According to the NIST reference model in Figure 1.2 [260], the entities involved in cloud computing

are the service consumer, the entity that maintains a business relationship with and uses service from

service providers; the service provider, the entity responsible for making a service available to service

consumers; the carrier, the intermediary that provides connectivity and transport of cloud services

between providers and consumers; the broker, an entity that manages the use, performance, and delivery

of cloud services and negotiates relationships between providers and consumers; and the auditor, a party

that can conduct independent assessment of cloud services, information system operations, performance,

and security of the cloud implementation. An audit is a systematic evaluation of a cloud system that

measures how well it conforms to a set of established criteria. For example, a security audit evaluates cloud security, a privacy-impact audit evaluates cloud privacy assurance, and a performance audit

evaluates cloud performance.

We start with the observation that it is difficult to distinguish the services associated with cloud

computing from those that any computer operations center would include [332]. Many of the services

discussed in this section could be provided by a cloud architecture, but note that they are available in

noncloud architectures as well.

Figure 1.3 presents the structure of the three delivery models, SaaS, PaaS, and IaaS, according to

the Cloud Security Alliance [98].

Software-as-a-Service (SaaS) gives the capability to use applications supplied by the service provider

in a cloud infrastructure. The applications are accessible from various client devices through a thin-client interface such as aWeb browser (e.g.,Web-based email). The user does not manage or control the underlying

cloud infrastructure, including network, servers, operating systems, storage, or even individual

application capabilities, with the possible exception of limited user-specific application configuration

settings. Services offered include:

• Enterprise services such as workflow management, groupware and collaborative, supply chain,

communications, digital signature, customer relationship management (CRM), desktop software,

financial management, geo-spatial, and search [32].

• Web 2.0 applications such as metadata management, social networking, blogs, wiki services, and

portal services.

The SaaS is not suitable for applications that require real-time response or those for which data is

not allowed to be hosted externally. The most likely candidates for SaaS are applications for which:

• Many competitors use the same product, such as email.

• Periodically there is a significant peak in demand, such as billing and payroll.

• There is a need for Web or mobile access, such as mobile sales management software.

• There is only a short-term need, such as collaborative software for a project.

Platform-as-a-Service (PaaS) gives the capability to deploy consumer-created or acquired applications

using programming languages and tools supported by the provider. The user does not manage

or control the underlying cloud infrastructure, including network, servers, operating systems, or storage.

The user has control over the deployed applications and, possibly, over the application hosting

environment configurations. Such services include session management, device integration, sandboxes,

instrumentation and testing, contentsmanagement, knowledge management, and UniversalDescription,

Discovery, and Integration (UDDI), a platform-independent Extensible Markup Language (XML)-based

registry providing a mechanism to register and locate Web service applications.

PaaS is not particulary useful when the application must be portable, when proprietary programming

languages are used, or when the underlaying hardware and software must be customized to improve

the performance of the application. The major PaaS application areas are in software development

where multiple developers and users collaborate and the deployment and testing services should be

automated.

Infrastructure-as-a-Service (IaaS) is the capability to provision processing, storage, networks, and

other fundamental computing resources; the consumer is able to deploy and run arbitrary software,

which can include operating systems and applications. The consumer does not manage or control the

underlying cloud infrastructure but has control over operating systems, storage, deployed applications,

and possibly limited control of some networking components, such as host firewalls. Services offered

by this delivery model include: server hosting, Web servers, storage, computing hardware, operating

systems, virtual instances, load balancing, Internet access, and bandwidth provisioning.

The IaaS cloud computing delivery model has a number of characteristics, such as the fact that the

resources are distributed and support dynamic scaling, it is based on a utility pricing model and variable

cost, and the hardware is shared among multiple users. This cloud computingmodel is particulary useful

when the demand is volatile and a new business needs computing resources and does not want to invest

in a computing infrastructure or when an organization is expanding rapidly.

A number of activities are necessary to support the three delivery models; they include:

1. Service management and provisioning, including virtualization, service provisioning, call center,

operations management, systems management, QoS management, billing and accounting, asset

management, SLA management, technical support, and backups.

2. Security management, including ID and authentication, certification and accreditation, intrusion

prevention, intrusion detection, virus protection, cryptography, physical security, incident response,

access control, audit and trails, and firewalls.

3. Customer services such as customer assistance and online help, subscriptions, business intelligence,

reporting, customer preferences, and personalization.

4. Integration services, including data management and development.

This list shows that a service-oriented architecture involves multiple subsystems and complex interactions

among these subsystems. Individual subsystems can be layered; for example, in Figure 1.2 we

see that the service layer sits on top of a resource abstraction layer, which controls the physical resource

layer.