

## **Warning**

This material has been reproduced and communicated to you by or on behalf of Monash University in accordance with s113P of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

**Do not remove this notice.**

**PART**

---

Foundations

1

This page intentionally left blank

# Introduction

# 1

Computing is being transformed into a model consisting of services that are commoditized and delivered in a manner similar to utilities such as water, electricity, gas, and telephony. In such a model, users access services based on their requirements, regardless of where the services are hosted. Several computing paradigms, such as grid computing, have promised to deliver this utility computing vision. *Cloud computing* is the most recent emerging paradigm promising to turn the vision of “computing utilities” into a reality.

Cloud computing is a technological advancement that focuses on the way we design computing systems, develop applications, and leverage existing services for building software. It is based on the concept of *dynamic provisioning*, which is applied not only to services but also to compute capability, storage, networking, and information technology (IT) infrastructure in general. Resources are made available through the Internet and offered on a *pay-per-use* basis from cloud computing vendors. Today, anyone with a credit card can subscribe to cloud services and deploy and configure servers for an application in hours, growing and shrinking the infrastructure serving its application according to the demand, and paying only for the time these resources have been used.

This chapter provides a brief overview of the cloud computing phenomenon by presenting its vision, discussing its core features, and tracking the technological developments that have made it possible. The chapter also introduces some key cloud computing technologies as well as some insights into development of cloud computing environments.

---

## 1.1 Cloud computing at a glance

In 1969, Leonard Kleinrock, one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET), which seeded the Internet, said:

*As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of ‘computer utilities’ which, like present electric and telephone utilities, will service individual homes and offices across the country.*

This vision of computing utilities based on a service-provisioning model anticipated the massive transformation of the entire computing industry in the 21<sup>st</sup> century, whereby computing services will be readily available on demand, just as other utility services such as water, electricity, telephone, and gas are available in today’s society. Similarly, users (consumers) need to pay providers

only when they access the computing services. In addition, consumers no longer need to invest heavily or encounter difficulties in building and maintaining complex IT infrastructure.

In such a model, users access services based on their requirements without regard to where the services are hosted. This model has been referred to as *utility computing* or, recently (since 2007), as *cloud computing*. The latter term often denotes the infrastructure as a “cloud” from which businesses and users can access applications as services from anywhere in the world and on demand. Hence, cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services supported by state-of-the-art data centers employing virtualization technologies for consolidation and effective utilization of resources.

Cloud computing allows renting infrastructure, runtime environments, and services on a pay-per-use basis. This principle finds several practical applications and then gives different images of cloud computing to different people. Chief information and technology officers of large enterprises see opportunities for scaling their infrastructure on demand and sizing it according to their business needs. End users leveraging cloud computing services can access their documents and data anytime, anywhere, and from any device connected to the Internet. Many other points of view exist.<sup>1</sup> One of the most diffuse views of cloud computing can be summarized as follows:

*I don't care where my servers are, who manages them, where my documents are stored, or where my applications are hosted. I just want them always available and access them from any device connected through Internet. And I am willing to pay for this service for as long as I need it.*

The concept expressed above has strong similarities to the way we use other services, such as water and electricity. In other words, cloud computing turns IT services into *utilities*. Such a delivery model is made possible by the effective composition of several technologies, which have reached the appropriate maturity level. *Web 2.0* technologies play a central role in making cloud computing an attractive opportunity for building computing systems. They have transformed the Internet into a rich application and service delivery platform, mature enough to serve complex needs. *Service orientation* allows cloud computing to deliver its capabilities with familiar abstractions, while *virtualization* confers on cloud computing the necessary degree of customization, control, and flexibility for building production and enterprise systems.

Besides being an extremely flexible environment for building new systems and applications, cloud computing also provides an opportunity for integrating additional capacity or new features into existing systems. The use of dynamically provisioned IT resources constitutes a more attractive opportunity than buying additional infrastructure and software, the sizing of which can be difficult to estimate and the needs of which are limited in time. This is one of the most important advantages of cloud computing, which has made it a popular phenomenon. With the wide deployment of cloud computing systems, the foundation technologies and systems enabling them are becoming consolidated and standardized. This is a fundamental step in the realization of the long-term vision

---

<sup>1</sup>An interesting perspective on the way cloud computing evokes different things to different people can be found in a series of interviews made by Rob Boothby, vice president and platform evangelist of Joyent, at the Web 2.0 Expo in May 2007. Chief executive officers (CEOs), chief technology officers (CTOs), founders of IT companies, and IT analysts were interviewed, and all of them gave their personal perception of the phenomenon, which at that time was starting to spread. The video of the interview can be found on YouTube at the following link: [www.youtube.com/watch?v=6PNuQHUiV3Q](http://www.youtube.com/watch?v=6PNuQHUiV3Q).

for cloud computing, which provides an open environment where computing, storage, and other services are traded as computing utilities.

### 1.1.1 The vision of cloud computing

Cloud computing allows anyone with a credit card to provision virtual hardware, runtime environments, and services. These are used for as long as needed, with no up-front commitments required. The entire stack of a computing system is transformed into a collection of utilities, which can be provisioned and composed together to deploy systems in hours rather than days and with virtually no maintenance costs. This opportunity, initially met with skepticism, has now become a practice across several application domains and business sectors (see [Figure 1.1](#)). The demand has fast-tracked technical development and enriched the set of services offered, which have also become more sophisticated and cheaper.

Despite its evolution, the use of cloud computing is often limited to a single service at a time or, more commonly, a set of related services offered by the same vendor. Previously, the lack of effective standardization efforts made it difficult to move hosted services from one vendor to another. The long-term vision of cloud computing is that IT services are traded as utilities in an open market, without technological and legal barriers. In this cloud marketplace, cloud service providers and consumers, trading cloud services as utilities, play a central role.

Many of the technological elements contributing to this vision already exist. Different stakeholders leverage clouds for a variety of services. The need for ubiquitous storage and compute power on demand is the most common reason to consider cloud computing. A scalable runtime for applications is an attractive option for application and system developers that do not have infrastructure or cannot afford any further expansion of existing infrastructure. The capability for Web-based access to documents and their processing using sophisticated applications is one of the appealing factors for end users.

In all these cases, the discovery of such services is mostly done by human intervention: a person (or a team of people) looks over the Internet to identify offerings that meet his or her needs. We imagine that in the near future it will be possible to find the solution that matches our needs by simply entering our request in a global digital market that trades cloud computing services. The existence of such a market will enable the automation of the discovery process and its integration into existing software systems, thus allowing users to transparently leverage cloud resources in their applications and systems. The existence of a global platform for trading cloud services will also help service providers become more visible and therefore potentially increase their revenue. A global cloud market also reduces the barriers between service consumers and providers: it is no longer necessary to belong to only one of these two categories. For example, a cloud provider might become a consumer of a competitor service in order to fulfill its own promises to customers.

These are all possibilities that are introduced with the establishment of a global cloud computing marketplace and by defining effective standards for the unified representation of cloud services as well as the interaction among different cloud technologies. A considerable shift toward cloud computing has already been registered, and its rapid adoption facilitates its consolidation. Moreover, by concentrating the core capabilities of cloud computing into large datacenters, it is possible to reduce or remove the need for any technical infrastructure on the service consumer side. This approach provides opportunities for optimizing datacenter facilities and fully utilizing their



**FIGURE 1.1**

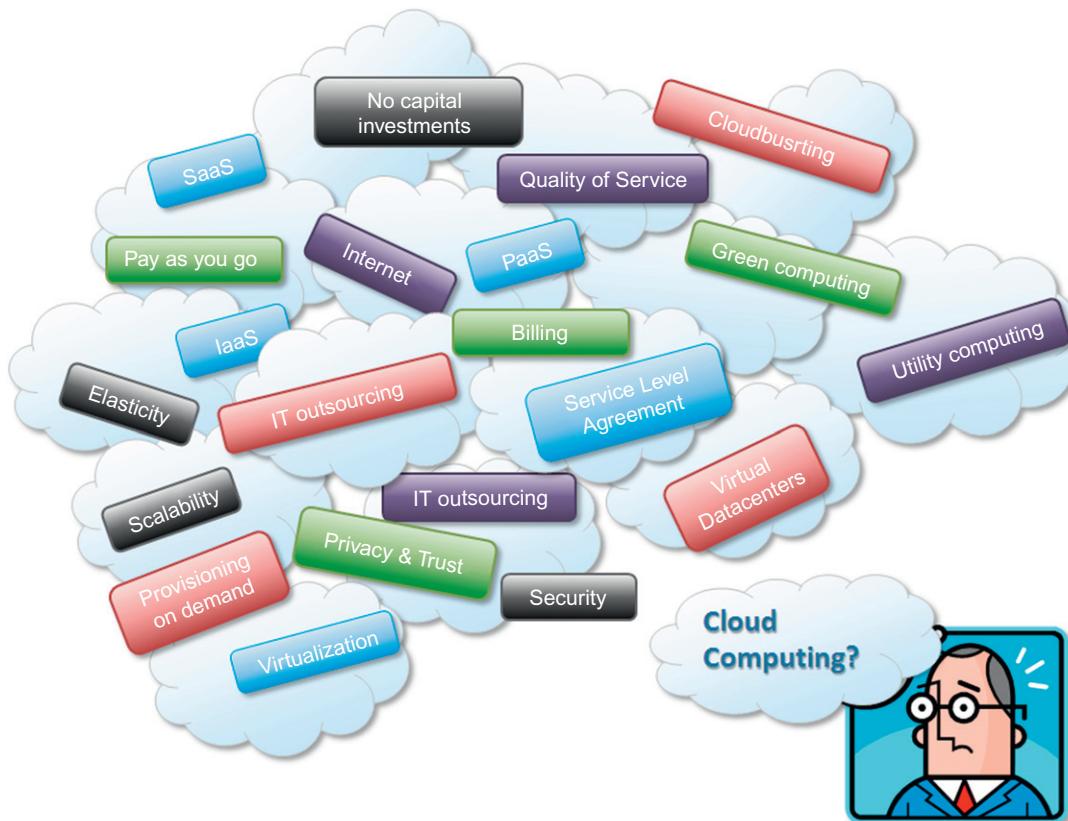
Cloud computing vision.

capabilities to serve multiple users. This consolidation model will reduce the waste of energy and carbon emissions, thus contributing to a greener IT on one end and increasing revenue on the other end.

### 1.1.2 Defining a cloud

Cloud computing has become a popular buzzword; it has been widely used to refer to different technologies, services, and concepts. It is often associated with virtualized infrastructure or hardware on demand, utility computing, IT outsourcing, platform and software as a service, and many other things that now are the focus of the IT industry. [Figure 1.2](#) depicts the plethora of different notions included in current definitions of cloud computing.

The term *cloud* has historically been used in the telecommunications industry as an abstraction of the network in system diagrams. It then became the symbol of the most popular computer network: the Internet. This meaning also applies to *cloud computing*, which refers to an Internet-centric way of



**FIGURE 1.2**

Cloud computing technologies, concepts, and ideas.

computing. The Internet plays a fundamental role in cloud computing, since it represents either the medium or the platform through which many cloud computing services are delivered and made accessible. This aspect is also reflected in the definition given by Armbrust et al. [28]:

*Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the datacenters that provide those services.*

This definition describes cloud computing as a phenomenon touching on the entire stack: from the underlying hardware to the high-level software services and applications. It introduces the concept of *everything as a service*, mostly referred as *XaaS*,<sup>2</sup> where the different components of a system—IT infrastructure, development platforms, databases, and so on—can be delivered, measured, and consequently priced as a service. This new approach significantly influences not only the way that we build software but also the way we deploy it, make it accessible, and design our IT infrastructure, and even the way companies allocate the costs for IT needs. The approach fostered by cloud computing is global: it covers both the needs of a single user hosting documents in the cloud and the ones of a CIO deciding to deploy part of or the entire corporate IT infrastructure in the public cloud. This notion of multiple parties using a shared cloud computing environment is highlighted in a definition proposed by the U.S. National Institute of Standards and Technology (NIST):

*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

Another important aspect of cloud computing is its utility-oriented approach. More than any other trend in distributed computing, cloud computing focuses on delivering services with a given pricing model, in most cases a “pay-per-use” strategy. It makes it possible to access online storage, rent virtual hardware, or use development platforms and pay only for their effective usage, with no or minimal up-front costs. All these operations can be performed and billed simply by entering the credit card details and accessing the exposed services through a Web browser. This helps us provide a different and more practical characterization of cloud computing. According to Reese [29], we can define three criteria to discriminate whether a service is delivered in the cloud computing style:

- The service is accessible via a Web browser (nonproprietary) or a Web services application programming interface (API).
- Zero capital expenditure is necessary to get started.
- You pay only for what you use as you use it.

Even though many cloud computing services are freely available for single users, enterprise-class services are delivered according a specific pricing scheme. In this case users subscribe to the service and establish with the service provider a service-level agreement (SLA) defining the

---

<sup>2</sup>XaaS is an acronym standing for *X-as-a-Service*, where the *X* letter can be replaced by one of a number of things: *S* for *software*, *P* for *platform*, *I* for *infrastructure*, *H* for *hardware*, *D* for *database*, and so on.

quality-of-service parameters under which the service is delivered. The utility-oriented nature of cloud computing is clearly expressed by Buyya et al. [30]:

*A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers.*

### 1.1.3 A closer look

Cloud computing is helping enterprises, governments, public and private institutions, and research organizations shape more effective and demand-driven computing systems. Access to, as well as integration of, cloud computing resources and systems is now as easy as performing a credit card transaction over the Internet. Practical examples of such systems exist across all market segments:

- *Large enterprises can offload some of their activities to cloud-based systems.* Recently, the *New York Times* has converted its digital library of past editions into a Web-friendly format. This required a considerable amount of computing power for a short period of time. By renting Amazon EC2 and S3 Cloud resources, the *Times* performed this task in 36 hours and relinquished these resources, with no additional costs.
- *Small enterprises and start-ups can afford to translate their ideas into business results more quickly, without excessive up-front costs.* Animoto is a company that creates videos out of images, music, and video fragments submitted by users. The process involves a considerable amount of storage and backend processing required for producing the video, which is finally made available to the user. Animoto does not own a single server and bases its computing infrastructure entirely on Amazon Web Services, which are sized on demand according to the overall workload to be processed. Such workload can vary a lot and require instant scalability.<sup>3</sup> Up-front investment is clearly not an effective solution for many companies, and cloud computing systems become an appropriate alternative.
- *System developers can concentrate on the business logic rather than dealing with the complexity of infrastructure management and scalability.* Little Fluffy Toys is a company in London that has developed a widget providing users with information about nearby bicycle rental services. The company has managed to back the widget's computing needs on Google AppEngine and be on the market in only one week.
- *End users can have their documents accessible from everywhere and any device.* Apple iCloud is a service that allows users to have their documents stored in the Cloud and access them from any device users connect to it. This makes it possible to take a picture while traveling with a smartphone, go back home and edit the same picture on your laptop, and have it show as updated on your tablet computer. This process is completely transparent to the user, who does not have to set up cables and connect these devices with each other.

How is all of this made possible? The same concept of IT services on demand—whether computing power, storage, or runtime environments for applications—on a pay-as-you-go basis

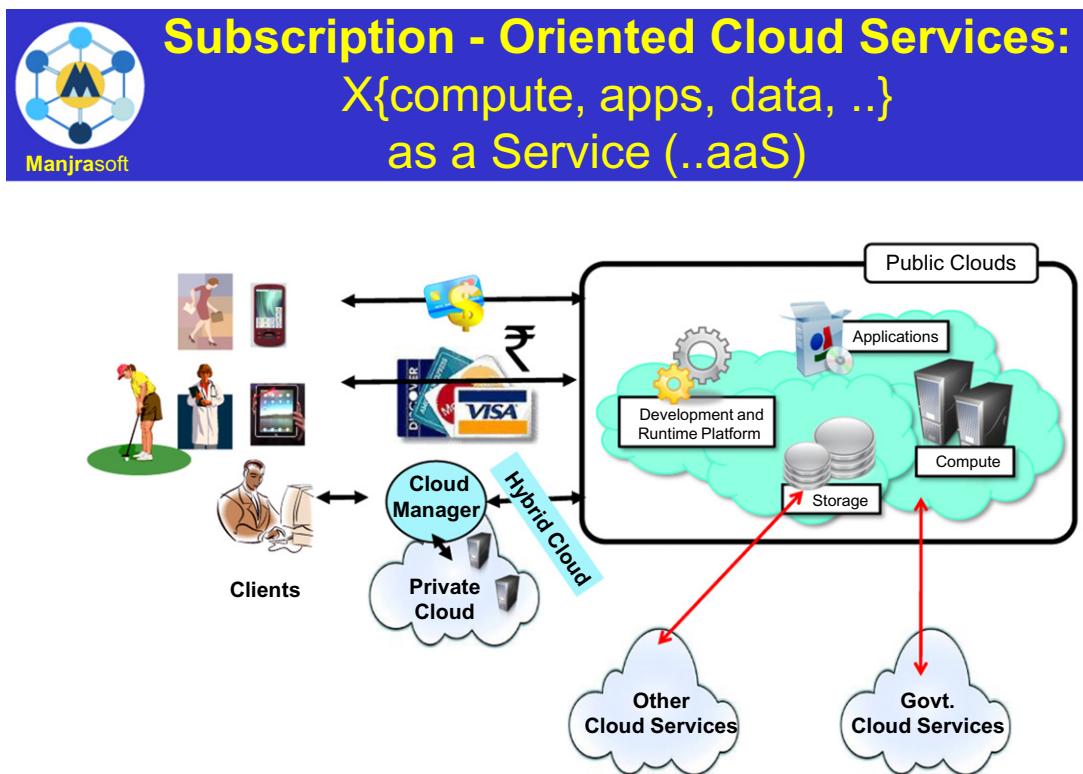
---

<sup>3</sup>It has been reported that Animoto, in one single week, scaled from 70 to 8,500 servers because of user demand.

accommodates these four different scenarios. Cloud computing does not only contribute with the opportunity of easily accessing IT services on demand, it also introduces a new way of thinking about IT services and resources: as utilities. A bird's-eye view of a cloud computing environment is shown in [Figure 1.3](#).

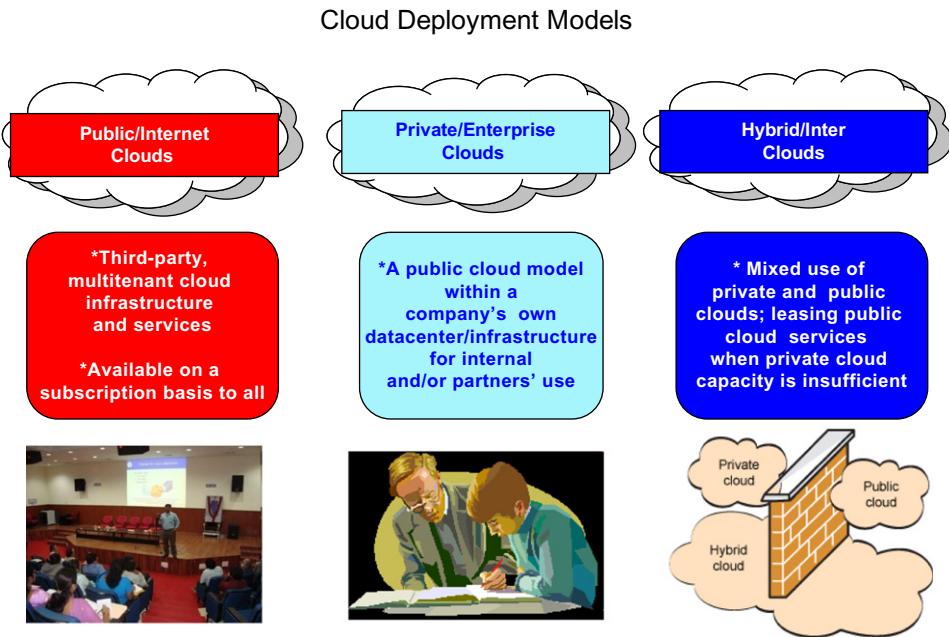
The three major models for deploying and accessing cloud computing environments are public clouds, private/enterprise clouds, and hybrid clouds (see [Figure 1.4](#)). *Public clouds* are the most common deployment models in which necessary IT infrastructure (e.g., virtualized datacenters) is established by a third-party service provider that makes it available to any consumer on a subscription basis. Such clouds are appealing to users because they allow users to quickly leverage compute, storage, and application services. In this environment, users' data and applications are deployed on cloud datacenters on the vendor's premises.

Large organizations that own massive computing infrastructures can still benefit from cloud computing by replicating the cloud IT service delivery model in-house. This idea has given birth to the concept of *private clouds* as opposed to public clouds. In 2010, for example, the U.S. federal government, one of the world's largest consumers of IT spending (around \$76 billion on more than



**FIGURE 1.3**

A bird's-eye view of cloud computing.



**FIGURE 1.4**

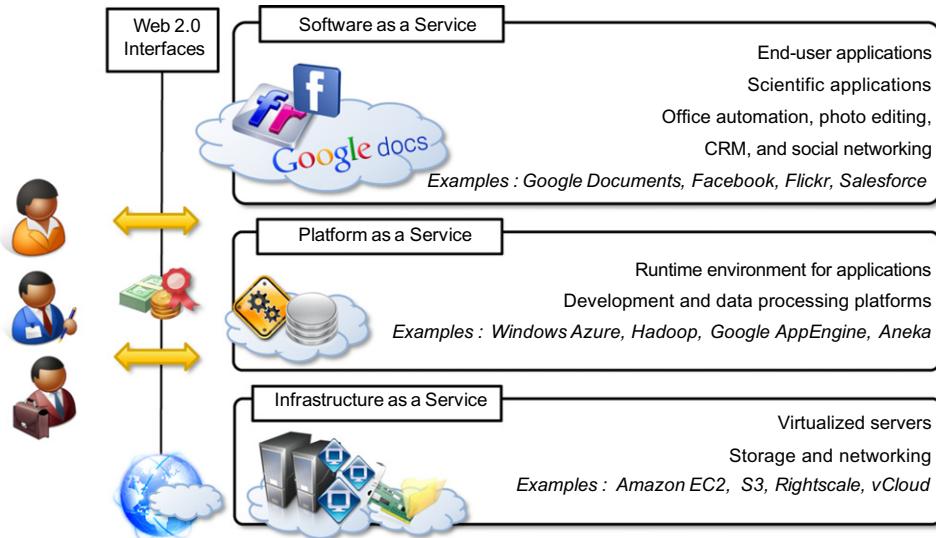
Major deployment models for cloud computing.

10,000 systems) started a cloud computing initiative aimed at providing government agencies with a more efficient use of their computing facilities. The use of cloud-based in-house solutions is also driven by the need to keep confidential information within an organization's premises. Institutions such as governments and banks that have high security, privacy, and regulatory concerns prefer to build and use their own private or enterprise clouds.

Whenever private cloud resources are unable to meet users' quality-of-service requirements, hybrid computing systems, partially composed of public cloud resources and privately owned infrastructures, are created to serve the organization's needs. These are often referred as *hybrid clouds*, which are becoming a common way for many stakeholders to start exploring the possibilities offered by cloud computing.

#### 1.1.4 The cloud computing reference model

A fundamental characteristic of cloud computing is the capability to deliver, on demand, a variety of IT services that are quite diverse from each other. This variety creates different perceptions of what cloud computing is among users. Despite this lack of uniformity, it is possible to classify cloud computing services offerings into three major categories: *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)*. These categories are related to each other as described in Figure 1.5, which provides an organic view of cloud computing. We refer to this diagram as the *Cloud Computing Reference Model*, and we will use it throughout the

**FIGURE 1.5**

The Cloud Computing Reference Model.

book to explain the technologies and introduce the relevant research on this phenomenon. The model organizes the wide range of cloud computing services into a layered view that walks the computing stack from bottom to top.

At the base of the stack, *Infrastructure-as-a-Service* solutions deliver infrastructure on demand in the form of virtual *hardware*, *storage*, and *networking*. Virtual hardware is utilized to provide compute on demand in the form of virtual machine instances. These are created at users' request on the provider's infrastructure, and users are given tools and interfaces to configure the software stack installed in the virtual machine. The pricing model is usually defined in terms of dollars per hour, where the hourly cost is influenced by the characteristics of the virtual hardware. Virtual storage is delivered in the form of raw disk space or object store. The former complements a virtual hardware offering that requires persistent storage. The latter is a more high-level abstraction for storing entities rather than files. Virtual networking identifies the collection of services that manage the networking among virtual instances and their connectivity to the Internet or private networks.

*Platform-as-a-Service* solutions are the next step in the stack. They deliver scalable and elastic runtime environments on demand and host the execution of applications. These services are backed by a core middleware platform that is responsible for creating the abstract environment where applications are deployed and executed. It is the responsibility of the service provider to provide scalability and to manage fault tolerance, while users are requested to focus on the logic of the application developed by leveraging the provider's APIs and libraries. This approach increases the level of abstraction at which cloud computing is leveraged but also constrains the user in a more controlled environment.

At the top of the stack, *Software-as-a-Service* solutions provide applications and services on demand. Most of the common functionalities of desktop applications—such as office

automation, document management, photo editing, and customer relationship management (CRM) software—are replicated on the provider's infrastructure and made more scalable and accessible through a browser on demand. These applications are shared across multiple users whose interaction is isolated from the other users. The SaaS layer is also the area of social networking Websites, which leverage cloud-based infrastructures to sustain the load generated by their popularity.

Each layer provides a different service to users. IaaS solutions are sought by users who want to leverage cloud computing from building dynamically scalable computing systems requiring a specific software stack. IaaS services are therefore used to develop scalable Websites or for background processing. PaaS solutions provide scalable programming platforms for developing applications and are more appropriate when new systems have to be developed. SaaS solutions target mostly end users who want to benefit from the elastic scalability of the cloud without doing any software development, installation, configuration, and maintenance. This solution is appropriate when there are existing SaaS services that fit users needs (such as email, document management, CRM, etc.) and a minimum level of customization is needed.

### 1.1.5 Characteristics and benefits

Cloud computing has some interesting characteristics that bring benefits to both cloud service consumers (CSCs) and cloud service providers (CSPs). These characteristics are:

- No up-front commitments
- On-demand access
- Nice pricing
- Simplified application acceleration and scalability
- Efficient resource allocation
- Energy efficiency
- Seamless creation and use of third-party services

The most evident benefit from the use of cloud computing systems and technologies is the increased economical return due to the reduced maintenance costs and *operational costs* related to IT software and infrastructure. This is mainly because IT assets, namely software and infrastructure, are turned into *utility costs*, which are paid for as long as they are used, not paid for up front. Capital costs are costs associated with assets that need to be paid in advance to start a business activity. Before cloud computing, IT infrastructure and software generated capital costs, since they were paid up front so that business start-ups could afford a computing infrastructure, enabling the business activities of the organization. The revenue of the business is then utilized to compensate over time for these costs. Organizations always minimize capital costs, since they are often associated with depreciable values. This is the case of hardware: a server bought today for \$1,000 will have a market value less than its original price when it is eventually replaced by new hardware. To make profit, organizations have to compensate for this depreciation created by time, thus reducing the net gain obtained from revenue. Minimizing capital costs, then, is fundamental. Cloud computing transforms IT infrastructure and software into utilities, thus significantly contributing to increasing a company's net gain. Moreover, cloud computing also provides an opportunity for small organizations and start-ups: these do not need large investments to start their business, but they can comfortably grow with it. Finally, maintenance costs are significantly reduced: by renting the

infrastructure and the application services, organizations are no longer responsible for their maintenance. This task is the responsibility of the cloud service provider, who, thanks to economies of scale, can bear the maintenance costs.

Increased agility in defining and structuring software systems is another significant benefit of cloud computing. Since organizations rent IT services, they can more dynamically and flexibly compose their software systems, without being constrained by capital costs for IT assets. There is a reduced need for capacity planning, since cloud computing allows organizations to react to unplanned surges in demand quite rapidly. For example, organizations can add more servers to process workload spikes and dismiss them when they are no longer needed. Ease of scalability is another advantage. By leveraging the potentially huge capacity of cloud computing, organizations can extend their IT capability more easily. Scalability can be leveraged across the entire computing stack. Infrastructure providers offer simple methods to provision customized hardware and integrate it into existing systems. Platform-as-a-Service providers offer runtime environment and programming models that are designed to scale applications. Software-as-a-Service offerings can be elastically sized on demand without requiring users to provision hardware or to program application for scalability.

End users can benefit from cloud computing by having their data and the capability of operating on it always available, from anywhere, at any time, and through multiple devices. Information and services stored in the cloud are exposed to users by Web-based interfaces that make them accessible from portable devices as well as desktops at home. Since the processing capabilities (that is, office automation features, photo editing, information management, and so on) also reside in the cloud, end users can perform the same tasks that previously were carried out through considerable software investments. The cost for such opportunities is generally very limited, since the cloud service provider shares its costs across all the tenants that he is servicing. Multitenancy allows for better utilization of the shared infrastructure that is kept operational and fully active. The concentration of IT infrastructure and services into large datacenters also provides opportunity for considerable optimization in terms of resource allocation and energy efficiency, which eventually can lead to a less impacting approach on the environment.

Finally, service orientation and on-demand access create new opportunities for composing systems and applications with a flexibility not possible before cloud computing. New service offerings can be created by aggregating together existing services and concentrating on added value. Since it is possible to provision on demand any component of the computing stack, it is easier to turn ideas into products with limited costs and by concentrating technical efforts on what matters: the added value.

### 1.1.6 Challenges ahead

As any new technology develops and becomes popular, new issues have to be faced. Cloud computing is not an exception. New, interesting problems and challenges are regularly being posed to the cloud community, including IT practitioners, managers, governments, and regulators.

Besides the practical aspects, which are related to configuration, networking, and sizing of cloud computing systems, a new set of challenges concerning the dynamic provisioning of cloud computing services and resources arises. For example, in the Infrastructure-as-a-Service domain, how many resources need to be provisioned, and for how long should they be used, in order to maximize the benefit? Technical challenges also arise for cloud service providers for the management of large computing infrastructures and the use of virtualization technologies on top of them. In

addition, issues and challenges concerning the integration of real and virtual infrastructure need to be taken into account from different perspectives, such as security and legislation.

Security in terms of confidentiality, secrecy, and protection of data in a cloud environment is another important challenge. Organizations do not own the infrastructure they use to process data and store information. This condition poses challenges for confidential data, which organizations cannot afford to reveal. Therefore, assurance on the confidentiality of data and compliance to security standards, which give a minimum guarantee on the treatment of information on cloud computing systems, are sought. The problem is not as evident as it seems: even though cryptography can help secure the transit of data from the private premises to the cloud infrastructure, in order to be processed the information needs to be decrypted in memory. This is the weak point of the chain: since virtualization allows capturing almost transparently the memory pages of an instance, these data could easily be obtained by a malicious provider.

Legal issues may also arise. These are specifically tied to the ubiquitous nature of cloud computing, which spreads computing infrastructure across diverse geographical locations. Different legislation about privacy in different countries may potentially create disputes as to the rights that third parties (including government agencies) have to your data. U.S. legislation is known to give extreme powers to government agencies to acquire confidential data when there is the suspicion of operations leading to a threat to national security. European countries are more restrictive and protect the right of privacy. An interesting scenario comes up when a U.S. organization uses cloud services that store their data in Europe. In this case, should this organization be suspected by the government, it would become difficult or even impossible for the U.S. government to take control of the data stored in a cloud datacenter located in Europe.

---

## 1.2 Historical developments

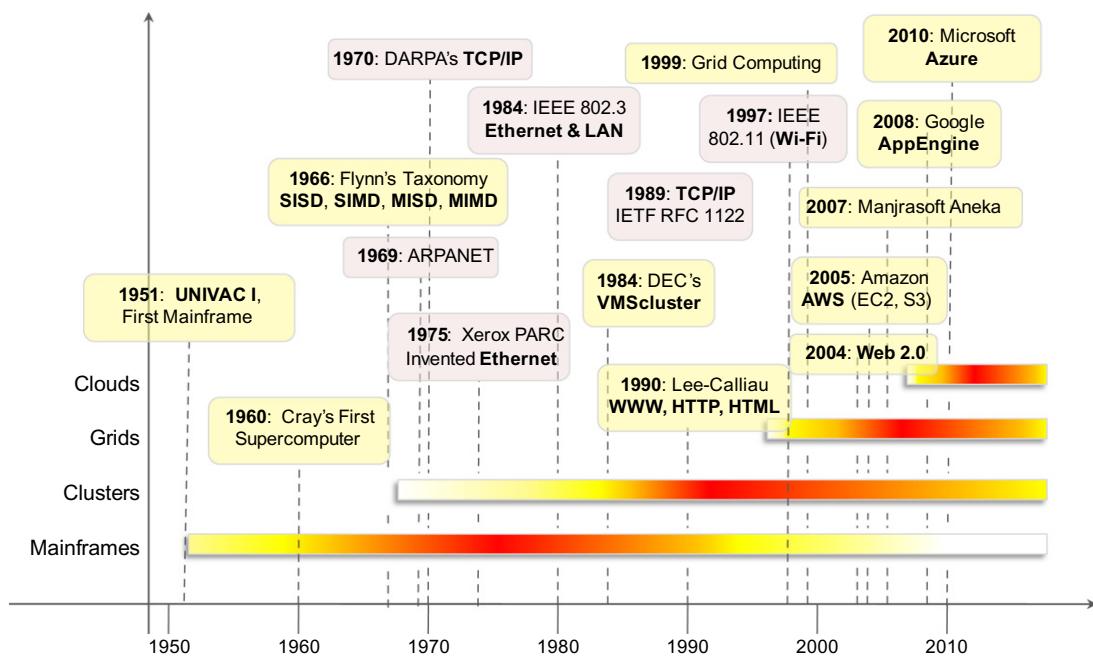
The idea of renting computing services by leveraging large distributed computing facilities has been around for long time. It dates back to the days of the mainframes in the early 1950s. From there on, technology has evolved and been refined. This process has created a series of favorable conditions for the realization of cloud computing.

Figure 1.6 provides an overview of the evolution of the distributed computing technologies that have influenced cloud computing. In tracking the historical evolution, we briefly review five core technologies that played an important role in the realization of cloud computing. These technologies are distributed systems, virtualization, Web 2.0, service orientation, and utility computing.

### 1.2.1 Distributed systems

Clouds are essentially large distributed computing facilities that make available their services to third parties on demand. As a reference, we consider the characterization of a distributed system proposed by Tanenbaum et al. [1]:

*A distributed system is a collection of independent computers that appears to its users as a single coherent system.*

**FIGURE 1.6**

The evolution of distributed computing technologies, 1950s–2010s.

This is a general definition that includes a variety of computer systems, but it evidences two very important elements characterizing a distributed system: the fact that it is composed of multiple independent components and that these components are perceived as a single entity by users. This is particularly true in the case of cloud computing, in which clouds hide the complex architecture they rely on and provide a single interface to users. The primary purpose of distributed systems is to share resources and utilize them better. This is true in the case of cloud computing, where this concept is taken to the extreme and resources (infrastructure, runtime environments, and services) are rented to users. In fact, one of the driving factors of cloud computing has been the availability of the large computing facilities of IT giants (Amazon, Google) that found that offering their computing capabilities as a service provided opportunities to better utilize their infrastructure. Distributed systems often exhibit other properties such as *heterogeneity*, *openness*, *scalability*, *transparency*, *concurrency*, *continuous availability*, and *independent failures*. To some extent these also characterize clouds, especially in the context of scalability, concurrency, and continuous availability.

Three major milestones have led to cloud computing: mainframe computing, cluster computing, and grid computing.

- **Mainframes.** These were the first examples of large computational facilities leveraging multiple processing units. Mainframes were powerful, highly reliable computers specialized for large

data movement and massive input/output (I/O) operations. They were mostly used by large organizations for bulk data processing tasks such as online transactions, enterprise resource planning, and other operations involving the processing of significant amounts of data. Even though mainframes cannot be considered distributed systems, they offered large computational power by using multiple processors, which were presented as a single entity to users. One of the most attractive features of mainframes was the ability to be highly reliable computers that were “always on” and capable of tolerating failures transparently. No system shutdown was required to replace failed components, and the system could work without interruption. Batch processing was the main application of mainframes. Now their popularity and deployments have reduced, but evolved versions of such systems are still in use for transaction processing (such as online banking, airline ticket booking, supermarket and telcos, and government services).

- *Clusters.* Cluster computing [3][4] started as a low-cost alternative to the use of mainframes and supercomputers. The technology advancement that created faster and more powerful mainframes and supercomputers eventually generated an increased availability of cheap commodity machines as a side effect. These machines could then be connected by a high-bandwidth network and controlled by specific software tools that manage them as a single system. Starting in the 1980s, clusters became the standard technology for parallel and high-performance computing. Built by commodity machines, they were cheaper than mainframes and made high-performance computing available to a large number of groups, including universities and small research labs. Cluster technology contributed considerably to the evolution of tools and frameworks for distributed computing, including Condor [5], Parallel Virtual Machine (PVM) [6], and Message Passing Interface (MPI) [7].<sup>4</sup> One of the attractive features of clusters was that the computational power of commodity machines could be leveraged to solve problems that were previously manageable only on expensive supercomputers. Moreover, clusters could be easily extended if more computational power was required.
- *Grids.* Grid computing [8] appeared in the early 1990s as an evolution of cluster computing. In an analogy to the power grid, grid computing proposed a new approach to access large computational power, huge storage facilities, and a variety of services. Users can “consume” resources in the same way as they use other utilities such as power, gas, and water. Grids initially developed as aggregations of geographically dispersed clusters by means of Internet connections. These clusters belonged to different organizations, and arrangements were made among them to share the computational power. Different from a “large cluster,” a computing grid was a dynamic aggregation of heterogeneous computing nodes, and its scale was nationwide or even worldwide. Several developments made possible the diffusion of computing grids: (a) clusters became quite common resources; (b) they were often underutilized; (c) new problems were requiring computational power that went beyond the capability of single clusters; and (d) the improvements in networking and the diffusion of the Internet made possible long-distance, high-bandwidth connectivity. All these elements led to the development of grids, which now serve a multitude of users across the world.

---

<sup>4</sup>MPI is a specification for an API that allows many computers to communicate with one another. It defines a language-independent protocol that supports point-to-point and collective communication. MPI has been designed for high performance, scalability, and portability. At present, it is one of the dominant paradigms for developing parallel applications.

Cloud computing is often considered the successor of grid computing. In reality, it embodies aspects of all these three major technologies. Computing clouds are deployed in large datacenters hosted by a single organization that provides services to others. Clouds are characterized by the fact of having virtually infinite capacity, being tolerant to failures, and being always on, as in the case of mainframes. In many cases, the computing nodes that form the infrastructure of computing clouds are commodity machines, as in the case of clusters. The services made available by a cloud vendor are consumed on a pay-per-use basis, and clouds fully implement the utility vision introduced by grid computing.

### 1.2.2 Virtualization

*Virtualization* is another core technology for cloud computing. It encompasses a collection of solutions allowing the abstraction of some of the fundamental elements for computing, such as hardware, runtime environments, storage, and networking. Virtualization has been around for more than 40 years, but its application has always been limited by technologies that did not allow an efficient use of virtualization solutions. Today these limitations have been substantially overcome, and virtualization has become a fundamental element of cloud computing. This is particularly true for solutions that provide IT infrastructure on demand. Virtualization confers that degree of customization and control that makes cloud computing appealing for users and, at the same time, sustainable for cloud services providers.

Virtualization is essentially a technology that allows creation of different computing environments. These environments are called *virtual* because they simulate the interface that is expected by a guest. The most common example of virtualization is *hardware virtualization*. This technology allows simulating the hardware interface expected by an operating system. Hardware virtualization allows the coexistence of different software stacks on top of the same hardware. These stacks are contained inside *virtual machine instances*, which operate in complete isolation from each other. High-performance servers can host several virtual machine instances, thus creating the opportunity to have a customized software stack on demand. This is the base technology that enables cloud computing solutions to deliver virtual servers on demand, such as Amazon EC2, RightScale, VMware vCloud, and others. Together with hardware virtualization, *storage* and *network virtualization* complete the range of technologies for the emulation of IT infrastructure.

Virtualization technologies are also used to replicate runtime environments for programs. Applications in the case of *process virtual machines* (which include the foundation of technologies such as Java or .NET), instead of being executed by the operating system, are run by a specific program called a *virtual machine*. This technique allows isolating the execution of applications and providing a finer control on the resource they access. Process virtual machines offer a higher level of abstraction with respect to hardware virtualization, since the guest is only constituted by an application rather than a complete software stack. This approach is used in cloud computing to provide a platform for scaling applications on demand, such as Google AppEngine and Windows Azure.

Having isolated and customizable environments with minor impact on performance is what makes virtualization a attractive technology. Cloud computing is realized through platforms that leverage the basic concepts described above and provides on demand virtualization services to a multitude of users across the globe.

### 1.2.3 Web 2.0

The Web is the primary interface through which cloud computing delivers its services. At present, the Web encompasses a set of technologies and services that facilitate interactive information sharing, collaboration, user-centered design, and application composition. This evolution has transformed the Web into a rich platform for application development and is known as *Web 2.0*. This term captures a new way in which developers architect applications and deliver services through the Internet and provides new experience for users of these applications and services.

Web 2.0 brings *interactivity* and *flexibility* into Web pages, providing enhanced user experience by gaining Web-based access to all the functions that are normally found in desktop applications. These capabilities are obtained by integrating a collection of standards and technologies such as *XML*, *Asynchronous JavaScript and XML (AJAX)*, *Web Services*, and others. These technologies allow us to build applications leveraging the contribution of users, who now become providers of content. Furthermore, the capillary diffusion of the Internet opens new opportunities and markets for the Web, the services of which can now be accessed from a variety of devices: mobile phones, car dashboards, TV sets, and others. These new scenarios require an increased dynamism for applications, which is another key element of this technology. Web 2.0 applications are extremely dynamic: they improve continuously, and new updates and features are integrated at a constant rate by following the usage trend of the community. There is no need to deploy new software releases on the installed base at the client side. Users can take advantage of the new software features simply by interacting with cloud applications. Lightweight deployment and programming models are very important for effective support of such dynamism. Loose coupling is another fundamental property. New applications can be “synthesized” simply by composing existing services and integrating them, thus providing added value. This way it becomes easier to follow the interests of users. Finally, Web 2.0 applications aim to leverage the “long tail” of Internet users by making themselves available to everyone in terms of either media accessibility or affordability.

Examples of Web 2.0 applications are *Google Documents*, *Google Maps*, *Flickr*, *Facebook*, *Twitter*, *YouTube*, *de.li.cious*, *Blogger*, and *Wikipedia*. In particular, social networking Websites take the biggest advantage of Web 2.0. The level of interaction in Websites such as Facebook or Flickr would not have been possible without the support of AJAX, Really Simple Syndication (RSS), and other tools that make the user experience incredibly interactive. Moreover, community Websites harness the collective intelligence of the community, which provides content to the applications themselves: Flickr provides advanced services for storing digital pictures and videos, Facebook is a social networking site that leverages user activity to provide content, and Blogger, like any other blogging site, provides an online diary that is fed by users.

This idea of the Web as a transport that enables and enhances interaction was introduced in 1999 by Darcy DiNucci<sup>5</sup> and started to become fully realized in 2004. Today it is a mature platform for supporting the needs of cloud computing, which strongly leverages Web 2.0. Applications

---

<sup>5</sup>In a column for *Design & New Media* magazine, Darci DiNucci describes the Web as follows: “The Web we know now, which loads into a browser window in essentially static screenfulls, is only an embryo of the Web to come. The first glimmerings of Web 2.0 are beginning to appear, and we are just starting to see how that embryo might develop. The Web will be understood not as screenfulls of text and graphics but as a transport mechanism, the ether through which interactivity happens. It will [...] appear on your computer screen, [...] on your TV set [...], your car dashboard [...], your cell phone [...], hand-held game machines [...], maybe even your microwave oven.”

and frameworks for delivering *rich Internet applications (RIAs)* are fundamental for making cloud services accessible to the wider public. From a social perspective, Web 2.0 applications definitely contributed to making people more accustomed to the use of the Internet in their everyday lives and opened the path to the acceptance of cloud computing as a paradigm, whereby even the IT infrastructure is offered through a Web interface.

### 1.2.4 Service-oriented computing

*Service orientation* is the core reference model for cloud computing systems. This approach adopts the concept of services as the main building blocks of application and system development. *Service-oriented computing (SOC)* supports the development of rapid, low-cost, flexible, interoperable, and evolvable applications and systems [19].

A *service* is an abstraction representing a self-describing and platform-agnostic component that can perform any function—anything from a simple function to a complex business process. Virtually any piece of code that performs a task can be turned into a service and expose its functionalities through a network-accessible protocol. A service is supposed to be *loosely coupled, reusable, programming language independent*, and *location transparent*. Loose coupling allows services to serve different scenarios more easily and makes them reusable. Independence from a specific platform increases services accessibility. Thus, a wider range of clients, which can look up services in global registries and consume them in a location-transparent manner, can be served. Services are composed and aggregated into a *service-oriented architecture (SOA)* [27], which is a logical way of organizing software systems to provide end users or other entities distributed over the network with services through published and discoverable interfaces.

Service-oriented computing introduces and diffuses two important concepts, which are also fundamental to cloud computing: *quality of service (QoS)* and *Software-as-a-Service (SaaS)*.

- Quality of service (QoS) identifies a set of functional and nonfunctional attributes that can be used to evaluate the behavior of a service from different perspectives. These could be performance metrics such as response time, or security attributes, transactional integrity, reliability, scalability, and availability. QoS requirements are established between the client and the provider via an SLA that identifies the minimum values (or an acceptable range) for the QoS attributes that need to be satisfied upon the service call.
- The concept of Software-as-a-Service introduces a new delivery model for applications. The term has been inherited from the world of application service providers (ASPs), which deliver software services-based solutions across the wide area network from a central datacenter and make them available on a subscription or rental basis. The ASP is responsible for maintaining the infrastructure and making available the application, and the client is freed from maintenance costs and difficult upgrades. This software delivery model is possible because economies of scale are reached by means of multitenancy. The SaaS approach reaches its full development with service-oriented computing (SOC), where loosely coupled software components can be exposed and priced singularly, rather than entire applications. This allows the delivery of complex business processes and transactions as a service while allowing applications to be composed on the fly and services to be reused from everywhere and by anybody.

One of the most popular expressions of service orientation is represented by Web Services (WS) [21]. These introduce the concepts of SOC into the World Wide Web, by making it consumable by applications and not only humans. Web services are software components that expose functionalities accessible using a method invocation pattern that goes over the HyperText Transfer Protocol (HTTP). The interface of a Web service can be programmatically inferred by metadata expressed through the *Web Service Description Language (WSDL)* [22]; this is an XML language that defines the characteristics of the service and all the methods, together with parameters, descriptions, and return type, exposed by the service. The interaction with Web services happens through *Simple Object Access Protocol (SOAP)* [23]. This is an XML language that defines how to invoke a Web service method and collect the result. Using SOAP and WSDL over HTTP, Web services become platform independent and accessible to the World Wide Web. The standards and specifications concerning Web services are controlled by the World Wide Web Consortium (W3C). Among the most popular architectures for developing Web services we can note ASP.NET [24] and Axis [25].

The development of systems in terms of distributed services that can be composed together is the major contribution given by SOC to the realization of cloud computing. Web services technologies have provided the right tools to make such composition straightforward and easily integrated with the mainstream World Wide Web (WWW) environment.

### 1.2.5 Utility-oriented computing

*Utility computing* is a vision of computing that defines a service-provisioning model for compute services in which resources such as storage, compute power, applications, and infrastructure are packaged and offered on a pay-per-use basis. The idea of providing computing as a *utility* like natural gas, water, power, and telephone connection has a long history but has become a reality today with the advent of cloud computing. Among the earliest forerunners of this vision we can include the American scientist John McCarthy, who, in a speech for the Massachusetts Institute of Technology (MIT) centennial in 1961, observed:

*If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility, just as the telephone system is a public utility ... The computer utility could become the basis of a new and important industry.*

The first traces of this service-provisioning model can be found in the mainframe era. IBM and other mainframe providers offered mainframe power to organizations such as banks and government agencies throughout their datacenters. The business model introduced with utility computing brought new requirements and led to improvements in mainframe technology: additional features such as operating systems, process control, and user-metering facilities. The idea of computing as utility remained and extended from the business domain to academia with the advent of cluster computing. Not only businesses but also research institutes became acquainted with the idea of leveraging an external IT infrastructure on demand. Computational science, which was one of the major driving factors for building computing clusters, still required huge compute power for addressing “Grand Challenge” problems, and not all the institutions were able to satisfy their computing needs internally. Access to external clusters still remained a common practice. The capillary diffusion of the Internet and the Web provided the technological means to realize utility computing on a

worldwide scale and through simple interfaces. As already discussed, computing grids provided a planet-scale distributed computing infrastructure that was accessible on demand. Computing grids brought the concept of utility computing to a new level: market orientation [15]. With utility computing accessible on a wider scale, it is easier to provide a trading infrastructure where grid products—storage, computation, and services—are bid for or sold. Moreover, e-commerce technologies [25] provided the infrastructure support for utility computing. In the late 1990s a significant interest in buying any kind of good online spread to the wider public: food, clothes, multimedia products, and online services such as storage space and Web hosting. After the *dot-com bubble*<sup>6</sup> burst, this interest reduced in size, but the phenomenon made the public keener to buy online services. As a result, infrastructures for online payment using credit cards became easily accessible and well proven.

From an application and system development perspective, service-oriented computing and *service-oriented architectures (SOAs)* introduced the idea of leveraging external services for performing a specific task within a software system. Applications were not only distributed, they started to be composed as a mesh of services provided by different entities. These services, accessible through the Internet, were made available by charging according to usage. SOC broadened the concept of what could have been accessed as a utility in a computer system: not only compute power and storage but also services and application components could be utilized and integrated on demand. Together with this trend, QoS became an important topic to investigate.

All these factors contributed to the development of the concept of utility computing and offered important steps in the realization of cloud computing, in which the vision of computing utilities comes to its full expression.

---

## 1.3 Building cloud computing environments

The creation of cloud computing environments encompasses both the development of applications and systems that leverage cloud computing solutions and the creation of frameworks, platforms, and infrastructures delivering cloud computing services.

### 1.3.1 Application development

Applications that leverage cloud computing benefit from its capability to dynamically scale on demand. One class of applications that takes the biggest advantage of this feature is that of *Web applications*. Their performance is mostly influenced by the workload generated by varying user demands. With the diffusion of Web 2.0 technologies, the Web has become a platform for developing rich and complex applications, including *enterprise applications* that now leverage the Internet as the preferred channel for service delivery and user interaction. These applications are

---

<sup>6</sup>The dot-com bubble was a phenomenon that started in the second half of the 1990s and reached its apex in 2000. During this period a large number of companies that based their business on online services and e-commerce started and quickly expanded without later being able to sustain their growth. As a result they suddenly went bankrupt, partly because their revenues were not enough to cover their expenses and partly because they never reached the required number of customers to sustain their enlarged business.

characterized by complex processes that are triggered by the interaction with users and develop through the interaction between several tiers behind the Web front end. These are the applications that are mostly sensible to inappropriate sizing of infrastructure and service deployment or variability in workload.

Another class of applications that can potentially gain considerable advantage by leveraging cloud computing is represented by *resource-intensive applications*. These can be either data-intensive or compute-intensive applications. In both cases, considerable amounts of resources are required to complete execution in a reasonable timeframe. It is worth noticing that these large amounts of resources are not needed constantly or for a long duration. For example, *scientific applications* can require huge computing capacity to perform large-scale experiments once in a while, so it is not feasible to buy the infrastructure supporting them. In this case, cloud computing can be the solution. Resource-intensive applications are not interactive and they are mostly characterized by batch processing.

Cloud computing provides a solution for on-demand and dynamic scaling across the entire stack of computing. This is achieved by (a) providing methods for renting compute power, storage, and networking; (b) offering runtime environments designed for scalability and dynamic sizing; and (c) providing application services that mimic the behavior of desktop applications but that are completely hosted and managed on the provider side. All these capabilities leverage service orientation, which allows a simple and seamless integration into existing systems. Developers access such services via simple Web interfaces, often implemented through representational state transfer (REST) Web services. These have become well-known abstractions, making the development and management of cloud applications and systems practical and straightforward.

### 1.3.2 Infrastructure and system development

Distributed computing, virtualization, service orientation, and Web 2.0 form the core technologies enabling the provisioning of cloud services from anywhere on the globe. Developing applications and systems that leverage the cloud requires knowledge across all these technologies. Moreover, new challenges need to be addressed from design and development standpoints.

Distributed computing is a foundational model for cloud computing because cloud systems are distributed systems. Besides administrative tasks mostly connected to the accessibility of resources in the cloud, the extreme dynamism of cloud systems—where new nodes and services are provisioned on demand—constitutes the major challenge for engineers and developers. This characteristic is pretty peculiar to cloud computing solutions and is mostly addressed at the middleware layer of computing system. Infrastructure-as-a-Service solutions provide the capabilities to add and remove resources, but it is up to those who deploy systems on this scalable infrastructure to make use of such opportunities with wisdom and effectiveness. Platform-as-a-Service solutions embed into their core offering algorithms and rules that control the provisioning process and the lease of resources. These can be either completely transparent to developers or subject to fine control. Integration between cloud resources and existing system deployment is another element of concern.

Web 2.0 technologies constitute the interface through which cloud computing services are delivered, managed, and provisioned. Besides the interaction with rich interfaces through the Web browser, Web services have become the primary access point to cloud computing systems from a

programmatic standpoint. Therefore, service orientation is the underlying paradigm that defines the architecture of a cloud computing system. Cloud computing is often summarized with the acronym *XaaS—Everything-as-a-Service*—that clearly underlines the central role of service orientation. Despite the absence of a unique standard for accessing the resources serviced by different cloud providers, the commonality of technology smoothes the learning curve and simplifies the integration of cloud computing into existing systems.

Virtualization is another element that plays a fundamental role in cloud computing. This technology is a core feature of the infrastructure used by cloud providers. As discussed before, the virtualization concept is more than 40 years old, but cloud computing introduces new challenges, especially in the management of virtual environments, whether they are abstractions of virtual hardware or a runtime environment. Developers of cloud applications need to be aware of the limitations of the selected virtualization technology and the implications on the volatility of some components of their systems.

These are all considerations that influence the way we program applications and systems based on cloud computing technologies. Cloud computing essentially provides mechanisms to address surges in demand by replicating the required components of computing systems under stress (i.e., heavily loaded). Dynamism, scale, and volatility of such components are the main elements that should guide the design of such systems.

### 1.3.3 Computing platforms and technologies

Development of a cloud computing application happens by leveraging platforms and frameworks that provide different types of services, from the bare-metal infrastructure to customizable applications serving specific purposes.

#### 1.3.3.1 Amazon web services (AWS)

AWS offers comprehensive cloud IaaS services ranging from virtual compute, storage, and networking to complete computing stacks. AWS is mostly known for its compute and storage-on-demand services, namely *Elastic Compute Cloud (EC2)* and *Simple Storage Service (S3)*. EC2 provides users with customizable virtual hardware that can be used as the base infrastructure for deploying computing systems on the cloud. It is possible to choose from a large variety of virtual hardware configurations, including GPU and cluster instances. EC2 instances are deployed either by using the AWS console, which is a comprehensive Web portal for accessing AWS services, or by using the Web services API available for several programming languages. EC2 also provides the capability to save a specific running instance as an image, thus allowing users to create their own templates for deploying systems. These templates are stored into S3 that delivers persistent storage on demand. S3 is organized into buckets; these are containers of objects that are stored in binary form and can be enriched with attributes. Users can store objects of any size, from simple files to entire disk images, and have them accessible from everywhere.

Besides EC2 and S3, a wide range of services can be leveraged to build virtual computing systems, including networking support, caching systems, DNS, database (relational and not) support, and others.

### 1.3.3.2 Google AppEngine

Google AppEngine is a scalable runtime environment mostly devoted to executing Web applications. These take advantage of the large computing infrastructure of Google to dynamically scale as the demand varies over time. AppEngine provides both a secure execution environment and a collection of services that simplify the development of scalable and high-performance Web applications. These services include in-memory caching, scalable data store, job queues, messaging, and cron tasks. Developers can build and test applications on their own machines using the AppEngine software development kit (SDK), which replicates the production runtime environment and helps test and profile applications. Once development is complete, developers can easily migrate their application to AppEngine, set quotas to contain the costs generated, and make the application available to the world. The languages currently supported are Python, Java, and Go.

### 1.3.3.3 Microsoft Azure

Microsoft Azure is a cloud operating system and a platform for developing applications in the cloud. It provides a scalable runtime environment for Web applications and distributed applications in general. Applications in Azure are organized around the concept of roles, which identify a distribution unit for applications and embody the application's logic. Currently, there are three types of role: *Web role*, *worker role*, and *virtual machine role*. The Web role is designed to host a Web application, the worker role is a more generic container of applications and can be used to perform workload processing, and the virtual machine role provides a virtual environment in which the computing stack can be fully customized, including the operating systems. Besides roles, Azure provides a set of additional services that complement application execution, such as support for storage (relational data and blobs), networking, caching, content delivery, and others.

### 1.3.3.4 Hadoop

Apache Hadoop is an open-source framework that is suited for processing large data sets on commodity hardware. Hadoop is an implementation of MapReduce, an application programming model developed by Google, which provides two fundamental operations for data processing: *map* and *reduce*. The former transforms and synthesizes the input data provided by the user; the latter aggregates the output obtained by the map operations. Hadoop provides the runtime environment, and developers need only provide the input data and specify the map and reduce functions that need to be executed. Yahoo!, the sponsor of the Apache Hadoop project, has put considerable effort into transforming the project into an enterprise-ready cloud computing platform for data processing. Hadoop is an integral part of the Yahoo! cloud infrastructure and supports several business processes of the company. Currently, Yahoo! manages the largest Hadoop cluster in the world, which is also available to academic institutions.

### 1.3.3.5 Force.com and Salesforce.com

[Force.com](#) is a cloud computing platform for developing social enterprise applications. The platform is the basis for [SalesForce.com](#), a Software-as-a-Service solution for customer relationship management. [Force.com](#) allows developers to create applications by composing ready-to-use blocks; a complete set of components supporting all the activities of an enterprise are available. It is also possible to develop your own components or integrate those available in *AppExchange* into your applications. The platform provides complete support for developing applications, from the

design of the data layout to the definition of business rules and workflows and the definition of the user interface. The [Force.com](#) platform is completely hosted on the cloud and provides complete access to its functionalities and those implemented in the hosted applications through Web services technologies.

#### 1.3.3.6 Manjrasoft Aneka

Manjrasoft Aneka [165] is a cloud application platform for rapid creation of scalable applications and their deployment on various types of clouds in a seamless and elastic manner. It supports a collection of programming abstractions for developing applications and a distributed runtime environment that can be deployed on heterogeneous hardware (clusters, networked desktop computers, and cloud resources). Developers can choose different abstractions to design their application: *tasks*, *distributed threads*, and *map-reduce*. These applications are then executed on the distributed service-oriented runtime environment, which can dynamically integrate additional resource on demand. The service-oriented architecture of the runtime has a great degree of flexibility and simplifies the integration of new features, such as abstraction of a new programming model and associated execution management environment. Services manage most of the activities happening at runtime: scheduling, execution, accounting, billing, storage, and quality of service.

These platforms are key examples of technologies available for cloud computing. They mostly fall into the three major market segments identified in the reference model: *Infrastructure-as-a-Service*, *Platform-as-a-Service*, and *Software-as-a-Service*. In this book, we use Aneka as a reference platform for discussing practical implementations of distributed applications. We present different ways in which clouds can be leveraged by applications built using the various programming models and abstractions provided by Aneka.

## SUMMARY

In this chapter, we discussed the vision and opportunities of cloud computing along with its characteristics and challenges. The cloud computing paradigm emerged as a result of the maturity and convergence of several of its supporting models and technologies, namely distributed computing, virtualization, Web 2.0, service orientation, and utility computing.

There is no single view on the cloud phenomenon. Throughout the book, we explore different definitions, interpretations, and implementations of this idea. The only element that is shared among all the different views of cloud computing is that cloud systems support dynamic provisioning of IT services (whether they are virtual infrastructure, runtime environments, or application services) and adopts a utility-based cost model to price these services. This concept is applied across the entire computing stack and enables the dynamic provisioning of IT infrastructure and runtime environments in the form of cloud-hosted platforms for the development of scalable applications and their services. This vision is what inspires the *Cloud Computing Reference Model*. This model identifies three major market segments (and service offerings) for cloud computing: *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)*. These segments directly map the broad classifications of the different type of services offered by cloud computing.

The long-term vision of cloud computing is to fully realize the utility model that drives its service offering. It is envisioned that new technological developments and the increased familiarity with cloud computing delivery models will lead to the establishment of a global market for trading computing utilities. This area of study is called *market-oriented cloud computing*, where the term *market-oriented* further stresses the fact that cloud computing services are traded as utilities. The realization of this vision is still far from reality, but cloud computing has already brought economic, environmental, and technological benefits. By turning IT assets into utilities, it allows organizations to reduce operational costs and increase revenues. This and other advantages also have downsides that are diverse in nature. Security and legislation are two of the challenging aspects of cloud computing that are beyond the technical sphere.

From the perspective of software design and development, new challenges arise in engineering computing systems. Cloud computing offers a rich mixture of different technologies, and harnessing them is a challenging engineering task. Cloud computing introduces both new opportunities and new techniques and strategies for architecting software applications and systems. Some of the key elements that have to be taken into account are virtualization, scalability, dynamic provisioning, big datasets, and cost models. To provide a practical grasp of such concepts, we will use Aneka as a reference platform for illustrating cloud systems and application programming environments.

---

## Review questions

1. What is the innovative characteristic of cloud computing?
2. Which are the technologies on which cloud computing relies?
3. Provide a brief characterization of a distributed system.
4. Define cloud computing and identify its core features.
5. What are the major distributed computing technologies that led to cloud computing?
6. What is virtualization?
7. What is the major revolution introduced by Web 2.0?
8. Give some examples of Web 2.0 applications.
9. Describe the main characteristics of a service orientation.
10. What is utility computing?
11. Describe the vision introduced by cloud computing.
12. Briefly summarize the Cloud Computing Reference Model.
13. What is the major advantage of cloud computing?
14. Briefly summarize the challenges still open in cloud computing.
15. How is cloud development different from traditional software development?

This page intentionally left blank