virtual machines or virtual servers along with virtual storages. The abstraction of these hardware resources is intended to provide the flexibility to the users. Internally, virtualization performs automated resource provisioning and optimizes the process of managing resources. The infrastructure layer act as a foundation for building the second layer called platform layer for supporting PaaS services.

The platform layer is responsible for providing readily available development and deployment platform for web applications to the cloud users without needing them to install in a local device. The platform layer has collection of software tools for development, deployment and testing the software applications. This layer provides an environment for users to create their applications, test operation flows, track the performance and monitor execution results. The platform must be ensuring to provide scalability, reliability and security. In this layer, virtualized cloud platform, acts as an "application middleware" between the cloud infrastructure and application layer of cloud. The platform layer is the foundation for application layer.

A collection of all software modules required for SaaS applications forms the application layer. This layer is mainly responsible for making on demand application delivery. In this layer, software applications include day-to-day office management softwares used for information collection, document processing, calendar and authentication. Enterprises also use the application layer extensively in business marketing, sales, Customer Relationship Management (CRM), financial transactions and Supply Chain Management (SCM). It is important to remember that not all cloud services are limited to a single layer. Many applications can require mixed - layers resources. After all, with a relation of dependency, the three layers are constructed from the bottom up approach. From the perspective of the user, the services at various levels need specific amounts of vendor support and resource management for functionality. In general, SaaS needs the provider to do much more work, PaaS is in the middle and IaaS requests the least. The best example of application layer is the Salesforce.com's CRM service where not only the hardware at the bottom layer and the software at the top layer is supplied by the vendor, but also the platform and software tools for user application development and monitoring.

3.2 NIST Cloud Computing Reference Architecture

In this section, we will examine and discuss the reference architecture model given by the National Institute of Standards and Technology (NIST). The model offers approaches for secure cloud adoption while contributing to cloud computing guidelines and standards.

The NIST team works closely with leading IT vendors, developers of standards, industries and other governmental agencies and industries at a global level to support effective cloud computing security standards and their further development. It is important to note that this NIST cloud reference architecture does not belong to any specific vendor products, services or some reference implementation, nor does it prevent further innovation in cloud technology.

The NIST reference architecture is shown in Fig. 3.2.1.

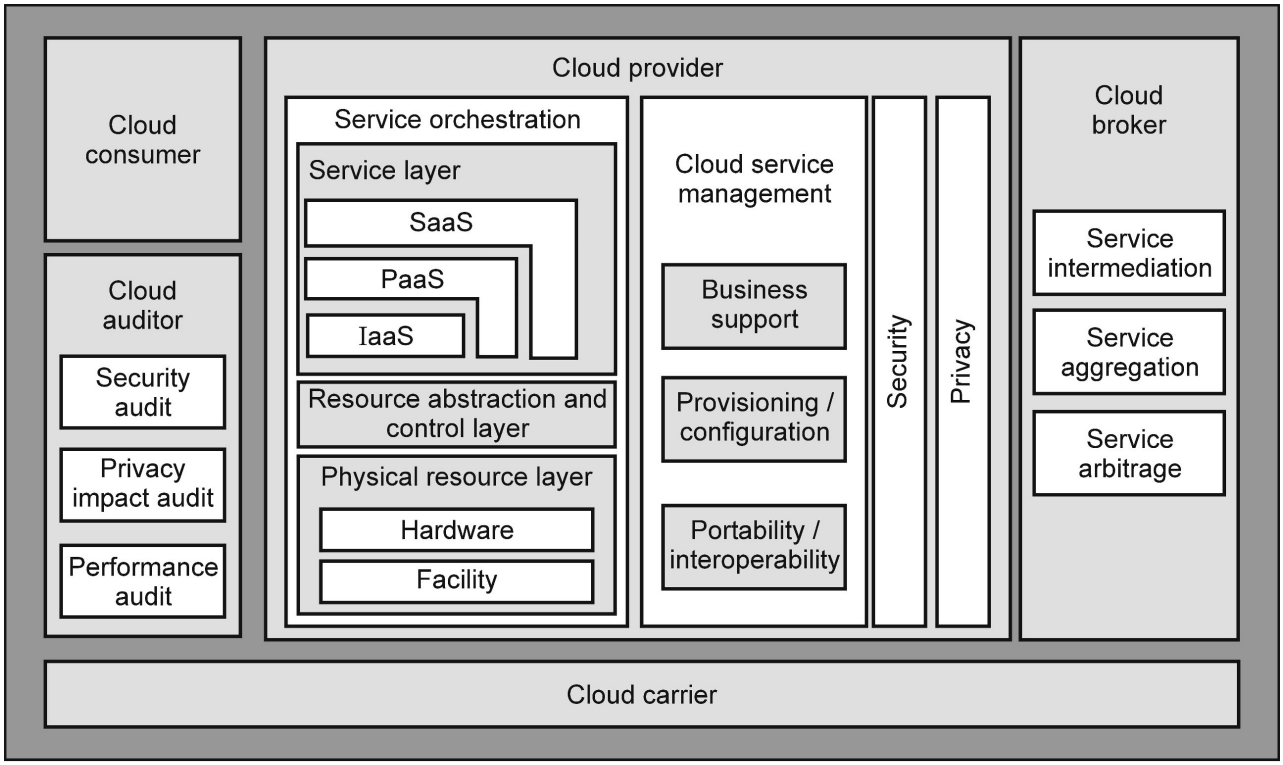

Fig. 3.2.1 : Conceptual cloud reference model showing different actors and entities

From Fig. 3.2.1, note that the cloud reference architecture includes five major actors :

- Cloud consumer
- Cloud provider
- Cloud auditor
- Cloud broker
- Cloud carrier

Each actor is an organization or entity plays an important role in a transaction or a process, or performs some important task in cloud computing. The interactions between these actors are illustrated in Fig. 3.2.2.

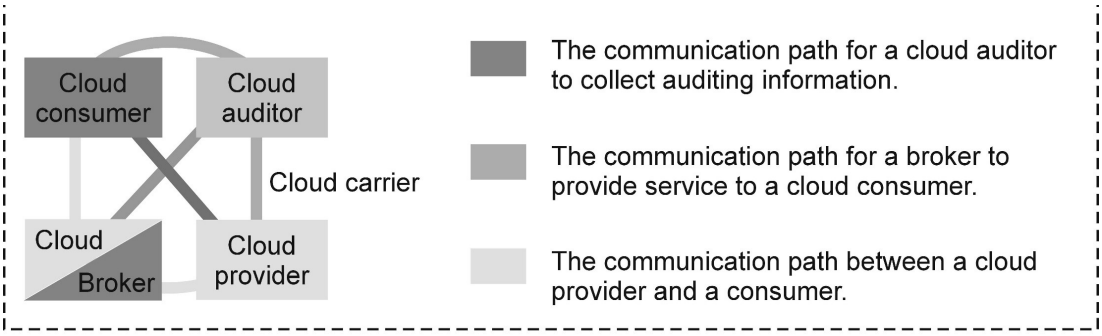

Fig. 3.2.2 : Interactions between different actors in a cloud

Now, understand that a cloud consumer can request cloud services directly from a CSP or from a cloud broker. The cloud auditor independently audits and then contacts other actors to gather information. We will now discuss the role of each actor in detail.

3.2.1 Cloud Consumer

A cloud consumer is the most important stakeholder. The cloud service is built to support a cloud consumer. The cloud consumer uses the services from a CSP or person or asks an organization that maintains a business relationship. The consumer then verifies the service catalogue from the cloud provider and requests an appropriate service or sets up service contracts for using the service. The cloud consumer is billed for the service used.

Some typical usage scenarios include :

Example 1 : Cloud consumer requests the service from the broker instead of directly contacting the CSP. The cloud broker can then create a new service by combining multiple services or by enhancing an existing service. Here, the actual cloud provider is not visible to the cloud consumer. The consumer only interacts with the broker. This is illustrated in Fig. 3.2.3.

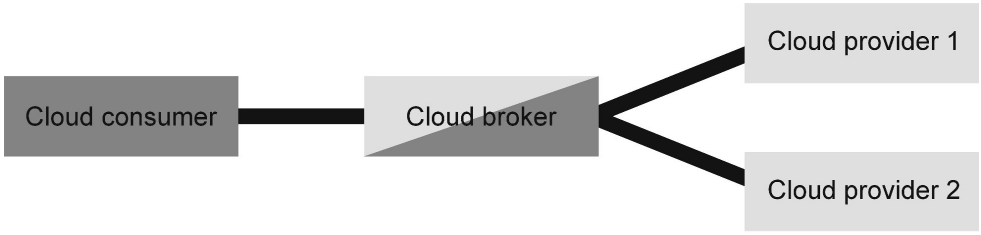

Fig. 3.2.3 : Cloud broker interacting with cloud consumer

Example 2 : In this scenario, the cloud carrier provides for connectivity and transports cloud services to consumers. This is illustrated in Fig. 3.2.4.

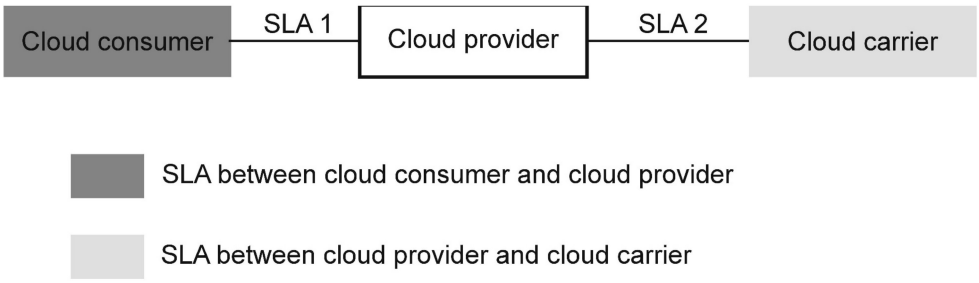

Fig. 3.2.4 : Scenario for cloud carrier

In Fig. 3.2.4, the cloud provider participates by arranging two SLAs. One SLA is with the cloud provider (SLA2) and the second SLA is with the consumer (SLA1). Here, the cloud provider will have an arrangement (SLA) with the cloud carrier to have secured, encrypted connections. This ensures that the services are available for the consumer at a consistent level to fulfil service requests. Here, the provider can specify the requirements, such as flexibility, capability and functionalities in SLA2 to fulfil essential service requirements in SLA1.

Example 3 : In this usage scenario, the cloud auditor conducts independent evaluations for a cloud service. The evaluations will relate to operations and security of cloud service implementation. Here the cloud auditor interacts with both the cloud provider and consumer, as shown in Fig. 3.2.5.

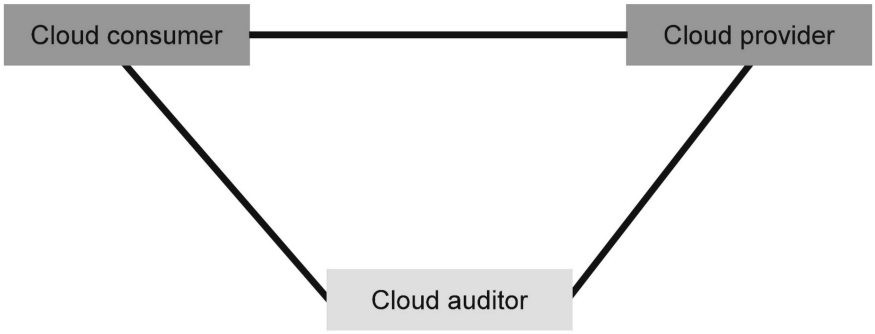

Fig. 3.2.5 : Usage scenario involving a cloud auditor

In all the given scenarios, the cloud consumer plays the most important role. Based on the service request, the activities of other players and usage scenarios can differ for other cloud consumers. Fig. 3.2.6 shows an example of available cloud services types.

In Fig. 3.2.6, note that SaaS applications are available over a network to all consumers. These consumers may be organisations with access to software applications, end users, app developers or administrators. Billing is based on the number of end users, the time of use, network bandwidth consumed and for the amount or volume of data stored.

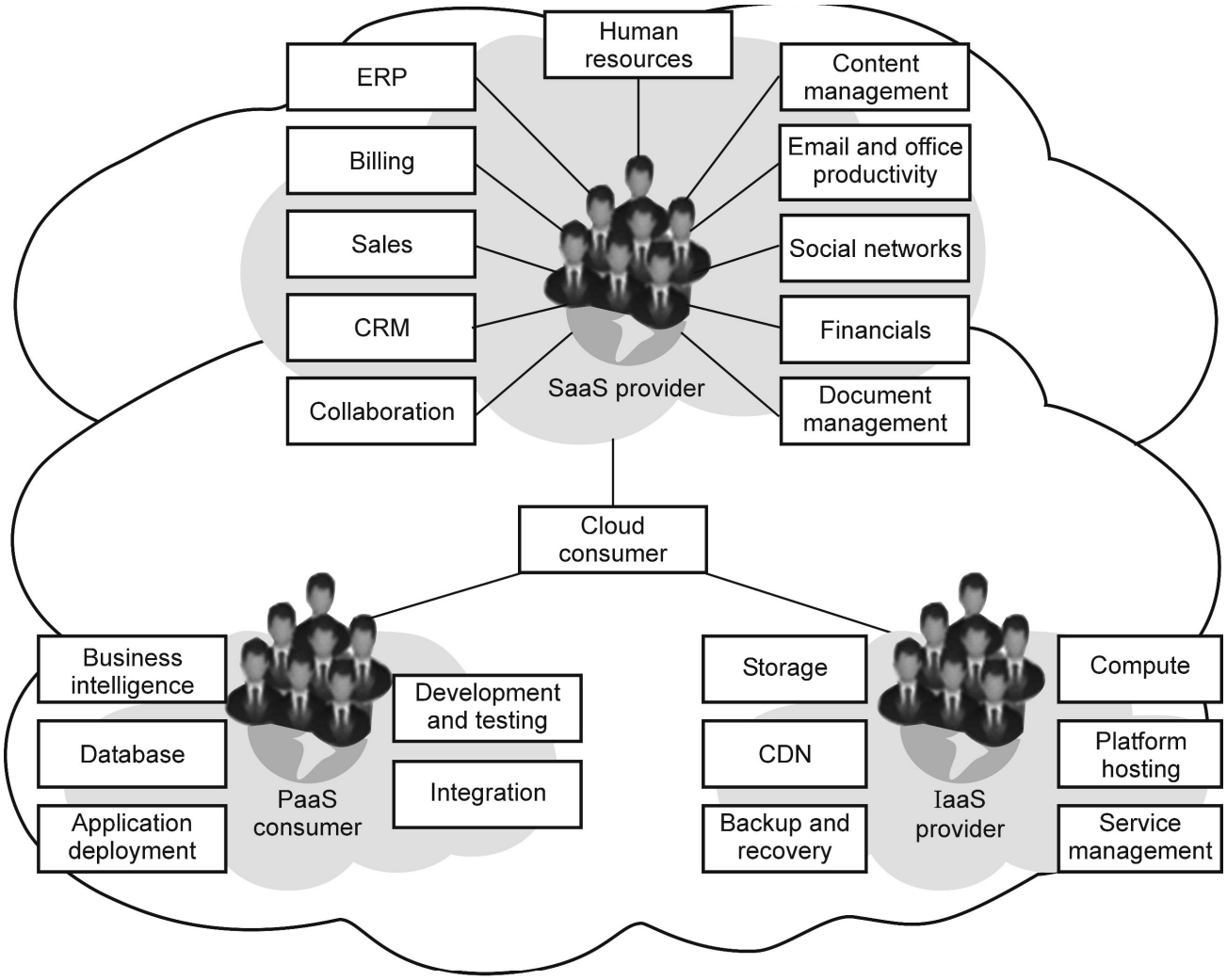

Fig. 3.2.6 : Example of cloud services available to cloud consumers

PaaS consumers can utilize tools, execution resources, development IDEs made available by cloud providers. Using these resources, they can test, develop, manage, deploy and configure many applications that are hosted on a cloud. PaaS consumers are billed based on processing, database, storage, network resources consumed and for the duration of the platform used.

On the other hand, IaaS consumers can access virtual computers, network - attached storage, network components, processor resources and other computing resources that are deployed and run arbitrary software. IaaS consumers are billed based on the amount and duration of hardware resources consumed, number of IP addresses, volume of data stored, network bandwidth, and CPU hours used for a certain duration.

3.2.2 Cloud Provider

Cloud provider is an entity that offers cloud services to interested parties. A cloud provider manages the infrastructure needed for providing cloud services. The CSP also runs the software to provide services and organizes the service delivery to cloud consumers through networks.

SaaS providers then deploy, configure, maintain and update all operations of the software application on the cloud infrastructure, in order to ensure that services are provisioned and to fulfill cloud consumer service requests. SaaS providers assume most of the responsibilities associated with managing and controlling applications deployed on the infrastructure. On the other hand, SaaS consumers have no or limited administrative controls.

PaaS cloud providers manage the computing infrastructure and ensure that the platform runs the cloud software and implements databases, appropriate runtime software execution stack and other required middleware elements. They support development, deployment and the management of PaaS consumers by providing them with necessary tools such as IDEs, SDKs and others. PaaS providers have complete control of applications, settings of the hosting environment, but have lesser control over the infrastructure lying under the platform, network, servers, OS and storage.

Now, the IaaS CSP aggregates physical cloud resources such as networks, servers, storage and network hosting infrastructure. The provider operates the cloud software and makes all compute resources available to IaaS cloud consumer via a set of service interfaces, such as VMs and virtual network interfaces. The IaaS cloud provider will have control over the physical hardware and cloud software to enable provisioning and possible infrastructure services.

The main activities of a cloud provider can be viewed in Fig. 3.2.7.

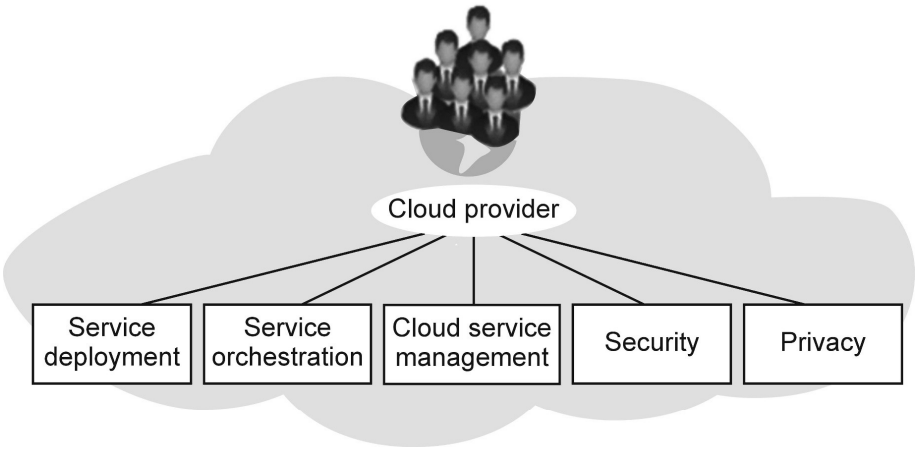

Fig. 3.2.7 : Major activities of a cloud provider

The major activities of a cloud provider include :

- **Service deployment :** Service deployment refers to provisioning private, public, hybrid and community cloud models.
- **Service orchestration :** Service orchestration implies the coordination, management of cloud infrastructure and arrangement to offer optimized capabilities of cloud services. The capabilities must be cost-effective in managing IT resources and must be determined by strategic business needs.
- **Cloud services management :** This activity involves all service-related functions needed to manage and operate the services requested or proposed by cloud consumers.
- **Security :** Security, which is a critical function in cloud computing, spans all layers in the reference architecture. Security must be enforced end-to-end. It has a wide range from physical to application security. CSPs must take care of security.
- **Privacy :** Privacy in cloud must be ensured at different levels, such as user privacy, data privacy, authorization and authentication and it must also have adequate assurance levels. Since clouds allow resources to be shared, privacy challenges are a big concern for consumers using clouds.

3.2.3 Cloud Auditor

The cloud auditor performs the task of independently evaluating cloud service controls to provide an honest opinion when requested. Cloud audits are done to validate standards conformance by reviewing the objective evidence. The auditor will examine services provided by the cloud provider for its security controls, privacy, performance, and so on.

3.2.4 Cloud Broker

The cloud broker collects service requests from cloud consumers and manages the use, performance, and delivery of cloud services. The cloud broker will also negotiate and manage the relationship between cloud providers and consumers. A cloud broker may provide services that fall into one of the following categories :

- **Service intermediation :** Here the cloud broker will improve some specific capabilities, and provide value added services to cloud consumers.
- **Service aggregation :** The cloud broker links and integrates different services into one or more new services.

- **Service Arbitrage :** This is similar to aggregation, except for the fact that services that are aggregated are not fixed. In service arbitrage, the broker has the liberty to choose services from different agencies.

3.2.5 Cloud Carrier

The cloud carrier tries to establish connectivity and transports cloud services between a cloud consumer and a cloud provider. Cloud carriers offer network access for consumers, by providing telecommunication links for accessing resources using other devices (laptops, computers, tablets, smartphones, etc.). Usually, a transport agent is an entity offering telecommunication carriers to a business organization to access resources. The cloud provider will set up SLAs with cloud carrier to ensure carrier transport is consistent with the level of SLA provided by the consumers. Cloud carriers provide secure and dedicated high - speed links with cloud providers and between different cloud entities.

3.3 Cloud Deployment Models

A cloud deployment models are defined according to where the computing infrastructure resides and who controls the infrastructure. The NIST have classified cloud deployment models into four categories namely,

- **Public cloud**
- **Private cloud**
- **Hybrid cloud**
- **Community cloud**

They describe the way in which users can access the cloud services. Each cloud deployment model fits different organizational needs, so it's important that you pick a model that will suit your organization's needs. The four deployment models are characterized based on the functionality and accessibility of cloud services. The four deployment models of cloud computing are shown in Fig. 3.3.1.

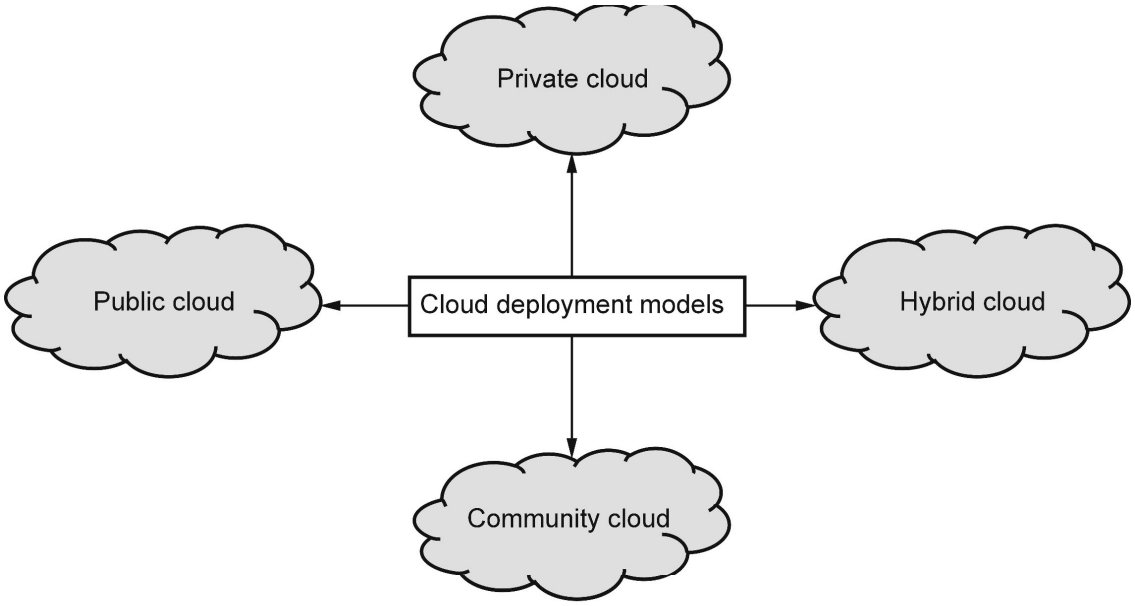

Fig. 3.3.1 : Four deployment models of cloud computing

3.3.1 Public Cloud

The public cloud services are runs over the internet. Therefore, the users who want cloud services have to have internet connection in their local device like thin client, thick client, mobile, laptop or desktop etc. The public cloud services are managed and maintained by the Cloud Service Providers (CSPs) or the Cloud Service Brokers (CSBs). The public cloud services are often offered on utility base pricing like subscription or pay-per-use model. The public cloud services are provided through internet and APIs. This model allows users to easily access the services without purchasing any specialize hardware or software. Any device which has web browser and internet connectivity can be a public cloud client. The popular public cloud service providers are Amazon web services, Microsoft azure and Google app engine, Salesforce etc.

Advantages of public cloud

1. It saves capital cost behind purchasing the server hardware's, operating systems and application software licenses.
2. There is no need of server administrators to take care of servers as they are kept at CSPs data center and managed by them.
3. No training is required to use or access the cloud services.
4. There is no upfront or setup cost is required.
5. A user gets easy access to multiple services under a single self - service portal.
6. Users have a choice to compare and select between the providers.

7. It is cheaper than in house cloud implementation because user have to pay for that they have used.
8. The resources are easily scalable.

Disadvantages of public cloud

1. There is lack of data security as data is stored on public data center and managed by third party data center vendors therefore there may be compromise of user's confidential data.
2. Expensive recovery of backup data.
3. User never comes to know where (at which location) their data gets stored, how that can be recovered and how many replicas of data have been created.

3.3.2 Private Cloud

The private cloud services are used by the organizations internally. Most of the times it run over the intranet connection. They are designed for a single organization therefore anyone within the organization can get access to data, services and web applications easily through local servers and local network but users outside the organizations cannot access them. This type of cloud services are hosted on intranet therefore users who are connected to that intranet get access to the services. The infrastructure for private cloud is fully managed and maintained by the organization itself. It is much more secure than public cloud as it gives freedom to local administrators to write their own security policies for user's access. It also provides good level trust and privacy to the users. Private clouds are more expensive than public clouds due to the capital expenditure involved in acquiring and maintaining them. The well-known private cloud platforms are Openstack, Open nebula, Eucalyptus, VMware private cloud etc.

Advantages of private cloud

1. Speed of access is very high as services are provided through local servers over local network.
2. It is more secure than public cloud as security of cloud services are handled by local administrator.
3. It can be customized as per organizations need.
4. It does not require internet connection for access.
5. It is easy to manage than public cloud.

Disadvantages of private cloud

1. Implementation cost is very high as setup involves purchasing and installing servers, Hypervisors, Operating systems.
2. It requires administrators for managing and maintaining servers.
3. The scope of scalability is very limited.

3.3.3 Hybrid Cloud

The hybrid cloud services are composed of two or more clouds that offers the benefits of multiple deployment models. It mostly comprises on premise private cloud and off-premise public cloud to leverage benefits of both and allow users inside and outside to have access to it. The Hybrid cloud provides flexibility such that users can migrate their applications and services from private cloud to public cloud and vice versa. It becomes most favored in IT industry because of its eminent features like mobility, customized security, high throughput, scalability, disaster recovery, easy backup and replication across clouds, high availability and cost efficient etc. The popular hybrid clouds are AWS with eucalyptus, AWS with VMware cloud, Google cloud with Nutanix etc.

The limitations of hybrid cloud are compatibility of deployment models, vendor-lock in solutions, requires a common cloud management software and management of separate cloud platforms etc.

3.3.4 Community Cloud

The community cloud is basically the combination of one or more public, private or hybrid clouds, which are shared by many organizations for a single cause. The community cloud is setup between multiple organizations whose objective is same. The Infrastructure for community cloud is to be shared by several organizations within specific community with common security, compliance objectives which is managed by third party organizations or managed internally. The well-known community clouds are Salesforce, Google community cloud etc.

3.3.5 Comparison between various Cloud Deployment Models

The comparison between different deployment models of cloud computing are given in Table 3.3.1.

Sr. No	Feature	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
1	**Scalability**	Very High	Limited	Very High	Limited
2	**Security**	Less Secure	Most Secure	Very Secure	Less Secure
3	**Performance**	Low to Medium	Good	Good	Medium
4	**Reliability**	Medium	High	Medium to High	Medium
5	**Upfront Cost**	Low	Very High	Medium	Medium
6	**Quality of Service**	Low	High	Medium	Medium
7	**Network**	Internet	Intranet	Intranet and Internet	Internet
8	**Availability**	For general public	Organizations internal staff	For general public and organizations internal Staff	For community members
9	**Example**	Windows Azure, AWS etc.	Openstack, VMware cloud, CloudStack, Eucalyptus etc.	Combination of Openstack and AWS	Salesforce community

Table 3.3.1 : Comparison between various Cloud Deployment Models

3.4 Cloud Service Models

A Cloud computing is meant to provide variety of services and applications for users over the internet or intranet. The most widespread services of cloud computing are categorised into three service classes which are called cloud service models or cloud reference models or working models of cloud computing. They are based on the abstraction level of the offered capabilities and the service model of the CSPs. The various service models are :

- **Infrastructure as a Service (IaaS)**
- **Platform as a Service (PaaS)**
- **Software as a Service (SaaS)**

The three service models of cloud computing and their functions are shown in Fig. 3.4.1.

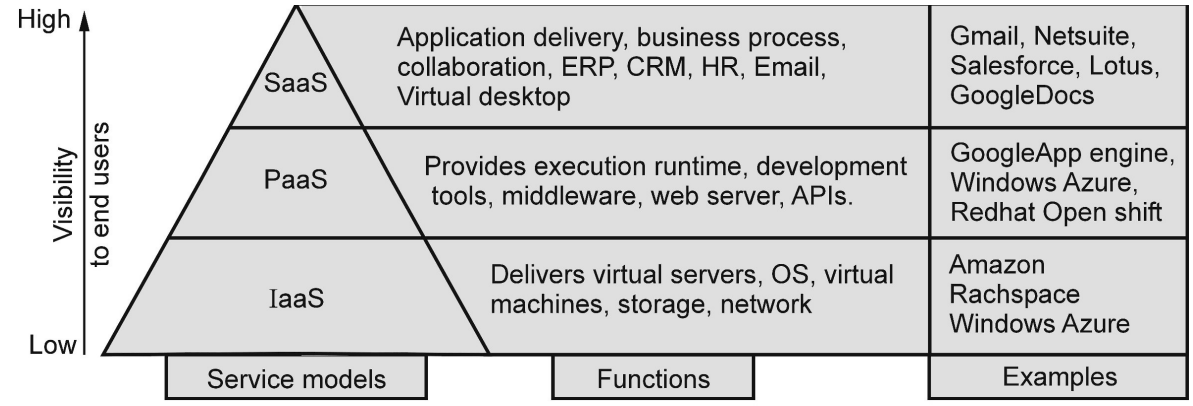

Fig. 3.4.1 : Cloud service models

From Fig. 3.4.1, we can see that the Infrastructure as a Service (IaaS) is the bottommost layer in the model and Software as a Service (SaaS) lies at the top. The IaaS has lower level of abstraction and visibility, while SaaS has highest level of visibility.

The Fig. 3.4.2 represents the cloud stack organization from physical infrastructure to applications. In this layered architecture, the abstraction levels are seen where higher layer services include the services of the underlying layer.

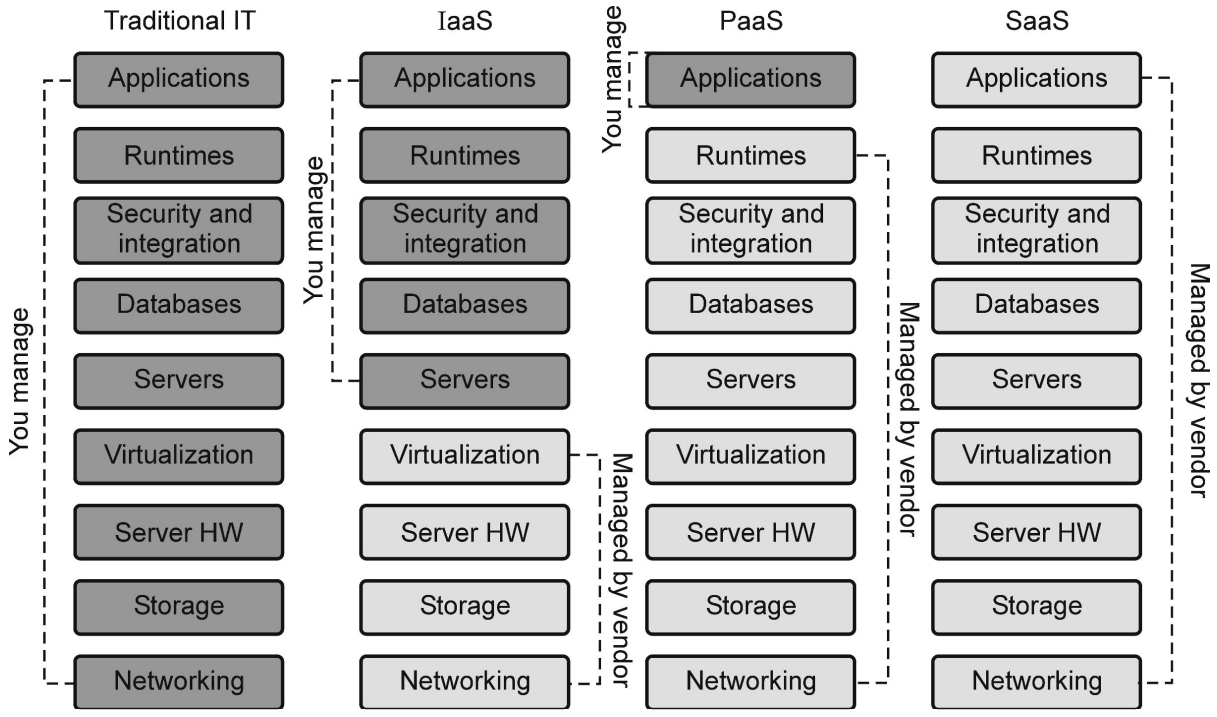

Fig. 3.4.2 : The cloud computing stack

As you can see in Fig. 3.4.2, the three services, IaaS, PaaS and SaaS, can exist independent of one another or may combine with one another at some layers. Different layers in every cloud computing model are either managed by the user or by the vendor

(provider). In case of the traditional IT model, all the layers or levels are managed by the user because he or she is solely responsible for managing and hosting the applications. In case of IaaS, the top five layers are managed by the user, while the four lower layers (virtualisation, server hardware, storage and networking) are managed by vendors or providers. So, here, the user will be accountable for managing the operating system via applications and managing databases and security of applications. In case of PaaS, the user needs to manage only the application and all the other layers of the cloud computing stack are managed by the vendor. Lastly, SaaS abstracts the user from all the layers as all of them are managed by the vendor and the former is responsible only for using the application.

The core middleware manages the physical resources and the VMs are deployed on top of them. This deployment will provide the features of pay-per-use services and multi-tenancy. Infrastructure services support cloud development environments and provide capabilities for application development and implementation. It provides different libraries, models for programming, APIs, editors and so on to support application development. When this deployment is ready for the cloud, they can be used by end-users/ organisations. With this idea, let us further explore the different service models.

3.4.1 Infrastructure as a Service (IaaS)

Infrastructure-as-a-Service (IaaS) can be defined as the use of servers, storage, computing power, network and virtualization to form utility like services for users. It is a cloud service model that provides hardware resources virtualized in the cloud. It provides virtual computing resources to the users through resource pool. In IaaS, the CSP owns all equipment, such as servers, storage disks, and network infrastructure. Developers use the IaaS service model to create virtual hardware on which the applications and/ or services are developed. We can understand that an IaaS cloud provider will create hardware utility service and make them available for users to provision virtual resources as per need. Developers can create virtual private storage, virtual private servers, and virtual private networks by using IaaS. The private virtual systems contain software applications to complete the IaaS solution. The infrastructure of IaaS consists of communication networks, physical compute nodes, storage solutions and the pool of virtualized computing resources managed by a service provider. IaaS provides users with a web-based service that can be used to create, destroy and manage virtual machines and storage. It is a way of delivering cloud computing infrastructure like Virtual servers, Virtual storage, Virtual network and Virtual operating systems as an on-demand service. Instead of purchasing extra servers, softwares, datacenter space or

network equipment, IaaS enables on-demand provisioning of computational resources in the form of virtual machines in cloud data center. Some key providers of IaaS are Amazon Web Services (AWS), Microsoft Azure, GoGrid, Joyent, Rackspace etc. and some of the private cloud softwares through which IaaS can be setup are Openstack, Apache Cloud Stack, Eucalyptus, and VMware VSphere etc.

You must understand that the virtualised resources are mapped to real systems in IaaS. This can be understood as when a user with IaaS service makes a request for a service from virtual systems, that request is redirected to the physical server that does the actual work. The structure of the IaaS model is shown in Fig. 3.4.3.

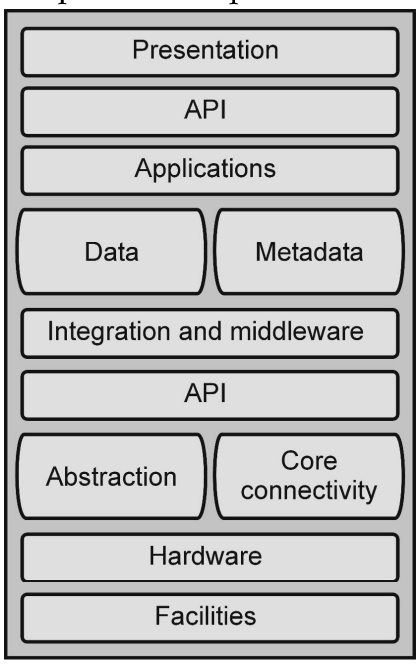

Fig. 3.4.3 : Components in IaaS service model (cloud security alliance)

In IaaS service delivery, workload is the fundamental component of the virtualised client. It simulates the capacity of a physical server to perform work. Hence, the work done is equal to the total number of Transaction Per Minute (TPM). Note that the workload also has other attributes, such as disk I/O (determined by I/O per second), RAM used in MB, latency and network throughput and others.

In the case of hosted applications, the client runs on a dedicated server inside a server rack. It may also run on a standalone server. In cloud computing, the provisioned server is known as an instance (or server instance), which is reserved by a customer, along with adequate computing resources required to fulfil their resource requirements. The user reserves an equivalent machine required to run workloads.

The IaaS infrastructure runs the instances of the server in the data centre offering the service. The resources for this server instance are drawn from a mix of virtualised systems, RAID disks, network and interface capacity. These are physical systems partitioned into logical smaller logical units.

The client in IaaS is allocated with its own private network. For example, Amazon EC2 enables this service to behave such that each server has its own separate network unless the user creates a virtual private cloud. If the EC2 deployment is scaled by adding additional networks on the infrastructure, it is easy to logically scale, but this can create an overhead as traffic gets routed between logical networks.

In IaaS, the customer has controls over the OS, storage and installed applications, but has limited control over network components. The user cannot control the underlying cloud infrastructure. Services offered by IaaS include web servers, server hosting, computer hardware, OS, virtual instances, load balancing, web servers and bandwidth provisioning. These services are useful during volatile demands and when there is a computing resource need for a new business launch or when the company may not want to buy hardware or if the organisation wants to expand.

3.4.2 Platform as a Service

The Platform as a Service can be defined as a computing platform that allows the user to create web applications quickly and easily and without worrying about buying and maintaining the software and infrastructure. Platform-as-a-Service provides tools for development, deployment and testing the softwares, middleware solutions, databases, programming languages and APIs for developers to develop custom applications; without installing or configuring the development environment. The PaaS provides a platform to run web applications without installing them in a local machine i.e. the applications written by the users can be directly run on the PaaS cloud. It is built on the top of IaaS layer. The PaaS realizes many of the unique benefits like utility computing, hardware virtualization, dynamic resource allocation, low investment costs and pre-configured development environment. It has all the application typically required by the client deployed on it. The challenge associated with PaaS is compatibility i.e. if user wants to migrate the services from one provider to other then they have checked the compatibility of execution engine and cloud APIs first. Some key providers of PaaS clouds are Google App Engine, Microsoft Azure, NetSuite, Red hat Open shift etc.

The PaaS model includes the software environment where the developer can create custom solutions using development tools available with the PaaS platform. The components of a PaaS platform are shown in Fig. 3.4.4. Platforms can support specific development languages, frameworks for applications and other constructs. Also, PaaS provides tools and development environments to design applications. Usually, a fully Integrated Development Environment (IDE) is available as a PaaS service. For PaaS to be a cloud computing service, the platform supports user interface development. It also has many standards such as HTML, JavaScript, rich media and so on.

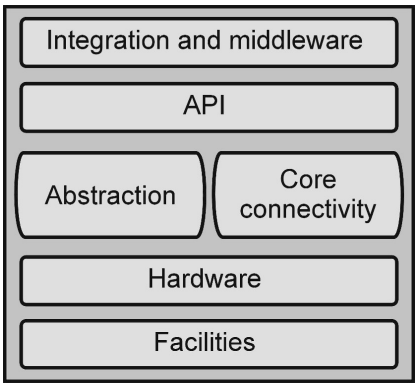

Fig. 3.4.4 : Components of PaaS

In this model, users interact with the software and append and retrieve data, perform an action, obtain results from a process task and perform other actions allowed by the PaaS vendor. In this service model, the customer does not own any responsibility to maintain the hardware and software and the development environment. The applications created are the only interactions between the customer and the PaaS platform. The PaaS cloud provider owns responsibility for all the operational aspects, such as maintenance, updates, management of resources and product lifecycle. A PaaS customer can control services such as device integration, session management, content management, sandbox, and so on. In addition to these services, customer controls are also possible in Universal Description Discovery and Integration (UDDI), and platform independent Extensible Mark-up Language (XML) registry that allows registration and identification of web service apps.

Let us consider an example of Google app engine.

The platform allows developers to program apps using Google's published APIs. In this platform, Google defines the tools to be used within the development framework, the file system structure and data stores. A similar PaaS offering is given by Force.com, another vendor that is based on the Salesforce.com development platform for the latter's SaaS offerings.Force.com provides an add - on development environment.

In PaaS, note that developers can build an app with Python and Google API. Here, the PaaS vendor is the developer who offers a complete solution to the user. For instance, Google acts as a PaaS vendor and offers web service apps to users. Other examples are : Google Earth, Google Maps, Gmail, etc.

PaaS has a few disadvantages. It locks the developer and the PaaS platform in a solution specific to a platform vendor. For example, an application developed in Python using Google API on Google App Engine might work only in that environment.

PaaS is also useful in the following situations :

- When the application must be portable.
- When proprietary programming languages are used.
- When there is a need for custom hardware and software.

Major PaaS applications include software development projects where developers and users collaborate to develop applications and automate testing services.

3.4.2.1 Power of PaaS

PaaS offers promising services and continues to offer a growing list of benefits. The following are some standard features that come with a PaaS solution :

- **Source code development :** PaaS solutions provide the users with a wide range of language choices including stalwarts such as Java, Perl, PHP, Python and Ruby.
- **Websites :** PaaS solutions provide environments for creating, running and debugging complete websites, including user interfaces, databases, privacy and security tools. In addition, foundational tools are also available to help developers update and deliver new web applications to meet the fast-changing needs and requirements of their user communities.
- **Developer sandboxes :** PaaS also provides dedicated "sandbox" areas for developers to check how snippets of a code perform prior to a more formal test. Sandboxes help the developers to refine their code quickly and provide an area where other programmers can view a project, offer additional ideas and suggest changes or fixes to bugs.

The advantages of PaaS go beyond relieving the overheads of managing servers, operating systems, and development frameworks. PaaS resources can be provisioned and scaled quickly, within days or even minutes. This is because the organisation does not have host any infrastructure on premises. In fact, PaaS also may help organisations reduce costs with its multitenancy model of cloud computing allowing multiple entities to share the same IT resources. Interestingly, the costs are predictable because the fees are pre-negotiating every month.

The following boosting features can empower a developer's productivity, if efficiently implemented on a PaaS site :

- **Fast deployment :** For organisations whose developers are geographically scattered, seamless access and fast deployment are important.
- **Integrated Development Environment (IDE) :** PaaS must provide the developers with Internet - based development environment based on a variety of languages, such as Java, Python, Perl, Ruby etc., for scripting, testing and debugging their applications.
- **Database :** Developers must be provided with access to data and databases. PaaS must provision services such as accessing, modifying and deleting data.
- **Identity management :** Some mechanism for authentication management must be provided by PaaS. Each user must have a certain set of permissions with the administrator having the right to grant or revoke permissions.
- **Integration :** Leading PaaS vendors, such as Amazon, Google App Engine, or Force.com provide integration with external or web-based databased and services. This is important to ensure compatibility.

- **Logs :** PaaS must provide APIs to open and close log files, write and examine log entries and send alerts for certain events. This is a basic requirement of application developers irrespective of their projects.
- **Caching :** This feature can greatly boost application performance. PaaS must make available a tool for developers to send a resource to cache and to flush the cache.

3.4.2.2 Complications with PaaS

PaaS can significantly affect an application's performance, availability and flexibility. However, there are critical issues to consider. The following are some of the complications or issues of using PaaS :

Interoperability : PaaS works best on each provider's own cloud platform, allowing customers to make the most value out of the service. But the risk here is that the customisations or applications developed in one vendor's cloud environment may not be compatible with another vendor and hence not necessarily migrate easily to it.

Although most of the times customers agree with being hooked up to a single vendor, this may not be the situation every time. Users may want to keep their options open. In this situation, developers can opt for open - source solutions. Open - source PaaS provides elasticity by revealing the underlying code and the ability to install the PaaS solution on any infrastructure. The disadvantage of using an open source version of PaaS is that certain benefits of an integrated platform are lost.

Compatibility : Most businesses have a restricted set of programming languages, architectural frameworks and databases that they deploy. It is thus important to make sure that the vendor you choose supports the same technologies. For example, if you are strongly dedicated to a .NET architecture, then you must select a vendor with native .NET support. Likewise, database support is critical to performance and minimising complexity.

Vulnerability and security : Multitenancy lets users to be spread over interconnected hosts. The providers must take adequate security measures in order to protect these vulnerable hosts from attacks, so that an attacker is not able to easily access the resources of host and also tenant objects.

Providers have the ability to access and modify user objects/systems. The following are the three ways by which security of an object can be breached in PaaS systems :

- A provider may access any user object that resides on its hosts. This type of attack is inevitable but can be avoided to some extent by trusted relations between the user and the provider.

- Co-tenants, who share the same resources, may mutually attack each other's objects.
- Third parties may attack a user object. Objects need to securely code themselves to defend themselves.
- Cryptographic methods namely, symmetric and asymmetric encryption, hashing and signatures are the solution for object vulnerability. It is the responsibility of the providers to protect the integrity and privacy of user objects on a host.

Vendor lock-in : Pertaining to the lack of standardisation, vendor lock-in becomes a key barrier that stops users from migrating to cloud services. Technology related solutions are being built to tackle this problem of vendor lock-in. Most customers are unaware of the terms and conditions of the providers that prevent interoperability and portability of applications. A number of strategies are proposed on how to avoid/lessen lock-in risks before adopting cloud computing.

Lock-in issues arise when a company decides to change cloud providers but is unable to migrate its applications or data to a different vendor. This heterogeneity of cloud semantics creates technical incompatibility, which in turn leads to interoperability and portability challenges. This makes interoperation, collaboration, portability and manageability of data and services a very complex task.

3.4.3 Software as a Service

Software-as-a-Service is specifically designed for on demand applications or software delivery to the cloud users. It gives remote access to softwares that resides on cloud server not on the user's device. Therefore, user does not need to install required softwares in their local device as they are provided remotely through network. The consumer of a SaaS application only requires thin client software such as a web browser to access the cloud-hosted application. This reduces the hardware requirements for end-users and allows for centralized control, deployment and maintenance of the software.

SaaS provides a model for complete infrastructure. It is viewed as a complete cloud model where hardware, software and the solution, all are provided as a complete service. You can denote SaaS as software deployed on the cloud or on a hosted service accessed through a browser, from anywhere over the internet. The user accesses the software, but all the other aspects of the service are abstracted away from the user. Some examples of popular SaaS applications are Google Docs, Hotmail, Salesforce and Gmail. The structure of the SaaS system is illustrated in Fig. 3.4.5.

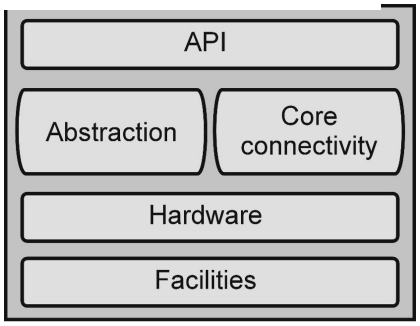

Fig. 3.4.5 : Structure of SaaS

SaaS provides the capability to use applications supplied by the service provider but does not follow control of platform or the infrastructure. Most of the users are familiar with SaaS systems because they offer a substitute for local software. Examples are : Google Calendar, Zoho Office Suite, Google Gmail.

SaaS applications come in a variety of applications to include custom software such as CRM applications, Helpdesk applications, HR applications, billing and invoicing applications and so on. SaaS applications may not be fully customisable, but there are many applications that provide APIs for developers to create customised applications.

The APIs allow modifications to the security model, data schema, workflow characteristics and other functionalities of services as experienced by the user. Few examples of SaaS platform enabled by APIs include Salesforce.com, Quicken.com and others. SaaS apps are delivered by CSPs. This further implies that the user does not have a hand in infrastructure management or individual app capabilities. Rather the SaaS apps can be accessed over a thin client web interface. SaaS provides the following services :

- Enterprise - level services
- Web 2.0 applications including social networking, blogs, wiki servers, portal services, metadata management and so on.

Some of the common characteristics found in SaaS applications are as follows :

- Applications deployed on SaaS are available over the internet and can be accessed from any location.
- Software can be licensed based on subscriptions or billed based on usage, usually on a recurring basis.
- The vendor monitors and maintains the software and the service.
- SaaS applications are cheaper because they reduce the cost of distribution and maintenance. End - user costs are also reduced significantly.
- SaaS enables faster rollout, as features such as automatic rollouts, upgrades, patch management and other tasks are easier to implement from a centralised system.
- SaaS applications can scale up or scale down based on demand and they have lower barrier entry compared to their locally installed competitors.
- All SaaS users can have the same version of the software, and hence the issue of