CHAPTER

3

Cloud as a Service

CONTENTS

3.1 INTRODUCTION

In today's economy, many businesses face the challenge of 'taking cost out' of their infrastructure while continuing to deliver new, innovative business services – basically they need to 'do more with less'. Many businesses face the necessity to bring fast changes to their IT infrastructure to manage requests during peak times. In order to overcome this situation, organizations are dealing with IT resource optimization and lowering the IT cost, and are looking for a way to manage these resources to meet such requirements. They are also trying to add rental-style capability to the IT resource usage.

Cloud computing defines a new way to manage IT resources by enabling self-service provisioning of IT resources, metering-style accounting based on use/time, the automation of IT management in a standard process environment.

Cloud computing is an emerging style of computing where user experience is utmost important while adopting a correct business model. Cloud computing is offered with compute and application in an appropriate service model over the Internet or a network. Cloud computing is based on infrastructure management coupled with platform development and is achieved by maintaining massive number of servers in a virtualized fashion from the same pool to deliver the services.

This chapter visualizes different cloud models with respect to services. The chapter also takes into account what service is all about and different types of infrastructure services that can be offered on the cloud as a service.

Common attributes of a cloud infrastructure are defined as follows:

1. **Flexible pricing:** It means utility pricing, variable payments, and pay-by-consumption; subscription models make pricing of IT services more flexible.
2. **Elastic scaling:** It means resources scale up and down by large factors as the demand changes.
3. **Rapid provisioning:** It means IT, and network capacity and capabilities are – ideally automatically – rapidly provisioned using Internet standards without transferring the ownership of resources.
4. **Standardized offerings:** It means uniform offerings readily available from a services catalog on a metered basis.

Cost optimization can be done in two ways, that is, by controlling operating and capital expenses. The problem is not to minimize the cost but to manage it for maintaining an equilibrium between the two expenses.

When you look at different cloud types, the common terminology that comes up is 'as a service', with infrastructure as a service being the most basic type of service. It includes compute power, storage, and file systems as services. At the next level, you have platform as a service, where a compute platform or middleware is provided.

At the next level, we have software as a service. Here the software capability is offered in a package – such as customer relationship management or e-commerce – and is delivered as a service. At the highest level is business process as a service. This is where a business can take a function that is considered to be a commodity, and not a differentiator, and can be completely

outsourced in a service-based model. So cloud computing has really taken a hold because of the fact that you take virtualization, standardization, and automation, and drive this automated delivery of services at a reduced cost.

Cloud computing introduces the concept of 'IT-as-a-Service'. To support this service, the cloud infrastructure must perform the following tasks:

1. **Abstraction:** Alleviate IT consumers from the operations of applications and allowing the end-users to focus instead on the execution of high-value activities.
2. **Virtualization:** On-demand access to services irrespective of location and resource availability.
3. **Dynamic workload management:** Workload placement, provisioning, and de-provisioning based on available resources and capacity management.
4. **Data management:** Fast, secure, reliable data access and mobility with integrated data protection and recovery management.

Because all the data reside on the same shared storage systems, effective and efficient data storage management becomes critical in a cloud deployment.

3.2 GAMUT OF CLOUD SOLUTIONS

There are a variety of cloud offerings in the broad spectrum of services (Fig. 3.1). Below are the common types under which most of the services can be viewed:

1. **Platform-as-a-Service (PaaS):** This refers to provisioning the developer's platform who writes custom application using the hardware, operating system (OS), database, and middleware. There are some software policies using which developers write and balance the scalability for application development.
2. **Software-as-a-Service (SaaS):** This refers to provisioning a purpose-built software over the Internet that uses underlying hardware, OS, and distribution capabilities.
3. **Infrastructure-as-a-Service (IaaS):** This is the provisioning of hardware or virtual computers where the organization has control over the OS, thereby allowing the execution of arbitrary software.
4. **Storage-as-a-Service (StaaS):** This is the provisioning of database-like services, billed on utility computing basis, for example, per gigabyte per month.
5. **Desktop-as-a-Service (DaaS):** This is the provisioning of desktop environment either within a browser or as a terminal server.

The distinction between these five categories of cloud offering is not necessarily clear cut. In particular, the transition from Infrastructure-as-a-Service to Platform-as-a-Service is a very gradual one.

3.2.1 *Platform-as-a-Service (PaaS)*

Instead of just offering applications over the Web in the form of SaaS, public cloud players are actually offering an entire PaaS. They provide the foundation to build highly scalable and robust Web-based applications in the same way the traditional operating systems such as

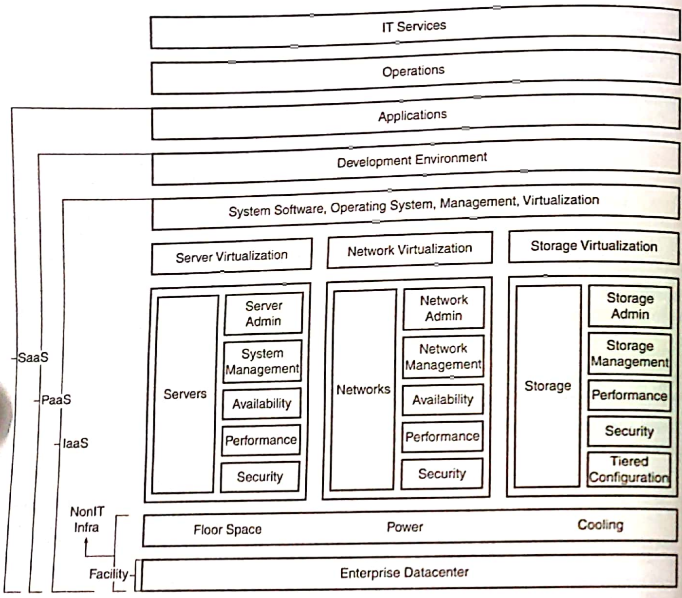

FIGURE 3.1 Cloud@datacenter.

Windows and Linux have done in the past for software developers. What is very different about this model is that the platform itself is no longer 'sold' to the customer who is then responsible for running and maintaining it. This model becomes more attractive with its operation on the hosting platform capabilities and with more capabilities in self-service model and billing. So PaaS has far-reaching implications on the business models of both PaaS vendors and their customers. One can use private clouds to speed up application deployments within minutes. It will help in tracking the usage for the chargeback and will give the option of cost-effective and secure appliance.

In order to optimize deployments, many organizations are looking to extend SOA to cloud services.

Cloud capabilities can improve the productivity of development and testing teams to roll out new applications and SOA services faster, and reduce the application backlog. The

cloud provides a catalog of virtual images and patterns all ready for immediate use and these patterns define a cluster of servers working together.

PaaS saves costs by reducing upfront software licensing and infrastructure costs as well as ongoing operational costs for development, testing, and hosting environments.

PaaS enhances the capability of development platforms and helps to remove the problem of integrating the development software and tools such as OS, database, middleware, security mechanism, frameworks, and virtually shared infrastructure. PaaS also reduces the cycle time of delivery and development iterations by providing development, testing, QA, and staging on the same platform. It brings collaboration between the developers and operation team. It actually helps to achieve bigger goals of project management and remove the gaps between the development and operation communities.

Currently, PaaS is in the maturity stage. There is a requirement to bundle the application services with the platform to make it more portable and create a standard platform across vendors to choose the services. The collaboration and component models should be adopted by all vendors and should support the legacy application on the same platform for the same cloud-based services.

There are different set of users for SaaS, PaaS, and IaaS. SaaS looks for the users who want to simplify the provisioning process of business services using hosted applications. PaaS is meant for the development community to support rapid development, testing, and delivery. IaaS is a managed services platform where infrastructure is provisioned to provide compute, storage, and network services.

3.2.2 *Software-as-a-Service (SaaS)*

SaaS reduces the investment by removing development effort, software licenses configuration, installation maintenance, infrastructure cost, delivery and release, and reducing operational cost and administration.

The SaaS platform takes lesser time to build the service compared to the traditional way of doing it. It becomes easier as all the tedious work of management, software maintenance, and support is passed to a vendor. This gives time to workers to focus more on the business' value-based features.

SaaS application does not need extensive customization and should not look for complex business-process applications. So, it is not well-suited for service-based deployment.

There are some issues in SaaS related to migration also. While moving from one on-premise network infrastructure to publically available network infrastructure, SaaS requires an upgrade in the infrastructure. It also requires upgradation of software versions and this can become a big problem for the vendors offering these services.

3.2.3 *Infrastructure-as-a-Service (IaaS)*

IaaS eliminates the cost by handling the provisioning process with the automation during the peak loads. So IaaS helps the server to flex-up and flex-down to meet the demands by reducing the operation cost and optimizing the support, administration, and maintenance cost. Organizations can manage their infrastructure requirements without increasing their infrastructure footprint. This also helps to control the resources required to manage the environment (Fig. 3.2).

The biggest drawback with the infrastructure provisioning is lengthy procurement cycles. IaaS helps to reduce the duration of these cycles from months to minutes. It is like shopping in an online store where one submits a request, gets it approved, and receives the services at his doorstep. These platforms follow open architecture with diverse set of supported operating systems, hypervisors, frameworks, and middlewares.

These compute resources are available on lease on the basis of pay-as-you-go model, typically based on the usage in terms of days, hours, and even in minutes. The application that needs 100% availability may not be the perfect suit for these types of services, but today even this can be managed as cloud service provides these services with the help of high availability and disaster recovery mechanisms. This infrastructure provides scaling options and therefore it is recommended to have the architecture and design of applications which can support scaling. There is no proper standardization mechanism currently for the integration and deployment of third-party software applications. This will be done in the coming years when all independent software vendors (ISVs) and cloud service providers will have a common understanding of standard software licenses and APIs to talk to diverse set of interfaces.

SaaS is considered to be more mature as a cloud offering than PaaS or IaaS. Even then, mainly small and medium businesses have adopted cloud services. Adoption by larger enterprises is

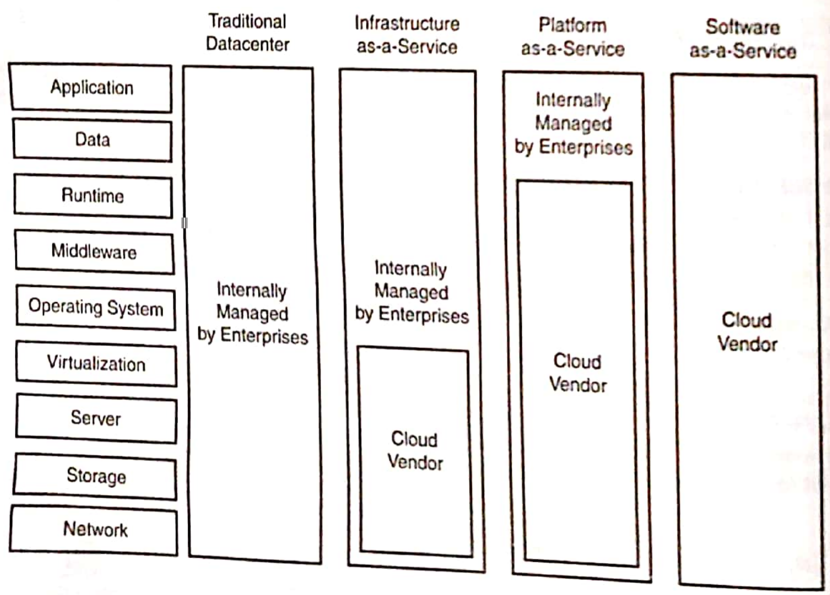

FIGURE 3.2 Cloud taxonomy.

still extremely low. A perception of cloud services as a high-risk technology option has led the large organizations to restrict the use of cloud services to a limited number of projects. PaaS is a more sophisticated service platform, and is still an emerging product. It needs to stabilize and mature before developers can use it for the extensive development of new SaaS applications. For IaaS, the entry of large vendors, such as Amazon and other cloud vendors, is driving up the maturity of the offering rather quickly, with Amazon leading the efforts to define the IaaS market.

There are now hundreds of vendors offering some new services of cloud computing. Other vendors have attached themselves to the 'Cloud bandwagon' by providing ancillary services to available offerings.

3.3 PRINCIPAL TECHNOLOGIES

The key to provide a dynamic cloud infrastructure is the virtualization layer that sits between the cloud instances and the physical hardware it runs on. The platform virtualization software – the hypervisor – allows multiple operating system instances to run as guests on the same server.

The main drivers for cloud computing are *cost*, *agility*, and *time to market*. By building cloud infrastructures using a cloud orchestrator and a provisioning engine, one can realize cost savings and improve time to market. The cloud orchestrator and a provisioning engine sit on top of the virtualization layer working on the network, server, and storage. It is a layer of software that (a) interacts with multiple servers, (b) enables IT departments to pool resources together across servers, and (c) defines standardized tiers of services called *virtual compute centers*. This helps break down infrastructure silos and drives the sharing of infrastructure.

Using the cloud orchestrator and provisioning engine, IT departments can define organizations and on-board users that can share the underlying cloud infrastructure in a secure multi-tenant environment. It can then create standardized collections of virtual machines (VMs) and set policies on how users can use these VMs. The users can login into the orchestrator and self-provision respective workloads which IT has setup already.

These orchestrators also transport with the API, which allows the cloud administrator and users to inter-relate with the cloud infrastructure in a systematic way. In addition, the cloud orchestrator and the provisioning engine allow writing workflows to automate the creation of the cloud infrastructure.

The increased pooling and sharing of resources, self-provisioning, and increased automation deliver greater savings in agility, IT infrastructure, and faster delivery of applications. So one can deliver the cloud benefits for today's applications and for the applications which will be developed in the future as the cloud orchestrator and the provisioning engine build the top of the hypervisor layer.

Virtualization is the foundation for the cloud. It consists of physical hardware with hypervisor layered on the top of it. The cloud orchestrator and the provisioning engine consist of one or more cells that communicate with a single database and offer a Web portal. Using this

Web portal, cloud administrators create the cloud infrastructure resources and the users' self-provision the cloud infrastructure resources in a secure multi-tenant fashion, thus enabling IaaS.

Chargeback and metering are a key area of on-premise cloud solution. In this process, the server talks to

(a) own database, (b) server database, and (c) cloud orchestrator and provisioning engine databases and allows to associate costs with the cloud and generate usage and billing reports. The server also needs to integrate with workflow systems, Lightweight Directory Access Protocol (LDAPs), approval process, etc. to provide the lifecycle management of the cloud environment.

3.4 CLOUD STRATEGY

This section provides a high-level guidance to define the cloud strategy and the artefacts that capture the architecture of a cloud-enabled application. These architectural artefacts are meant for the implementation planning phase of the cloud for enabling an application. Only the high-level architecture of the system that is to be cloud-enabled is captured in these artefacts.

The implementation planning phase of the cloud enables an application lying between the business strategy definition for the adoption of the cloud and the design, development, and implementation phases of the application that are being implemented to be offered on the cloud platform. This phase also takes care of linking the business strategy that is defined for a business to adopt a cloud-based strategy and the IT requirements for the applications on the cloud that are needed to support this strategy. So, this critical piece of the implementation planning translates the business intent to a set of IT requirements for the cloud-based application, deriving the high-level structures of the cloud-based application and defining a roadmap for the implementation of the application.

So, the primary input for the cloud implementation planning phase comes from the cloud strategy for the business that is driving the cloud-based implementation of one or more applications on the cloud.

The key steps in the cloud implementation planning are as follows:

1. Understand the cloud strategy.
2. Define the cloud application requirements.
3. Assess the cloud readiness.
4. Define high-level cloud architecture.
5. Identify changes in management requirements.
6. Develop a roadmap and implementation plan.

Of these phases, the artefacts defined here relate to the phase of 'defining the high-level cloud architecture'. In this phase, the high-level structure of the cloud-based application is defined from the inputs of the previous step, the 'define cloud application requirements'.

The first step in the architecture development is the usage of existing asset analysis to understand which of the components, including components of the existing application, can be used for building the application and how they can be leveraged in the cloud environment. The next

step is to derive the high-level structure of the solution from the IT requirements for the cloud in the form of the following artefacts. The non-functional characteristics that the application must support and deliver are captured in the artefact 'Non-Functional Requirements'.

The roadmap for strategy and planning for cloud deployment helps in:

1. Identification of the business justification for adopting the cloud.
2. Starting the architectural work for cloud computing.
3. Application readiness and fitment based on workload.
4. Workload assessment and recommendation for the steps of migration.
5. Selection of correct solution and implementation.
6. Acquiring required skills and bridging technology gap.
7. Management of roles, responsibilities, and customer-relationship.
8. Development of high-level cloud roadmap and value proposition.

This roadmap helps in cloud deployment with the following benefits:

1. **Reduced risk and faster deployment:** It leverages cloud vendor assets, skills, and experience to reduce the risk. It also accelerates the development and implementation by identifying gaps, activities, and risks, and defines mitigation strategies within an implementation roadmap.
2. **Improved service:** It identifies the optimal delivery model, mixes and prioritizes the workloads to migrate to the cloud to achieve the business and IT objectives.
3. **Lower cost:** It identifies the opportunities to reduce capital and operating expenses across the infrastructure.

3.5 CLOUD DESIGN AND IMPLEMENTATION USING SOA

SOA is a very useful architectural style for implementing applications in the cloud. Adoption of SOA is the best way to leverage and consume the application services provided by the cloud. A cloud-based application consists of many granular coarse-grained services offered on the cloud. These cloud offerings may in turn integrate and leverage services and systems from different environments. The coarse-grained services that cloud offerings leverage can come from the traditional IT environment in the form of standard services that are already provided and are internal to the cloud.

Standardization across these different IT environments is not possible; hence they mostly consist of heterogeneous environments. So a cloud service offering should be based on open standards that can be consumed and leveraged by this environment. Language- and platform-independent services can be provided using standards-based, platform-agnostic SOA architecture with the support for appropriate industry and technology standards.

The cloud-based application services consist of a coherent set of business processes that are aligned to the business boundaries, provide a coherent, integrated set of operations that adheres to the business intent and provides value to the users. These services also comprise consumable interfaces which may be user interfaces on different devices, coarse-grained Web-service interfaces, feeds, or widgets.

The cloud application services should be mapped to the business processes that they consist of. These business processes could be at different levels. Drill down the processes to the levels that make business sense.

These processes – along with the business strategy, the modular business alignment model, and the process scenarios to be supported – form the input to Service-Oriented Modeling and Architecture (SOMA) methodology. SOMA is applied with a meet-in-the-middle approach. Business processes and the business strategy along with business goals stated are used to arrive at the service portfolio using SOMA. While deriving the services, a bottom-up approach is also followed, taking into consideration the existing assets, which consist of existing application components and services available on public clouds and internal to the cloud, as well as industry cloud component maps.

3.5.1 *Architecture Overview*

The purpose of the architecture overview artefact in a cloud-based implementation is to communicate to the sponsor and external stakeholders a conceptual understanding of the architectural goals of the cloud implementation. It offers a layered conceptual model of the application services to be cloud-enabled and provides a high-level vision of the cloud architecture and its scope to developers. It is easy to explore and evaluate alternative architectural options for the cloud implementation. This stage enables early recognition and validation of the implications of the cloud-based architectural approach and facilitates effective communication between different communities of stakeholders and developers.

This developed document will help to understand the conceptual points of the cloud implementation and its dependencies on internal or external cloud ecosystem. The document may include both the cloud and non-cloud infrastructures such as offerings, components, relationships, external systems, and elements for support systems.

This artefact can also provide key stakeholders with the first high-level view of the cloud architecture landscape of their transformed cloud-based implementation.

Typically, the artefact is produced after multiple iterations and as the project moves through the solution definition and design phases, the conceptual models get clearer and this document is kept up-to-date and governed to always have the current conceptual elements.

An architecture summary depiction represents the governing ideas and candidate building blocks of a cloud-based offering and enterprise architecture. This depiction also provides an overview of the main conceptual elements and relationships in the architecture. The main purpose of such a depiction is to communicate a simple, brief, clear, and understandable overview of the target IT system.

At the enterprise-level, the architecture summary depiction is often produced as a part of an overall IT strategy to move to a cloud-based offering model. At this instance, it is used to describe the vision of the business and IT capabilities required by an organization that needs the offering to be hosted on the cloud. The architecture summary depiction provides an overview of the main offerings and the relationships with other offerings, external systems, components, nodes, connections, users, and systems, and a definition of the key characteristics and requirements.

At the system- and component-level, the architecture summary depiction is developed very early in the project (possibly pre-proposal) and influences the initial cloud hosting vision with the component model and operational model. It is intended that design commitments should be based on this conceptual overview, while the (more detailed) component model and operational model are developed and validated. Subsequently, the component model and the operational model are the primary models used for implementation activities, while their compliance with the architecture summary depiction is maintained continually. Changes to the architecture summary depiction are made using a governance process.

For most infrastructure cloud engagements, the project scope is something less than the client's enterprise architecture. Therefore, the depiction will represent the governing ideas and candidate building blocks of the offerings that are to be hosted on the cloud, and the focus of the engagement. There can be, then, multiple views of the cloud-based IT environment represented by architecture summary depiction – the current and future view, views by waves of transformation, or views by types of clouds – for each alternative solution.

3.6 CONCEPTUAL CLOUD MODEL

So far, we have discussed different service paradigms for cloud deployment. Now let us discuss the conceptual cloud model based on cloud services.

This model gives the structure of offerings in the cloud on the basis of the components of hardware and software with their relevant roles, interdependencies, and interfaces to provide the functionality. The cloud component model for implementation planning is specified at the conceptual level.

The highest level of the conceptual model is the set of cloud-based offerings that make up the cloud-based business solution. These highest-level conceptual components are derived on the basis of business intent and business functionality. These conceptual components, which form offerings, align tightly to the business intent that initiated the creation of cloud-based offerings.

Conceptual offerings that provide the cloud-based implementation of the solution can be further broken down and depicted in a layered composition. The next level of conceptual component model broken down below the offering consists of the following elements:

1. High-level service components that form the services provided by the offerings.
2. Resources that support cloud services.
3. Technical components that provide technical underpinnings of the cloud service and support non-functional needs of the cloud application.
4. External and internal services that are leveraged by cloud application services.

The high-level conceptual component model is the main artefact that provides an abstract view of the design of a cloud application to business stakeholders. This abstract view describes how the business needs will be met by the cloud application components without delving into their technical details. Components identified can be decomposed into further layered conceptual component structures to convey further details of the respective components (Fig. 3.3).

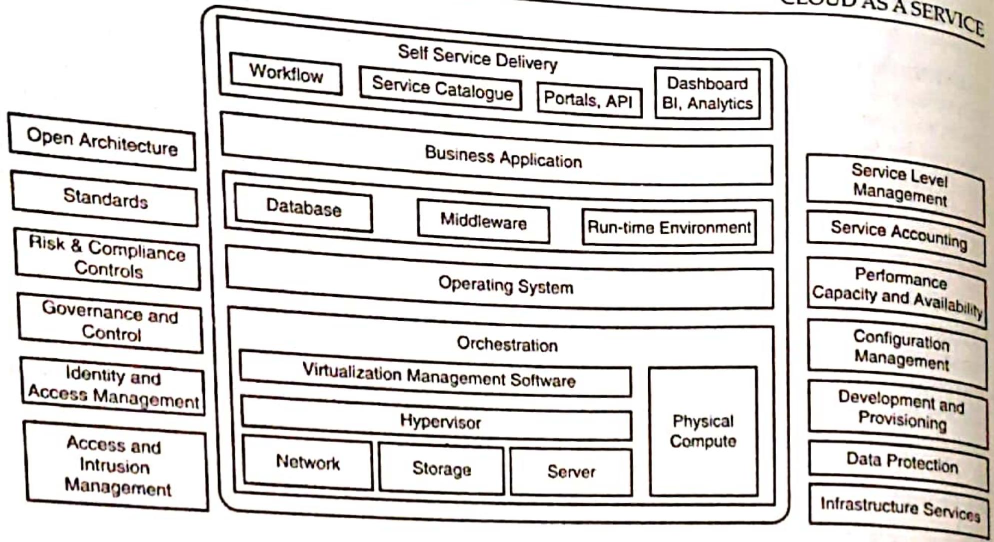

FIGURE 3.3 Cloud model.

The cloud conceptual component model should contain the following elements:

1. The conceptual structure of the cloud application.
2. Dynamic interactions and dependencies between various conceptual components.
3. Components that comprise the cloud services provided by the application, each of which may be made up of sub-components.

At this level, the conceptual components are not converted to physical components but retained at the conceptual level only.

3.6.1 *Cloud Application Security and Privacy Principles*

Cloud application security principles contain the high-level guidance to cover the security and privacy characteristics identified in the Information Asset Profile. This is a summary document of the information asset required for the architecture. Cloud application security principles deliverable contains a number of security principles that occur in a typical application. Typically, quite a number of the characteristics identified in the information asset profile effort will relate to the existing enterprise security elements of the client.

In addition to the elements from the existing enterprise security program, consider the following areas that tend to change during the transition to the cloud environment.

3.6.2 *Governance*

How will the decisions like different SLAs be made between cloud vendors and cloud customers for the cloud environment? These decisions are typically made by either another infrastructure group within the client IT organization or an external vendor who provides cloud infrastructure management for either a managed private cloud or a public cloud environment.

Authentication and Access Control

As an application moves to the cloud, the methods used to authenticate and authorize users require examination. Existing methods may require additional support to work in the cloud environment, such as providing connectivity to an existing server, and may require significant extension to accommodate the requirements of an external vendor providing infrastructure support.

Data Protection

With the move to a cloud environment, the existing methods of data protection – protecting the data from disclosure or modification both on the network and on a storage medium – are likely to require changes with the move to a virtualized storage and with the introduction of additional infrastructure administrators, often from a vendor organization.

Logging and Alerting

Logging is the ability to tie actions to an individual. Alerting is the method of recognizing activities that may indicate a malicious act and bringing those to the attention of the security staff. With the move to a cloud infrastructure, with different network and additional administrators, often from an external vendor, the process of tying actions to specific individuals requires additional attention.

3.7 CLOUD SERVICE DEFINED

This section highlights the aspects of service, its scope, and cloud-based different platform integration and deployment services.

3.7.1 *Service Definitions*

Let us begin with a cursory glance on what 'service' is:

1. **Service:** A specific IT deliverable that provides customer value. It is measurable in 'customer' terms and provides the basis for doing business with the customer. It is delivered through a series of processes and/or activities.
2. **Service portfolio:** The collection of services provided by IT in aggregation to represents all the 'value add' activities performed by IT.
3. **Service component:** A logically grouped set of activities that represent a part of a service that touches the customer. Service components are grouped together to create deliverables that form the basis for doing the business with customers.
4. **Service owner:** The individual who is accountable for ensuring that the customer receives the identified value of the service. They take customer's end-to-end view of IT activities by working with process owners to ensure that all the required delivery components fit together smoothly.
5. **Process:** A collection of related activities that take inputs, transform them, and produce outputs that support an enterprise goal.
6. **Enablers:** The decomposed components of a service (process, organization, and technology) that are combined to create *service deliverables*. The collection of activities (and their supporting roles and technology) become the workflow needed for *service delivery*.

7. **Service-level agreements:** It is a grouping of services or service components that have specific delivery commitments and roles identified with the customer. SLAs can be grouped together in different ways to represent various products. Examples would include e-mail and service delivery.
8. **Service-level management:** It is the governing mechanism for coordination, agreement, monitoring, reporting, and reviewing the steady-state operations service agreements, so that the service is maintained at a given level and with continuous improvement. It is the trust building activity between provider and consumer. This mechanism is required in any organization to maintain the IT service level needed to support the business. Service-level management also keeps a check on what is achieved or what is not, and the reason behind that.
9. **Service-level management objectives:** The objectives related to SLA maintain the quality, agreement clauses, penalties with the help of constant monitoring and reporting. This also helps to investigate the causes for not meeting the SLAs because of the poor service. It develops a better relationship between the IT department and its customers.

SLAs should be established for all IT services being provided. Underpinning Contracts (UCs) and Operational Level Agreements (OLAs) should also be in place with those suppliers (external and internal) upon whom the delivery of service is dependent.

SLAs are likely to be a service if the 'what' is separated from the 'how', and we can change the underlying assets – processes, technologies, data – as well as suppliers, but still provide the promised business outcomes and value. It is a service if its description and design highlight separate roles for customers, users, providers, and suppliers, and is offered with a pre-defined value proposition stating specific price points and performance metrics tied to business (not just IT operational) outcomes. It is something that makes sense as a customer/user-selectable item with corresponding service requests available in a menu or service catalog.

3.7.2 *Services Scope Overview*

In this section we discuss about the service scope for platform-based management operations.

Platform Development Services

These services provide a set of project services for planning, design, procurement, assembly, integration, site installation, and project management of the deployment mainstream and special-purpose end-user devices, and also include a number of asset lifecycle services. These services integrate and customize multiple devices – including PCs, wireless and mobile devices, kiosks, ATMs, point-of-sale (POS) devices, and printers – as per end-user specifications. These services follow a factory approach for off-customer-site build and integration services, using build and integration centers around the globe.

Software Platform Management Services

These services provide a set of project and annuity services to manage end-user software platforms, including image development and management, application software packaging and distribution, and services to manage the availability of the end-user platform proactively. This includes services to design and migrate end-user platforms, for example, Microsoft XP to Vista or Linux. These services follow a factory approach for off-customer-site platform management services, using management and configuration centers.

3.7.3 *Platform Integration and Deployment Component Services*

This section discusses some of the very important platform integration and deployment component services.

Order Management

This service handles the procurement of hardware and/or software on behalf of the customer (regardless of asset ownership), and the fulfilment (delivery) of that hardware and software to a central build facility prior to platform build and pre-load.

Warehousing and Stock Management

This service gives provision of central warehousing facilities to store hardware and other agreed components before and after build, and before the shipment to the site. It provides services for receiving and warehousing, and ensures that the inventory is efficiently maintained and protected while the products are in the storage.

Platform Build and Test

This service provides services to build, integrate, customize, prepare, and test the hardware and software platforms before the shipment to the customer site or the end-user location.

Base Backup

This service provides a base backup to be taken during platform pre-build (assuming that the software platform supports this feature).

Data and Personality Migration

This service migrates data and personal settings (e.g., desktop wallpaper, Internet favorites, desktop layout, etc.) from the original to a replacement platform.

Asset Tagging and Custom Labelling

This service provides custom asset tagging of hardware components during build and pre-delivery preparation at a central build facility.

Asset Inventory Update and Report

This service includes adding a new asset to the customers or managed asset database.

Logistics and Delivery

This service includes the shipment of the hardware to a respective site for building the platform.

Installation

This service provides the deployment of the platform into the customer's operational environment. This could be either at the end-users' desk or at the agreed deployment center location for the machines that are to be collected or deployed by the customer.

Extended Project Management

This is an extension to base project management services that cloud uses to manage internal functions and activities. This service extends project management scope to cover overall management of the platform deployment program, including customer and customer third-party resources and activities.

Platform Removal and Return

This includes decommission of platform from the customer location and return to cloud vendors for refurbishment or disposal (but not including refurbishment or disposal activities).

Asset Refurbishment

This service checks the suitability of the hardware platform, refurbishment and upgrade as appropriate, and integration into the deployment process.

Asset and Data Disposal

This service provides removal of sensitive data from the hardware platform to varying levels of security (e.g., to the department of Defense standards), as well as safe environmental disposal to international standards and/or asset value recovery.

Emergency Replacement

This service provides the emergency replacement of hardware platforms (pre-built to the customer's standards) to agreed service levels.

Software Platform Services

This section explains the software platform management services in detail.

1. **Software platform design consulting:** This service helps the client to understand the business and technology needs for a new software platform, and provides the design specification.
2. **Software platform creation and customization:** This service creates and tests a software platform to support the needs of the business.
3. **Software support and maintenance:** This service provides ongoing support and maintenance of the client's software platform, including platform updates and management, and ongoing problem support.

Application Scripting

This service creates application software unattended installation scripts to accommodate the customer environment. This service also includes new software applications and updates for existing packages.

Application Discovery

This service discovers which applications are in use across the organization.

Application Portfolio Management

This service helps the clients to manage their application portfolio.

Software Delivery

This service schedules and transports application packages to target end points. It includes tl capability to PUSH software out to user PCs from a central distribution and/or the ability f the users to PULL software to their PCs using a Web interface.

Antivirus Management

These services manage the delivery of antivirus signature files. They provide real-time prote tion from malicious virus attacks that can cause potentially disastrous system.

Patch Management

This service ensures that end-user devices have the latest patch releases installed. This serv also provides real-time deployment of critical operating system patches for protection agai flaws and vulnerabilities.

Health Check Services

These services ensure that end-user devices are in a good state of health of the system, sucl infrastructure and PC checks. They also perform remote monitoring of supported workstation critical hardware alerts and initiate scripted responses as alerts are received. Examples of he check services are identification of low memory, detection of spyware, identify low disk space,

Compliance Services

These services ensure that end-user devices are compliant with client standards. They perl remote monitoring of supported workstations to perform activities such as detecting pee peer software detection of games. Until there is an agreement to industry standard defin of what an IT service is, one must agree with the client on a definition of 'service'. So, diff type of the services visibility with respect to the customer will be considered.

External Services

These services are visible and seen by the customer, and include business services (bus intelligence, logistics, receiving orders, marketing services, invoicing, accounting, etc.) user services (desktop support, maintenance, education, etc.).

Internal Services

These services are invisible or less visible to the customer, but essential for the delive IT services. They include infrastructure services (hosting services, storage, availability, retention, or recovery) and network services (network, remote access, mobile or wireless services, etc.). Internal services also take care of application services (integration, testing, design, maintenance, optimization, etc.).

User-Initiated Service Request

This request includes service request handling in incident management, for example, the service request progresses through its lifecycle exactly as an incident. Most companies separate user service requests and incidents, for example, a service request follows a different process and use a different tool to track it throughout its lifecycle. Users request services for which their businesses have already contracted with a service provider and to which they are already entitled. Some people have referred to the list of services from which a user can order services as a service catalog. Perhaps it is a 'User Services Catalog'. These service requests and this type of listing of services are appropriately offered through a single point of contact for the users of IT support, that is, the service desk.

Customer-Initiated Service Request – A Service Catalog-Based Request

It includes the concept of a service catalog as a list of services that the customer can order. The customer is the one who pays for services. When customers order a service, their users are entitled to receive services under that agreement. Each individual request made by a user is a user initiated service request. The user is entitled to services that their business has ordered through a service catalog request. This type of request is from the customer to some account representative or 'business relationship manager' who responds to this request and initiates the service provisioning for that customer.

3.8 SUMMARY

This chapter visualizes different cloud models with respect to services. The chapter also takes into account what service is all about and different types of infrastructure services that can be offered as the cloud as a service.

CHAPTER 4

Cloud Solutions

CONTENTS

4.1 INTRODUCTION

The cloud environment presents an opportunity to enhance user experience by providing a broader communication path for reaching out to the user or providing a series of business services to the user via application features.

Deploying an application to the cloud is somewhat different since the deployment process will not be done locally within the enterprise and the existence of the provisioned image as well as a series of deployment steps are needed to deploy the application and validate the deployment.

Development and testing environments are readily available within the cloud environment. The advantages of these environments, especially from a cost perspective, are numerous as there is no need to purchase and deploy servers within the normal enterprise environment. If a proof of concept (POC) was being developed and the project was cancelled, no software, hardware, or even development tools would have to be purchased, only to be thrown away later, as the cloud supports development and testing of applications.

4.1.1 Cloud Application Planning

Cloud application design and planning exercise requires many unique considerations, which are as follows:

1. Business functions.
2. Application architecture.
3. Security for cloud computing.
4. Cloud delivery model.
5. User experience.
6. Development, testing, and run-time environments.

Application architecture is selected through an appropriate criteria evaluation. The key thing to talk about from the security aspect is the enhancement to the existing security model where data protection and isolation of the data becomes important from other areas of the cloud environment. Encryption is one possibility to further enhance the security model; however, the enterprise would not necessarily invoke that option. Another aspect of security would be to further authenticate and authorize users of the application and services they have been entitled to use.

4.1.2 Cloud Business and Operational Support Services

The elements that are required by the cloud service provider to run the business operations are called business support services (BSS). These elements are mostly associated with service providers having a utility-based service model. These services include:

1. Accepting customer orders.
2. Managing customer data.
3. Managing order data.
4. Billing.
5. Service offered.

The IT system used by service providers comes under the category of operational support services (OSS). It is mostly associated with the network infrastructure and maintenance services such as:

1. Provisioning.
2. Installation and configuration.
3. Fault management.

BSS and OSS components need to be externalized so that the application's supporting services being transformed to the cloud environment can capitalize on the various functions provided by OSS and BSS. For example, provisioning can be adapted to support the application's provisioning requirements instead of creating self-provisioning service from scratch. The ability to tap into the monitoring and metering events and keep track of the activity within the cloud environment can assist the application to continue to provide service levels and quality of service. Each of the OSS and BSS offered by the cloud environment will continue to support the application and its consumers in maintaining the key characteristics of cloud computing.

Applications built for the cloud have a layered architecture and are composed of infrastructure, security, business services, application programming interfaces (APIs) to devise the optimal solution. The service provided in the cloud model is offered by service providers to their consumers. Typically, the services offered in this area are infrastructure, software, platform, and business processes, and they come with rapid deployment, scaling, self-service, pay per use and a shared infrastructure model.

The service provider is accountable for offering service instances of various categories to cloud service consumers. This also requires ongoing operations, with an option of self-servicing, to manage cloud service instances. All technical configurations are standardized and capsulated in a template that also describes the OSS, an artifact to describe the services with respect to the service provider and cloud services.

To fulfill these services, specific tools and software are required over the cloud layers. For infrastructure as a service, a hypervisor, cloud orchestration and a provisioning engine are needed; whereas for platform as a service, middleware and development platforms are required; similarly for software as a service, multi-tenancy–enabled distribution software is required. Based on services, various cloud service instances represent various entities.

Cloud services can be built on top of each other, for example, a software service could use a platform service or infrastructure service as its foundation, and a platform service could use an infrastructure service as its foundation. However, this is not always true because a software service could also directly be built on top of the 'traditional' infrastructure, clearly inheriting all the constraints associated with such an infrastructure. In general, the basic cloud architecture postulates to share as much resources as possible across cloud services with respect to the management platform and underlying infrastructure. However, the cloud does not require to have only one single, fully homogeneous infrastructure. Of course, this would be the ideal goal, but given different infrastructural requirements, this is not possible. For example, if a particular cloud service has very specific infrastructure needs, then this cloud service will be allowed to run on a dedicated infrastructure (e.g., the Google search engine or HPC cloud services would always run on a purpose-built physical infrastructure due to performance and efficiency reasons; however, they would not run on a virtualized compute cloud service).

When we build cloud services in a layered fashion, it is very important to understand the difference between OSS and BSS available on a common platform. These differences should be studied across multiple cloud services and their usage capabilities.

Whatever be the scenario, any cloud-based service can be treated as an OSS or BSS for cloud vendors. Therefore, a service is developed after gathering a lot of requirement data and standardization based on business decisions. It is followed by go-to-market (GTM) activities with appropriate and proven business models for the development and operations of cloud services.

4.2 CLOUD ECOSYSTEM

It is very important to understand the relationship between a cloud service and the artefacts that can be developed on the basis of and within the boundaries of an ecosystem-focused infrastructure-as-a-service (IaaS) or platform-as-a-service (PaaS) cloud service. Taking any service to a potential market requires huge budget allocation and then needs to be enabled with appropriate chargeback and metering mechanism to support the business model. At the time of development, it is difficult for developers to understand the need to make it as flexible as possible for unpredictable scenarios. A cloud ecosystem is shown in Fig. 4.1.

Interpretation	Self-Service	Transparency	Visibility	Agility	Automation
Consumer	Cost Mitigation	TCO/ROI	Managed Services	Platform Integration	SLAs Time to Market
Provider	Business Model	Value Proposition	Revenue Model	Service Accounting	Orchestration
Platform	Device	Security	QOS	Collaboration	Utilities
Infrastructure	Standards	Tools	Server	Network	Storage

FIGURE 4.1 Cloud ecosystem.

Once a service provider thinks of launching services, it calls the exercise of illustrating all the requirements of its functional and non-functional relationships. Once the ecosystem is developed, there is a limited scope of change as most of the components existing on the top layers of the cloud ecosystem are frozen. This is not a negative phenomenon as it provides stability to the system. It also helps in providing strict guidelines for the ecosystem and lays down the principles for using the services to reduce the overall cost of the system. The more time it takes to create the service instance, the less will be the chances of it being successful in the market.

Therefore, it is very important to first visualize the relationship between the business and the technical understanding and then develop the artifacts on the existing layers of the ecosystem that will focus on the service based on boundaries. Also, most of the time, cloud service

is termed as a service product with the deployment capability in various environments and comes with the benefit of choosing suitable services that fit best in the business function.

This means that an organization is not tied to one solution for cloud computing, but can rather use multiple solutions. So, depending on business needs, the application, and what the application has to offer, cloud requirements help in selecting proper cloud environments.

Cloud-based environments come handy especially when used to develop, test, and run your application for the following reasons:

1. Easy availability in your own private cloud environment or on the public cloud.
2. Rapid access to a configurable development and test environment to speed time to market.
3. A self-service portal for multi-tenancy, provisioning, and account management.
4. Pay-as-you-go pricing, with capacity and workload management.
5. Security with compliance and regulations to protect the environment.
6. A rich service catalog with images.
7. Faster provisioning to reduce the cycle time.

4.3 CLOUD BUSINESS PROCESS MANAGEMENT

Business process management (BPM) looks after business processes across the organization which may be cross-functional and directly or indirectly customer focused. BPM is implemented by the strategy based on the core business objectives and deployment of resources across the organization to create customer-centric values. It integrates all vertical-based core functions such as supply chain and product development in an integrated way within multiple line of businesses (LOBs). In this way, BPM is distinguished from other traditional approaches.

BPM is based on the principle of continuous improvement to increase value for market sustenance of the organization. It defines and aligns operations, IT, and organization. Therefore, the cloud can help in the following ways:

1. **Integration of core business:**
   - Is holistic.
   - Crosses organizational functions and boundaries (height and breadth).
   - Includes business and technology.
2. **Value-focused efficiency:**
   - Customer-centric perspective.
   - Bottom-line success.
   - Return on investment (ROI) delivery speed.
   - Performance measurement.
3. **Continuous:**
   - Based on longer period of intervals pertaining to cloud business.
   - Continual improvement.
4. **Cultural:**
   - Cultural considerations of the organization
   - Geographical and cultural factors at the time of due diligence of requirement.

4.3.1 *Identifying BPM Opportunities*

This section discusses the opportunities required for successful cloud BPM and the characteristics of cloud deployment offerings. The answers to the following questions can help you identify cloud opportunities better:

1. Are the strategic value proposition and capabilities defined for your organization?
2. How does your overall strategy drive the design and execution of your business processes? Is there a traceability of execution to goals?
3. How do you manage your core business processes?
4. What are your current process initiatives?
5. What are your current process governance facilities?
6. Are your existing organizational structures aligned to enable efficient process operations?
7. How do your customers measure and assess the performance of your processes?
8. How does your process performance compare with that of your competitors?
9. How effectively does your current technology (information, systems, tools, machines, etc.) enable the enterprise's core business processes?
10. What risks and challenges does your current technology present for current and future process capabilities?
11. What type of products does your organization have?
12. What are the notable pieces of your IT portfolio?
13. Has your organization adopted service-oriented architecture (SOA)?
14. How are the processes currently modelled in your organization? What is included in this model?
15. What design/development tools are currently used in the organization?
16. What testing tools are currently used by the organization? What are the strengths/weaknesses of these tools?
17. Please describe the different business processes that are automated in the organization.

Cloud application development offerings provide

1. Cloud application reference architecture.
2. Unmatched experience of developing high-performing, secure applications across a wide range of technologies of the cloud vendor.
3. Unmatched application security expertise.
4. Leadership in cloud-related technologies such as multi-tenancy, virtualization, and pervasive computing.
5. Significant expertise with cloud business models.
6. Ability to integrate a portfolio of related cloud services.

4.3.2 *Cloud Technical Strategy*

This section describes the technical strategy on how cloud customers can enable cloud deployment.

Cloud services enable our cloud users to build middleware clouds in their datacenter and utilize public clouds, where it makes sense to provide the following cloud-enabled middleware services:

1. Infrastructure services.
2. Platform services.
3. Application services.

Cloud strategy enables our organizations to do the following:

1. Build middleware clouds in their datacenter.
2. Utilize public clouds where it makes sense.

4.3.3 *Cloud Strategy Support Areas*

The cloud strategy does so by providing support in the following areas:

1. **Cloud-enabled middleware services:**
   - Infrastructure services.
   - Platform services.
   - Application services.
   - Serving on-premise and public clouds.

So, what does it mean to develop an application service for the cloud? For one, it means product features development similar to on-premise software. This means:

1. **Enabling the software for cloud essentially implies**
   - Support for collaborative multi-tenancy.
   - Self-service registration.
   - Customers and their entitlements management.
   - Single sign-on.
   - Additional security concerns.
2. **Integration with the datacenter:**
   - Firewalls and reverse proxy configurations.
   - Certificate and domain names.
   - Management of services and patch procedures.
   - Isolation, recovery, and backup issues.

4.3.4 *Cloud Use Cases*

IaaS or Test/Development

Problem: Development teams require unpredictable amount of infrastructural facilities to get their job done. In the majority of cases, getting all of these resources in place before they are actually required in the development cycle can be quite a challenge. Purchasing the hardware consumes project budget, and procurement of it is often quite slow. Static development and testing resources require manual re-provisioning to re-purpose resources for use, or new resources need to be purchased to meet demand. In cases where project timelines are short, project managers often choose not to set up much of an environment because it depletes their budget or jeopardizes the project's delivery schedule. Actual usage of the system(s) can be quite short in terms of absolute time. Different types of projects require different kinds of development components (such as SQL server, SharePoint, and BizTalk) depending on the architecture of the solution. Besides the testing environment, the team usually needs a system that looks quite similar to the production environment. This system helps to perform various simulations for load, stress testing for various deliveries and training.

Solution: Companies can create standardized service catalog items for common infrastructure requirements and enable development and test teams to access infrastructure in a self-service model (IaaS). Overall control over the process is maintained with business policies around quotas, reservations, reclamation, and standardized offerings.

Standardized Development Platforms/Middleware (PaaS Enabled)

Problem: Developers deliver their code without involving operations into architectural decisions or code reviews. Enterprise architects recognize the large costs associated with non-standard development platforms. To simplify the ongoing maintenance and streamline development operations, many companies are creating corporate standards around development stacks that include middleware/applications. However, most companies are hesitant to an external PaaS offering due to the constraints of having to rewrite their internal applications to fit the external PaaS API sets.

Solution: Companies can create standardized development platform definitions for use by development teams to standardize and streamline their efforts. This improves corporate IT productivity by helping them build a 'private cloud' that provides a common foundation for building custom applications running securely behind their own firewall.

Application Cloud

Problem: Companies want to move beyond self-service for infrastructure and provide application owners the ability to define, instantiate, and manage complex multi-tier applications. This includes configuring the application for production usage and monitoring performance within the application for Service-Level Agreement (SLA) optimization.

Solution: End-users can access complete application definitions and manage them according to their quotas and preferences defined by cloud administrators. Applications in production can be monitored across multiple factors and automatically scaled up or down according to business policies.

Software-as-a-Service (SaaS) to End Customers

Problem: Many companies or independent software vendor (ISVs) want to deliver their applications to end-users as a service. However, creating a multi-tenant SaaS offering requires substantial development to support security, performance, and scalability needs. Due to these high costs, companies cannot offer new services based upon existing applications.

Solution: Companies can provision a unique application instance per customer, with private cloud automation capabilities. Environments are provisioned according to business policies, and unique SLAs can be delivered per customer according to the business arrangements.

The cloud engine should provide the automation to provision on-demand complex application and configuration environments required along with the dynamic application scaling. The cloud engine should also deliver unique business policy selections, including custom placement and high availability across multiple datacenters.

4.4 CLOUD SERVICE MANAGEMENT

The cloud service management system offers standardization, operation control, support desk, monitoring, and consistent service delivery required for public, private, and hybrid cloud platforms.

1. **Simplify user interaction with IT:**
   - Self-service portals speed up the delivery consistently.
   - Service catalog enables standards which drive consistent service delivery.
2. **Enable policies to lower the cost with provisioning:**
   - Automated provisioning and de-provisioning speed up service delivery.
   - Provisioning and de-commissioning of assets.
3. **Enhance system administrator efficiency:**
   - Migration from various management isolations to full-fledged service management system.

The emergence of cloud deployment is prompting enterprises to either assemble in-house teams to manage specialized cloud service providers or look for third-party cloud brokers chiefly due to the following reasons:

1. Every service-oriented approach needs a mechanism to enable discovery and end-point resolution.
2. Registry/repository technology provides specialized cloud service and third-party cloud brokers where the service delivery is inside the firewall.
3. Cloud services delivered across firewalls need something similar – a third party that serves as a 'service broker'.

Leveraging service brokers will probably become a critical success factor in cloud computing as cloud services multiply and expand faster than the ability of cloud consumers to manage or govern them. The growth of service brokerage businesses will increase the ability of cloud consumers to use services in a trustworthy manner. Cloud service providers are expected to begin to partner with cloud brokers to ensure that they can deliver the services they promote. These brokers help companies to get the right platform and deploy applications among multiple service providers as well as arbitrage services to migrate and move from one platform to the other based on the best pricing model. There can be three categories of opportunities fo cloud brokers:

1. **Cloud service intermediation:** This requires the right cloud platform and becomes th basis for building new services.
2. **Cloud aggregation:** Deploying customer services over multiple cloud platforms.
3. **Cloud service arbitrage:** Supplying flexibility, providing 'opportunistic choices', an fostering competition between clouds.

It will be similar to cloud services under one umbrella except that services being aggregate will not be fixed. This flexibility will be important while doing chores such as providing mul tiple e-mail services through one service provider or providing a credit-scoring service tha checks multiple scoring agencies and then selects the best score.

The ability to federate an application across multiple clouds will become important as in the case when one service goes down, another can be started and the service broker will just simplify it. To help federate the clouds, a 'storefront' site can be created with services that are pre-screened to meet government procurement guidelines. The new site can be expected to cut red tape and make it easier for government agencies to quickly deploy the latest technology.

4.4.1 *Key Cloud Solution Characteristics*

The key characteristic capabilities of cloud orchestrator and provisioning engine are as follows:

1. **Scalability:** The cloud orchestrator should maintain an index of the resources that are acquired from the hypervisor, giving the master a low overhead and enabling it to scale across tens of thousands of machines across multiple geographies.
2. **High availability:** The cloud orchestrator should play for the master node to sup- 'Active–Passive' as well as 'Active–Active' scenarios for availability and disaster port recovery (DR). It should also monitor individual physical servers for availability, and in case of a physical resource server failure, it should restart the virtual machine (VM) on another running server to meet the requirements.
3. **Application life cycle:** The cloud orchestrator should offer complete application life cycle support, from the creation of infrastructure to installation, configuration, and launching an application to its deletion or expiration. This allows applications to be instantiated, removed, or flexed very quickly to respond to real-time demand for the same.
4. **Multi-tenancy/Role-based administration:** The cloud orchestrator should also sup- port multi-tenant capability with specific user permissions. It should have multiple personas such as cloud administrator, account owner, and user. Application definitions are only 'published' to specific users. The application owner or administrator logs in with his/her credentials and can view (user) the application VMs that have been allo- cated to him/her as well as perform admin operations (account owner) on those VMs if needed. Role-based administration allows fine-grained control of what each person can or cannot do in terms of cloud orchestrator features.
5. **Policies:** The cloud orchestrator should provide a rich set of policies that can be enabled. These policies can be modified or new ones can be created to take effect at the global level on applications, VMs, hosts, etc. These policies can also be embedded in service definitions to take effect of automatically depending on the metric thresh- old. For example, a policy can be created to allow an application to flex up to 10 VMs during high load or demand and to be reduced to only 2 running VMs during low load or demand. This policy frees up resources which can be used by other applications that are experiencing high load.
6. **Alarms:** The cloud orchestrator should provide pre-defined alarms that can be set at the global level for applications, VMs, hosts, etc. These alarms can be used to notify individual users or application owners regarding the application thresh- olds being reached. For example, an alert can be sent if the response time of an application is below the SLA but there are no more resources for the application to flex up.

7. **Application awareness and policy-based allocation:** The cloud orchestrator should be aware of application requirements and optimize the placement of the application accordingly, such as placing the VMs running the application close to each other to reduce latency. Cloud orchestrator should also support major application servers.
8. **Resource awareness and policy-based allocation:** The cloud orchestrator should opti- mize the usage of the cloud infrastructure through intelligent resource allocation poli- cies and allow load balancing of VMs.
9. **Elasticity based on performance (flex-up/flex-down):** The cloud orchestrator should provide out-of-box functionality to flex-up or flex-down an application instance or resource based on performance metrics.
10. **Reporting and accounting:** The cloud orchestrator should also provide metering and billing reports on resource allocation and actual usage. Additionally, this data can be used to create reports on inventory capacity and consumption. This allows differ- ent business owners to create reports on how much or how little the application is used, and administrators can then adjust the resources allocated to each application accordingly.
11. **Self-service portal:** The cloud orchestrator should enable a self-service portal for application owners. Application owners can request machines or entire multi-machine application environment, and monitor and control them through this portal. The portal should drive the workflow necessary to create the environment and provide run-time environment management to support application elasticity. For example, the owner of the auditing application may request more resources for his application during a busy period.

4.5 ON-PREMISE CLOUD ORCHESTRATION AND PROVISIONING ENGINE

On-premise cloud orchestration and provisioning engine can be a bundled offering that includes hardware, software, and all the services one needs to get started with cloud comput- ing. This bundled offering should include all the elements in a service ecosystem. The offering must have a self-service portal which should include automation and should track as well as control all the resources.

The other objective is to completely integrate and include a service and, allow users to add other services to do integration or other types of cloud work. It can be a pre-packaged private cloud offering that can bring together the hardware, software, and services needed to establish a private cloud or to accelerate selling efforts and effectiveness.

Cloud orchestration and provisioning engine should be designed from client cloud implementation experiences and integrated with the service management software system along with servers, storage, and services to enable a private cloud in the IT environment. This approach will help in doing away with the guess work of establishing a private cloud by pre- installing and configuring the necessary software on the hardware and leveraging services for customization as per the environment. This is required to install applications and utilize the benefits of cloud computing such as pooling, self-service, agility, and service accounting.

4.5.1 *Benefits/Value Proposition*

1. Powers faster time to innovation and lowers cost per unit of innovation.
2. Dramatically improves business values and IT's effect on time to market by enabling business workloads to be deployed rapidly as well as accurately, when and where they are needed.
3. Helps in improving productivity and reducing the capex investment.
4. Minimization of human error based on automation.

4.5.2 *Cloud Orchestration and Provisioning Requirement Analysis*

To know cloud orchestration and provisioning engine requirements, we need to understand the test and development requirements of cloud deployment. Therefore, we should be sure about the automation of testing and development cycles to reduce the deployment time. To do so, we can initiate discussion with the cloud customer about the opportunity for the deployment of the cloud orchestration and provisioning engine. After this, we can set the boundaries of the environment. About 30–50% of any given IT environment is devoted to test/development purposes. It can take developers days, weeks, or even months to procure and configure appropriate hardware, networking, management software, storage, etc. for testing. With cloud orchestration and provisioning engine, a developer can log into a self-service portal, select the required resources along with the time frame, select an image to provision from the service catalog, and be ready to go within hours, as opposed to months. Customer datacenters support hundreds or thousands of distinct composite applications representing business workloads. They leverage multiple types of servers, storage, networks, middleware, and operating systems. About 70% of the IT datacenter budget is spent on assembling and re-assembling existing infrastructure, thus leaving fewer resources for innovation.

Cloud solutions are based on its service management solutions. The first solution in a series of cloud orchestration and provisioning engine solutions that should be considered is workload-specific deployment. It will be a great entry point for users who want to get into cloud computing, as users can roll out cloud orchestration and provisioning engine into their environment without adversely effecting anything else in the environment, and use it as their initial pilot project to start running cloud services.

Cloud orchestration and provisioning engine in any IT environment represents an alternative entry point into cloud computing. So, in some cases, the users want to turn their existing environment into a cloud. In that case, we go into the datacenter, install a cloud management platform, and assign the existing resources to the cloud. One reason behind this transition may be that the organization already has plenty of equipment that are not being used efficiently and the organization sees a lot of benefit in turning its existing investment into a cloud. So, cloud orchestration and provisioning engine can work as the established platform to start the cloud-based services and can then be bundled with the compute and storage to jumpstart the services.

Entry points:

1. **Turn existing environment into a cloud:**
   - Install a cloud management platform and assign existing resources to the cloud.
   - Scenario – you already have enough equipment or the cloud orchestration and provisioning engine offering is not the right platform.

2. **Jumpstart your cloud:**
   - Hardware, software, and services required for a quick startup.
   - Can use a 'seed and grow' model – start with the cloud orchestration and provisioning engine and then add more.

So what solution should one use? Let us look at some scenarios. When can you opt for a cloud orchestration and provisioning engine? The answer to this question is that one can opt when you wish to get started with cloud computing and recognize the advantages of having a cloud management platform bundled with the resources. This is the most rapid way to get a cloud platform up and running. In a scenario where you want to transfer your existing resources into a cloud, you use this service. Also, a user who uses cloud orchestration and provisioning engine will be benefitted when the cloud vendor can help him plan, design, and implement services, whether he wants to implement either infrastructure or platform as a service, or else turn it into a production cloud.

So, in some cases, it is appropriate to opt for cloud orchestration and provisioning engine, business development and test services, because if you just start with the provisioning engine only, you may eventually want to build out your service catalog, integrate it into your directory, cloud vendor or third-party service management products, and extend the capability of the platform.

4.5.3 *Cloud Infrastructure Security*

The security aspect of the cloud infrastructure goes side by side SOA security. It can be introduced as a layered approach. At the top, the service layer with the run-time-secure virtualized environment can be seen. This layer is available as a distributed service environment. These services include administrative and security aspects across different clouds as well as within a single cloud. This environment should gel with the Web services stack, and it is important that we bind the internal resources with other cloud services to offer better hybrid cloud services for successful cloud deployments. We will discuss more about this in Chapter 9.

SOA helps to integrate multiple services from different providers. Cloud computing is the biggest buyer of this characteristic as cloud accommodates various tenants, sub-providers, standards, and distributors.

This type of support is required to manage the SOA system based on mutual trust and agility, and by adopting dynamism within it. The audience may be open, which makes it really difficult to establish the relationship between the provider and the consumer.

The virtual system layer has secure access to the data store. This run-time system differs from classic run-time systems as it operates on VM images, rather than on individual applications. This run-time system provides security services such as anti-virus, introspection, and externalized security services around virtual images. While the foundations of secure virtualized run-time pre-date SOA security and are built on decades of experience with mainframe architectures, the development of secure virtualized run-time is still very much in flux. Cloud vendors continuously invest in research and development of stronger isolation at all levels of the network, server, hypervisor, process, and storage infrastructure to support massive multi-tenancy.

Cloud orchestration and provisioning engine–integrated service management is offered with network, servers, storage, services, and financing as an integrated offering for client test platforms. It

1. Delivers consistent and rapid services from a preloaded system.
2. Helps to improve the value of the business system by effective freedom of choice and faster services.
3. Reduces capital expenses and optimally utilizes the resources.
4. Based on automation and standardized platform, minimizes the chance of human error.
5. Has open architecture–based workload management.
6. Is a complete platform with infrastructure, service management, middleware and distributed system.
7. Is scalable and modular.
8. Offers virtualization for compute and storage.
9. Offers self-service–based delivery.
10. Works as 'lights-out' or 'without human intervention' operations.

Cloud orchestration and provisioning engine is offered as a service engagement which can build a solution to the client's needs, including creation of custom virtual images for dispensing. Summary points are given below:

1. Reduction in configuration and installation time.
2. Creation of new environments in minutes.
3. **Reduction in risk by codifying infrastructure:**
   - Freeze-dry best practices for repeated, consistent deployments.
   - Life cycle–based security.
4. **Simplify maintenance and management:**
   - Flexibly manage and update the components of your patterns.
   - Ensure consistency in versions across development, testing, and production.
5. Spend less time in administrating and more time in developing new solutions.

4.6 COMPUTING ON DEMAND

On-demand computing (CoD) is the need of the hour. It is very essential even in the supercomputing environment. On-demand computing can be implemented using various virtualization techniques. The cloud gives an option to leverage the computing infrastructure without actually buying the hardware. This helps to transfer the workload, if the resources are not able to support it; and at other times, others can utilize the resources that are lying idle. In this way, it is possible to use the resources in the most efficient manner. There may be many spikes in the utilization, but cloud experts can make it smoother.

The uniquely rich set of features offered by on-demand computing enables the service seeker to deploy a true utility. The platform allows users to:

1. Align cost with utilization so that users can scale costs down as well as up. This allows a workload to start with minimal upfront costs and scale the cost as the demand grows, without paying a penalty to increase capacity. Additionally, users can benefit from not incurring the disruption to move to a larger machine.

2. Increase end-users availability significantly. As the workload can be moved dynamically, it is possible to move the workload from one server to another without any interruption, so the remedial work can be carried out if server downtime is required.
3. Balance workload dynamically across multiple servers without taking applications offline. Using workload mobility features, customers can align their costs by ensuring that the workload is deployed in such a way so as to optimize the system's resources.
4. React to short-term resource requirements almost instantly. If a workload has to be deployed at a short notice, a VM can be created on the server and resources can be allocated instantaneously using the dynamic capacity model.
5. Reduce the physical footprint in the datacenter. Consolidation of the workload on to a smaller number of servers will improve space, power, and cooling metrics.
6. Confidently increase system utilization to over 75%, without the fear of degrading performance for end-users.
7. Develop a simple charging model that reflects usage for end-users as the service delivery culture continues to mature.
8. Double the workload delivered in the power and cooling envelope.

4.6.1 *Pre-Provisioning*

For the on-demand computing requirement, pre-provisioning is a viable option as it helps the organization in meeting dynamic datacenter requirements. Organizations would like to reduce the time they take to commission servers when new workload is to be deployed. This approach is ideal when

1. The sizing and capacity planning is fully understood.
2. The workload is fairly constant, ensuring good utilization levels are being achieved.
3. There are business reasons that require physical separation of the workload.
4. The workload can be scaled horizontally.

4.6.2 *On-Demand CPU/Memory/VM Resources*

In the dynamic environment, it is important to track the requirements of CPU, memory, and VMs. It is based on the common pool concept, where resources are allocated and de-allocated as the requirement is over. This approach is ideal when

1. Workloads are trending upward so that investment can be aligned with utilization.
2. Peaks in workload are long term.
3. The workload scales vertically.
4. There is a chance to 'buy out' dynamic capacity, as it proves advantageous then.

4.6.3 *Dynamic Capacity*

Dynamic capacity is based on utility where the additional processing power for a short period of time can be shared from the resource pool. The service is measured via a service accounting tool based on the utilization and finally the billing is done on the basis of the usage. This approach is ideal when

1. The workload is very variable and multiple workloads can be hosted on to a single machine, thus leveling out the utilization.
2. The workload has short periods where system utilization increases massively, but for the majority of the time, it is not resource intensive.
3. Workloads can share a physical platform.
4. The workload is designed to scale vertically only.
5. Users want to dynamically balance workloads across servers.
6. Users want to continue to run very small workloads without incurring the overheads associated with running a physical server to support it.

Benefits

1. Partition mobility, significantly reduced power/cooling footprint, donation of unused processor cycles of VMs with dedicated processors to uncapped partitions, and, at the same time, guaranteed performance of these VMs.
2. Very short deployment time (time-to-market optimized).
3. Lowest possible cost for the deployment of small workloads.
4. Less management effort, for example, when using VMs.
5. Most granular charging scheme, for example, pay for the CPU and memory cycles when actually used.
6. Complete decommissioning of partitions, with resources being available for other purposes.
7. Flexible workload management where workloads can compensate for each other, thus reducing overall utilization.
8. Ideal for environments with identical systems management, utilities for development and testing.

Limitations

1. Dynamic capacity shows short peaks, must not exceed certain limits, and needs to be monitored (via Web interface) to ensure that the best value is obtained.
2. Utility CoD provides processor resources only to the uncapped partitions.

However, one of the most important advantages of the dynamic capacity model is the flexibility, that is, to allow the switching of CPU capacity on and off as needed, which can reduce costs significantly. This is not reflected in the calculation as it requires input from application owners.

4.6.4 Cloud Platform Characteristics Based on CoD

This section discusses cloud platform characteristics on the basis of low-end, on-demand, and dynamic-capacity–based servers.

Low-End Servers

1. Physical segregation of servers.
2. High administration cost due to the management of more physical servers.
3. Limited and complex scalability – maximum eight processors per server, slower turn-around time for server deployment.

4. Longer lead time for server deployment from ordering of servers to setting up of infrastructure.
5. Non-ideal for short product life cycle application due to fixed-cost expenditure for the hardware.
6. Wastage of hardware resources for applications that react to a volatile market.
7. Inability to share resources between applications.
8. Wastage of unused processing cycles if the application does not fully utilize the resources.
9. No hardware/application interdependency forcing downtime on application owners.
10. No capacity on demand capability. Downtime is required for adding new hardware.
11. Low price per CPU cycle purchased, but higher cost per CPU cycle actually used.

On-Demand Platform

1. Physical or logical segregation of servers or partition implementation.
2. Lower administration cost due to less physical server management.
3. Possibility to lead to quick turnaround time for new application deployment or increase the capacity according to business requirements.
4. Enhanced time to market for new product launch, with immediate availability of CPU/memory capacity.
5. Non-ideal for short product life cycle application due to fixed-cost expenditure for the hardware.
6. Wastage of hardware resources for applications which react to a volatile market.
7. Ability to share I/O, CPU, and memory resources between applications.
8. Ability to take advantage of unused CPU/memory if dynamic VM reallocation or share pool methodology is implemented.
9. For provisioning of a platform of independent hardware and application forcing down-time on application owners, careful capacity planning and management is required.
10. Capacity on demand capability – no downtime is required if COD CPU/memory is sufficient.
11. Higher price per CPU cycle, but lower cost per CPU cycle actually used.

Dynamic Capacity Platform

1. Choose VM or workload VM implementation for application consolidation.
2. Lower administration cost due to less physical and logical server management.
3. Can cater to quick turnaround time for new application deployment.
4. Enhanced time-to-market for new product launch, with immediate availability of infrastructure and setup.
5. Ability to scale up and down, which is ideal for applications with short product life cycle.
6. Ability to cater to applications which react to a volatile market, that is, scaling-up and -down capacity.
7. Ability to share I/O, CPU, and memory resources between applications.
8. Ability to take advantage of unused processing cycle of other applications.
9. No hardware/application interdependency forcing downtime on application owners as workload can be dynamically moved to facilitate maintenance, etc.

10. Capacity on-demand capability. No downtime is required as the machine is fully configured.
11. Higher price per CPU cycle, but lower cost per CPU cycle actually used. Average price due to the ability to optimize utilization and rapidly deploy workload.

4.7 CLOUD SOURCING

Today, we are living in an era of optimizing hardware resources and moving toward larger enterprises day by day; so, cloud computing is becoming the major ingredient of infrastructure deployments. Now you may not just need cloud computing, you also need the entire consulting, implementation, and management solutions.

The new wave that is igniting cloud deployments in the service industry is cloud sourcing - outsourcing an end-to-end solution using cloud methodology through public cloud, infrastructure, and platforms. This will be a more planned approach as it will comprise the whole service cycle of outsourcing business with cloud principles through the help of strategized connected cloud platforms matching the overall enterprise requirements.

This includes the whole cloud implementation, IT business consulting, integration, and configuration of the business. This will give the option of enjoying the benefits of service industry with the benefits of the cloud, thus providing an innovative approach of paying resources over subscription.

Real deployment of cloud sourcing will require a business model with the impact of cloud customer and cloud vendor requirements.

With respect to cloud customers, it is important to note that there is no control on the infrastructure layout of cloud deployments or over the place from where data services are offered by the cloud vendor. It is also known that cloud customers do not have to think about the operational staff for the deployments.

Thus, cloud sourcing will play a vital role in the next generation of cloud implementation. With the availability of new open-source tools, it is like icing on the cake, integrated with partner cloud solutions, platforms, and infrastructure. Also, the new charging models, such as services on both project and subscription basis, will generate a new wave to deploy and adopt cloud sourcing models. This will help to customize applications on cloud infrastructure. This will be primarily offered as a public cloud, and all these offerings will be available as managed services. These services will be prototype-based, that is, developed internally on the product and working applications. Therefore, they will provide a good opportunity to use intellectual property for developing different business vertical solutions easily.

4.8 SUMMARY

In this chapter, we have pointed out the main features of cloud orchestration and provisioning engine, BPM clouds, cloud sourcing, and requirements of service management. The next chapter will discuss different types of cloud offerings.