1

Edge Architectures

1.1. The three levels of Edge Networking

The four-tier distributed multi-cloud architecture is becoming the standard. The levels correspond to four types of data centers that form the basis for the digitalization of various companies and organizations. Large data centers symbolize the Cloud with almost infinite computing and storage power. However, these data centers are far from the users with high latency to reach them. Many applications cannot run with these latencies. In addition, the energy consumption per client is very high since the information flows have to pass through many intermediate machines. However, these data centers are indispensable for applications that require power and for particularly consistent storage. The most important data centers belong to the GAFAM companies (Google, Apple, Facebook, Amazon and Microsoft), but also the BATX (Baidu, Alibaba, Tencent and Xiaomi).

For several years, data centers at the edge of networks have become increasingly important for many reasons. Latency allows the introduction of new services that are becoming preponderant for the digitalization, intelligence and security of companies, administrations and public services. For example, Industry 4.0, with IIoT (Industrial IoT), smart grid control, short-latency applications, etc. are examples of these services. These applications have been highlighted by a number of companies. These applications have been highlighted by 5G, which must have an important place in the digital infrastructure of the Edge. Next-generation Wi-Fi should not be forgotten as convergence with 5G has now clearly started. The diagram in Figure 1.1 shows the Edge environment of a network.

Edge Networking,
by Khaldoun AL AGHA, Pauline LOYGUE and Guy PUJOLLE. © ISTE Ltd 2022.

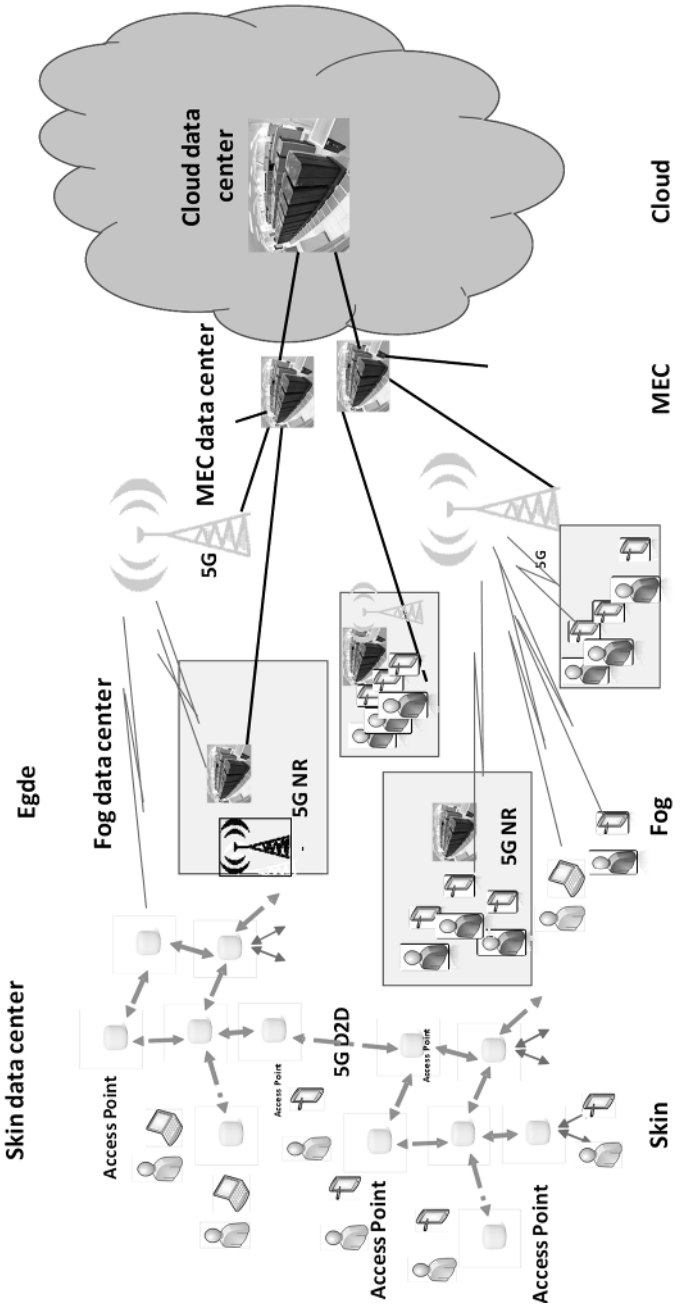

Figure 1.1. *The distributed cloud environment. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Three levels of Edge Computing are being implemented to define a distributed multi-Cloud. The most advanced in its definition and deployment concerns the MEC (Mobile/Multi-access Edge Computing) data centers of 5G operators, which enable the virtualization of numerous telecom devices and the introduction of new services that are well described in the 3GPP specifications and which concern the Internet of Things, mission-critical services with high reliability and very high speeds in mobility.

The second level, defined a long time ago, concerns Fog Networking. The word Fog was introduced by Cisco in 2013. Initially, Fog Computing was used to manage streams from objects to pre-process them before sending them to a central Cloud. Today, Fog Networking is linked to enterprise data centers that virtualize local network equipment such as routers, switches, firewalls, authentication servers and many middleboxes with control and management functions. The vCPE (virtual customer premises equipment) uses Fog data centers.

The third level is the one that is the focus of this book. It has various names that have not yet stabilized, such as Very Edge Networking, the home data center, Mist-Computing or Skin Computing, which we will use in the following. These structures are generally related to the participatory Internet, although we can see star systems around a data center. For a participatory network, the tiny data centers are located very close to the user or even on the user. These are small but powerful servers that support virtualization. These Skin data centers must be light, autonomous and consume very little energy since they can be mobile and follow the users like firemen during a fire. The advantages come from the extremely short latency time, their low energy consumption and their mobility, not to mention their security by disconnecting them from the Internet thanks to their autonomy as soon as necessary.

The four levels we have described overlap, and no one level is expected to be dominant in the future. The distributed Cloud was born from this observation. To optimize performance, services must be positioned as virtual machines (VM) or containers that users need at the right level of the architecture. This level is highly dependent on the application and its performance, control, management and security characteristics.

These digital infrastructure services, whether infrastructure or application services, are divided into microservices encapsulated in virtual machines or containers. We will see in the next chapter the platform architectures and this decomposition into microservices. An even finer decomposition is emerging in the early 2020s with the decomposition into functions. A microservice is, therefore, a set of functions.

1.2. Edge Computing architectures

The three main architectures associated with the services, carried by the data centers of the different levels of the Edge, come from the OIF (Open Infrastructure Forum), the Cloud Native of the CNCF (Cloud Native Computing Foundation) and the EECC (European Edge Computing Consortium). In Figures 1.2A and B, we have depicted the main elements of these three environments.

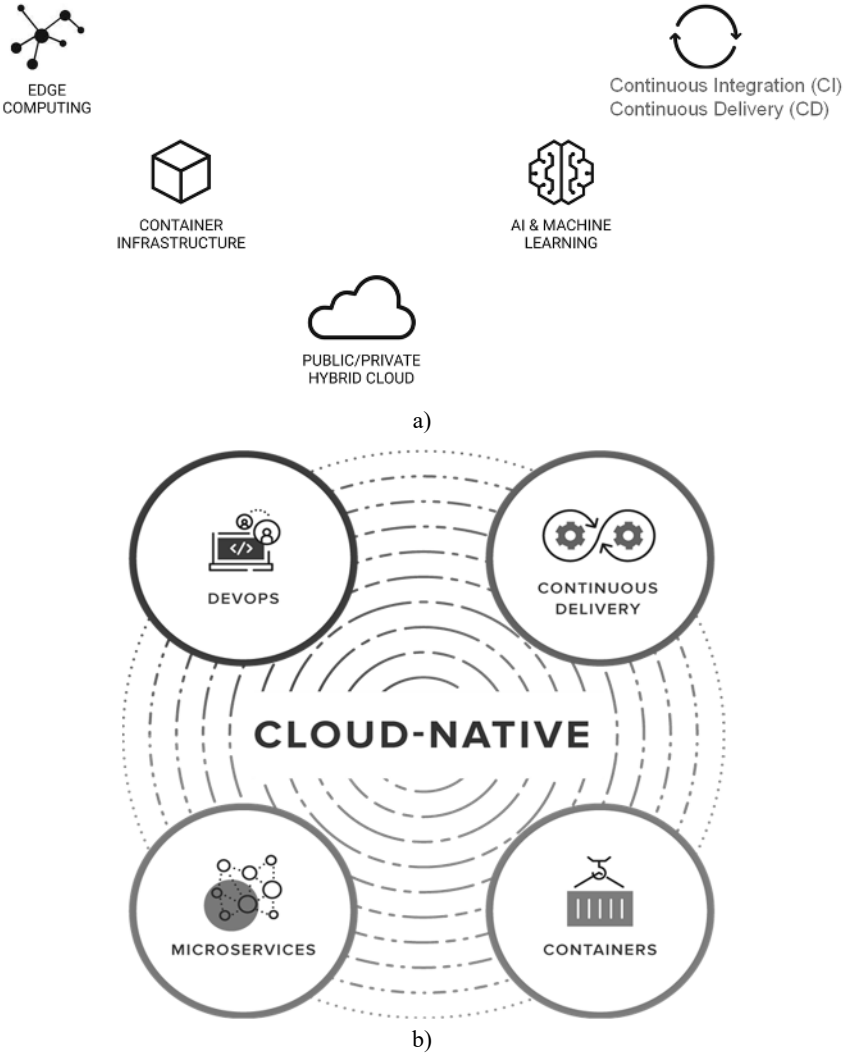

Figure 1.2A. *The basic elements of OIF (a) and CNCF (b)*

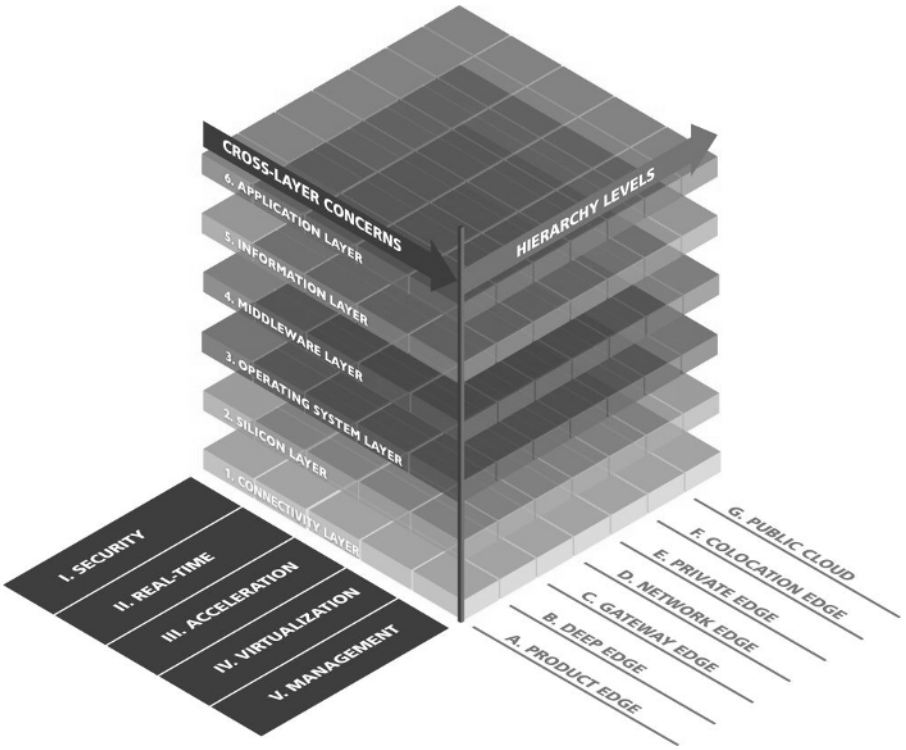

Figure 1.2B. *The basic elements of the EECC (continued). For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Notably, the Telco Edge Cloud (TEC) is an initiative that started in 2020 and aims to define an Edge architecture allowing interoperability between operators.

An intelligent digital infrastructure on the Edge requires the deployment of a distributed multi-Cloud distributed framework using 5G and Wi-Fi integration. This generation also relies on infrastructure and application services that must make businesses and organizations more agile, flexible and independent of the web giants. Figure 1.3 shows the evolution of these architectures up to the 2020s that we are interested in here.

In Figure 1.3, looking at the architecture of the 2020s, we see many changes. The physical infrastructure, or technical infrastructure, now contains only three elements: the antennas, optical fibers and data centers. Everything else is virtualized in the data centers. All the boxes have disappeared from this architecture. Indeed, the functions associated with these boxes are virtualized in one of the data centers with urbanization

to determine its level Edge or Cloud. This vision of the physical infrastructure is not relevant for participatory networks at the endpoints, which can be mobile and which we will describe a little further on.

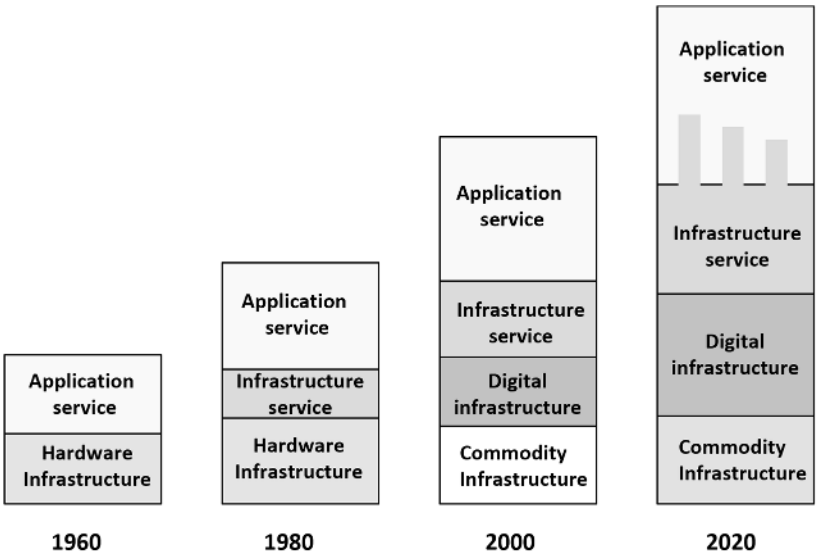

Figure 1.3. *The evolution of architectures. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Above the physical infrastructure is the digital infrastructure, which contains all the functions related to the physical infrastructure to make it work. These are the functions found in boxes such as Wi-Fi boxes, routers, switches, firewalls, etc. Again, these functions are positioned at the different levels associated with Skin, Fog, MEC or Cloud data centers.

Infrastructure services correspond to all the basic functions that can be used by the application services. Infrastructure services are the services provided to meet general requirements, including technical and software solutions, such as interoperability, security, middleware and network or distributed systems services. Infrastructure services also include the management and control of application services. These services are becoming part of the application services as they integrate more and more functions that used to be put directly into the application service but are now available directly to all applications that want to use them rather than being found in parallel at the level of each application.

We are now going to take a closer look at the two most widely used digital infrastructures, which initially came from the major Web industries but now have the support of almost all industries: the OIF and the CNCF.

OIF took over the OpenStack Alliance in 2020. Open Stack is a Cloud management system that is widely used by Cloud providers and operators. However, the Cloud is not enough to define an infrastructure. This is the reason for the birth of OIF: to add to the Cloud and to virtualization the ingredients to build a complete infrastructure.

OpenStack was born in 2010 with the merger of two projects, Rackspace (Storage) and NASA (Compute). OpenStack is developed in Python and distributed under the Apache 2.0 license. The versions started with A, then B, etc. The latest are:

– Queens (2018, 02);

– Rocky (2018, 08);

– Stein (2019, 04);

– Train (2019, 10);

– Ussuri (2020, 05);

– Victoria (2020, 10);

– Wallaby (2021, 04).

OpenStack contains a large number of modules that allow you to build a complete Cloud. Some of these modules are described in Figure 1.4. They include:

– heat: orchestration component;

– neutron: network and addressing management;

– nova: management of the computing resources;

– cinder: block storage;

– swift: object storage;

– glance: disk image service;

– ceilometer: metric-based telemetry;

– keystone identity service.

OpenStack is a complete environment that is increasingly being introduced in commercial software. However, while it supports virtualization techniques well, it needs

to be complemented by other open-source Cloud infrastructure software. For example, IOF integrates Magma for Edge infrastructure, offering both cellular (5G) and Wi-Fi services and the OpenInfra Labs. It includes, for example, the Wenjua project that provides services to shorten the introduction time needed to integrate artificial intelligence. The OIF architecture includes, of course, Kubernetes to orchestrate the containers.

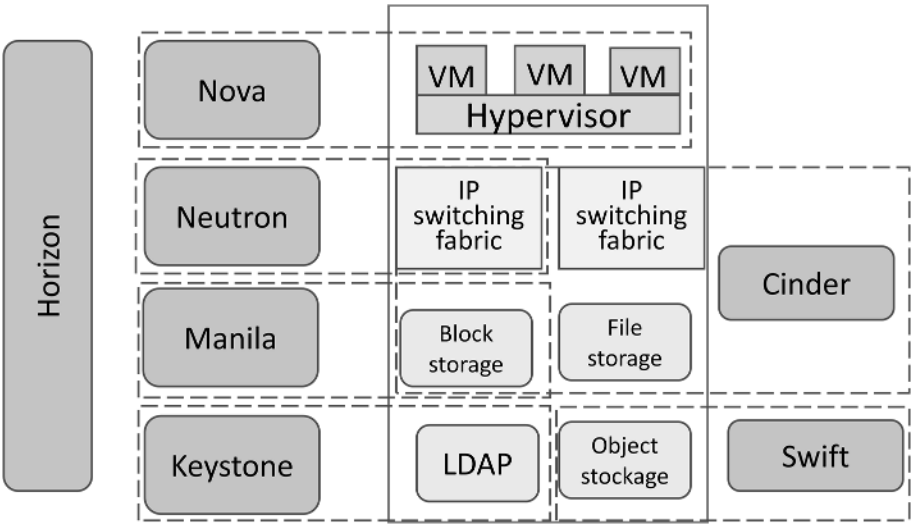

Figure 1.4. *Some modules of the OpenStack open-source software. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

The second major architecture is the Cloud Native Computing Foundation (CNCF). This is a project of the Linux Foundation that was founded in 2015 to help advance container technology and align the technology industry around its evolution. Founding members include Google, CoreOS, Mesosphere, Red Hat, Twitter, Huawei, Intel, Cisco, IBM, Docker, Univa and VMware. Today, the CNCF is supported by nearly 500 members.

CNCF technology projects are cataloged with a Sandbox, Incubated and Graduated maturity level in ascending order. The CNCF process incorporates projects as Incubated projects and then aims to progress them to Graduated, which involves a level of process and technology maturity.

The main properties of the CNCF architecture are shown in Figure 1.5.

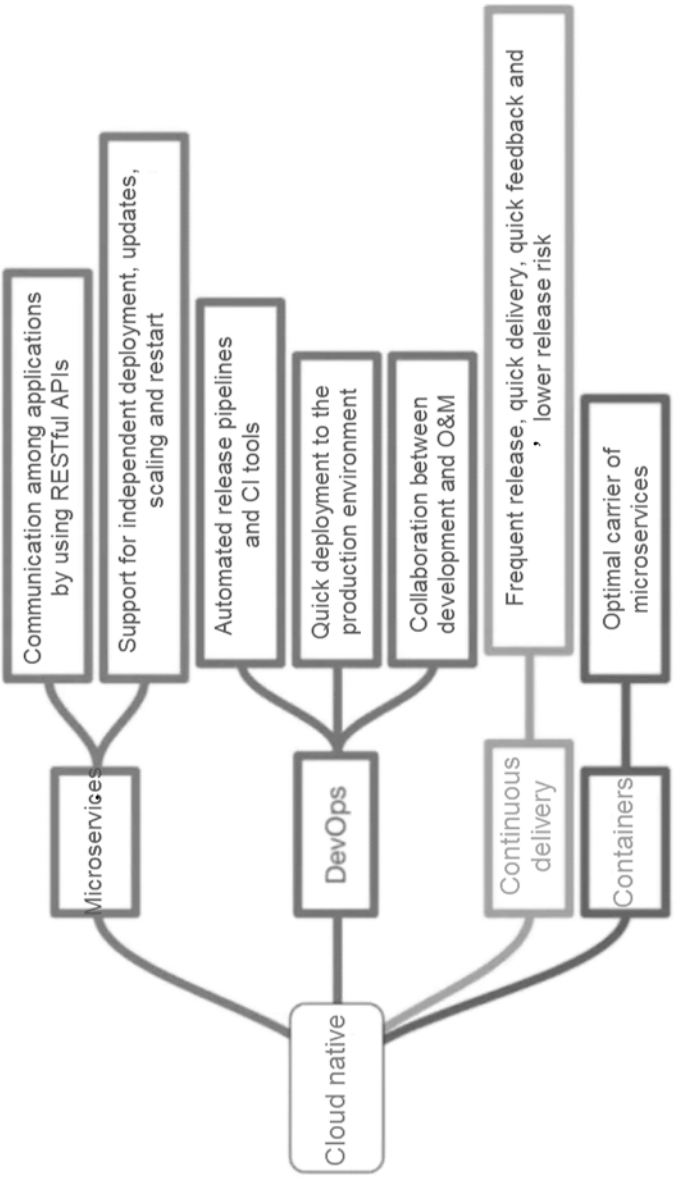

Figure 1.5. *The main properties of the CNCF architecture. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Cloud-native architecture brings the fluidity, resiliency and scalability of Cloud architectures, which contributes to their appeal to organizations that favor a DevOps philosophy. A Cloud-native approach offers the benefits we describe below.

The use of loosely coupled services instead of a monolithic stack. The use of loosely coupled services, instead of a monolithic stack, gives development teams the ability to choose the framework, language or system that best meets the specific goals of an organization or project. The portability of microservices in containers ensures that an organization is not overly dependent on one Cloud provider. The complexity of troubleshooting is reduced since an open-source container orchestration platform like Kubernetes makes it easy to identify a problematic container without shutting down the entire application.

Because microservices operate independently of each other, developers can optimize them one by one for core functionality and ultimately enrich the end-user experience. Using microservices in software development facilitates continuous integration and continuous delivery efforts, reducing the development lifecycle and the risk of human error with automated processes. In addition, a container orchestrator can automatically schedule and allocate resources based on demand to increase efficiency.

The use of microservices for application architecture allows developers to make changes to a microservice or offer new features without affecting the overall application and its availability. The overall CNCF architecture is described in Figure 1.6.

Of the several tens of billions of containerized services in operation by the end of 2021, two-thirds migrate daily. Most of these migrations (70%), take place within the same data center, essentially to cluster and unbundle them with the objective of minimizing energy consumption. Between different data centers, the figure is down to 10%, but the arrival of the distributed Cloud will certainly change the situation.

The process of migrating services and microservices must be optimized and secured in the Edge and Cloud Computing environment. Migrations allow services and microservices to be moved from one level to another, taking into account an urbanization algorithm. These architectures are moving toward a container environment orchestrated by Kubernetes and using CaaS (container as a service) or FaaS (function as a service) techniques in a serverless environment. This environment is deployed on the different levels of the architecture.

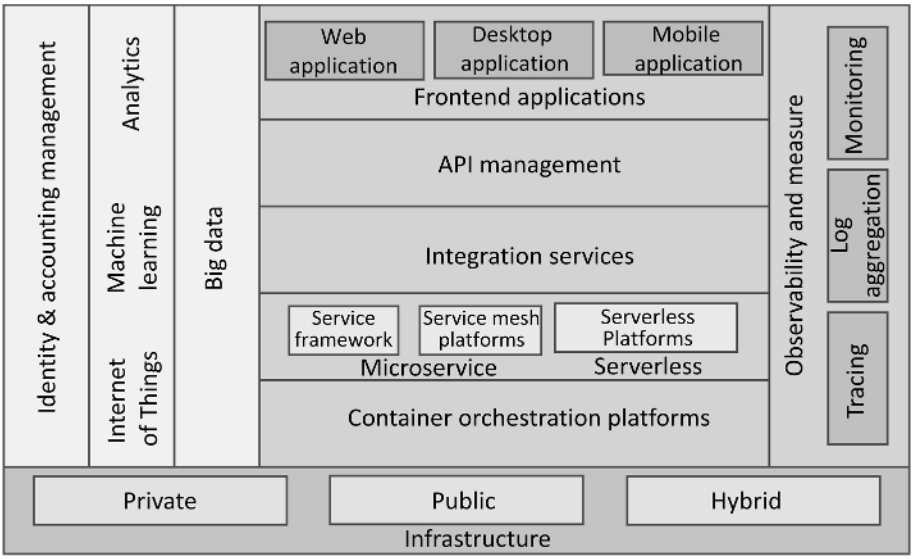

Figure 1.6. *The global architecture of the CNCF. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

All modern architectures are moving toward containerization, that is, the use of containers that form the equivalent of virtual machines running on the same operating system, usually Linux. To dynamically allocate resources to the microservices that are encapsulated in the containers, you need an orchestration system that automatically allocates resources. Kubernetes is open-source software that has become essential. Figure 1.7 describes the Kubernetes container orchestrator.

Kubernetes was originally designed by Google and is now maintained by the CNCF. It aims to provide a platform for automating the deployment, scaling and operations of application containers. Kubernetes works with a range of tools to set up containers in a cluster, often with images created using Docker.

Kubernetes defines a set of primitives that collectively provide mechanisms that deploy, maintain and scale applications based on CPU, memory or custom metrics. Kubernetes is loosely coupled and extensible to meet different workloads. This scalability is provided in large part by the Kubernetes API, which is used by internal components as well as by the extensions and containers that run on top of Kubernetes. The platform exercises control over computational and storage resources by defining

resources as objects that can then be managed as such. Kubernetes components can be divided into those that manage an individual node and those parts of the control plane.

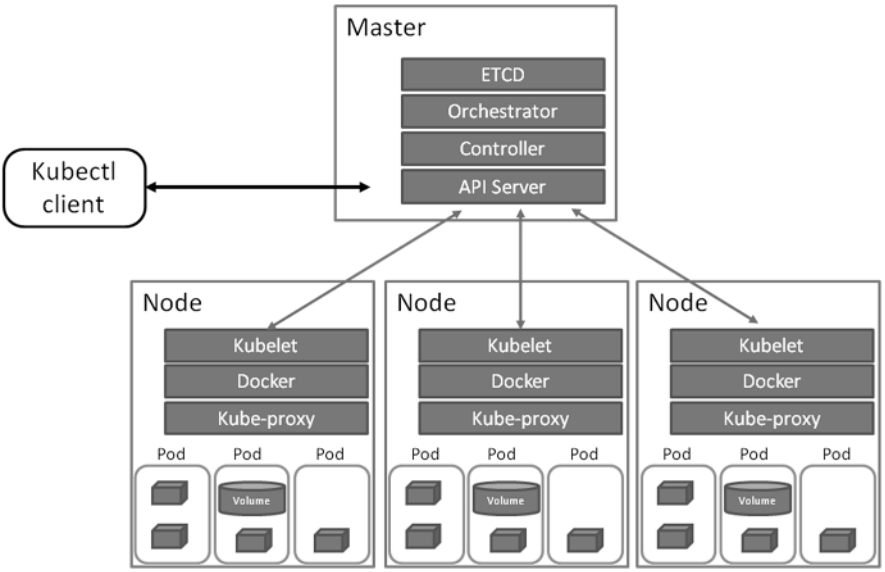

Figure 1.7. *The Kubernetes orchestrator. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Many Cloud services offer a Kubernetes-based platform as a service (PaaS) or infrastructure as a service (IaaS) on which Kubernetes can be deployed as a platform delivery service. Many providers also offer their own Kubernetes distributions.

The basic scheduling unit in Kubernetes is a pod, as shown in Figure 1.7. A pod is a grouping of containerized components. A pod consists of one or more containers that are guaranteed to be co-located on the same node. Each pod in Kubernetes is assigned a unique IP address within the cluster, allowing applications to use ports without the risk of conflict. All containers can reference each other within the pod, but a container in one pod has no way to directly address another container in another pod. If it wishes to perform such an action, it must use the IP address of the pod.

A pod can define a volume, such as a local disk directory or a network disk, and expose it to the pod containers. Pods can be managed manually via the Kubernetes API, or their management can be delegated to a controller that is located in the master node.

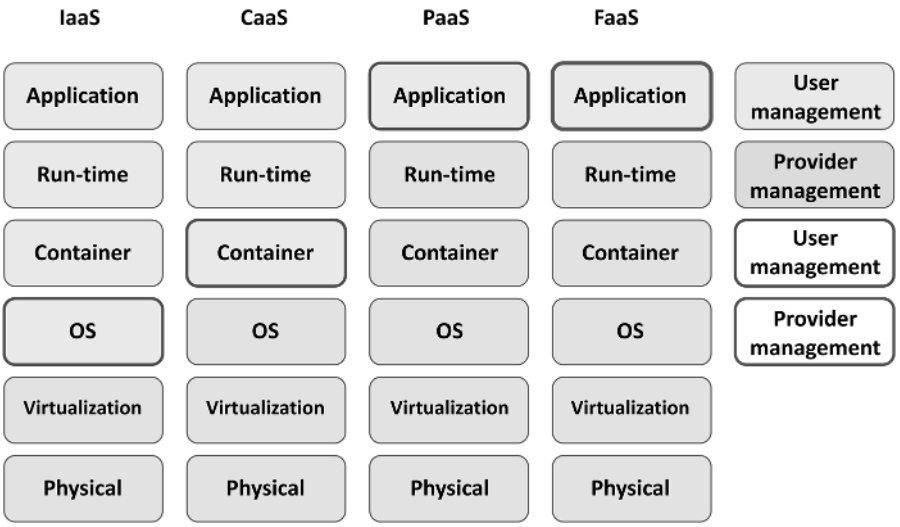

Figure 1.8. *CaaS and FaaS architectures. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

Two solutions have been developed strongly since 2020 to complete the environment on top of Kubernetes: CaaS and FaaS. These two solutions are described in Figure 1.8.

The CaaS solution is the one most often seen today. The FaaS solution is expected to become the standard in the coming years. The CaaS solution, as shown in Figure 1.8, requires the digital infrastructure provider to support the hardware, virtualization and operating system, while the customer developing on this environment must support the containers, the runtime environment and, of course, the application. FaaS technology completely hides the containers, so-called serverless techniques, by allowing developers to work with simple functions that form a subset of microservices.

We describe in Figure 1.9 the three solutions for obtaining a development platform. The first one, the most classical so far, concerns monolithic developments on a PaaS platform. The most popular solution today is CaaS, given the breakthrough of containers in Cloud computing architecture. The solution that is starting to gain ground is FaaS which should greatly simplify development on Edge and Cloud platforms.

All architectures today integrate containers and the Kubernetes orchestrator. It is becoming more and more standard to use serverless technology on top of Kubernetes.

On top of this, several open-source solutions are available such as Knative (Cloud-Native serverless). Knative allows serverless deployment of applications and their execution on all Kubernetes platforms. Istio completes the previous set. It is an open-source platform that allows controlling the way data is shared between microservices. These different solutions are described in Figure 1.9.

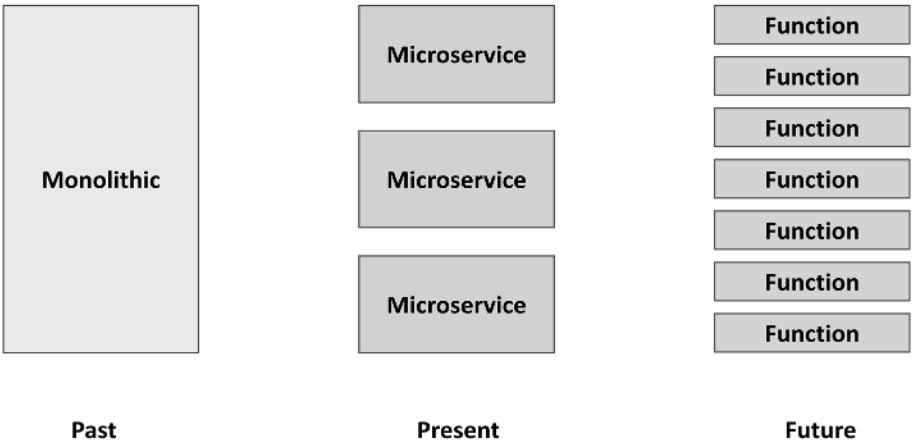

Figure 1.9. *Service architectures. For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

1.3. Security and domain name system on Edge

Security on Edge will be ensured by solutions provided by 5G and will be based mainly on embedded SIM (eSIM) and the new generation of SIMs: integrated SIM (iSIM). The eSIM is a SIM card integrated into a device (mobile, tablet or connected objects). It is no longer necessary to insert a physical SIM card to allow a device to transmit information using the connections of a mobile network. An eSIM environment is described in Figure 1.10.

The eSIM contains the smartphone's internal component (embedded universal integrated circuit card, eUICC), which allows managing profiles corresponding to specific SIM cards which are managed remotely in the subscription manager–data preparation (SM-DP+) server or even in remote servers through a discovery service offered by a subscription manager-discovery server (SM-DS). This is sometimes referred to as a virtual SIM since the profiles are located remotely and no longer directly in the smartphone's SIM card.

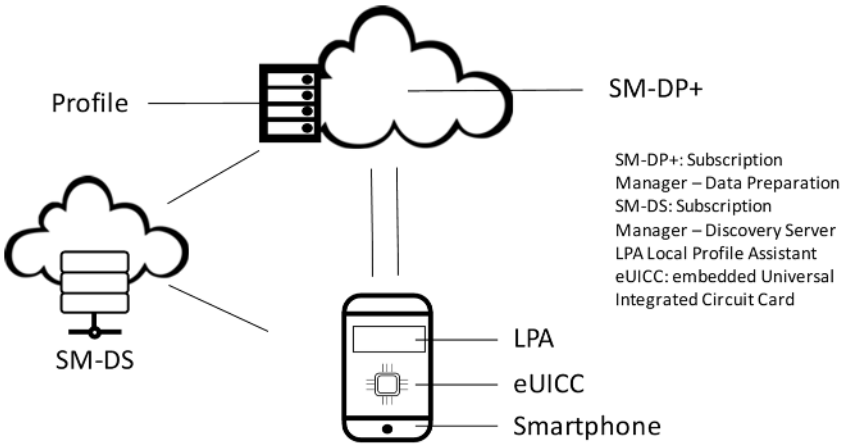

Figure 1.10. *An eSIM environment*

An iSIM card is a miniaturized SIM card with an equally small modem. This assembly is grafted onto the platform that hosts the processor and memory, commonly called the System on a Chip (SoC). The advantage of this solution comes from the integration of a SIM card in each hardware, whether it is tiny or large. In particular, each communicating object will be able to have its own SIM card, which will make it possible to easily introduce a high level of security compared to today, when many objects are very easily attacked because they are protected by software passwords.

The domain name system (DNS) or domain name resolution system is one of the essential building blocks of the Internet today and one of the vectors for spying on and attacking the various Web services. DNS resolution consists of associating a Web address with an IP address in order to route traffic correctly. It is a distributed database system built in a hierarchical manner. DNS servers allow the publication of the correspondence of a DNS address with an IP address through the publication of a DNS zone. But, in order to be able to identify the DNS servers allowing the resolution of a domain, it is essential to have DNS resolvers.

The resolver is the building block for identifying the DNS servers that contain the zone information for a given domain through a recursive search of the DNS servers. Resolvers can be accessible only in a given network, in which case they are identified as private resolvers, or they can be private resolvers, or they can be publicly accessible, in which case they are called public resolvers.

Until now, DNS resolution has been done via the UDP protocol, but several new DNS resolution methods have appeared to complement the initial protocol in order to better secure communications between the resolver and the client, guarantee the reliability of the resolution and allow private resolution. Among the extensions to the DNS protocol that make it more secure is DNS over HTTPS (DoH), which allows DNS requests to be transported via the HTTPS protocol. The DoH protocol is relatively recent and will be integrated via browsers such as Chrome and Firefox.

1.4. The digital infrastructure of the participatory Internet

The participatory Internet is based on a particular physical and digital infrastructure since it can be mobile. The participatory Internet combines Edge Computing with its Skin data centers located close to the users and interconnection of these data centers by a mobile, flexible and autonomous network, the whole forming a distributed Edge Cloud close to the users. The users participate in this flexibility, and their resources are added to the digital infrastructure of the participatory Internet. Participatory Internet infrastructure is described in Figure 1.11.

In the diagram in Figure 1.11, five smart boxes, or Skin data centers, are interconnected by wireless technology such as Wi-Fi, but any technology is possible as long as it allows communication between boxes within an acceptable distance. The boxes contain the digital infrastructure, the infrastructure services and the application services in such a way that even without connection to the Internet, the system continues to function. The interest in this infrastructure also comes from the mobility of the boxes, their possible disappearance without impact on the network and the connection of new boxes, which must be done automatically.

Users can be participants in a tactical bubble, in a smart environment like a smart city, or employees of a company to build the company's digital infrastructure or individuals to build a cluster where resources are shared.

The word participatory comes from the boxes that can be owned by the users, and that participate in the network infrastructure. These boxes are the nodes of the network. They are small data centers capable of managing virtualization to allow the transfer of a service or microservice from one box to another. However, the participatory Internet owes its own infrastructure to be able to detach itself from the central Internet. Since the network is mobile and nodes can appear and disappear, the services included in the infrastructure must be distributed in order to continue to function, regardless of the number of active and interconnected devices. The infrastructure adopts a mesh technology for interconnecting nodes with the possibility of mobility or not. The

most interesting case is obviously that of mobility, where all the nodes can move and follow the users as they move.

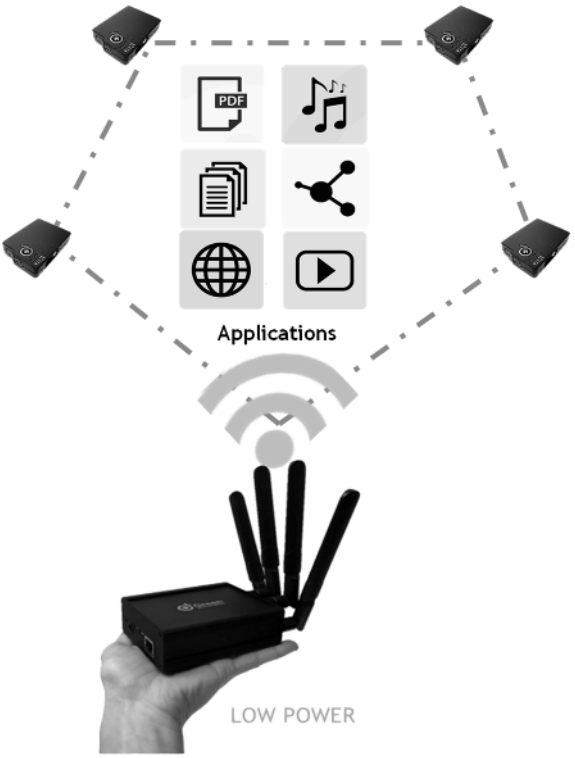

Figure 1.11. *Participatory Internet infrastructure (©Green Communications). For a color version of this figure, see www.iste.co.uk/alagha/networking.zip*

The most classical infrastructures of the participatory Internet can be symbolized by the vehicular networks that we will study in Chapter 7. Chapter 4 will also be devoted to the digital infrastructure of the participatory Internet in order to describe it in detail.

1.5. Conclusion

Infrastructure is going digital. While it has taken its place first by the large Cloud data centers, it is now being defined on the Edge and concerns consumer and enterprise users. This new digital infrastructure is positioned on the edges of the network. It must enable users to connect their terminals and provide them directly,

quickly and securely with all the functions they need. To do this, these functions are virtualized in data centers to introduce great flexibility, agility and the ability to deploy any service very easily and almost instantly.

1.6. References

Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T. (2018). Mobile edge computing: A survey. *IEEE Internet Things*, 5(1), 450–465.

Ahmed, E. and Rehmani, M.H. (2017). Mobile edge computing: Opportunities, solutions, and challenges. *Future Generation Computer Systems*, 70, 59–63.

Ahmed, E., Gani, A., Sookhak, M., Ab Hamid, S.H., Xia, F. (2015a). Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges. *Journal of Network and Computer Applications*, 52, 52–68.

Ahmed, E., Akhunzada, A., Whaiduzzaman, M., Gani, A., Ab Hamid, S.H., Buyya, R. (2015b). Network-centric performance analysis of runtime application migration in mobile cloud computing. *Simulation Modelling Practice and Theory*, 50, 42–56.

Ahmed, E., Naveed, A., Gani, A., Ab Hamid, S.H., Imran, M., Guizani, M. (2017a). Process state synchronization for mobility support in mobile cloud computing. *Proceedings IEEE Xplore, International Conference on Communications (ICC)*, IEEE, New York.

Ahmed, E., Ahmed, A., Yaqoob, I., Shuja, J., Gani, A., Imran, M., Shoaib, M. (2017b). Bringing computation closer toward the user network: Is edge computing the solution? *IEEE Communications Magazine*, 55(11), 138–144.

Ahmed, A., Naveed, A., Gani, A., Ab Hamid, S.H., Imran, M., Guizani, M. (2019). Process state synchronization-based application execution management for mobile edge/cloud computing. *Future Generation Computer Systems*, 91, 579–589.

Kraemer, F.A., Braten, A.E., Tamkittikhun, N., Palma, N. (2017). Fog computing in healthcare – A review and discussion. *IEEE Access*, 5, 9206–9222.

Moura, J. and Hutchison, D. (2019). Game theory for multi-access edge computing: Survey, use cases, and future trends. *IEEE Commun. Surveys Tuts.*, 21(1), 260–288.

Open Edge Computing (2022). Open Edge Computing [Online]. Available at: https://www.openedgecomputing.org/ [Accessed 4 October 2019].

Pan, J. and McElhannon, J. (2018). Future edge cloud and edge computing for Internet of Things applications. *IEEE Internet of Things Journal*, 5(1), 439–449.

Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., Taleb, T. (2018). Survey on multi-access Edge computing for Internet of Things realization. *IEEE Commun. Surveys Tuts.*, 20(4), 2961–2991.

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39.

Singh, H. (2017). *Implementing Cisco Networking Solutions*. Packt Publishing, Birmingham.

Sun, X. and Ansari, N. (2016). EdgeIoT: Mobile edge computing for the Internet of Things. *IEEE Communications Magazine*, 54(12), 22–29.

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Commun. Surveys Tuts*., 19(3), 1657–1681.