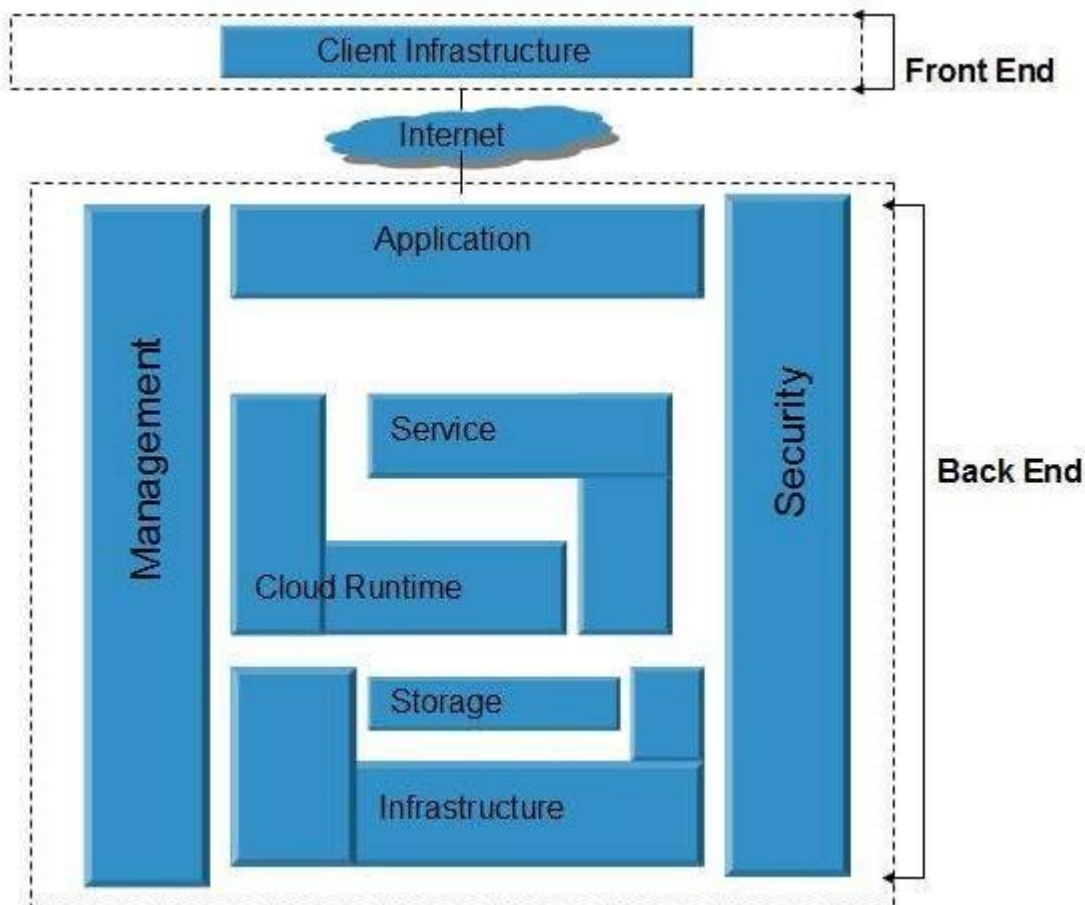


CLOUD COMPUTING ARCHITECTURE

Each of the ends is connected through a network, usually Internet. The following diagram shows the graphical view of cloud computing architecture:



Front End

The **front end** refers to the client part of cloud computing system. It consists of interfaces and applications that are required to access the cloud computing platforms, Example - Web Browser.

Back End

The **back end** refers to the cloud itself. It consists of all the resources required to provide cloud computing services. It comprises of huge data storage, virtual machines, security mechanism, services, deployment models, servers, etc.

Note

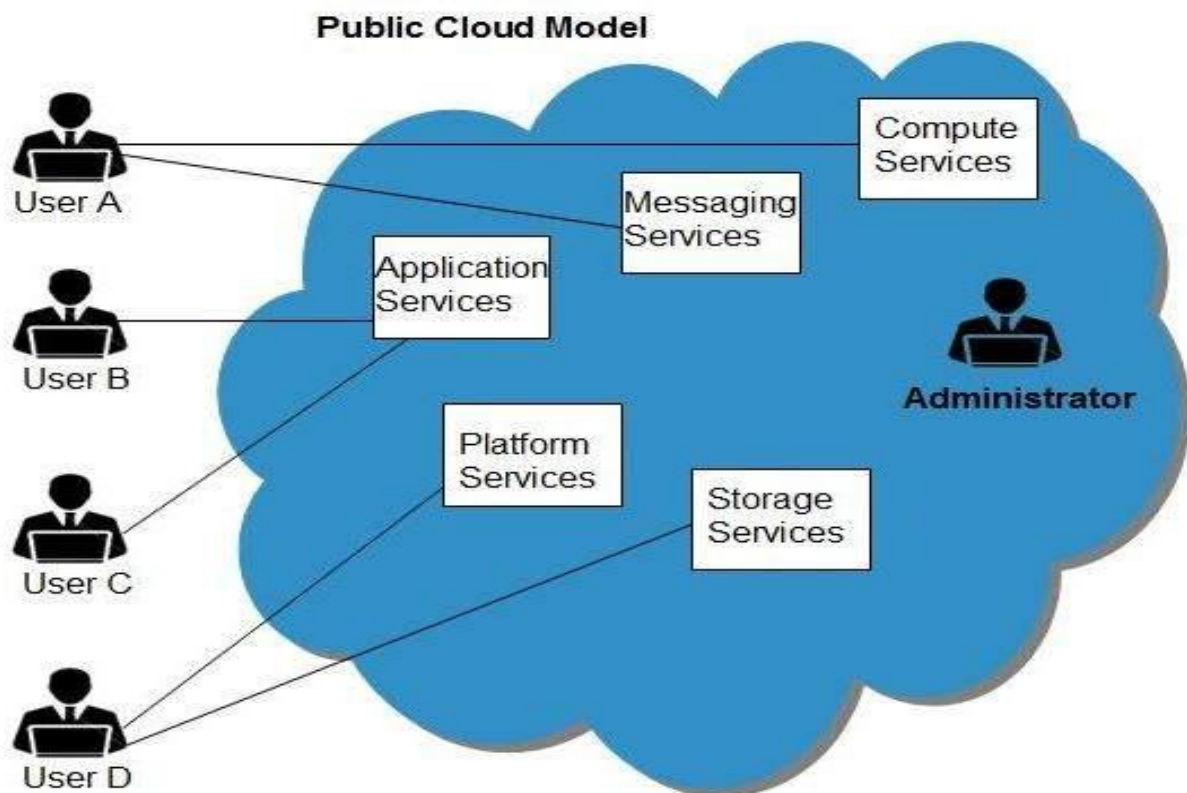
- It is the responsibility of the back end to provide built-in security mechanism, traffic control and protocols.
- The server employs certain protocols known as middleware, which help the connected devices to communicate with each other.

Deployment Models

1. Public Cloud.
2. Private Cloud
3. Hybrid cloud.
4. Community Cloud.

PUBLIC CLOUD MODEL

Public Cloud allows systems and services to be easily accessible to general public. The IT giants such as **Google**, **Amazon** and **Microsoft** offer cloud services via Internet. The Public Cloud Model is shown in the diagram below.



Benefits

There are many benefits of deploying cloud as public cloud model. The following diagram shows some of those benefits:

Cost Effective

Since public cloud shares same resources with large number of customers it turns out inexpensive.

Reliability

The public cloud employs large number of resources from different locations. If any of the resources fails, public cloud can employ another one.

Flexibility

The public cloud can smoothly integrate with private cloud, which gives customers a flexible approach.

Location Independence

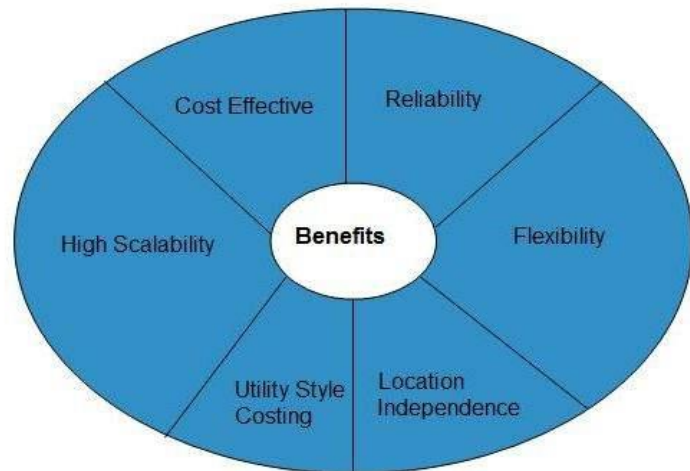
Public cloud services are delivered through Internet, ensuring location independence.

Utility Style Costing

Public cloud is also based on pay-per-use model and resources are accessible whenever customer needs them.

High Scalability

Cloud resources are made available on demand from a pool of resources, i.e., they can be scaled up or down according the requirement.



Disadvantages

Low Security

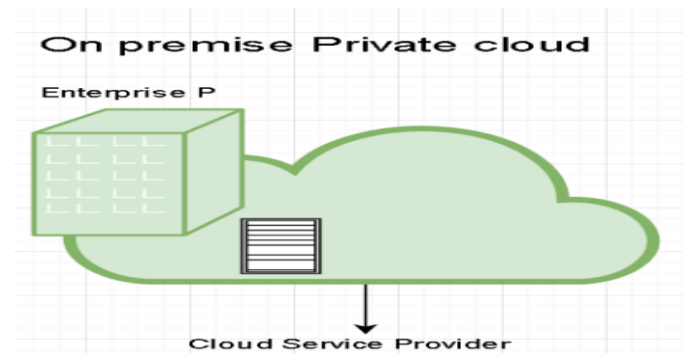
In **public cloud model**, data is hosted off-site and resources are shared publicly, therefore does not ensure higher level of security.

Less Customizable

It is comparatively less customizable than private cloud.

PRIVATE CLOUD MODEL

Private Cloud allows systems and services to be accessible within an organization. The Private Cloud is operated only within a single organization. However, it may be managed internally by the organization itself or by third-party. The private cloud model is shown in the diagram below.



Benefits

There are many benefits of deploying cloud as private cloud model. The following diagram shows some of those benefits:

High Security and Privacy

Private cloud operations are not available to general public and resources are shared from distinct pool of resources. Therefore, it ensures high **security** and **privacy**.

More Control

The **private cloud** has more control on its resources and hardware than public cloud because it is accessed only within an organization.

Cost and Energy Efficiency

The **private cloud** resources are not as cost effective as resources in public clouds but they offer more efficiency than public cloud resources.

Disadvantages

Here are the disadvantages of using private cloud model:

Restricted Area of Operation

The private cloud is only accessible locally and is very difficult to deploy globally.

High Priced

Purchasing new hardware in order to fulfill the demand is a costly transaction.

Limited Scalability

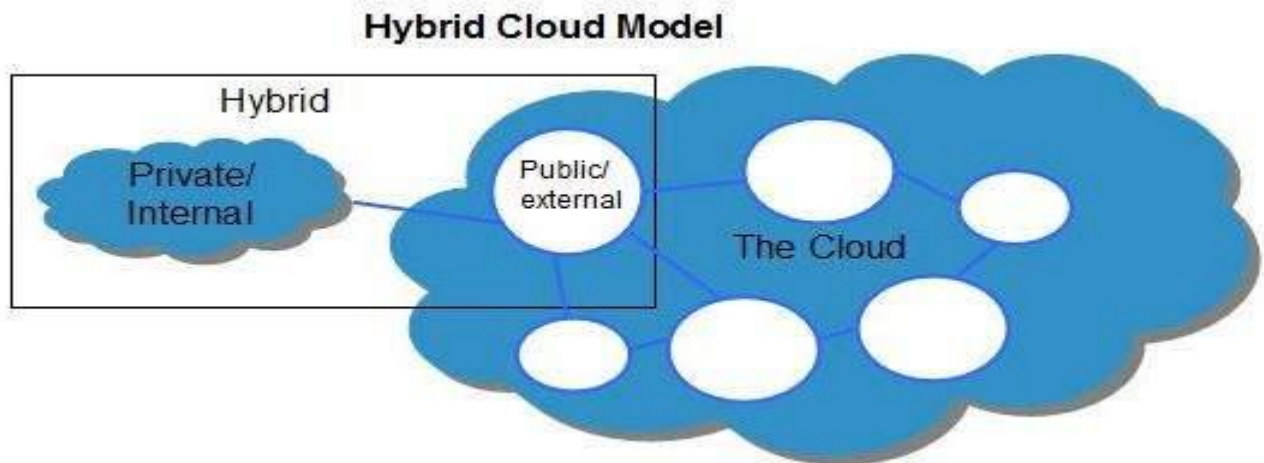
The private cloud can be scaled only within capacity of internal hosted resources.

Additional Skills

In order to maintain cloud deployment, organization requires skilled expertise.

Hybrid Cloud Model

Hybrid Cloud is a mixture of **public** and **private** cloud. Non-critical activities are performed using public cloud while the critical activities are performed using private cloud. The Hybrid Cloud Model is shown in the diagram below.



BENEFITS

There are many benefits of deploying cloud as hybrid cloud model. The following diagram shows some of those benefits:

Scalability

It offers features of both, the public cloud scalability and the private cloud scalability.

Flexibility

It offers secure resources and scalable public resources.

Cost Efficiency

Public clouds are more cost effective than private ones. Therefore, hybrid clouds can be cost saving.

Security

The private cloud in hybrid cloud ensures higher degree of security.

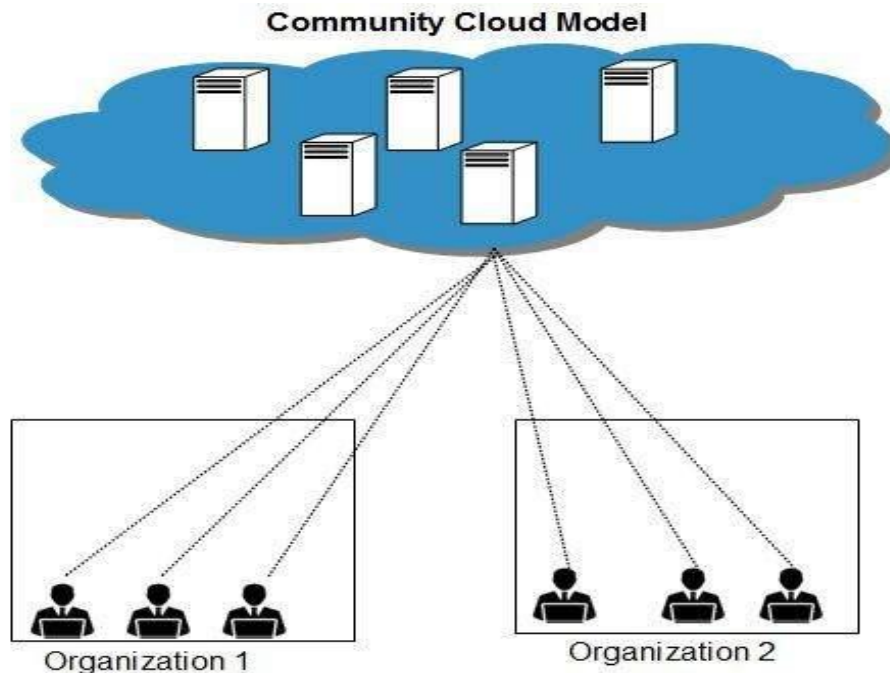
DISADVANTAGES

1. Networking Issues
2. Security Compliance
3. Infrastructure Dependency

The **hybrid cloud model** is dependent on internal IT infrastructure, therefore it is necessary to ensure redundancy across data centers.

Community Cloud Model

Community Cloud allows system and services to be **accessible by group of organizations**. It shares the infrastructure between **several organizations from a specific community**. It may be managed internally by organizations or by the third-party. The Community Cloud Model is shown in the diagram below.



Benefits

There are many benefits of deploying cloud as **community cloud model**.

Cost Effective

Community cloud offers same advantages as that of private cloud at low cost.

Sharing Among Organizations

Community cloud provides an infrastructure to share cloud resources and capabilities among several organizations.

Security

The community cloud is comparatively more secure than the public cloud but less secured than the private cloud.

Issues

- Since all data is located at one place, one must be careful in storing data in community cloud because it might be accessible to others.
- It is also challenging to allocate responsibilities of governance, security and cost among organizations.

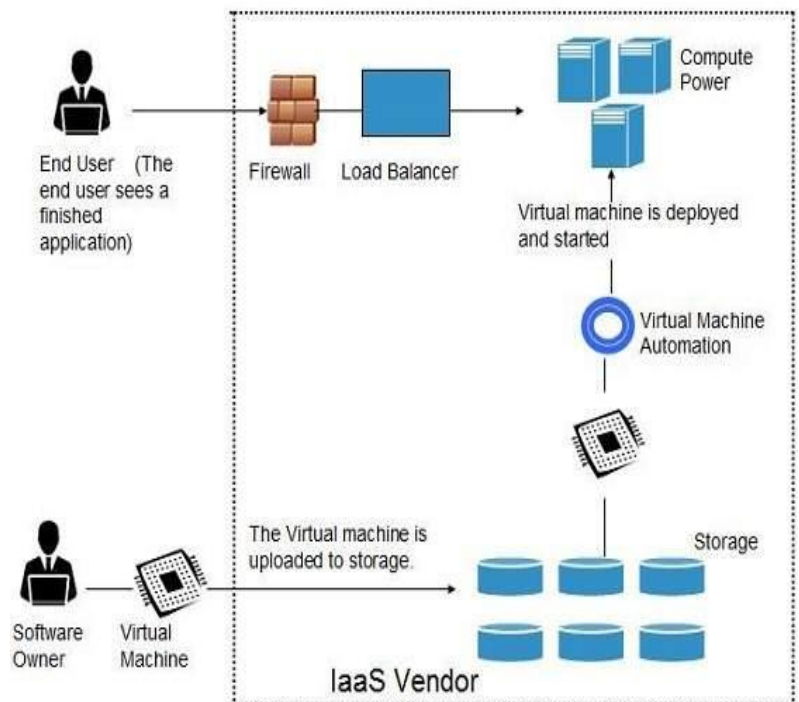
Cloud Computing Services Model

1. IAAS
2. PAAS
3. SAAS

Cloud Computing Infrastructure as a Service (IaaS)

Infrastructure-as-a-Service provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc. Apart from these resources, the IaaS also offers:

- Virtual machine disk storage
- Virtual local area network (VLANs)
- Load balancers
- IP addresses
- Software bundles



Benefits

IaaS allows the cloud provider to freely locate the infrastructure over the Internet in a cost-effective manner.

Some of the key benefits of IaaS are listed below

1. Full control of the computing resources through administrative access to VMs.
2. Flexible and efficient renting of computer hardware.
3. Portability, interoperability with legacy applications.(network and email types app)

Issues

Compatibility with legacy security vulnerabilities

customer to run legacy software in provider's infrastructure, it exposes customers to all of the security vulnerabilities of such legacy software.

Virtual Machine sprawl

Automatically update such VM software may not be ON Mode,

Robustness of VM-level isolation

IaaS offers an isolated environment to individual customers through hypervisor. Hypervisor is a software layer that includes hardware support for virtualization **to split a physical computer** into multiple virtual machines.

Data erase practices

the customer **releases the resource**, the cloud provider must ensure that next customer to rent the resource does not observe data residue from previous customer.

Characteristics

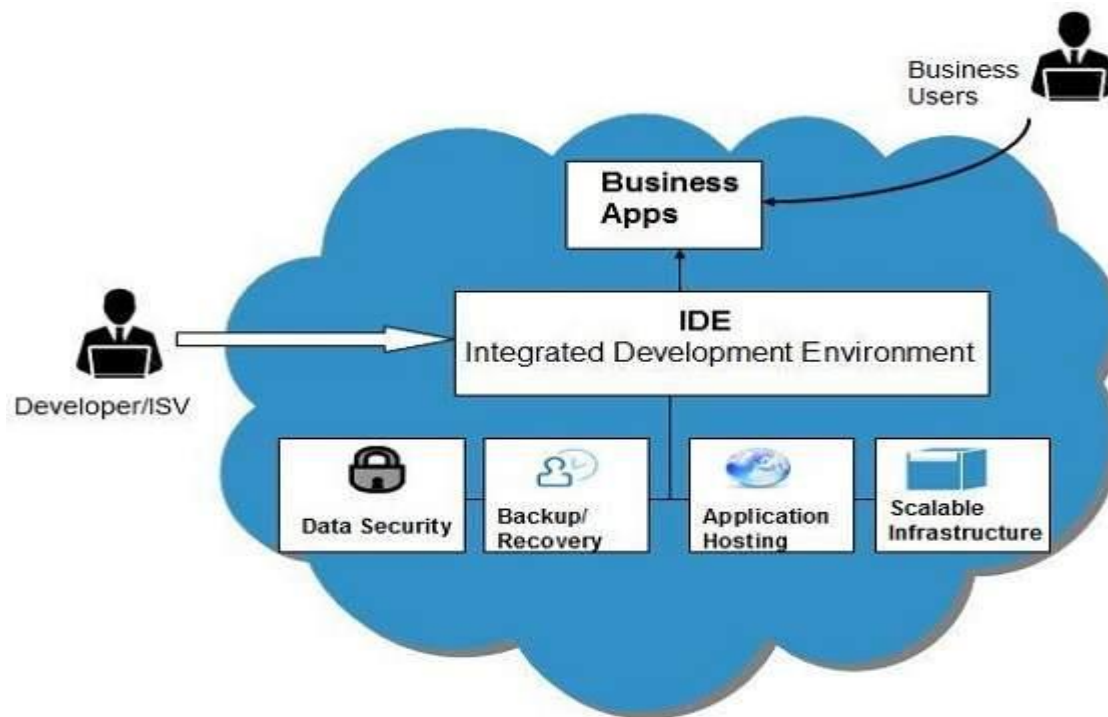
Here are the characteristics of IaaS service model:

- Virtual machines with pre-installed software.
- Virtual machines with pre-installed operating systems such as Windows, Linux, and Solaris.
- On-demand availability of resources.
- Allows to store copies of particular data at different locations.
- The computing resources can be easily scaled up and down.

Platform as a Service (PaaS)

Platform-as-a-Service offers the runtime environment for applications. It also offers development and deployment tools required to develop applications. PaaS has a feature of **point-and-click** tools that enables non-developers to create web applications.

App Engine of Google and **Force.com** are examples of PaaS offering vendors. Developer may log on to these websites and use the **built-in API** to create web-based applications.



Benefits

Following are the benefits of PaaS model:

1. Lower administrative overhead.
2. **Lower total cost of ownership.**
3. Scalable solutions.
4. More current system software (**cloud provider to maintain software versions and patch installations.**)

Issues:

1. Lack of portability between PaaS clouds
2. Event based processor scheduling
3. Security engineering of PaaS applications

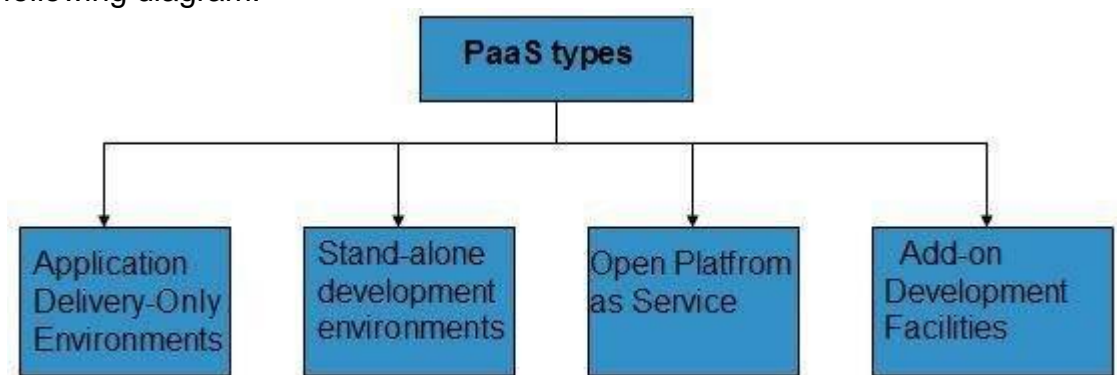
Characteristics

Here are the characteristics of PaaS service model:

- PaaS offers **browser based development environment**. It allows the developer to create database and edit the application code either via Application Programming Interface or point-and-click tools.
- PaaS provides **built-in security, scalability, and web service interfaces**.
- PaaS provides built-in tools for defining **workflow, approval processes**, and business rules.
- It is easy to integrate PaaS with other applications on the same platform.
- PaaS also provides web services interfaces that allow us to connect the applications outside the platform.

PaaS Types

- Based on the functions, PaaS can be classified into four types as shown in the following diagram:



Software as a Service (SaaS)

Software-as-a-Service (SaaS) model allows to provide software application as a service to the end users. It refers to a software that is deployed on a host service and is accessible via Internet. There are several SaaS applications listed below:

- Billing and invoicing system
- Customer Relationship Management (CRM) applications
- Help desk applications
- Human Resource (HR) solutions

Characteristics

Here are the characteristics of SaaS service model:

- SaaS makes the software available over the Internet.
- The software applications are maintained by the vendor.
- The license to the software may be subscription based or usage based. And it is billed on recurring basis.
- SaaS applications are cost-effective since they do not require any maintenance at end user side.
- They are available on demand.
- They can be scaled up or down on demand.
- They are automatically upgraded and updated.
- SaaS offers shared data model. Therefore, multiple users can share single instance of infrastructure. It is not required to hard code the functionality for individual users.
- All users run the same version of the software.

Benefits

Some of the benefits are listed below:

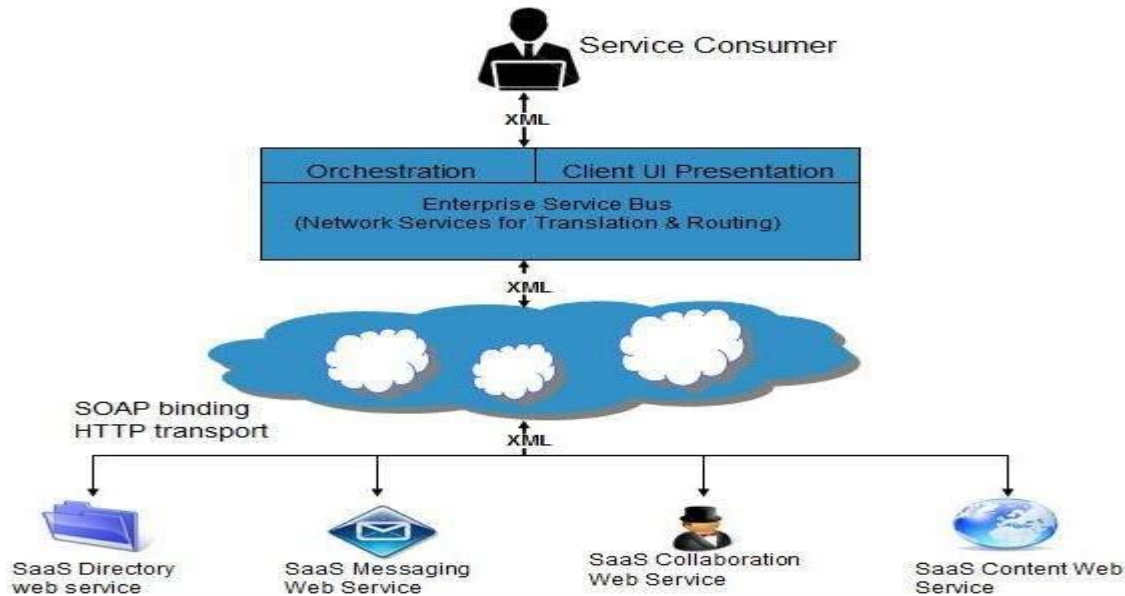
- Modest software tools
- Efficient use of software licenses
- Centralized management and data
- Platform responsibilities managed by provider
- Multitenant solutions

Issues

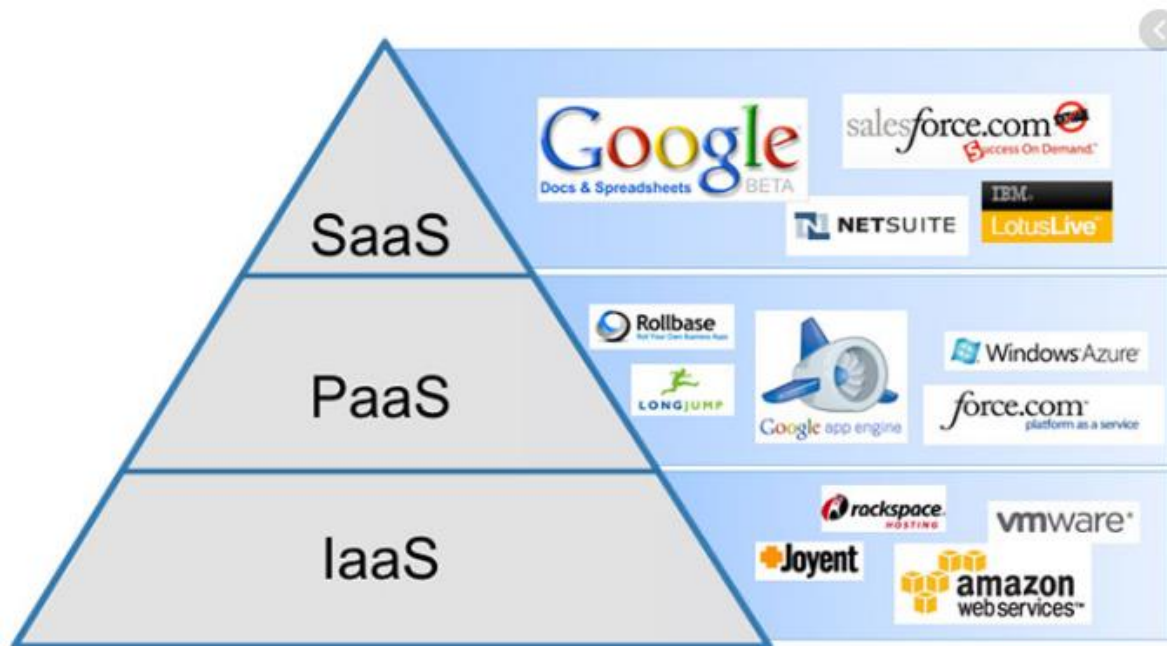
There are several issues associated with SaaS, some of them are listed below:

- Browser based risks
- Network dependence
- Lack of portability between SaaS clouds

The following diagram shows the SaaS implementation based on SOA:



Cloud Service Provider List



What is xml

- **Xml** (eXtensible Markup Language) is a mark up language.
- XML is designed to store and transport data.
- Xml was released in late 90's. it was created to provide an easy to use and store self describing data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

HTML vs XML

No.	HTML	XML
1)	HTML is used to display data and focuses on how data looks.	XML is a software and hardware independent tool used to transport and store data . It focuses on what data is.
2)	HTML is a markup language itself.	XML provides a framework to define markup languages .
3)	HTML is not case sensitive .	XML is case sensitive .
4)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.
5)	HTML has its own predefined tags .	You can define tags according to your need .
6)	In HTML, it is not necessary to use a closing tag .	XML makes it mandatory to use a closing tag .
7)	HTML is static because it is used to display data.	XML is dynamic because it is used to transport data.

XML Example

XML documents create a hierarchical structure looks like a tree so it is known as XML Tree that starts at "the root" and branches to "the leaves"

1. `<?xml version="1.0" encoding="ISO-8859-1"?>`
2. `<note>`
3. `<to>Tove</to>`
4. `<from>Jani</from>`
5. `<heading>Reminder</heading>`
6. `<body>Don't forget me this weekend!</body>`
7. `</note>`

XML: Books

`<bookstore>`

```
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
```

```
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

```
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
```


</book>

</bookstore>

XML Validation

XML document can be validated against **DTD or Schema**.

A well-formed XML document is an XML document with correct syntax.

Valid XML document

It must be well formed (satisfy all the basic syntax condition)

It should be behave according to predefined DTD or XML schema

Rules for well formed XML

- It must begin with the XML declaration.
- It must have one unique root element.
- All start tags of XML documents must match end tags.
- XML tags are case sensitive.
- All elements must be closed.
- All elements must be properly nested.
- All attributes values must be quoted.
- XML entities must be used for special characters.

XML DTD

A DTD defines the legal elements of an XML document

DTD defines the document structure with a list of legal elements and attributes.

XML schema is a **XML based alternative** to DTD.

Its main purpose is to **define the structure of an XML document**

Web Services

A Web service is a method of **communication between two electronic devices** over a network.

It is a software system for the **interoperable machine to machine** communication.

It is a **collection of standards or protocols for exchanging information between two devices** or application.

Types of Web Services

There are mainly two types of web services.

1. SOAP web services.
2. RESTful web services

Web Service Features

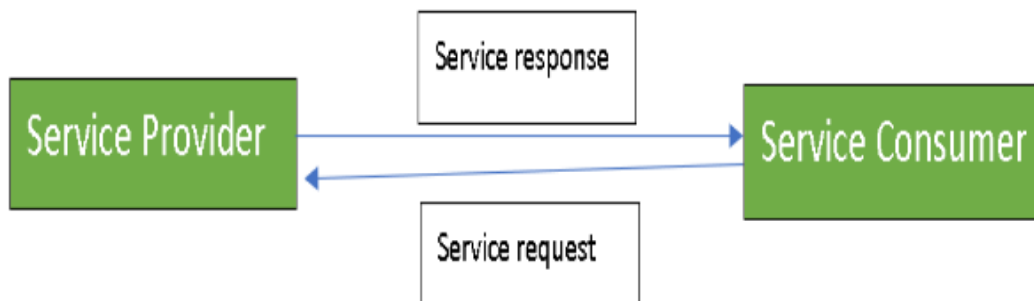
1. XML-Based
2. Loosely Coupled
3. Ability to be Synchronous or Asynchronous
4. Supports Remote Procedure Calls (RPCs)
5. Supports Document Exchange

Service Oriented Architecture Protocol(SOAP)

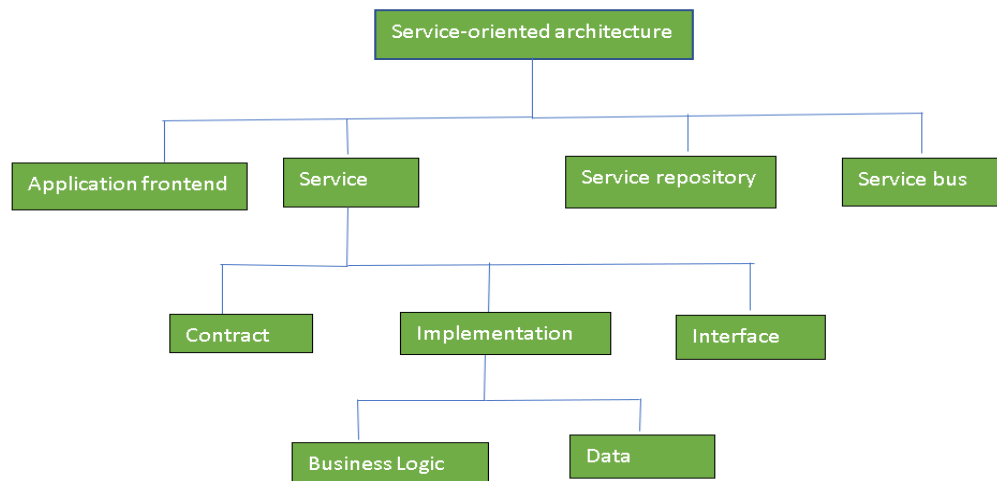
Service-Oriented Architecture helps to **use applications as a service** for other applications. It is not dependent on **the type of vendor, product or technology**. Therefore, it is possible to exchange the **data between applications of different vendors without additional programming** or making changes to services.

There are two major roles within Service-oriented Architecture:

1. **Service provider:** The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. the provider can **publish registry, service contract that specifies the nature of the service, how to use it,**
2. **Service consumer:** The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.



Components of SOA:



Advantages of SOA:

- **Service reusability:** .
- **Easy maintenance:** As services are independent of each other they can be updated and modified easily without affecting other services.
- **Platform independent:** SOA services independent of the platform.
- **Availability:** SOA facilities are easily available to anyone on request.
- **Reliability:** SOA applications are more reliable because it is **easy to debug small services** rather than huge codes
- **Scalability:** Services can run on different servers within an environment, this increases scalability

Disadvantages of SOA:

- **High overhead:** A validation of input parameters of services is done whenever services interact this **decreases performance** as it **increases load and response time**.
- **High investment:** A huge initial investment is required for SOA.
- **Complex service management:** task to handle a large number of messages.

Practical applications of SOA:

SOA is used in many ways around us whether it is mentioned or not.

1. SOA infrastructure is **used by many armies and air force** to deploy situational awareness systems.
2. SOA is used to **improve the healthcare delivery**.
3. Many games use inbuilt functions to run. For example, an app might need GPS so it uses inbuilt GPS functions of the device. This is SOA in mobile solutions.
4. SOA helps **maintain museums a virtualized storage pool** for their information and content.

Challenges in Cloud Computing

- Security and Privacy
- Portability
- Interoperability
- Computing Performance
- Reliability and Availability

Hypervisor

Hypervisor is a **firmware** or **low-level program** that acts as a Virtual Machine Manager. It allows to share the single physical instance of cloud resources between several tenants

The hypervisor is a firmware or low-level program that acts as a **Virtual Machine Manager**.

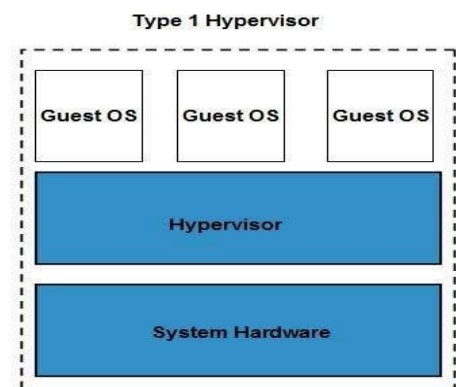
There are two types of hypervisor:

A hypervisor is a form of virtualization software used in Cloud hosting **to divide and allocate the resources on various pieces of hardware**. The program which provides partitioning, isolation or abstraction is called **virtualization hypervisor**

The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time.

Type 1 hypervisor executes on bare system. LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogic VLX are examples of Type 1 hypervisor. The following diagram shows the Type 1 hypervisor.

The **type1 hypervisor** does not have any host operating system because they are installed on a bare system.



Type 2 hypervisor is a software interface that emulates the devices with which a system normally interacts.

Containers, KVM, Microsoft Hyper V, VMWare Fusion, Virtual Server 2005 R2, Windows Virtual PC and **VMWare workstation 6.0** are examples of Type 2 hypervisor. The following diagram shows the Type 2 hypervisor.

HYPERVERSOR REFERENCE MODEL :

There are 3 main modules coordinates in order to emulate the underlying hardware:

1. DISPATCHER:2.ALLOCATOR 3 INTERPRETER:

