A. Lisdorf, *Cloud Computing Basics*
https://doi.org/10.1007/978-1-4842-6921-3_1

1. Cloud Foundations

Anders Lisdorf[1]
(1) Copenhagen, Denmark

In this chapter, we lay the foundation for understanding what the cloud actually is. We start by considering the history of the term. Then we look at how to define it in order to get a firmer handle on what we mean by cloud computing. We also investigate different ways to conceptualize the cloud. It has, for example, been suggested that the cloud is a utility or a service, which highlights certain important aspects of the cloud but also mischaracterizes it in other ways. The chapter seeks to establish an understanding of what it means when we talk about the *cloud*.

The History of the Term “Cloud”

Many ideas come together in the cloud. They can be traced from multiple sources, which we do in a later chapter, but the origin of the term “cloud” in itself is to many people a bit mysterious. Why would a cloud convey an idea about technology being accessed through a network? The idea of a communication network precedes the talk of the cloud. The first mention of the cloud was in the 90s, but before that it was already customary for engineers to use a cloud to denote a network.

Why a Cloud?

To understand why the cloud metaphor came to denote the technological revolution we are now seeing, we have to look at what a network actually is from an engineering perspective. When you connect multiple computers in a network, this happens through multiple nodes. These are computers or other electronic devices that have the capability to connect to other devices or computers and route communication traffic between computers. Every node in the network does little more than receive data from one end point and route it to another. The familiar radio towers that we see if we look hard around us are all just such nodes that form a network that our cell phones can use to connect to other cell phones.

The identity and connections of these nodes, how they route traffic, is not important in order to understand the properties of the network. If you had to describe how one cell phone connects to another in a diagram, it would be inconvenient to document all the radio towers that form the network through which these phones communicate.

Rather than try to capture all these details and entities that constitute the network, a metaphor was needed to symbolize an amorphous collection of machines. What better symbol of something amorphous than a cloud? This was also relatively easy to sketch out by hand. Engineers often communicate ideas by drawing sketches on whiteboards or on paper. Drawing a cloud was a convenient and fast way to illustrate that something connected to a network to which other things were connected as well.

The Origin of the Term “Cloud”

The first public mention of the cloud was, as far as we can tell, in 1994 in a *Wired* article by author and journalist Steven Levy. The article was about a company called General Magic. The company was founded by the creators of the Macintosh computer, Bill Atkinson and Andy Herzfeld. By 1994, they had spent four years trying to create an ambitious new communications platform to change everyone's life. They imagined that "a lot of different areas are converging on an electronic box that's in your pocket, that's with you all the time, that supports you in some way." Obviously at that time no "electronic box" would be able to do much computing that would be very interesting, so that had to be done on a server to which the device connected through a network. The product General Magic envisioned was called Telescript and was the interface people would use to connect. It was a system to tie together all sorts of different networks into one standardized interface. They imagined that:

> *"Now, instead of just having a device to program, we now have the entire Cloud out there, where a single program can go and travel to many different sources of information and create sort of a virtual service".*
>
> *—Bill and Andy's Excellent Adventure II, Wired 1994*

It is interesting that in this quote we see not only the word "cloud" but also clearly the contours of the concept of the cloud: the ability to access information and functionality from any device at any location.

The particular technology offered by General Magic, however, did not catch on. Even though they were visionaries in the field, the company ceased operations and was liquidated at the start of the 2000s, before cloud computing really caught on. The word "cloud" was not adopted by the general public immediately.

It wasn't until 1996 that a major company took the concept and word to heart and built a strategy on it. At the time, the Internet and browsers like Netscape were catching on. Internet business was the hot thing. In an office park at the outskirts of Houston, Texas, a group of technology executives built a new strategy for their company based on exactly that assumption.

Marketing executive Steve Favoloro and technologist Sean O'Sullivan envisioned that business software and file storage would move to the Internet and they were scheming on how their company, Compaq, would benefit from this. This was the start of a multibillion-dollar business selling servers to Internet providers. It is uncertain exactly which of the two, Favoloro or O'Sullivan, came up with the term, but it is found in an internal document titled "Internet Solutions Division Strategy for Cloud Computing" from November 14, 1996. Although the term cloud was conceptualized as a marketing effort, Compaq eventually decided against using the term in part due to concerns from the PR department.

Sean O'Sullivan went on to build an online educational services company, NetCentric, in 1997. He even filed a trademark application for the term "cloud computing," which would have been convenient for him today, had it gone through. But it did not.

Compaq profited greatly from the strategy by selling hardware to support this new cloud thing and the Internet as well as "proto cloud" services like web-based email. In addition, the cloud computing juggernaut, Salesforce, was founded in 1999. Even so, the cloud didn't really catch on as a term.

It wasn't until 10 years later, in 2006, that the term reached prime time. Eric Schmidt introduced the term at the Search Engine Strategies Conference:

> *"What's interesting [now] is that there is an emergent new model, and you all are here because you are part of that new model. I don't think people have really understood how big this opportunity really is. It starts with the premise that the data services and architecture should be on servers. We call it cloud computing – they should be in a 'cloud' somewhere".*
>
> *—Conversation with Eric Schmidt hosted by Danny Sullivan*

Here for the first time the full scope of the term was introduced. Major technology companies that would later develop into key cloud vendors, such as Amazon, Microsoft, and IBM, started using the term as well. The following year an article in *The New York Times* cemented the term in the public eye, with the headline: "IBM to Push 'Cloud Computing,' Using Data from Afar" in the November 15, 2007 issue. From that time, the use of the term, as well as the industry itself, has gone just one way: up.

The Birth of the Cloud Computing Concept

The use of the term "cloud" started as a convenient way to refer to an abstracted network used by engineers. Once people realized that computing and data storage would move the individual devices onto centralized servers, the abstraction of the network access from those individual servers came to conveniently denote the whole concept of the cloud. Another important aspect is that the term was conceived from a marketing perspective and used as a general term to describe very different solutions that all had one thing in common: the use of the Internet. Today it is such a common term that we talk about cloud stocks, see it referred to casually in movies and frequently in headlines in the general stream of news media. It has become a mainstream term that most people have some vague understanding of, but in order to get a firmer grip of the concept and technologies that power this term, we need to narrow it down and consider it with some more precision.

Definitions of Cloud Computing

While definitions are not truths about the world, they are important sources for understanding how certain aspects of the world are conceptualized.

The IT consultancy Gartner was one of the first to define cloud computing in 2008. It has subsequently been slightly updated:

> *"Cloud computing is a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies."*
>
> *—Gartner Glossary*

This definition points to some of the key aspects of cloud computing: scalability, elasticity, and delivered as a service over the Internet. Scalability aims at the property of the cloud in contrast to on-premise computing that ramping up computing and storage is very easy. You don't have to order machines, unwrap them, and connect them to your data center. Anything is easily scaled up. Elasticity, on the other hand, aims at the fact that this scalability goes both ways: it will also scale down when capacity is no longer needed.

Later we go deeper into the importance of the service concept in cloud computing. This definition is quite open and draws the contours of the cloud, but it does not tell us a lot of specifics.

A more precise and comprehensive definition is the one by The National Institute of Standards in Technology (NIST). Virtually every exposition of cloud refers to this definition and it has become the de facto standard definition of what is and what is not cloud. Therefore, it may be a good idea to look into its background.

One of the key focus areas of NIST is to make standards to further innovation, especially for government agencies. Under the Obama administration there was a push for transitioning from the costly hosting and licensing models of the prevailing on-premise computing to the promise of the cheaper and more flexible cloud. However, that did not come by itself. It was difficult for agencies as well as private companies at the time to distinguish old fashioned hosting from cloud computing. At the time the definition was finished, the purpose was clearly stated by NIST computer scientist Peter Mell:

> *"When agencies or companies use this definition (..) they have a tool to determine the extent to which the information technology implementations they are considering meet the cloud characteristics and models. This is important because by adopting an authentic cloud, they are more likely to reap the promised benefits of cloud—cost savings, energy savings, rapid deployment, and customer empowerment"*
>
> *—NIST press release, October 2011*

The work had been going on for more than three years through 15 drafts. The structure of the definition is around three sections:

- *Essential characteristics*—Described five key characteristics that defined cloud computing: On-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.
- *Service models*—Concerned with the different ways cloud resources could be consumed: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).
- *Deployment models*—About how cloud infrastructure could be provisioned.

Although this formed the basic vocabulary of the cloud and is still used as the definition of cloud, there are certain aspects that bear witness to the specific time and context and may no longer be as useful. Let's look at the three sections to understand what was meant by them and how relevant they are today.

Essential Characteristics

The essential characteristics were meant as the properties a cloud solution should have. If it did not have the following five characteristics, it was not considered a cloud solution.

On-demand self-service highlights the need for the end user to be able to provision computing resources by themself. This was in contrast to the typical model at the time of hosting by one of the large hosting providers or on-premise. To provision computing infrastructure at the time, it was necessary to order servers and software from vendors or hosting companies. This was a slow process that did not align with the wish for agility and flexibility. It should not be necessary to involve anybody else as you would do when submitting a purchase order and waiting for someone to execute it. It should happen automatically. Today some of the biggest cloud vendors have services that need to be activated on request, which can take a while.

Broad network access in practice refers basically to services being available through the Internet since most cloud computing is public cloud. Basically, anyone with a laptop, table, or cell phone should be able to access the computing resources. This characteristic rules out an on-premise data center behind a company firewall. The network access needs not be Internet. There could be other types of ways to connect to the cloud, which we are beginning see in the area of IoT (Internet of Things). It has also become common for cloud providers to provide a dedicated fiber connection between the customer and provider data center. Some vendors also offer their cloud services in "boxes" that are completely cut off from any network.

Resource pooling has more to do with how you achieve the effect. The intent here is to say that in cloud computing, several users use the same pool of computing infrastructure. They don't get one computer each. Without this the goal of energy savings would not be realized. This is still important, but in practice it is possible through cloud providers to gain access to single dedicated machines, sometimes called bare metal or dedicated instance. Consequently, this is not in practice an essential characteristic.

Rapid elasticity may at first seem a bit odd, because how do elasticity and silicon computers fit together? Obviously, the point is not that the computers should bend around corners. The meaning is metaphorical. If a service like a website suddenly has a lot more users because of a flash sale, for example, it needs to be able to scale up the computing power to handle this quickly. Then, when the sale is over, it should scale back down like an elastic band that stretches and retracts when force is no longer applied. Here it is important to be aware that not all cloud services come with this automatically. Rather it often needs to be built by the customer.

Measured service basically means that the customer should only pay for what they use to avoid paying for a machine standing idle in a data center. This should be transparent to the customer. In practice the unit used to measure can differ greatly. It could be time, storage, processing capacity, number of users, number of instructions, or any other measurable property of relevance to the service. The most common are time and number of users. In practice vendors are not always making it as transparent, as NIST would have wanted them to. It can be very difficult or impossible for a customer to validate the correctness of the metering.

Although these characteristics were helpful a decade ago for distinguishing between old-fashioned hosting and this new cloud thing, the lines are more blurred today. In practice, these characteristics are good guidelines of what you can expect from most cloud solutions today, but they are not an accurate reflection of all cloud computing. It is perhaps better to think of these characteristics as common characteristics that we would expect to find most often. If, say, only two apply we would be hard pressed to call the service a cloud service, but if three or four applies it is clearer. It is important also to understand that it does not necessarily detract from the solution when it does not fulfill all criteria. For example, having a dedicated instance reserved for a year or two is much cheaper. Also, connecting through a direct connection to the cloud vendor's data center rather than through the Internet is a crucial piece of infrastructure that drives adoption of cloud computing. These solutions do not fit all the essential characteristics.

Service Models

The three service models have become standard parlance in modern cloud computing, even if the boundaries between them are breaking down. They all rely on the service concept, which we look into in more detail at a later point.

- **Software as a Service (SaaS)** is a model where everything is managed by the vendor. You cannot program anything yourself, you only configure and use it through a web browser. Common examples are Google's services like Docs, Calendar, and Sheets or Microsoft's Office 365. These provide the user with only limited options to configure the software. Many enterprise applications fall into this category, like SAP's SuccessFactors, Oracle HCM, ServiceNow, Zendesk, and more.
- **Platform as a Service (PaaS)** is a bit less straightforward and the examples are more heterogeneous. This service model refers to platforms that can be used to program applications by the consumer. It is possible to write code and configure the service, but the vendor manages the underlying infrastructure.
- **Infrastructure as a Service (IaaS)** is the most basic form where basic computing resources are provided and the consumer installs and manages the needed software. This model gives the most control

but also requires the most work.

Although these are good ideal types, there are new models that fall somewhere in between. For example, so called serverless computing or Function as a Service (FaaS), which is the capability of being able to write a piece of code that executes based on triggers, like a web service call. This is somewhere between PaaS and SaaS. Another example is containers, which are a level above the operating system and somewhere between IaaS and PaaS. Some types of software come in all variants, like databases. You can install a database on an IaaS machine, use the same database as a PaaS offering and, in some cases, even as a SaaS product. It is sometimes contested whether it is one or the other. Consequently, it is important to look at the service in question and evaluate whether it fits the needs more than whether it fits the label.

Deployment Models

There are different ways to deploy cloud infrastructures or more accurately, different ways to allow access to them. The deployment models specify different types of clouds.

- **Private cloud** is when the cloud services are offered on a private infrastructure. Although this is certainly a theoretical option and some organizations have made headway into offering a limited array of cloud services as a private cloud to internal developers, this is not a widespread model and misses most if not all of the benefits of the cloud. It is essentially just another way of running an on-premise data center.
- **Community cloud** is when a specific community of consumers band together to build a cloud that they can use similar, to a private club cloud. This is also a theoretical possibility. It was much more commonly talked about a decade ago, but today little cloud computing is deployed in this way. I have not been able to verify any existing large-scale deployments of this type. If you stretch the definition, it could be argued that the large cloud vendors' so-called GovClouds act as community clouds. Subsets of their cloud offerings are tailor made to government customers, so they might fit this description. However, they also fit the description of a specific variant of a public cloud.
- **Public cloud** is the common model that we all know. Virtually all cloud computing today is deployed according to this model. A user can access and use this through the public Internet. The difference between now and when the definition was made is perhaps increased capabilities for keeping the public cloud private with the aid of data encryption in transit and at rest.
- **Hybrid cloud** is a combination of two or more of these models. Because the first two are mostly theoretical constructs, this one is also irrelevant at least in its NIST formulation. The idea of a hybrid cloud in a different sense however is much more common, since most organizations run hybrid infrastructures with multiple cloud providers and their own (non-cloud) data centers. In a sense, the hybrid cloud is very widespread. Just not in the original definition offered by NIST.

Toward a Concept of the Cloud

As you can see from the previous sections, the definition of the cloud is not as clear-cut as that of say a carbon atom. There are a number of aspects that are commonly associated with cloud products but in specific cases some of them can be lacking and still "count" as cloud-based. We saw that the characteristics were more common than essential and that it is a fluctuating landscape. Rather than come up with a clear-cut definition that will never receive agreement, we are better served to try to understand the cloud phenomenon and the market that drives its development, which is the subject of the next chapter. In the remainder of this chapter, we focus on a few concepts that are key to understanding what is special about the cloud and how it is conceptualized. These concepts are utility, service, and layers.

The Cloud as a Utility

Along with the taming of electricity came the modern concept of a utility as a way to deliver a service to the public. It is important to notice that we are not talking about the economic concept of utility as a measure of worth. We are talking about a public utility—like gas, electricity, and water—that is offered to consumers in a standardized way. A public utility is not necessarily a governmental institution. It can be, and often is, privately owned. It maintains an infrastructure that offers a public service its consumers.

The idea that computing could be a utility arose, as so many other key ideas, in the 60s. John McCarthy, speaking at the MIT Centennial in 1961, said:

> *"(..) computing may some day 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."*
>
> *—Architects of the Information Society*

This can be seen as a rallying cry for what we today call cloud computing. In many ways this vision of computing as a utility has come true, but in other ways it is also a misleading metaphor for what cloud computing is and how it works.

There is no clear definition of what a public utility is. However, most definitions just assert that it is an organization that provides a basic service to the public and ends up listing the typical cases like water, gas, electricity, and transportation. Let's therefore try to compare a few aspects of traditional utilities and cloud computing.

Product

If a utility offers gas, that is what you get. No more, no less. Similar for water, you get just water. The product being offered is undifferentiated and the consumer has no options to choose from. It could be said that for telephony, it differs somewhat. Still, when you buy a cell phone connection, even though you can get different plans, you just get an undifferentiated connection. The difference in plans is more related to consumption patterns than the nature of the service, which remains a plain wireless connection. The method of consumption is similarly simple. For a typical utility, you just have to connect to the infrastructure and the product starts flowing. It will not stop until you discontinue the service or forget to pay the bills.

In the cloud, some individual services may offer a similar undifferentiated product such as storage, but the vast majority of services are not undifferentiated. Very little just starts flowing from the taps when you connect to the cloud. Even with a basic infrastructure service like computing, you have to decide on the type of CPU, memory, and operating system, and you often have to be ready to upgrade to secure service continuity. When it comes to Platform as a Service, there is even bigger differentiation: an MS SQL database is not the same as an Oracle database or a PostgreSQL database and these are even comparable relational databases that differ even more from other databases like graph databases.

For Software as a Service, there are very few if any similarities between the vendors. SAP HR, which is an HR system, differs significantly from Oracle HCM, which is also an HR system. Furthermore, anyone who has tried to implement an HR system can testify to the fact that the service doesn't just start flowing from the tap when you connect to it. It will often take months or even years before you can start using it, because it needs configuration and migration of data.

Path Dependency

In physics and mathematics, the concept of path dependency refers to a system whose state depends on the history of that system. If you heat a bowl of water, it will eventually boil regardless of how you heated it. This is not path-dependent. The same goes for other utilities: gas will come into your house given the right pressure regardless of where it came from or how it was treated. Utilities in general show no path dependency since the state of the system does not have to take into account any particular historical factors.

This is very rarely the case for cloud computing. It could be the case when you are starting something completely new. For example, when you build a new application. If you are transferring your HR system from one provider to another, there is path dependency in the configuration of the service. The state of the system reflects all the different HR events that have taken place in the system and is therefore path dependent. Almost all systems containing data in the cloud are consequently path dependent. Purely functional systems, like containers or some virtual machines, however, do not need to be path dependent if data is stored outside in a database for example and may be transferred easily to another provider. This is a key aspect that shows the limitations of the cloud as a utility. Data makes the cloud path dependent.

Transferability

The fact that utilities have no path dependency positively affects the transferability of a service. This allows the consumer to transfer to any provider without additional work. In the case of electricity or cell phone coverage, the consumer typically just has to sign the contract and will never know exactly when he or she transferred to the new provider.

The cloud, on the other hand, has path dependency for all data carrying applications. It is necessary to migrate the data to the new service because of this path dependency. This is actually a key parameter built into many vendors' business models that consumers are often wary of, namely vendor lock-in. The greater the vendor lockin, the less transferable the service. This is perhaps one of the biggest contrasts to the thinking of traditional utilities and their history of being regulated state monopolies. They had to be easily transferable in order to maintain competition. There is, however, a slow convergence toward higher transferability of certain types of services, especially from on-premise to the cloud but also between clouds. This only works when there is standardization and no path dependency.

Configuration

A utility like gas, water, and electricity needs no configuration. It is offered according to the specifications that are common in the service area. Water and gas will have a specific pressure and electricity a specific voltage and the consumer just needs to buy products that fit these specifications.

Again, this is very rarely the case in the cloud. Most services need to be configured in one way or another. Even with the simplest like storage, you have to decide what kind of storage you need. For higher-level SaaS products, the configuration takes on the scale of development of a new system, because there are so many parameters that need to be set before the service is operational. A customer typically has to hire a systems integrator for months to configure the service for use.

Service Continuity

For utilities, the service will never change. The electricity will remain the at same voltage. The water will remain water and will not become pink lemonade or Aperol spritz. The consumer will never have to do anything to adapt or to ensure service continuity. The utility will keep flowing without variance.

This is not the case in the cloud. Because it is software, it will in some cases need to be upgraded, which to varying degrees is up to the consumer. On this point, Software as a Service is better but new features will continually be made available, or the design will change, which will require the consumer to adapt. For other types of services, they will often become technically obsolete and be discontinued from the provider, which again requires the consumer to react in order to retain service continuity. This actually happens a lot faster than most organizations are used to running an on-premise data center. A database or machine will after all usually run until you turn it off, even if support has stopped for the product. This is not necessarily the case in the cloud.

Regulation

Utilities are often heavily regulated in terms of specifications for the service provided, such as water quality, and in terms of price or rules for transferability, in the case of cell phones. The reason for this is that often they have tended toward natural monopolies that need to be managed in order to maintain a competitive market. In the case of electricity, it is not feasible that a new market entrant can start building its own network, and the same is true for water and gas. If it belongs to a single corporation, they could take advantage of this monopoly to increase prices, which would harm the consumer.

Although the cloud is not a natural monopoly in the same way, it is close to it. It is not feasible for a new market entrant to start offering the same services as say AWS, and building the same infrastructure necessary to compete on that market. There is, however, competition from incumbents that we will look at later, but nothing restricts this handful of companies from agreeing tacitly or explicitly to a certain price because they together control a monopoly. Another complication compared to utilities and regulation is that cloud computing by its very nature is a global phenomenon and utilities have always been regulated at the national or state level. The same goes for quality of service. The consumer has very little power to complain since cloud offerings are often offered on a best effort basis. Consequently, the lack of regulation that traditional utilities have leaves the cloud consumer vulnerable.

Is the Cloud a Utility? Or More Like a Supermarket?

As can be seen in Table 1-1, many of the aspects we typically associate with a utility and perhaps what John McCarthy had in mind in the beginning of the 1960s are not comparable. The cloud may, in a select few cases, come close to properties that are typical of utilities, but in the big picture it would be wrong and misleading to think of the cloud as a utility.

Table 1-1 Comparison of Public Utilities and the Cloud

Aspect	Public Utility	Cloud
Product	Undifferentiated	Mostly differentiated but some undifferentiated products exist

Aspect	Public Utility	Cloud
Path dependency	No path dependency	All data carrying applications are path dependent, some functional ones are not
Transferability	Easily transferable	Mostly not easily transferable
Configuration	No configuration needed	Configuration needed before a service will be available
Service continuity	Consumer does not need to actively manage the service to retain service continuity	Customer needs to actively manage the service to retain service continuity
Regulation	Heavily regulated	Unregulated

That does not mean that the cloud is not valuable or cannot move more toward being a utility. It just means that if someone advertises the cloud as a utility, it makes sense to be critical and ask in what sense that is meant. Because by and large the cloud is not a utility.

A better way of thinking about the cloud is as a form of retail. In the retail industry, we have a handful of very large general-purpose retailers and many smaller specialized ones. If we think about fast-moving consumer goods, another pattern is important to be aware of. When we shop for groceries there is a range of products that are identical across the different supermarkets: milk, eggs, bacon, orange juice, beans, canned tomatoes, etc. The precise range is different from country to country. Not only are the products virtually identical with only the brand name differing, the price is too. These are the staples that draw customers into the store.

Retailers do not typically earn a lot if anything from these items, rather they earn their money on the other products that the customer grabs on the way to the register. These are more particular like vegan paleo granola or bacon ice cream. These products are difficult to compare and therefore more difficult for the consumer to evaluate in terms of price. Consequently, this is where the store makes the bulk of its profit.

Similarly, in the cloud there are consumer staples particularly in the Infrastructure as a Service world: virtual machines, blob storage, and block storage. These are offered with very little if any variation across the different cloud providers, and at a similar price. Then there are the more specialized offerings like the databases, security, and integration solutions. Because they differ more, it is difficult to compare the price and a higher mark-up can be made. Incidentally, the modern cloud market was created by a retailer, that is, Amazon with its Amazon Web Services. Subsequent entrants in this market have followed the same model.

When you think about it, this is also a better description of what cloud providers do. Retail is originally from the Old French word *tailler*, meaning “a piece cut off.” Its modern meaning is “selling in smaller quantities." This precisely describes what cloud providers do. They buy in bulk, for example racks of servers, and sell access in smaller portions as individual virtual machines as needed by the consumer. For larger systems they operate them as multi-tenancy, where one installation is managed for multiple consumers using a small portion of the total system resources. This, I believe, is a much better concept to have in mind than a utility when engaging with the cloud.

The Cloud as a Service

As we saw in our earlier definition of cloud, it is common to refer to different segments of the cloud as a service: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). New types are also gaining traction, such as Function as a Service (FaaS) and Business Process as a Service (BPaaS). This is not a trivial observation. Let's look at what we mean by "as a Service," since it seems to be an important concept for the cloud.

In economics there is a fundamental distinction between goods and services. The precise distinction between the two remains disputed. In classical economics, the focus was on the goods as physical objects that had value, and over which one could have ownership rights. A good would be in the possession of the owner and could be bought and sold on a market.

Adam Smith in *The Wealth of Nations,* published in 1776, distinguished between two types of labor: productive and unproductive. Productive labor led to the production of tangible goods that could be stored and sold. This aided in the creation of wealth. Unproductive labor, however, did not produce any tangible goods but rather produced a service that perished at the time it was produced.

Today a service is considered something intangible that does not result in ownership. It is merely consumed by the buyer. Some definitions claim that a service does not have to do with anything physical, but that does not hold in all cases, as we saw with utilities. Water is indeed physical and can also be owned as a matter of fact. Goods, on the other hand, are physical and can be owned but can also be immaterial such as intellectual property rights. There is therefore still some overlap.

Because most of the services we see in the cloud are automated and do not directly depend on labor as classical services, we don't have to concern ourselves with classical economics' discussions of whether the labor is productive or not or weather it contributes to wealth. The key point about a service in the context of the cloud is that it is something that is not owned, but rather can be used. It is also intangible because it depends on software and data being communicated through a network.

The different kinds of services are essentially different packages of functionality offered to the consumer. Infrastructure as a Service (IaaS), for example, offers the functionality of infrastructure to the consumer. The provider handles everything to support the service. This is similar to a restaurant, which handles everything needed to provide the meal for consumption, like shopping, preparation, the room, and the furniture. The consumer of IaaS similarly doesn't own the infrastructure consumed (although data and code on that infrastructure can be owned, but that is different). The consumer still needs to do a lot of things with IaaS because it is just like renting a blank computer. The consumer has to install all the applications and configure it to perform the needed functions.

At the other end of the scale, we have Software as a Service (SaaS), where a neatly packaged product is offered to the consumer, who does not have to do anything in terms of managing, installing, or developing software. This is also something intangible, a service that is not owned by the consumer. It is a package of functionality but a qualitatively different one that implies a lot less responsibility on the part of the consumer in order to make it functional. It also implies a lot less flexibility in terms of functionality, since the service cannot be customized to the same degree. No special functions that are unique to the consumer can be supplied. The more you move from IaaS over PaaS to SaaS, the less responsibility you have as a customer, but you also have less flexibility. Consequently, you have to find the right level of service.

Think of it like going on vacation. Some people want maximum flexibility and just buy the tickets to the destination and then go out to explore. Maybe they already know the territory or have friends or they brought a tent or they reserved a hotel somewhere else. This is similar to IaaS. It definitely allows for flexibility but also requires responsibility and action.

Other people maybe buy a complete package with plane and hotel and transport from the airport to the hotel. They still have the flexibility to go and explore restaurants at the destination or rent a car and go for a trip somewhere. The travel agency handles only the travel and hotel the consumer the rest. This is similar to PaaS.

That last kind of traveler buys the all-inclusive package, where all the trips have been planned in advance and all meals are served at the destination hotel. The consumer is not responsible for anything and the traveler only has to animate the vacation with their good spirits and visit the pool or the beach or stay up all night singing Karaoke. It will not be possible to suddenly go bungee jumping or walk the Inca trail with this model. This is similar to SaaS.

The Cloud as Layers

In technology in general, the concept of layers is important. It is an abstraction that does not have any real physical basis but guides and encapsulates functionality in manageable bits. An example is the Open Systems Interconnection (OSI) model for communication between computer systems. It's used as a reference for communication over computer networks and therefore also the cloud. It consists of seven layers:

- Layer 1: Physical Layer—Transmission of raw streams of bits in a physical medium
- Layer 2: Data Link Layer—The connection between the two nodes
- Layer 3: Network Layer—Provides the functionality to transfer data sequences
- Layer 4: Transport Layer—Provides the functionality of controlling and maintaining the quality of a connection
- Layer 5: Session Layer—Controls the exchange of data in a session between two computers
- Layer 6: Presentation Layer—Provides a mapping between the application layer and the underlying layers
- Layer 7: Application Layer—The layer closest to the end user

In order for something to be transmitted, it has to go through the physical layer. There is no way around it. But it is inconvenient for the end user to be coding the message into strings of bits. Consequently, the different layers wrap a particular functional area of concern that can interface with the layers above and below it. There is no natural or physical reason it has to be like this, but in order to work productively this has proven helpful.

Layering allows for division of labor too. A web developer can focus only on the functions and commands offered at Layer 7 and has to know nothing about any of the other layers. The electrical engineer who is designing parts for a cellphone needs to know about Layers 1 and 2, while the engineer who is designing routing software needs to know about Layers 3 and 4.

In the cloud, which builds on these communication layers, the situation is similar, although there is no commonly shared standard similar to the OSI model that defines responsibilities and protocols for the different layers. A common way to look at it is the following. IT is often used in the context of explaining the cloud:

1. Network—Connection of the physical property to other networks and the internal network of the data center
2. Storage—The functionality necessary for storing data
3. Servers—Machines with a CPU that can to process data
4. Virtualization—A virtualization of the machine resources
5. Operating system—The operating system offered to higher level functionality
6. Middleware—Software that provides functionality to applications beyond those of the operating system
7. Runtime—The functionality to run a program
8. Data—Representation of binary information in a format readable by programs
9. Applications—The programs that define functionality

As you can see, this model is far from as neat and sequential as the OSI model. For example, why is Layer 8 (data) not on top of Layer 2 (storage)? And why is storage lower than the server? One would think that storage and data might be parallel. However, the purpose of the layering model is much the same as the OSI model: to delineate areas of responsibility and division of labor.

Again, an application programmer should only be concerned with the application layer. The operations professional would be concerned with Layers 6 and 7, and the infrastructure professionals with Layers 2 to 5. Network specialists would focus on Layer 1. Each of these groups is able to focus only on their area and disregard what goes on in the other layers. Operations specialists do not need to know anything about the application layer in order to do their job.

Such is the power of layering in cloud computing and it has allowed increased division of labor and specialization. The earliest programmers had to be masters of all layers in order to get any result. Now, thanks to the layering and partial standardization of the layers, it is possible to work independently of other surrounding layers. This becomes particularly important for the cloud since cloud providers assume responsibility and offer a layer as a service. This allows the consumers to choose at which level they want to take advantage of the cloud.

Because each layer is offered as a service, we can take advantage of the insight described here. Some are adventurous backpackers who want great flexibility and have the know how to manage lower levels. Others just want some sun and curacao drinks at the bar and opt for the higher levels when it comes to consumption. The right level depends on the context, which we return to later in the book. For now, it is important to understand the nature of the choice.

Summary

In this chapter we saw how an abstraction used for sketching technical solutions resulted in the cloud becoming the symbol for cloud computing. This symbol was taken up more widely and used for building new business models focused on the Internet. It was a conceived of as a marketing term that took a decade to catch on in the wider population. But once it did catch on, it quickly became a dominating concept.

The definitions of the cloud arose in a context where potential customers needed support to navigate the market to figure out what true cloud computing was. The NIST definition has come to delineate much of the vocabulary we use today around cloud computing, but certain aspects no longer fit perfectly with the cloud market.

We saw that a number of concepts were important in how we conceptualize the cloud. An old and persistent version had the cloud pitched as a utility. Closer inspection, however, revealed that this is a bit of a stretch. A better way to look at the cloud is as a form of retail.

The concept of the cloud as a service that stems from the earliest definitions continues to be important. Although it differs somewhat from a classical economical concept of a service, it does point to important aspects of the cloud—that it is based on consumption and does not entail ownership of the computing resources.

Finally, we saw how the concept of layering has aided the cloud in developing areas of specialization to support division of labor between different groups of specialists. By supplying layers as services, this has fueled development in the cloud where consumers can choose the level of flexibility/responsibility they want.