MODULE – II
Cloud Computing Architecture
Chapters – 4

4.1 Introduction
Utility-oriented data centers are the first outcome of cloud computing, and they serve as the infrastructure through which the services are implemented and delivered.
Commonly, clouds are built by relying on one or more datacenters. In most cases hardware resources are virtualized to provide isolation of workloads and to best exploit the infrastructure.
According to the specific service delivered to the end user, different layers can be stacked on top of the virtual infrastructure: a virtual machine manager, a development platform, or a specific application middleware.
A broad definition of the phenomenon could be as follows:
"Cloud computing is a utility-oriented and Internet-centric way of delivering IT services on demand. These services cover the entire computing stack: from the hardware infrastructure packaged as a set of virtual machines to software services such as development platforms and distributed applications."

4.2 The cloud reference model
Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. Such characterization includes quite different aspects: infrastructure, development platforms, application and services.

4.2.1 Architecture

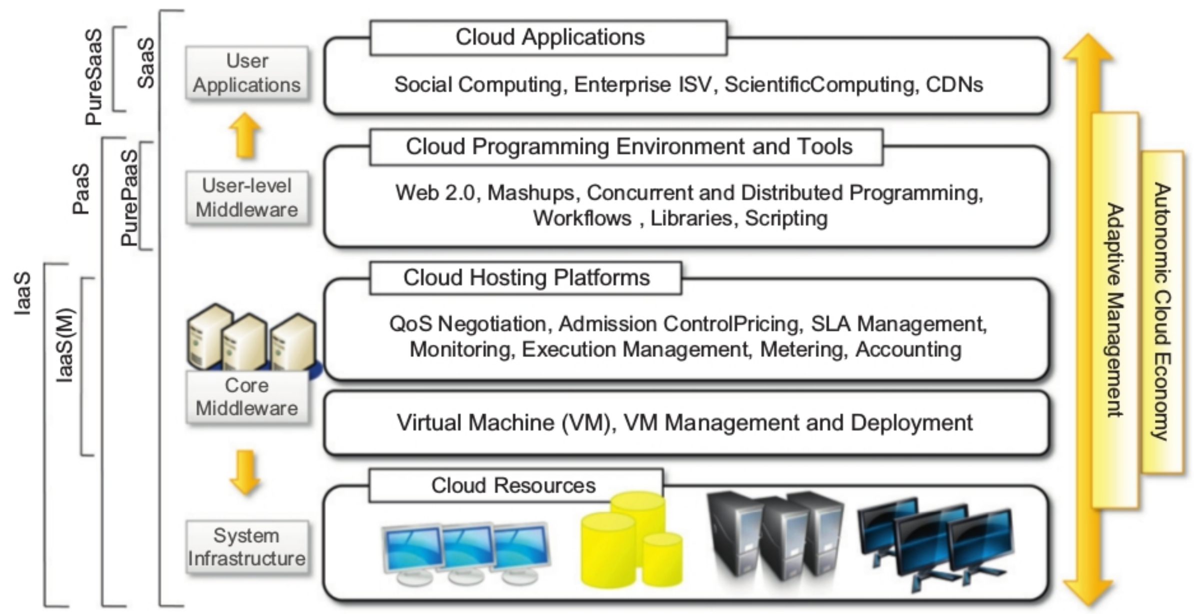

FIGURE 4.1
The cloud computing architecture.

It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack (see Figure 4.1), from hardware appliances to software systems. Cloud resources are harnessed to offer "computing horsepower" required for providing services. Cloud infrastructure can be heterogeneous in nature because a variety of resources, such as clusters and even networked PCs, can be used to build it.

The physical infrastructure is managed by the core middleware, the objectives of which are to provide an appropriate runtime environment for applications and to best utilize resources. At the bottom of the stack, virtualization technologies are used to guarantee runtime environment customization, application isolation, sandboxing, and quality of service. Hardware virtualization is most commonly used at this level. Hypervisors manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines. By using virtual machine technology it is possible to finely partition the hardware resources such as CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications. This solution is generally paired with storage and network virtualization strategies, which allow the infrastructure to be completely virtualized and controlled.

Infrastructure management is the key function of core middleware, which supports capabilities such as negotiation of the quality of service, admission control, execution management and monitoring, accounting, and billing.

The combination of cloud hosting platforms and resources is generally classified as a Infrastructure-as-a-Service (IaaS) solution. We can organize the different examples of IaaS into two categories: Some of them provide both the management layer and the physical infrastructure; others provide only the management layer (IaaS (M)).

In this second case, the management layer is often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.

IaaS solutions are suitable for designing the system infrastructure but provide limited servicesto build applications. Such service is provided by cloud programming environments and tools,which form a new layer for offering users a development platform for applications.

The range of tools include Web-based interfaces, command-line tools, and frameworks for concurrent and distributed programming. In this scenario, users develop their applications specifically for the cloud by using the API exposed at the user-level middleware. For this reason, this approach is also known as Platform-as-a-Service (PaaS) because the service offered to the user is a development platform rather than an infrastructure.

The top layer of the reference model depicted in **Figure 4.1** contains services delivered at the application level. These are mostly referred to as Software-as-a-Service (SaaS). In most cases these are Web-based applications that rely on the cloud to provide service to end users. The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.

Table 4.1 summarizes the characteristics of the three major categories used to classify cloud computing solutions. In the following section, we briefly discuss these characteristics along with some references to practical implementations.

Table 4.1 Cloud Computing Services Classification

Category	Characteristics	Product Type	Vendors and Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizen.com (project management) Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

4.2.2 Infrastructure- and hardware-as-a-service

Infrastructure- and Hardware-as-a-Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing. They deliver customizable infrastructure on demand. The available options within the IaaS offering umbrella range from single servers to entire infrastructures, including network devices, load balancers, and database and Web servers.

The main technology used to deliver and implement these solutions is hardware virtualization:

one or more virtual machines opportunely configured and interconnected define the distributed system on top of which applications are installed and deployed. Virtual machines also constitute the atomic components that are deployed and priced according to the specific features of the virtual hardware: memory, number of processors, and disk storage.

From the perspective of the customer it reduces the administration and maintenance cost as well as the capital costs allocated to purchase hardware.

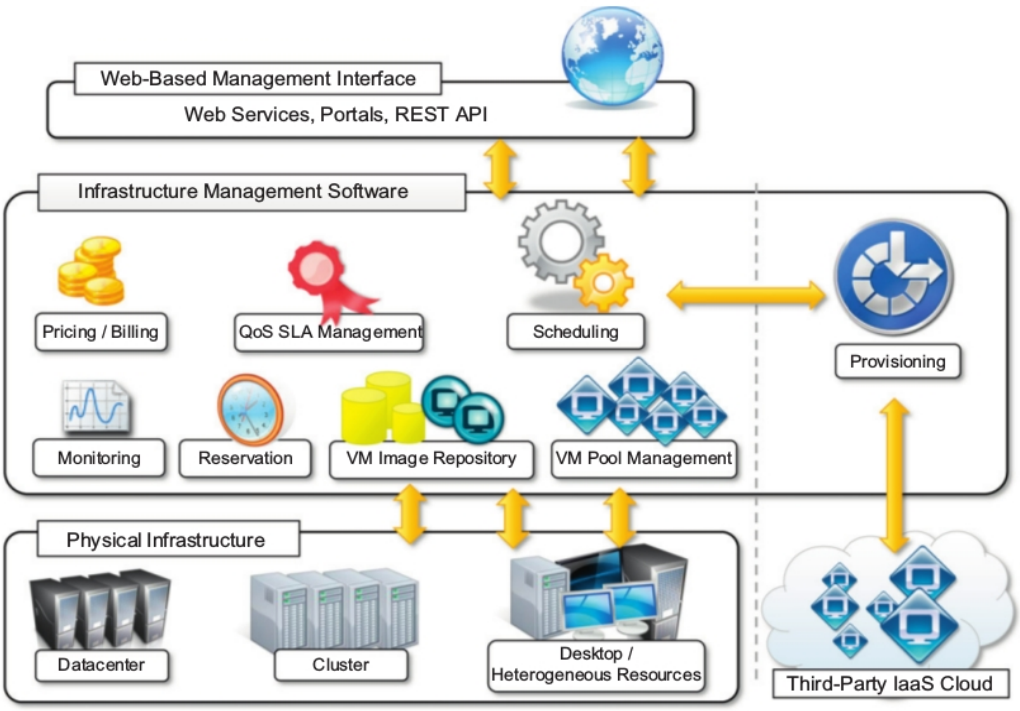

FIGURE 4.2

Infrastructure-as-a-Service reference implementation.

Figure 4.2 provides an overall view of the components forming an Infrastructure-as-a-Service solution. It is possible to distinguish three principal layers: the physical infrastructure, the software management infrastructure, and the user interface.

At the top layer the user interface provides access to the services exposed by the software management infrastructure. Such an interface is based on Web 2.0 technologies: Web services, RESTful APIs, and mash-ups. These technologies allow either applications or final users to access the services exposed by the underlying infrastructure. Web 2.0 applications allow developing full-featured management consoles completely hosted in a browser or a Web page. Web services and RESTful APIs allow programs to interact with the service without human intervention, thus providing complete integration within a software system.

The core features of an IaaS solution are implemented in the infrastructure management software layer. In particular, management of the virtual machines is the most important function performed by this layer. A central role is played by the scheduler, which is in charge of allocating the execution of virtual machine instances. The scheduler interacts with the other components that perform a variety of tasks:

- The pricing and billing component takes care of the cost of executing each virtual machine instance and maintains data that will be used to charge the user.
- The monitoring component tracks the execution of each virtual machine instance and maintains data required for reporting and analyzing the performance of the system.
- The reservation component stores the information of all the virtual machine instances that have been executed or that will be executed in the future.

If support for QoS-based execution is provided, a QoS/SLA management component will maintain a repository of all the SLAs made with the users; together with the monitoring component, this component is used to ensure that a given virtual machine instance is executed with the desired quality of service.

- The VM repository component provides a catalog of virtual machine images that users can use to create virtual instances. Some implementations also allow users to upload their specific virtual machine images.
- A VM pool manager component is responsible for keeping track of all the live instances.

Finally, if the system supports the integration of additional resources belonging to a third-party IaaS provider, a provisioning component interacts with the scheduler to provide a virtual machine instance that is external to the local physical infrastructure directly managed by the pool.

The bottom layer is composed of the physical infrastructure, on top of which the management layer operates. As previously discussed, the infrastructure can be of different types; the specific infrastructure used depends on the specific use of the cloud. A cloud infrastructure developed in house, in a small or medium-sized enterprise or within a university department, will most likely rely on a cluster. At the bottom of the scale it is also possible to consider a heterogeneous environment where different types of resources—PCs, workstations, and clusters—can be aggregated.

4.2.3 Platform as a service

Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud. They constitute the middleware on top of which applications are built. A general overview of the features characterizing the PaaS approach is given in Figure 4.3. Application management is the core functionality of the middleware. PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure. They automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases, and managing system change based on policies set by the user.

The core middleware is in charge of managing the resources and scaling applications on demand or automatically, according to the commitments made with users.
From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud. These can be in the form of a Web-based interface or in the form of programming APIs and libraries.

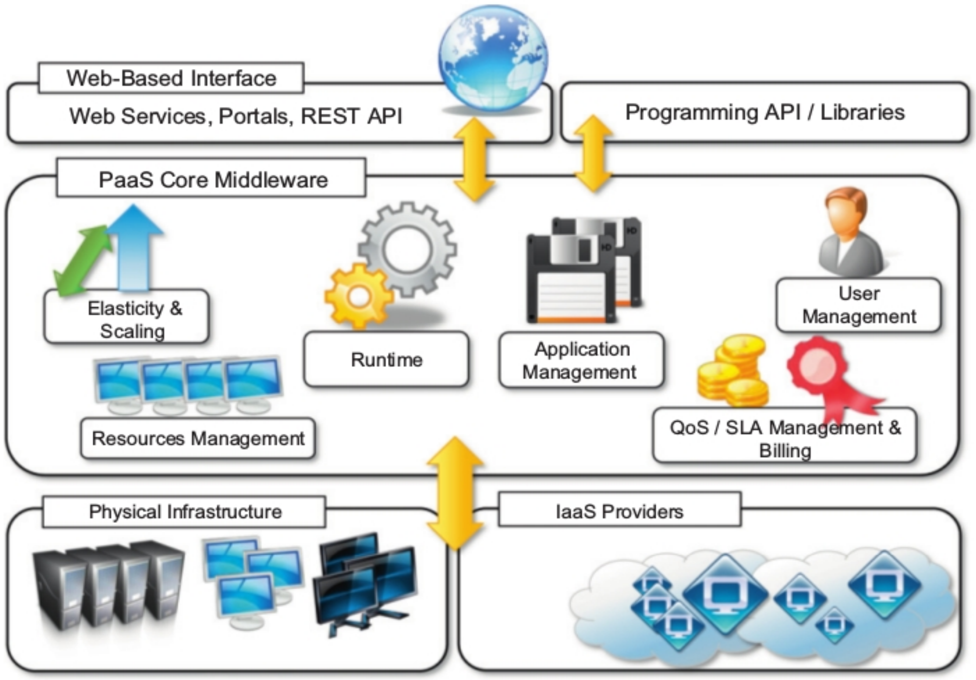

FIGURE 4.3

The Platform-as-a-Service reference model.

It is possible to find integrated developed environments based on 4GL and visual programming concepts, or rapid prototyping environments where applications are built by assembling mash-ups and user-defined components and successively customized. Other implementations of the PaaS model provide a complete object model for representing an application and provide a programming language-based approach. This approach generally offers more flexibility and opportunities but incurs longer development cycles. PaaS solutions can offer middleware for developing applications together with the infrastructure or simply provide users with the software that is installed on the user premises. In the first case, the PaaS provider also owns large datacenters where applications are executed; in the second case, referred to in this book as Pure PaaS, the middleware constitutes the core value of the offering.
Table 4.2 provides a classification of the most popular PaaS implementations. It is possible to organize the various solutions into three wide categories: PaaS-I, PaaS-II, and PaaS-III.

As noted by Sam Charrington, product manager at Appistry.com, there are some essential characteristics that identify a PaaS solution:
1. Runtime framework. This framework represents the "software stack" of the PaaS model. The runtime framework executes end-user code according to the policies set by the user and the provider.
2. Abstraction. PaaS solutions are distinguished by the higher level of abstraction that they provide. In the case of PaaS the focus is on the applications the cloud must support. PaaS solutions offer a way to deploy and manage applications on the cloud rather than a bunch of virtual machines on top of which the IT infrastructure is built and configured.

Table 4.2 Platform-as-a-Service Offering Classification

Category	Description	Product Type	Vendors and Products
PaaS-I	Runtime environment with Web-hosted application development platform. Rapid application prototyping.	Middleware + Infrastructure Middleware + Infrastructure	Force.com Longjump
PaaS-II	Runtime environment for scaling Web applications. The runtime could be enhanced by additional components that provide scaling capabilities.	Middleware + Infrastructure Middleware Middleware + Infrastructure Middleware + Infrastructure Middleware + Infrastructure Middleware	Google AppEngine AppScale Heroku Engine Yard Joyent Smart Platform GigaSpaces XAP
PaaS-III	Middleware and programming model for developing distributed applications in the cloud.	Middleware + Infrastructure Middleware Middleware Middleware Middleware Middleware	Microsoft Azure DataSynapse Cloud IQ Manjrasof Aneka Apprenda SaaSGrid GigaSpaces DataGrid

3. Automation. PaaS environments automate the process of deploying applications to the infrastructure, scaling them by provisioning additional resources when needed. This process is performed automatically and according to the SLA made between the customers and the provider.
4. Cloud services. PaaS offerings provide developers and architects with services and APIs, helping them to simplify the creation and delivery of elastic and highly available cloud applications.

4.2.4 Software as a service

Software-as-a-Service (SaaS) is a software delivery model that provides access to applications through the Internet as a Web-based service.
It provides a means to free users from complex hardware and software management by offloading such tasks to third parties, which build applications accessible to multiple users through a Web browser.
In this scenario, customers neither need install anything on their premises nor have to pay considerable up-front costs to purchase the software and the required licenses.
The SaaS model is appealing for applications serving a wide range of users and that can be adapted to specific needs with little further customization. This requirement characterizes SaaS as a "one-to-many" software delivery model, whereby an application is shared across multiple users.
This is the case of CRM 3 and ERP 4 applications that constitute common needs for almost all enterprises, from small to medium-sized and large business. Every enterprise will have the same requirements for the basic features concerning CRM and ERP; different needs can be satisfied with further customization.
ASPs (application service providers) has some of the core characteristics of SaaS:

- The product sold to customer is application access.
- The application is centrally managed.
- The service delivered is one-to-many.
- The service delivered is an integrated solution delivered on the contract, which means provided as promised.

ASPs provide access to packaged software solutions that addressed the needs of a variety of

customers.
The SaaS approach introduces a more flexible way of delivering application services that are fully customizable by the user by integrating new services, injecting their own components, and designing the application and information workflows.
The benefits delivered are the following:

1. Software cost reduction and total cost of ownership (TCO) were paramount
2. Service-level improvements
3. Rapid implementation
4. Standalone and configurable applications
5. Rudimentary application and data integration
6. Subscription and pay-as-you-go (PAYG) pricing

4.3 Types of clouds

Clouds constitute the primary outcome of cloud computing. They are a type of parallel and distributed system harnessing physical and virtual computers presented as a unified computing resource.
A more useful classification is given according to the administrative domain of a cloud. It is then possible to differentiate four different types of cloud:
1. Public clouds. The cloud is open to the wider public.
2. Private clouds. The cloud is implemented within the private premises of an institution and generally made accessible to the members of the institution or a subset of them.
3. Hybrid or heterogeneous clouds. The cloud is a combination of the two previous solutions and most likely identifies a private cloud that has been augmented with resources or services hosted in a public cloud.
4. Community clouds. The cloud is characterized by a multi-administrative domain involving different deployment models (public, private, and hybrid), and it is specifically designed to address the needs of a specific industry.

4.3.1 Public clouds

Public clouds are a realization of the canonical view of cloud computing in which the services offered are made available to anyone, from anywhere, and at any time through the Internet. From a structural point of view they are a distributed system, most likely composed of one or more datacenters connected together, on top of which the specific services offered by the cloud are implemented. Any customer can easily sign in with the cloud provider, enter her credential and billing details, and use the services offered.
Historically, public clouds were the first class of cloud that were implemented and offered. They offer solutions for minimizing IT infrastructure costs and serve as a viable option for handling peak loads on the local infrastructure.
A fundamental characteristic of public clouds is multitenancy. A public cloud is meant to serve a multitude of users, not a single customer. Any customer requires a virtual computing environment that is separated, and most likely isolated, from other users.
QoS management is a very important aspect of public clouds.
Hence, a significant portion of the software infrastructure is devoted to monitoring the cloud resources, to bill them according to the contract made with the user, and to keep a complete history of cloud usage for each customer.
A public cloud can offer any kind of service: infrastructure, platform, or applications. For example, Amazon EC2 is a public cloud that provides infrastructure as a service; Google AppEngine is a public cloud that provides an application development platform as a service; and SalesForce.com is

a public cloud that provides software as a service.

4.3.2 Private clouds

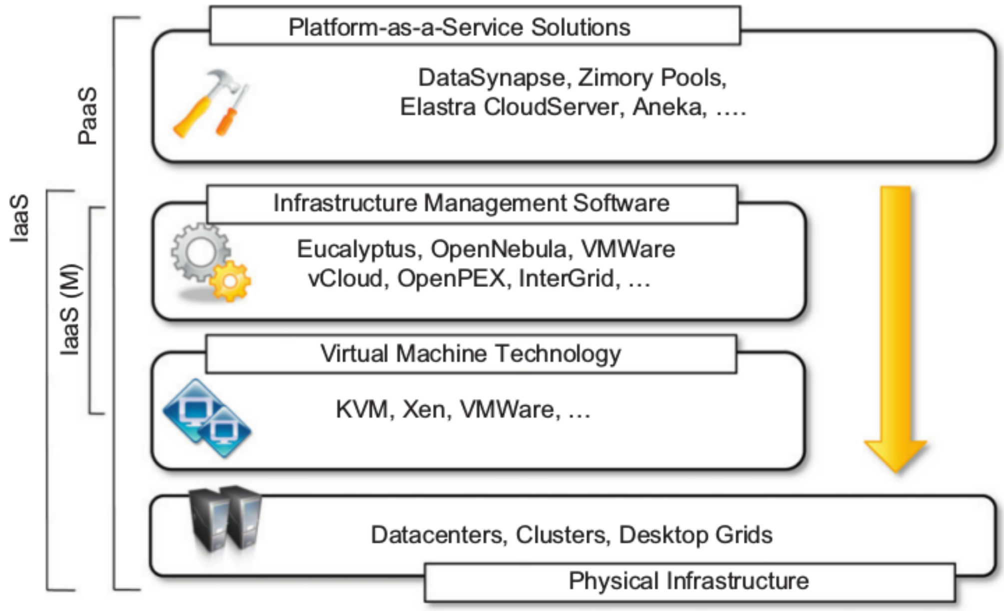

FIGURE 4.4

Private clouds hardware and software stack.

In the case of public clouds, the provider is in control of the infrastructure and, eventually, of the customers' core logic and sensitive data. Even though there could be regulatory procedure in place that guarantees fair management and respect of the customer's privacy, this condition can still be perceived as a threat or as an unacceptable risk that some organizations are not willing to take.
Figure 4.4 provides a comprehensive view of the solutions together with some reference to the most popular software used to deploy private clouds. At the bottom layer of the software stack, virtual machine technologies such as Xen, KVM, and VMware serve as the foundations of the cloud. Virtual machine management technologies such as VMware vCloud, Eucalyptus, and OpenNebula can be used to control the virtual infrastructure.

Private clouds are virtual distributed systems that rely on a private infrastructure and provide internal users with dynamic provisioning of computing resources. Instead of a pay-as-you-go model as in public clouds, there could be other schemes in place, taking into account the usage of the cloud and proportionally billing the different departments or sections of an enterprise. Private clouds have the advantage of keeping the core business operations in-house by relying on the existing IT infrastructure and reducing the burden of maintaining it once the cloud has been set up. In this scenario, security concerns are less critical, since sensitive information does not flow out of the private infrastructure. Moreover, existing IT resources can be better utilized because the private cloud can provide services to a different range of users. Another interesting opportunity that comes with private clouds is the possibility of testing applications and systems at a comparatively lower price.
Key advantages of using a private cloud computing infrastructure:

1. Customer information protection. Despite assurances by the public cloud leaders about security, few provide satisfactory disclosure or have long enough histories with their cloud

offerings to provide warranties about the specific level of security put in place on their systems. In-house security is easier to maintain and rely on.
2. Infrastructure ensuring SLAs. Quality of service implies specific operations such as appropriate clustering and failover, data replication, system monitoring and maintenance, and disaster recovery, and other uptime services can be commensurate to the application needs. Although public cloud vendors provide some of these features, not all of them are available as needed.
3. Compliance with standard procedures and operations. If organizations are subject to third-party compliance standards, specific procedures have to be put in place when deploying and executing applications. This could be not possible in the case of the virtual public infrastructure.

DataSynapse provides a flexible environment for building private clouds on top of datacenters.
Elastra Cloud Server is a platform for easily configuring and deploying distributed application infrastructures on clouds.

4.3.3 Hybrid clouds

Public clouds are large software and hardware infrastructures that have a capability that is huge enough to serve the needs of multiple users, but they suffer from security threats and administrative pitfalls.

One of the major drawbacks of private deployments is the inability to scale on demand and to efficiently address peak loads. In this case, it is important to leverage capabilities of public clouds as needed.

Hybrid clouds allow enterprises to exploit existing IT infrastructures, maintain sensitive information within the premises, and naturally grow and shrink by provisioning external resources and releasing them when they’re no longer needed. Security concerns are then only limited to the public portion of the cloud that can be used to perform operations with less stringent constraints but that are still part of the system workload.

Figure 4.5 provides a general overview of a hybrid cloud: It is a heterogeneous distributed system resulting from a private cloud that integrates additional services or resources from one or more public clouds. For this reason they are also called heterogeneous clouds. As depicted in the diagram, dynamic provisioning is a fundamental component in this scenario. Hybrid clouds address scalability issues by leveraging external resources for exceeding capacity demand. These resources or services are temporarily leased for the time required and then released. This practice is also known as cloudbursting.

Dynamic provisioning is most commonly implemented in PaaS solutions that support hybrid clouds.

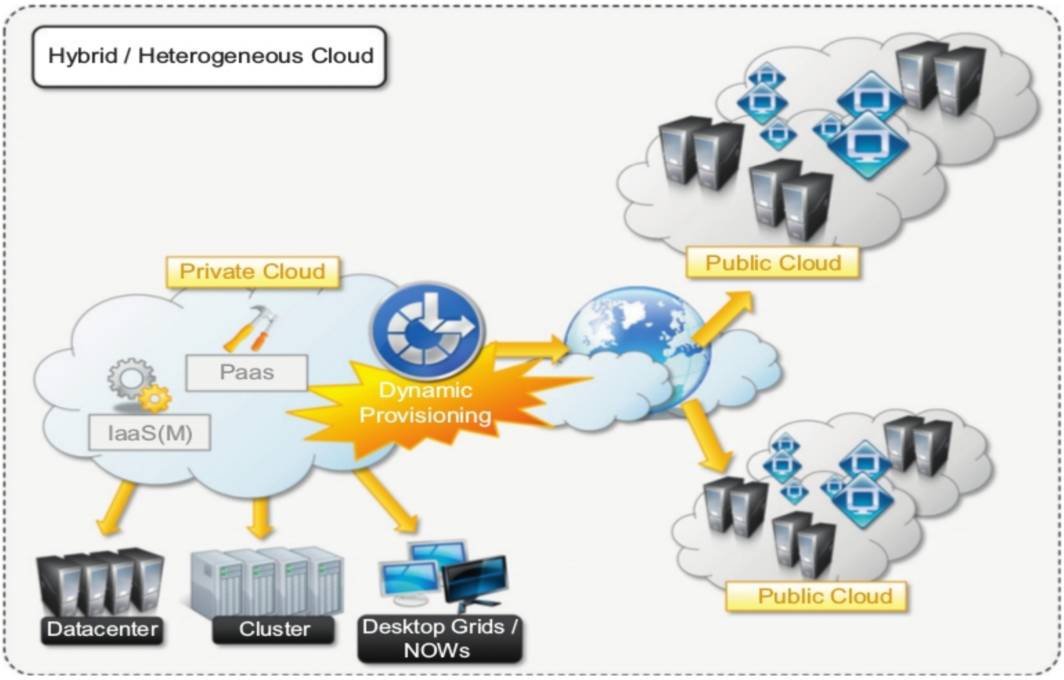

FIGURE 4.5

Hybrid/heterogeneous cloud overview.

4.3.4 Community clouds

Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector. The National Institute of Standards and Technologies (NIST) [43] characterizes community clouds as follows:

"The infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise."

Figure 4.6 provides a general view of the usage scenario of community clouds, together with reference architecture. The users of a specific community cloud fall into a well-identified community, sharing the same concerns or needs; they can be government bodies, industries, or even simple users, but all of them focus on the same issues for their interaction with the cloud.

This is a different scenario than public clouds, which serve a multitude of users with different needs.

Community clouds are also different from private clouds, where the services are generally delivered within the institution that owns the cloud.

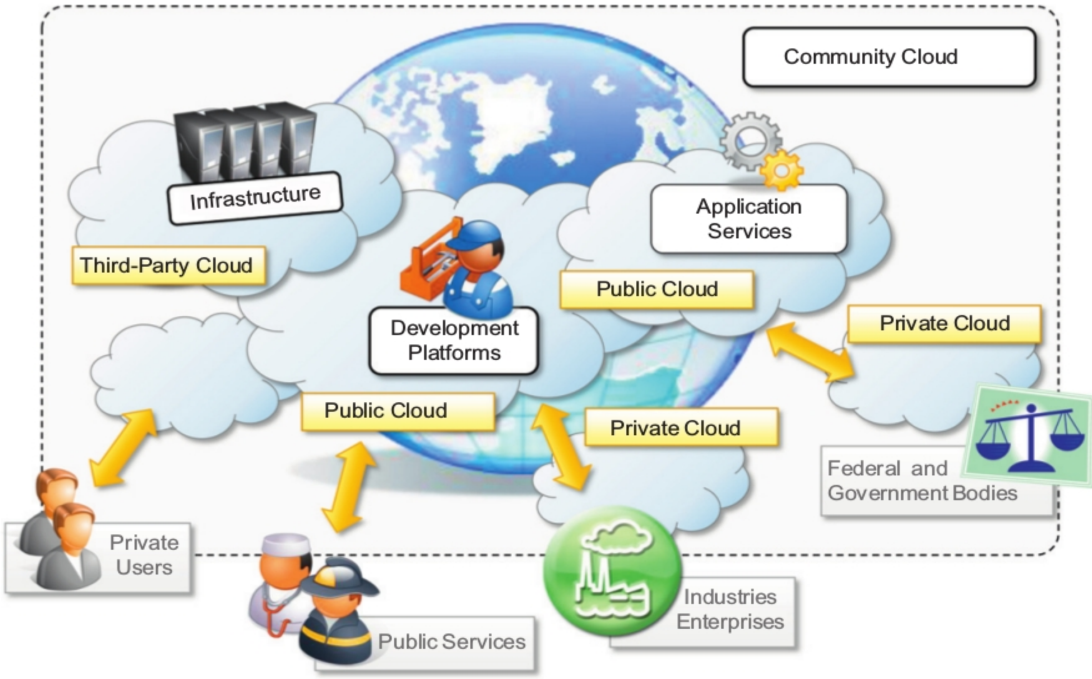

FIGURE 4.6

A community cloud.

Candidate sectors for community clouds are as follows:

1. Media industry - looking for low-cost, agile, and simple solutions to improve the efficiency of content production.

2. Healthcare industry - In the healthcare industry, there are different scenarios in which community clouds could be of use. In particular, community clouds can provide a global platform on which to share information and knowledge without revealing sensitive data maintained within the private infrastructure.

3. Energy and other core industries - In these sectors, community clouds can bundle the comprehensive set of solutions that together vertically address management, deployment, and orchestration of services and operations.

4. Public sector - Legal and political restrictions in the public sector can limit the adoption of public cloud offerings. Moreover, governmental processes involve several institutions and agencies and are aimed at providing strategic solutions at local, national, and international administrative levels.

5. Scientific research - Science clouds are an interesting example of community clouds. In this case, the common interest driving different organizations sharing a large distributed infrastructure is scientific computing.

The benefits of these community clouds are the following:

1. Openness. By removing the dependency on cloud vendors, community clouds are open systems in which fair competition between different solutions can happen.

2. Community. Being based on a collective that provides resources and services, the infrastructure turns out to be more scalable because the system can grow simply by expanding its user base.

3. Graceful failures. Since there is no single provider or vendor in control of the infrastructure, there is no single point of failure.

4. Convenience and control. Within a community cloud there is no conflict between convenience and control because the cloud is shared and owned by the community, which makes all the decisions through a collective democratic process.

5. Environmental sustainability. The community cloud is supposed to have a smaller carbon footprint because it harnesses underutilized resources. Moreover, these clouds tend to be more organic by growing and shrinking in a symbiotic relationship to support the demand of the community, which in turn sustains it.

4.4 Economics of the cloud

The main drivers of cloud computing are economy of scale and simplicity of software delivery and its operation. In fact, the biggest benefit of this phenomenon is financial: the pay-as-you-go model offered by cloud providers. In particular, cloud computing allows:

1. Reducing the capital costs associated to the IT infrastructure
2. Eliminating the depreciation or lifetime costs associated with IT capital assets
3. Replacing software licensing with subscriptions
4. Cutting the maintenance and administrative costs of IT resources.

A capital cost is the cost occurred in purchasing an asset that is useful in the production of goods or the rendering of services. Capital costs are one-time expenses that are generally paid up front and that will contribute over the long term to generate profit.

IT resources constitute a capital cost for any kind of enterprise. It is good practice to try to keep capital costs low because they introduce expenses that will generate profit over time; more than that, since they are associated with material things they are subject to depreciation over time, which in the end reduces the profit of the enterprise because such costs are directly subtracted from the enterprise revenues.

One of the advantages introduced by the cloud computing model is that it shifts the capital costs previously allocated to the purchase of hardware and software into operational costs inducted by renting the infrastructure and paying subscriptions for the use of software. These costs can be better controlled according to the business needs and prosperity of the enterprise. Cloud computing also introduces reductions in administrative and maintenance costs. That is, there is no or limited need for having administrative staff take care of the management of the cloud infrastructure.

In terms of the pricing models introduced by cloud computing, we can distinguish three different strategies that are adopted by the providers:

1. Tiered pricing. In this model, cloud services are offered in several tiers, each of which offers a fixed computing specification and SLA at a specific price per unit of time. This model is used by Amazon for pricing the EC2 service.

2. Per-unit pricing. This model is more suitable to cases where the principal source of revenue for the cloud provider is determined in terms of units of specific services, such as data transfer and memory allocation. In this scenario customers can configure their systems more efficiently according to the application needs. This model is used, for example, by GoGrid, which makes customers pay according to RAM/hour units for the servers deployed in the GoGrid cloud.

3. Subscription-based pricing. This is the model used mostly by SaaS providers in which users pay a periodic subscription fee for use of the software or the specific component services that are integrated in their applications.

4.5 Open challenges

Cloud computing presents many challenges for industry and academia. There is a significant amount of work in academia focused on defining the challenges brought by this phenomenon.

In this section, we highlight the most important ones.
1 Cloud definition
2 Cloud interoperability and standards
3 Scalability and fault tolerance
4 Security, trust, and privacy
5 Organizational aspects

1 Cloud definition

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

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

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

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

2 Cloud interoperability and standards

To fully realize this goal, introducing standards and allowing interoperability between solutions offered by different vendors are objectives of fundamental importance. Vendor lock-in constitutes one of the major strategic barriers against the seamless adoption of cloud computing at all stages.

The presence of standards that are actually implemented and adopted in the cloud computing community could give room for interoperability and then lessen the risks resulting from vendor lock-in.

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

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

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

3 Scalability and fault tolerance

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

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

4 Security, trust, and privacy

Security, trust, and privacy issues are major obstacles for massive adoption of cloud computing.
The traditional cryptographic technologies are used to prevent data tampering and access to sensitive information. The massive use of virtualization technologies exposes the existing system to new threats, which previously were not considered applicable.
Information can be stored within a cloud storage facility using the most advanced technology in cryptography to protect data and then be considered safe from any attempt to access it without the required permissions.
The lack of control over data and processes also poses severe problems for the trust we give to the cloud service provider and the level of privacy we want to have for our data.

5 Organizational aspects

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

1. What is the new role of the IT department in an enterprise that completely or significantly relies on the cloud?
2. How will the compliance department perform its activity when there is a considerable lack of control over application workflows?
3. What are the implications (political, legal, etc.) for organizations that lose control over some aspects of their services?
4. What will be the perception of the end users of such services?

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