments. At the same time, a service orientation requires that contracts and schema remain stable over time, since it would be possible to propagate changes to all its possible clients. To address this issue, contracts and schema are defined in a way that allows services to evolve without breaking already deployed code. Technologies such as XML and SOAP provide the appropriate tools to support such features rather than class definition or an interface declaration.
- Services compatibility is determined based on policy. Service orientation separates structural compatibility from semantic compatibility. Structural compatibility is based on contracts and schema and can be validated or enforced by machine-based techniques. Semantic compatibility is expressed in the form of policies that

## Q4a) Explain about the parallel processing and its appealing computing alternatives

Processing of multiple tasks simultaneously on multiple processors is called *parallel processing*. The parallel program consists of multiple active processes (tasks) simultaneously solving a given problem. A given task is divided into multiple subtasks using a divide-and-conquer technique, and each subtask is processed on a different central processing unit (CPU). Programming on a multiprocessor system using the divide-and-conquer technique is called *parallel programming*.

Many applications today require more computing power than a traditional sequential computer can offer. Parallel processing provides a cost-effective solution to this problem by increasing the number of CPUs in a computer and by adding an efficient communication system between them. The workload can then be shared between different processors. This setup results in higher computing power and performance than a single-processor system offers.

The development of parallel processing is being influenced by many factors. The prominent among them include the following:

- Computational requirements are ever increasing in the areas of both scientific and business computing. The technical computing problems, which require high-speed computational power, are related to life sciences, aerospace, geographical information systems, mechanical design and analysis, and the like.
- Sequential architectures are reaching physical limitations as they are constrained by the speed of light and thermodynamics laws. The speed at which sequential CPUs can operate is reaching saturation point (no more vertical growth), and hence an alternative way to get high computational speed is to connect multiple CPUs (opportunity for horizontal growth).

- Hardware improvements in pipelining, superscalar, and the like are nonscalable and require sophisticated compiler technology. Developing such compiler technology is a difficult task.
- Vector processing works well for certain kinds of problems. It is suitable mostly for scientific problems (involving lots of matrix operations) and graphical processing. It is not useful for other areas, such as databases.
- The technology of parallel processing is mature and can be exploited commercially; there is already significant R&D work on development tools and environments.
- Significant development in networking technology is paving the way for heterogeneous computing.

**Q4b) Discuss about the technologies that enable service computing**

Different books and different organizations provide different definitions to Web Services. Some of them are listed here.

- A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response. As all communication is in XML, web services are not tied to any one operating system or programming language—Java can talk with Perl; Windows applications can talk with Unix applications.
- Web services are self-contained, modular, distributed, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains. These applications can be local, distributed, or web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML.
- Web services are XML-based information exchange systems that use the Internet for direct application-to-application interaction. These systems can include programs, objects, messages, or documents.
- A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

To summarize, a complete web service is, therefore, any service that −

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

## Components of Web Services

The basic web services platform is XML + HTTP. All the standard web services work using the following components −

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

All these components have been discussed in the Web Services Architecturechapter.

## Web Service Work

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of −

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

You can build a Java-based web service on Solaris that is accessible from your Visual Basic program that runs on Windows.

You can also use C# to build new web services on Windows that can be invoked from your web application that is based on JavaServer Pages (JSP) and runs on Linux.

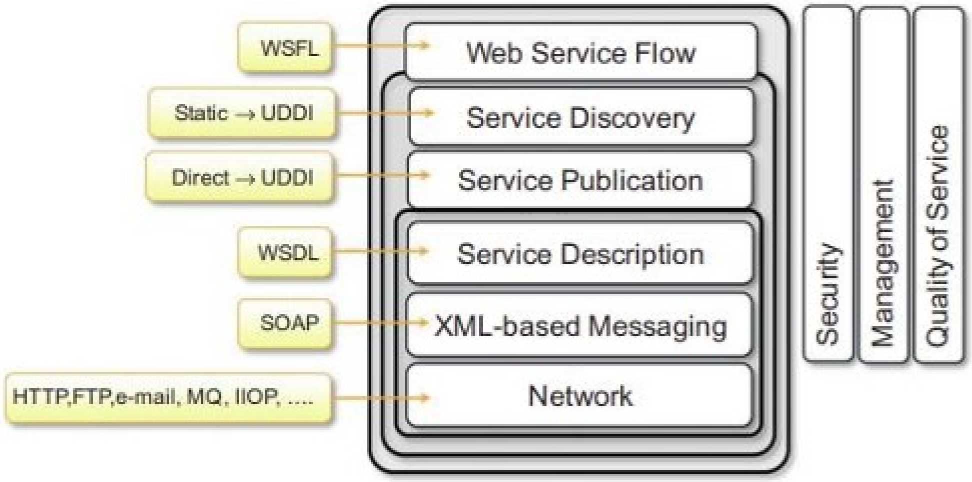
## Example

Consider a simple account-management and order processing system. The accounting personnel use a client application built with Visual Basic or JSP to create new accounts and enter new customer orders.

The processing logic for this system is written in Java and resides on a Solaris machine, which also interacts with a database to store information.

The steps to perform this operation are as follows −

- The client program bundles the account registration information into a SOAP message.
- This SOAP message is sent to the web service as the body of an HTTP POST request.
- The web service unpacks the SOAP request and converts it into a command that the application can understand.
- The application processes the information as required and responds with a new unique account number for that customer.
- Next, the web service packages the response into another SOAP message, which it sends back to the client program in response to its HTTP request.
- The client program unpacks the SOAP message to obtain the results of the account registration process.




## Q5 a) What are the characteristics of virtual environments?

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.

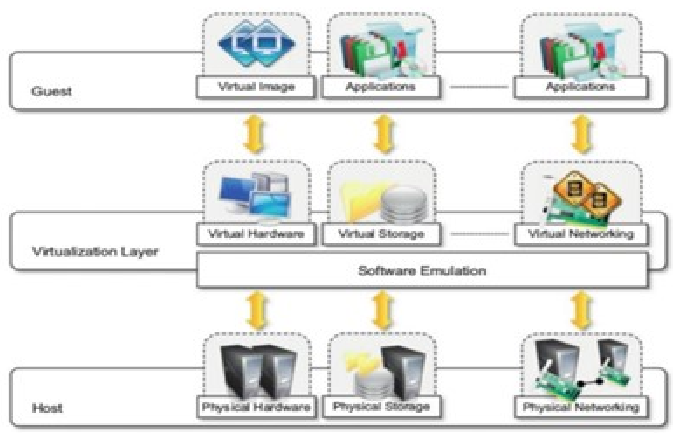



**FIGURE 3.1**

The virtualization reference model.

The virtualization layer is responsible for recreating the same or a different environment where the guest will operate (see Figure 3.1).

**The characteristics of virtualized solutions are:**

1 Increased security

2 Managed executions

3 Portability

**1. Increased security**

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. This level of indirection allows the virtual machine manager to control and filter the activity of the guest, thus preventing some harmful operations from being performed. For example, applets downloaded from the Internet run in a sandboxed 3 version of the Java Virtual Machine (JVM), which provides them with limited access to the hosting operating system resources. Both the JVM and the .NET runtime provide extensive security policies for customizing the execution environment of applications.

**2 Managed executions**

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

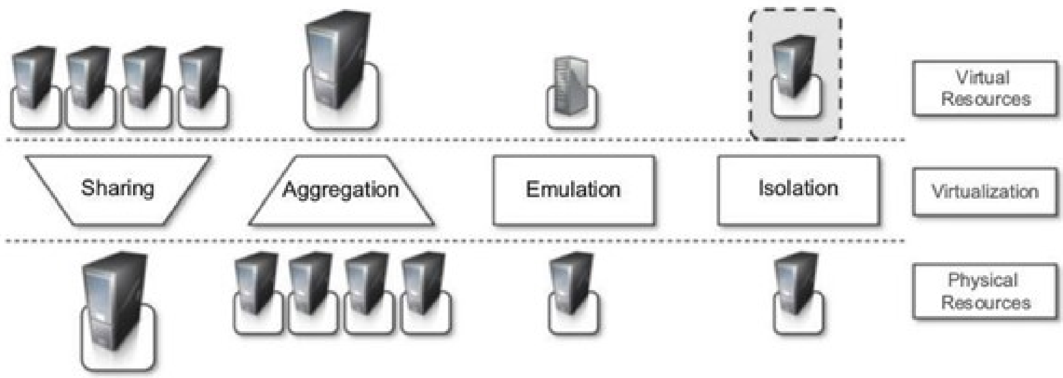



**FIGURE 3.2**

Functions enabled by managed execution.

are the most relevant features (see Figure 3.2).

**(a) Sharing -** Virtualization allows the creation of a separate computing environments within the same host. In this way it is possible to fully exploit the capabilities of a powerful guest, which would otherwise be underutilized.

**(b) 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.

**(c) 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.

**(d) 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.

**3 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.

In the case of programming-level virtualization, as implemented by the JVM or the .NET runtime, the binary code representing application components (jars or assemblies) can be run without any recompilation on any implementation of the corresponding virtual machine. This makes the application development cycle more flexible and application deployment very straightforward: One version of the application, in most cases, is able to run on different platforms with no changes.

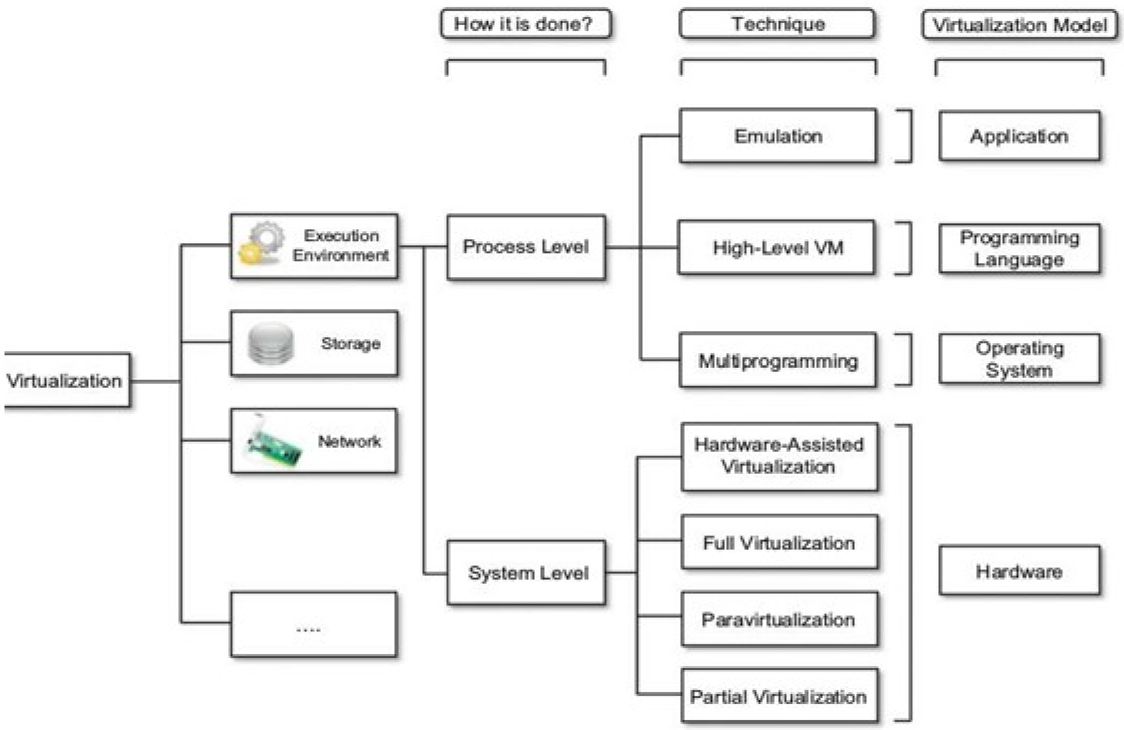
**Q5 b) Discuss about the classification of taxonomy of virtualization at various levels**

Virtualization covers a wide range of emulation (computing action of a different computer, software system) techniques that are applied to different areas of computing.

A classification of these techniques helps us better understand their characteristics and use (see Figure 3.3).

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. Therefore, it deserves major investigation and a further categorization




URE 3.3

**Q6 a) Discuss about the Machine Reference model for execution in 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. Modern computing systems can be expressed in terms of the reference model described in Figure 3.4.

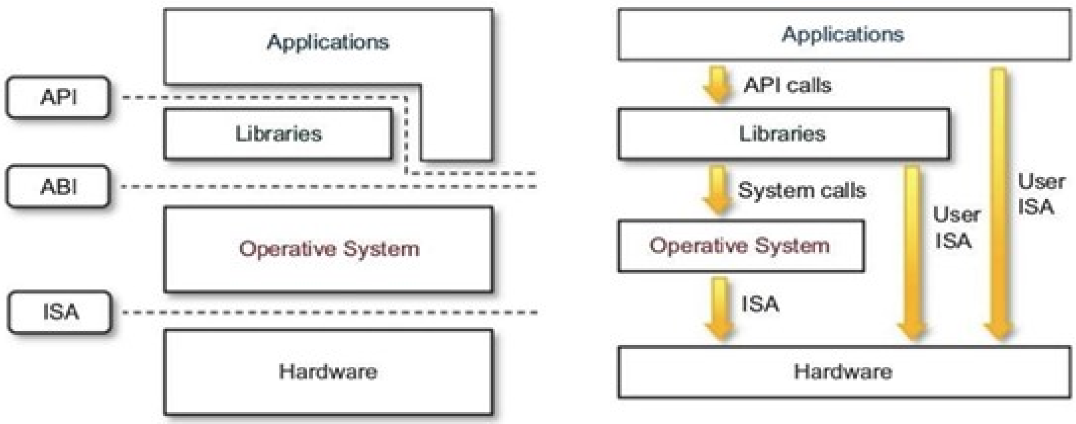



**FIGURE 3.4**

A machine reference model.

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.

## Q6 b) What is Xen? Discuss the elements of virtualization

**Xen** is an open-source **hypervisor** based on **paravirtualization**. It is the most popular application of paravirtualization. Xen has been extended to compatible with full virtualization using hardware-assisted virtualization. It enables high performance to execute **guest operating system**.

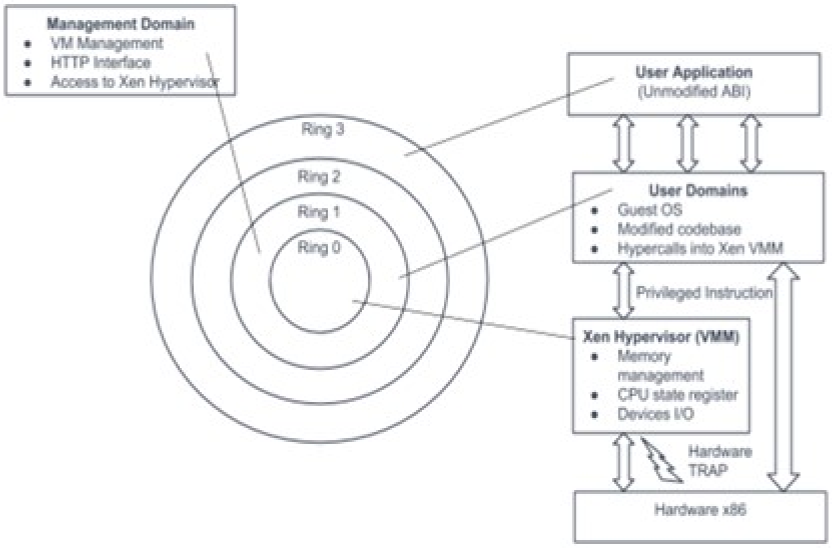



**Figure** – Xen Architecture and Guest OS Management

Above figure describes the Xen Architecture and its mapping onto a **classic x86 privilege model**. A Xen based system is handled by Xen hypervisor, which is executed in the most privileged mode and maintains the access of guest operating system to the basic hardware. Guest operating systems are run between domains, which represents virtual machine instances.

In addition, **particular control software**, which has **privileged access** to the host and handles all other guest OS, runs in a special domain called Domain 0. This the only one loaded once the virtual machine manager has fully booted and hosts an HTTP server that delivers requests for virtual machine creation, configuration, and termination.

This component establishes the primary version of a shared virtual machine manager (VMM), which is a necessary part of Cloud computing system delivering Infrastructure-as-a-Service (IaaS) solution

Almost all the frequently used Operating system uses only two levels i.e., Ring 0 for the Kernel code and Ring 3 for user application and non-privilege OS program. This provides a chance to the Xen to implement paravirtualization. This enables Xen to control unchanged the

Application Binary Interface (ABI) thus allowing a simple shift to Xen- virtualized solutions, from an application perspective.

Paravirtualization demands the OS codebase be changed, and hence all operating systems cannot be referred to as guest OS in a Xen-based environment. This condition holds where hardware-assisted virtualization cannot be free, which enables to run the hypervisor in Ring 1 and the guest OS in Ring 0. Hence, Xen shows some limitations in terms of legacy hardware and in terms of legacy OS.

## Q6 c) Discuss Hyper-V's architecture. Illustrate its application in cloud computing

Hyper – V is an infrastructure virtualization solution developed by Microsoft for server virtualization. As the name recalls, it uses a hypervisor-based approach for hardware virtualization, which leverages several techniques to support a variety 2008 R2 that installs the hypervisor as a role within the server.

### 1. Architecture

Hyper – V supports multiple and concurrent execution of the guest operating system by means of partitions. A partition is a completely isolated environment in which an operating system is installed and run.

Fig. 3.17 provides an overview of the architecture of Hyper – V. Hyper – V takes control of the hardware, and the host operating system becomes a virtual machine instance with special privileges, called parent partition. The parent partition (also called root partition) is the only one that has direct access to the hardware, it runs the virtualized stack, host all the drivers required to configure guest operating systems and creates child partitions through the hypervisor. Child partitions are used to host guest operating systems and do not have access to the underlying hardware, but their interaction with it is controlled by either the parent partition or the hypervisor itself.

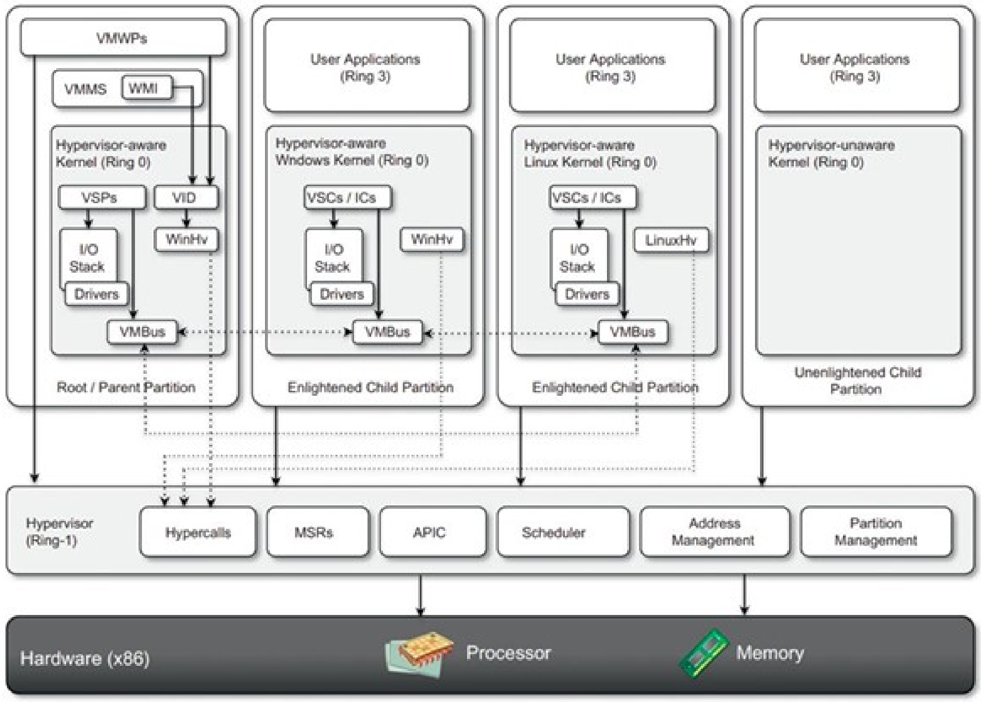



**FIGURE 3.17**

Microsoft Hyper-V architecture.

**(a) Hypervisor:** the hypervisor Is the component that directly manages the underlying

**(b)** hardware(processors and memory). It is logically defined by the following components:

- Hypercalls interface
- Memory service routines (MSRs)
- Advanced programming interrupt controller (APIC)
- Schedular
- Address manager
- Partition manager

**Q7 a) What is “Xaas” abbreviation stands for? Eplain in detail**

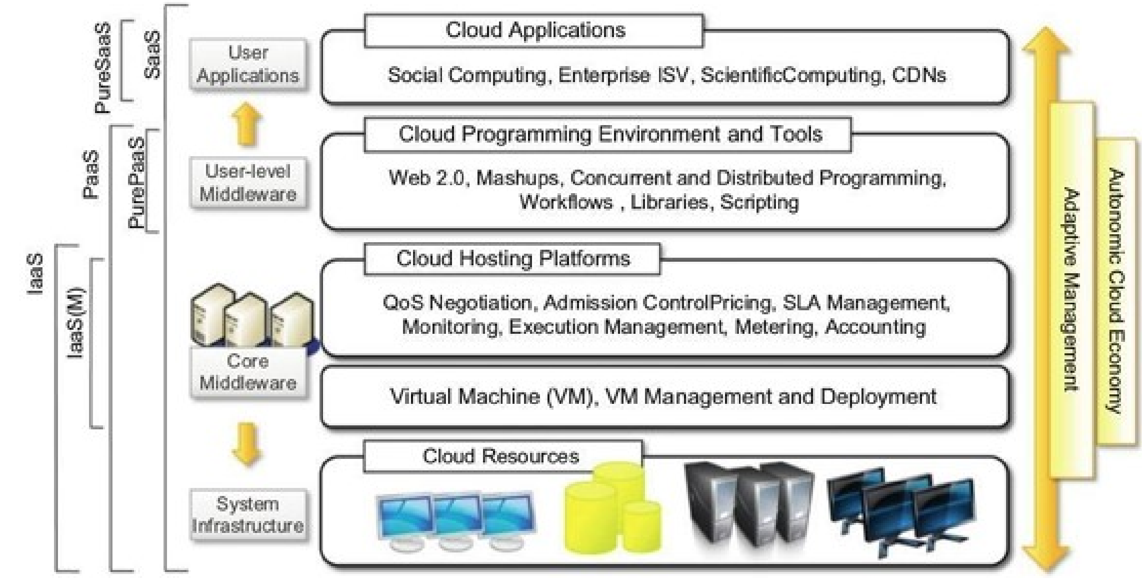



**FIGURE 4.1**

The cloud computing architecture.

The reference model described in Figure 4.1 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. 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 the case of virtual Web farms: a distributed system composed of Web servers, database servers, and load balancers on top of which prepackaged software is installed to run Web applications. This possibility has made cloud computing 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.

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)<br>Clarizen.com (project management)<br>Google Apps |
| *PaaS* | Customers are provided with a platform for developing applications hosted in the cloud. | Programming APIs and frameworks<br>Deployment systems | Google AppEngine<br>Microsoft Azure<br>Manjrasoft Aneka<br>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<br>Storage management<br>Network management | Amazon EC2 and S3<br>GoGrid<br>Nirvanix |

**Q7 b) Explain briefly about cloud reference model and essential components**

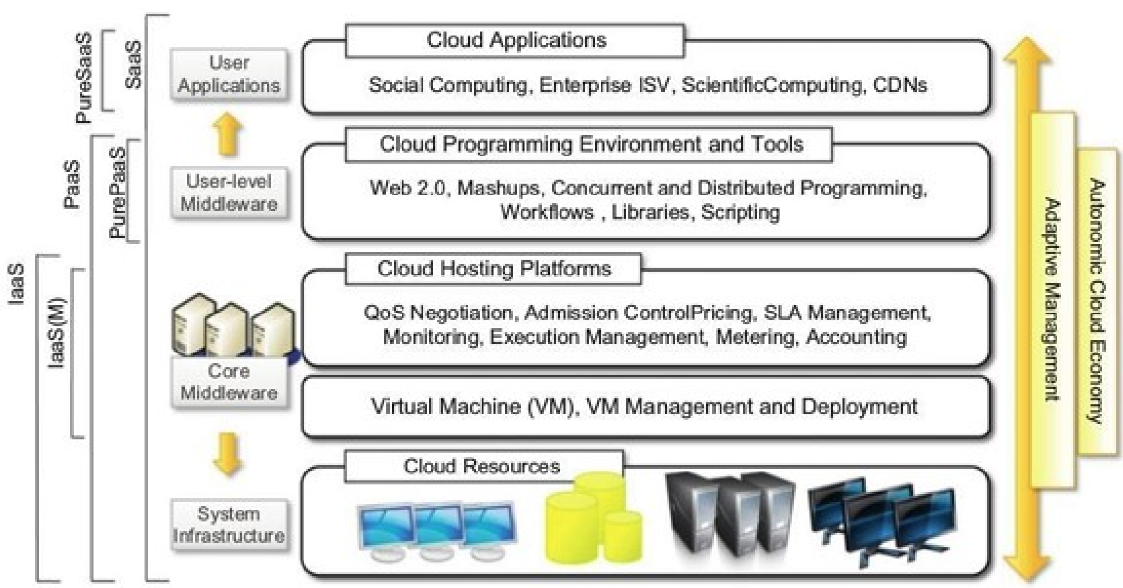



**FIGURE 4.1**

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 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 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 tofinely partition the hardware resources such as CPU and memory and to virtualize specific devices,thus meeting the requirementsof 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)).
- **User-level middleware:** 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 is **User Application level:** These are referred 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.

**Q8 a) Explain about the different forms of cloud and its classification**

A more useful classification is given according to the administrative domain of a cloud. It is then possible to differentiate four different types of cloud:

1. **Public clouds.** The cloud is open to the wider public.
2. **Private clouds.** The cloud is implemented within the private premises of an **institution** and generally made **accessible to the members of the institution or a subset** of them.
3. **Hybrid or Heterogeneous clouds.** The cloud is a combination of the two previous (Public + Private) solutions and most likely identifies a private cloud that has been augmented with resources or services hosted in a public cloud.
4. **Community clouds.** The cloud is characterized by a multi-administrative domain involving different deployment models (public, private, and hybrid), and it is specifically designed to address the needs of a specific industry.

**Q8 b) Discuss about the challenges of cloud computing. List out the application development concept.**

Cloud computing presents many challenges for industry and academia. There is a significant amount of work in academia focused on defining the challenges brought by this phenomenon.
In this section, we highlight the most important ones.

1 Cloud definition
2 Cloud interoperability and standards
3 Scalability and fault tolerance
4 Security, trust, and privacy
5 Organizational aspects

## 1 Cloud definition

There have been several attempts made to define cloud computing and to provide a classification of all the services and technologies identified as such.

NSIT characterizes cloud computing as on-demand **self-service**, **broad network access, resource-pooling, rapid elasticity, and measured service**; classifies services as **SaaS, PaaS, and IaaS**; and categorizes deployment models as **public, private, community, and hybrid clouds**.

Alternative taxonomies for cloud services. David Linthicum, founder of Blue Mountains Labs, provides a more detailed classification, which comprehends 10 different classes and better suits the vision of cloud computing within the enterprise.

These characterizations and taxonomies reflect what is meant by cloud computing at the present time but being in its infancy the phenomenon is constantly evolving, and the same will happen to the attempts to capture the real nature of cloud computing.

## 2 Cloud interoperability and standards

To fully realize this goal, introducing standards and allowing interoperability between solutions offered by different vendors are objectives of fundamental importance. Vendor lock-in constitutes one of the major strategic barriers against the seamless adoption of cloud computing at all stages. The presence of standards that are implemented and adopted in the cloud computing community could give room for interoperability and then lessen the risks resulting from vendor lock-in.

The first steps toward a standardization process have been made, and a few organizations, such as the Cloud Computing Interoperability Forum (CCIF), the Open Cloud Consortium, and the DMTF Cloud Standards Incubator, are leading the path.

Another interesting initiative is the Open Cloud Manifesto, which embodies the point of view of various stakeholders on the benefits of open standards in the field.

The Open Virtualization Format (OVF) is an attempt to provide a common format for storing the information and metadata describing a virtual machine image. Even though the OVF provides a full specification for packaging and distributing virtual machine images incompletely platform-independent fashion, it is supported by few vendors that use it to import static virtual machine images.

### 3 Scalability and fault tolerance

The ability to scale on demand constitutes one of the most attractive features of cloud computing. Clouds allow scaling beyond the limits of the existing in-house IT resources, whether they are infrastructure (compute and storage) or applications services. To implement such a capability, the cloud middleware has to be designed with the principle of scalability along different dimensions in mind— for example, **performance, size, and load.**

The cloud middleware manages a huge number of resource and users, which rely on the cloud to obtain the horsepower. In this scenario, the ability to tolerate failure becomes fundamental, sometimes even more important than providing an extremely efficient and optimized system. Hence, the challenge in this case is designing highly scalable and fault-tolerant systems that are easy to manage and at the same time provide competitive performance.

### 4 Security, trust, and privacy

Security, trust, and privacy issues are major obstacles for massive adoption of cloud computing. The traditional cryptographic technologies are used to prevent data tampering and access to sensitive information. The massive use of virtualization technologies exposes the existing system to new threats, which previously were not considered applicable.

Information can be stored within a cloud storage facility using the most advanced technology in cryptography to protect data and then be considered safe from any attempt to access it without the required permissions.

The lack of control over data and processes also poses severe problems for the trust we give to the cloud service provider and the level of privacy we want to have for our data.

### 5 Organizational aspects

More precisely, **storage, compute power, network infrastructure, and applications are delivered as metered services over the Internet**. This introduces a billing model that is new within typical enterprise IT departments, which requires a certain level of cultural and organizational process maturity.

**In particular, the following questions must be considered:**

1. What is the new role of the IT department in an enterprise that completely or significantly relies on the cloud?

2. How will the compliance department perform its activity when there is a considerable lackof control over application workflows?

3. What are the implications (political, legal, etc.) for organizations that lose control over some aspects of their services?

4. What will be the perception of the end users of such services?

From an organizational point of view, the lack of control over the management of data and processes poses not only security threats but also new problems that previously did not exist.

**Q9 a) What is AWS? What are types of services does it provide?**

Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

1) **Compute services**
    a. Amazon machine image
    b. EC2 Environment
    c. AWS CloudFormation, AWS elastic beanstalk, Amazon elastic MapReduce
2) **Storage services**
    a. S3
    b. Amazon elastic block store
    c. Amazon ElasticCache
    d. Structured storage solution
    e. Amazon Cloud Front
3) **Communication Services**
    a. Virtual networking
    b. Messaging

**Q9 b) Discuss the storage services provided by windows AZURE in detail**

Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage. Compared to local storage, these services can be accessed by multiple clients at the same time and from everywhere, thus becoming a general solution for storage.

**2.1) Blobs**

Azure allows storing large amount of data in the form of binary large objects (BLOBs) by means of the blobs service. This service is optimal to store large text or binary files. Two types of blobs are available:

• Block blobs. Block blobs are composed of blocks and are optimized for sequential access; therefore they are appropriate for media streaming. Currently, blocks are of 4 MB, and a single block blob can reach 200 GB in dimension.

• Page blobs. Page blobs are made of pages that are identified by an offset from the beginning of the blob. A page blob can be split into multiple pages or constituted of a single page. This type of blob is optimized for random access and can be used to host data different from streaming. Currently, the maximum dimension of a page blob can be 1 TB.

Blobs storage provides users with the ability to describe the data by adding metadata. It is also possible to take snapshots of a blob for backup purposes. Moreover, to optimize its distribution, blobs storage can leverage the Windows Azure CDN so that blobs are kept close to users requesting them and can be served efficiently.

**2.2) Azure drive**

Page blobs can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file. This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage. A page blob mounted as part of an NTFS tree is called an Azure Drive.

**2.3) Tables**

Tables constitute a semi-structured storage solution, allowing users to store information in the form of entities with a collection of properties. Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table. Users can insert, update, delete, and select a subset of the rows stored in the table. Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relationships among entities. For this reason, tables are more similar to spreadsheets rather than SQL tables. The service is designed to handle large amounts of data and queries returning huge result sets. This capability is supported by partial result sets and table partitions.

**2.5) Queues**

Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages. Applications enter messages into a queue, and other applications can read them in a first-in, first-out (FIFO) style. To ensure that messages get processed, when an application reads a message it is marked as invisible; hence it will not be available to other clients. Once the application has completed processing the message, it needs to explicitly delete the message from the queue. This two-phase process ensures that messages get processed before they are removed from the queue, and the client failures do not prevent messages from being processed. At the same time, this is also a reason that the queue does not enforce a strict FIFO model: Messages that are read by applications that crash during processing are made available again after a timeout, during which other messages can be read by other clients. An alternative to reading a message is peeking, which allows retrieving the message but letting it stay visible in the queue. Messages that are peeked are not considered processed. All the services described are geo-replicated three times to ensure their availability in case of major disasters. Geo-replication involves the copying of data into a different datacenter that is hundreds or thousands of miles away from the original datacenter.

## Q10 a) Describe how cloud computing technologies can be used to facilitate remote ECG monitoring

Healthcare is a domain in which computer technology has found several and diverse applications: from supporting the business functions to assisting scientists in developing solutions to cure diseases. Cloud computing technologies allow remote monitoring of a patients heartbeat data, its analysis in minimum time, and the notification of first-aid personal and doctors should this data reveal potentially dangerous conditions. This way patient at risk can be constantly monitored without going to hospital for ECG analysis. At the same time doctors and patients can be instantly be notified with cases that require their attentions An illustration of the infrastructure and model for supporting remote ECG monitoring is shown in Figure 10.1. Wearable computing devices equipped with ECG sensors constantly monitor the patient's heartbeat. Such information is transmitted to the patient's mobile device, which will eventually forward it to the cloud-hosted Web service for analysis.

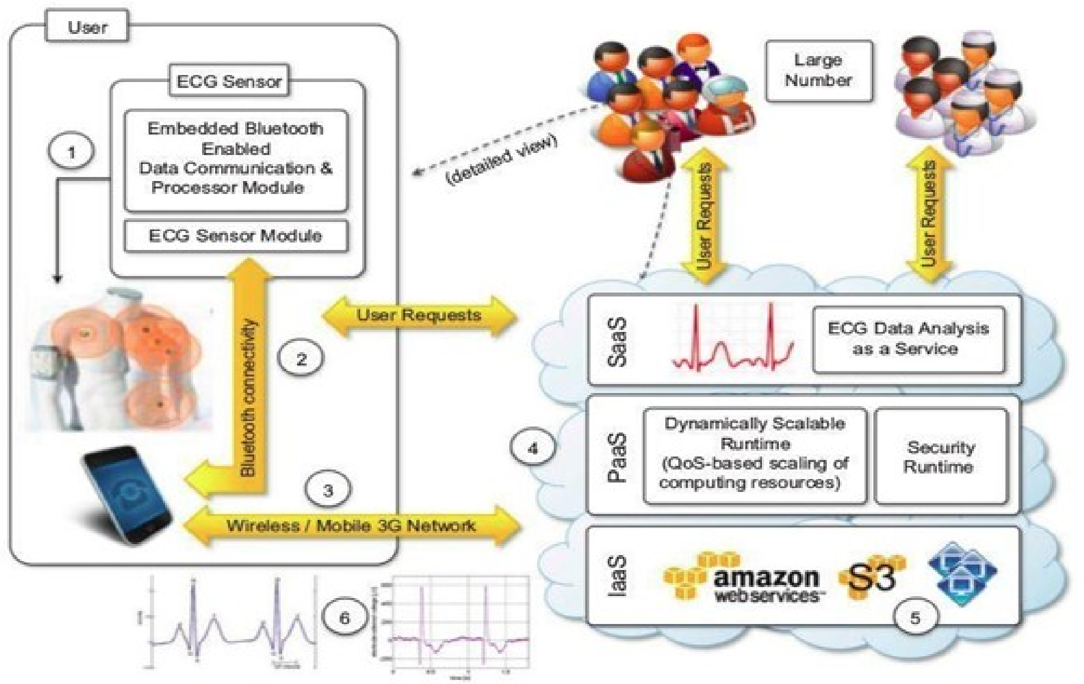



**FIGURE 10.1**

An online health monitoring system hosted in the cloud.

layers of cloud computing stack: SaaS, PaaS, and IaaS. The Web service constitute SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud platform. The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka. The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances used to execute the single tasks defined by the workflow engine for a single ECG processing job.

**Advantages**

1. The elasticity of cloud infrastructure that can grow and shrink according to the requests served. As a result, doctors and hospitals do not have to invest in large computing infrastructures designed after capacity planning, thus making more effective use of budgets.

2. Ubiquity. Cloud computing technologies are easily accessible and promise to deliver systems with minimum or no downtime. Computing systems hosted in cloud are accessible from any Internet device through simple interfaces (such as SOAP and REST-based Web services). This makes systems easily integrated with other systems maintained on hospital's premises.

3. Cost savings. Cloud services are priced on a pay-per-use basis and with volume prices for large numbers of service requests.

## Q10 b) What are the most significant benefits of cloud technologies for social networking application?

Social networking applications have grown considerably in the last few years to become the most active sites on the Web. To sustain their traffic and serve millions of users seamlessly, services such as Twitter and Facebook have leveraged cloud computing technologies.

**Facebook**

Facebook is probably the most evident and interesting environment in social networking. With more than 800 million users, it has become one of the largest Websites in the world. To sustain this incredible growth, it has been fundamental that Facebook be capable of continuously adding capacity and developing new scalable technologies and software systems while maintaining high performance to ensure a smooth user experience.

Currently, the social network is backed by two data centers that have been built and optimized to reduce costs and impact on the environment. On top of this highly efficient infrastructure, built and designed out of inexpensive hardware, a completely customized stack of opportunely modified and refined open-source technologies constitutes the back-end of the largest social network.

The reference stack serving Facebook is based on LAMP (Linux, Apache, MySQL, and PHP). This collection of technologies is accompanied by a collection of other services developed in- house. These services are developed in a variety of languages and implement specific functionalities such as search, news feeds, notifications, and others. While serving page requests, the social graph of the user is composed.

The social graph identifies a collection of interlinked information that is of relevance for a given user. Most of the user data are served by querying a distributed cluster of MySQL instances, which mostly contain key-value pairs.