Available on-line at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Edge Computing: An Overview of Framework and Applications

Ezhilmathi Krishnasamy[a,*], Sebastien Varrette[a], Michael Mucciardi[b]

[a]University Du Luxembourg, SnT, UL HPC, Luxembourg.
[b]Barcelona Supercomputing Center, Spain.

Abstract

This report gives an overview of the Edge Computing paradigm and its applications. Indeed, with the advent of the Internet of Things (IoT) era, many electronic devices and sensors produce a vast volume of data which should be processed in a timely manner and this novel computing model is nowadays seen as a pertinent answer to this open challenge. This report thus explains why Edge Computing is needed and how the edge architecture is typically structured. It further presents the technologies that help this cutting-edge model to function properly. Since Edge Computing involves a heterogeneous architecture, it requires to adapt to a few technological recommendations for optimal performance. In this context, this report reviews the latest hardware technology trends tied to Edge Computing developments, and points out technical challenges implementing this innovative computing model. In particular, we analyse how High Performance Computing and Cloud Computing infrastructures can be efficiently organised to design an Edge Computing-based framework able to tackle cutting-edge issues solved by Artificial Intelligence techniques. Finally, this report presents selected real-world applications of the Edge Computing paradigm across multiple domains affecting our daily life, *i.e.,* healthcare, smart city and grids, industry 4.0 and public safety.

*Corresponding author.
tel. +352 46 66 44 6265 fax. +352 46 66 44 36265 e-mail. ezhilmathi.krishnasamy@uni.lu

Contents

1. Introduction

This technical report is part of a series of reports published in the Work Package "HPC Planning and Commissioning" (WP5) of the PRACE-6IP project. The series aims to describe the state-of-the-art and mid-term trends of the technology and market landscape in the context of High Performance Computing (HPC) and Artificial Intelligence (AI), Edge-, Cloud- and Interactive Computing, Big Data and other related technologies. It provides information and guidance useful for decision makers at different levels: PRACE aisbl, PRACE members, EuroHPC sites and the EuroHPC advisory groups "Infrastructure Advisory Group" (INFRAG) and "Research & Innovation Advisory Group" (RIAG) and other European HPC sites. Further reports published so far cover "State-of-the-Art and Trends for Computing and Network Solutions for HPC and AI" [1] and "Data Management Services and Storage Infrastructures" [2]. The series will be continued in 2021 with further selected highly topical subjects.

Edge Computing aims to process the data very close to the source where it is produced. Many electronic devices are currently connected to the Internet of Things (IoT), which will produce a massive volume of data, and it might be even larger with mobile phones in a 5G network. Cisco Global Cloud Index estimated in 2019 that IoT devices will generate around 500 zettabytes of data [3]. Furthermore, data traffic will be approximately 10.4 zettabytes, which is up from 3.4 zettabytes in 2014 [4]. Moreover, by 2020, 50 million streaming IoT devices will be in use [5].

Edge Computing refers to processing the data on the device and very close to the device. This massive volume of data might be hard to handle entirely on the Cloud Computing network. To face this challenge, Edge Computing offers the full computation or part of the computation that can process the data at the Edge network, which is in very close proximity to the data source. It enables low latency, faster response, and more comprehensive data analysis.

Usually, devices connected to the IoT provide the service in healthcare, smart cities, smart grid, transportation, multimedia, and security. In general, those services depend on AI methodologies, which are compute-intensive and use massive data. A few years back, these devices usually sent the data to the cloud or local data centre to process the data. With ongoing development in Edge Computing, the part of the data at the Edge node can be processed, thus minimising the application's overall latency.

The rest of this technical report is organised as follows: Section 2 focuses on why Edge Computing is needed. Section 3 gives a quick overview of the difference between Edge Computing and Cloud Computing. Section 4 explains the technologies and architectures that are available for Edge Computing. Furthermore, virtualisation, resource management, and development for Edge Computing platforms and how these categories define the proper working functionality of Edge Computing are described. Section 5 focuses on Edge Computing's latest architecture trends, giving more detail about Nvidia Jetson, Raspberry PI, Tinker, and Kalray MPPA. Section 6 presents how Edge Computing still depends on supercomputers/data centres or Cloud Computing for data-intensive applications, particularly AI methodologies. Section 7 focuses on the applications of Edge Computing in real-world applications. Four applications are categorised with more examples in each category, such as healthcare, smart city, industrial applications, smart grid, and public safety. Finally, Section 8 presents a few challenges while implementing Edge Computing, specifically about naming and programmability.

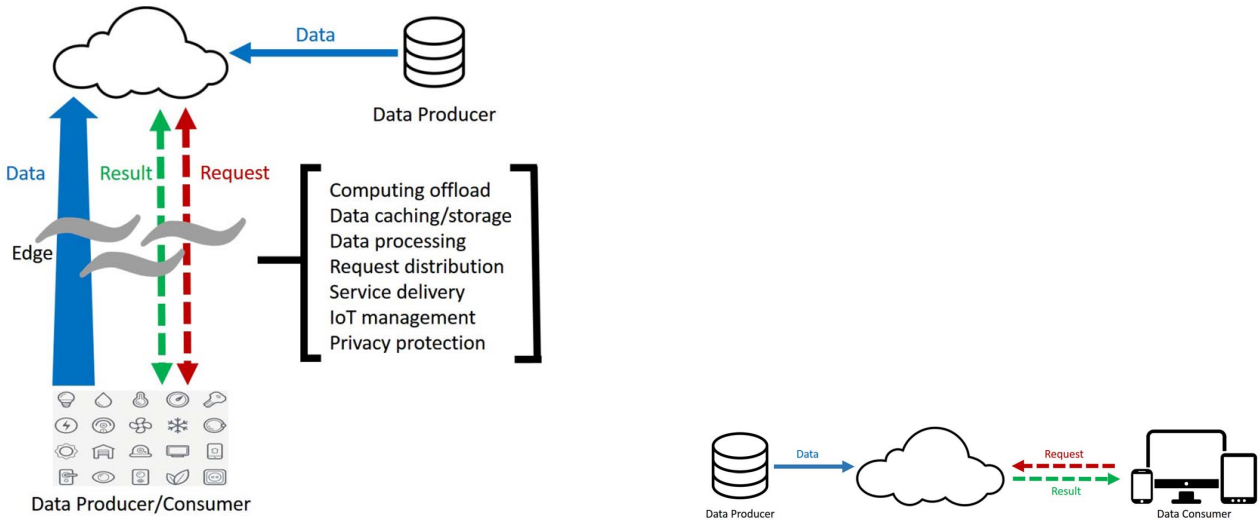
2. Why Edge Computing?

Currently, the entire world is going towards digitalisation, and lots of data is produced in various fields. Moreover, in most cases, this data needs to be processed in a short time to facilitate the present technology (real-time applications). A few years back, cloud technologies have been introduced, gradually reducing the need for small- and medium-scale companies and research institutes to own a computer to do the computations. Nevertheless, the end-users still need to send and receive the data to and from the location where the machine is located. In contrast, Edge Computing is an alternative option for doing computations where the data is located, and is especially suited for real-time applications.

In particular, part of the IoT might require short response time, private data, and Big Data, which could be challenging for the network. However, Cloud Computing cannot handle few of these challenges. Figure 1a and 2 show the paradigm and schematic model of Edge Computing.

Edge Computing is not a direct competition to Cloud Computing or supercomputers, but it is certainly sharing computational burden with cloud technology and supercomputers. If the present trend continues, more robust and energy-efficient small/embedded machines will improve the computations in the future. The following items provide information about why Edge Computing is needed:

- ***Push from the Cloud Services:*** In general, Cloud Computing has proven to be very efficient in terms of computation, but in some situations, there has to be an alternative solution to avoid data transfer bottlenecks. Edge Computing is solving this problem. For example, a Boeing 787 produces 5 Gigabytes (GBs) of data every minute [6], and transferring this data to a satellite or the ground is not efficient for data processing. Yet another example could be autonomous vehicles, and its information needs to be processed very frequently to steer the vehicle in the right direction. It is not feasible to process these data in the cloud as the bandwidth of the network is a bottleneck. Moreover, Edge Computing consumes less energy compared to cloud technology due to the minimal consumption of embedded devices.



(a) Edge Computing Paradigm.

(b) Cloud Computing Paradigm.

Figure 1: Workflow Model of Cloud and Edge Computing Paradigm [7].

- ***Push from the Internet of Things:*** Presently, electronic devices such as LEDs, surveillance cameras, and air quality sensors are part of the IoT, and they produce and consume a lot of data. In future, there will be even more electronic devices that will be connected to the IoT. It is not feasible to process all the data in the cloud due to the bandwidth and latency. This means some of the data need to be processed at the level of Edge devices. Moreover, privacy is a big concern for cloud solutions, Edge Computing can minimise this concern by restricting the data within the Edge. Figure 1b shows the traditional cloud computing paradigm, where raw data is produced and transferred to the cloud and consumers are sending the request to access the data from the cloud. Basically, this structure is not optimal since a large amount of data needs to be transferred, and in some situations, data privacy is also a concern.

- ***Change from a Data Consumer to a Producer:*** A device at the Edge not only consumes the data from the cloud, but it also produces the data and uploads the data to the cloud. Watching a YouTube video from your mobile phone, using Facebook and Instagram, are examples where Edge users pull the data from the cloud. At the same time, Edge devices produce the data, such as taking pictures or recording videos. When the Edge users try to upload this data to the cloud, it could be a lot of data depending on the resolution. This would occupy even more bandwidth for the uploading. In a situation like this, the resolution can be adjusted at the Edge device before uploading the data to the cloud.

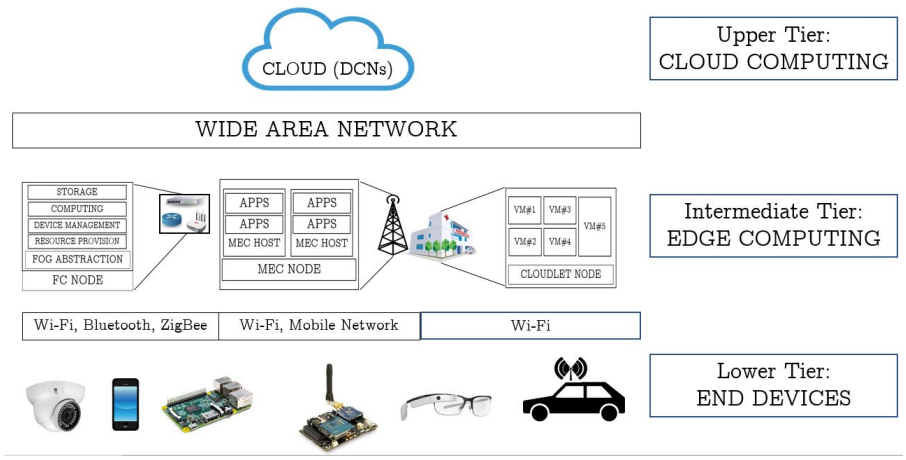


Figure 2: Edge Computing Example [8].

3. Comparison between the Cloud and Edge Computing Paradigms

Edge Computing is a paradigm relying on similar concepts deployed within Cloud models. In Cloud Computing, either in public, private or hybrid types of accessibility models, data processing or computation occurs at the data centre, where it has a substantial computational resource and data needs to be transferred back and forth. The cost model that made this paradigm so popular since the last decade is that the end-users only pay for the resources they used, whether in terms of computing, storage or data transfer capacities. In practice, the following deployment models are traditionally considered within the Cloud Computing paradigm [9]:

SaaS "Software as a Service", this refers to using the existing software or applications from the cloud, for example, using Gmail, Office 365, etc. Here, a cloud provider is controlling software or applications, and end-users use the software.

PaaS "Platform as a Service", this refers to using your software but using the cloud resource as hardware, for example, using the cloud's hardware and operating system. Here, a cloud provider offers middleware, development tools, operating systems, hardware and other business tools for end-users or customers.

IaaS "Infrastructure as a Service", refers to providing infrastructure, such as computing, storage, and cloud technology to the end-users. Here, end-users can scale down or scale up their computational platform as they want.

Several derived models were recently proposed, such are HaaS (Hardware as a Service), but their developments are considered out of scope for this report. In all cases, Figure 3 illustrates the available Cloud service models. Furthermore, Cloud technologies lead to several benefits for customers: 1) no need to employ anyone to maintain the computer, 2) no need to update the hardware and 3) minimisation of overall processing costs. At the same time, this model exhibits some drawbacks as well, which are: 1) slow network connectivity and internet traffic which make cloud technologies slower 2) security concerns, and 3) local data which can be stored in foreign countries, not subjected to the same data protection regulation as the GDPR in European countries.

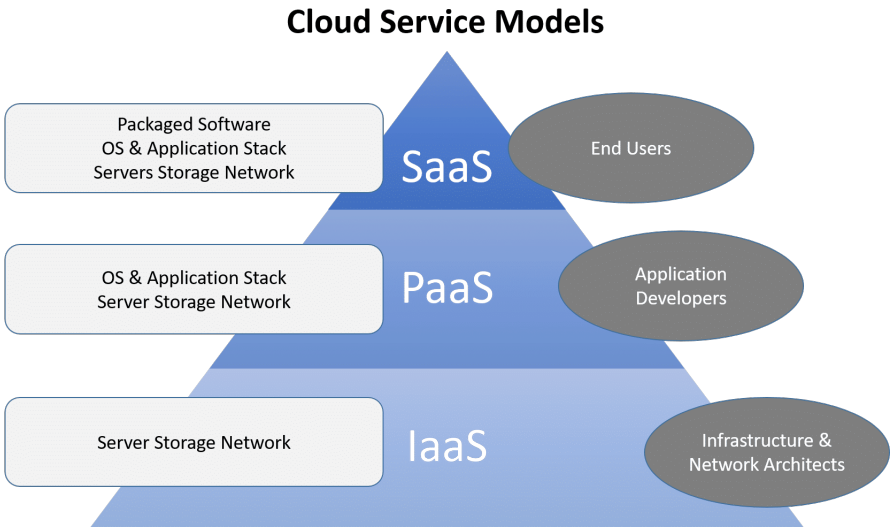


Figure 3: Overview of the main Cloud Computing deployment models [10].

It follows that even though Cloud Computing may be considered faster for data processing or computation, data transfer is a major bottleneck for Big Data analytics workflows. This is particularly relevant for IoT [11, 12, 13] environments, which tend to produce huge volume of data across interdisciplinary disciplines like technology, healthcare, environment, and transportation. In this case, a novel distributed and large-scale computing paradigm is required to effectively treat and analyse such large-scale dataset in a timely manner. That is how the Edge Computing model was introduced, with as its heart the idea to bring data storage and compute power closer to the device or data source where it is mostly needed. More specifically, the Edge Computing paradigm allows computing resources and application services to be distributed along the communication path, via decentralised computing infrastructures organised to treat in a hierarchical fashion the data analytic workflow. The hierarchy coupled with the distribution of computing capabilities aims at solving the bandwidth bottleneck identified for general Cloud architectures.

4. Edge Computing Architecture and Technology

4.1. Edge Computing Architecture

Edge Computing architectures are traditionally composed by several layers playing an essential role in the successful execution of the associated paradigm. Figure 4 and Table 1 show the schematic architecture and

characteristics of an Edge Computing environment, which are thus categorised according to the following deployment models:

- ***Cloudlet Computing.*** This refers to computing resources (small cluster) connected via WLAN to the end-users. In general, it can be considered as a "data centre in a box" which provides support (computing and storage) to the end-users over the WLAN network. Cloudlet Computing is based on three layers: the component layer, the node layer, and the cloudlet layer. This is designed to have higher bandwidth, thus lowering the latency for the applications.
- ***Fog Computing***, a decentralised computing resource that can be placed anywhere between the cloud and the end-users. It is based on the so-called Fog Computing Nodes (FCNs) [14]. All of these FCNs are heterogeneous, including switches, routers, and access points. FCNs heterogeneous environment facilitates the devices at different protocol layers and non-IP based technologies to communicate between the FCNs and the end device. These FCNs are hidden for the end-users, thus ensuring security.
- ***Multi-access Edge Computing (MEC)***, which refers to implementing Edge Computing within the Radio Access Network to reduce the latency. Formally known as Mobile Edge Computing, it is an ETSI-defined network architecture located closer to the Radio Network Controller or macro base station. The edge orchestrator organises the MEC, provides network information about load and capacity, and offers information to the end-users about their location and network information.
- ***IoT* (Internet of Things)** contains a large set of devices and sensors that produce a huge volume of data. These also exchange the data through a modern communication network and monitor and control the infrastructure. Typically, end-users at the Edge use the IoT devices and sensors.

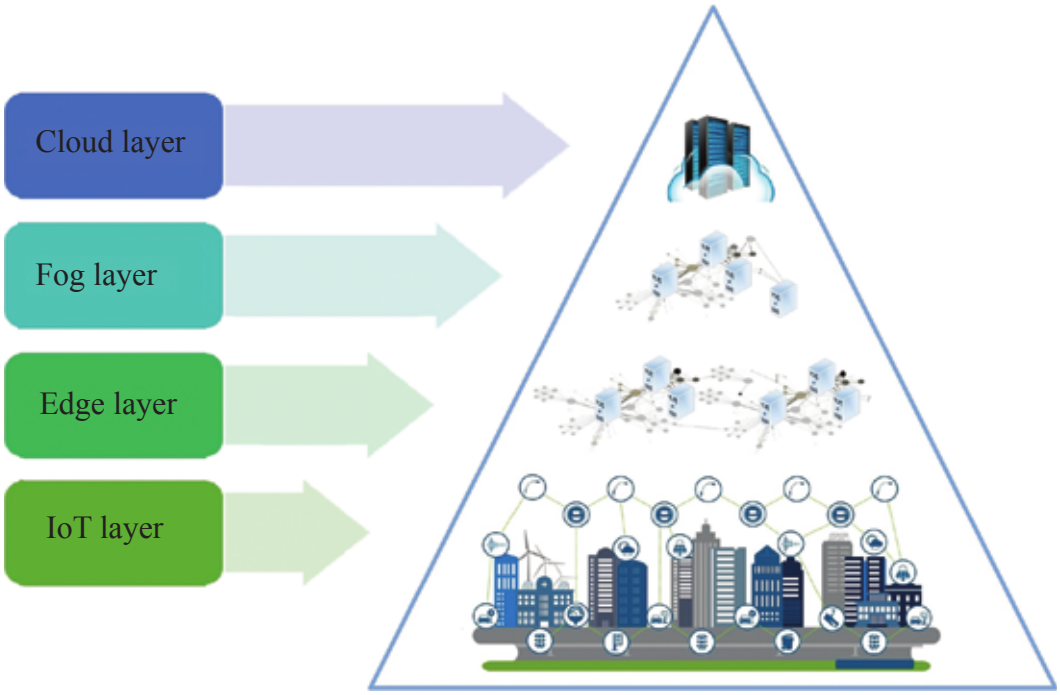


Figure 4: Overview of Edge Computing Architectures [15].

In general, Edge Computing involves complex or heterogeneous architecture. It is hard to ultimately make use of this complex architecture for some Edge Computing applications. However, many software platforms help to make Edge Computing work correctly and effectively. The following list provides more description about some of the important software platforms in Edge Computing.

4.2. Virtualisation

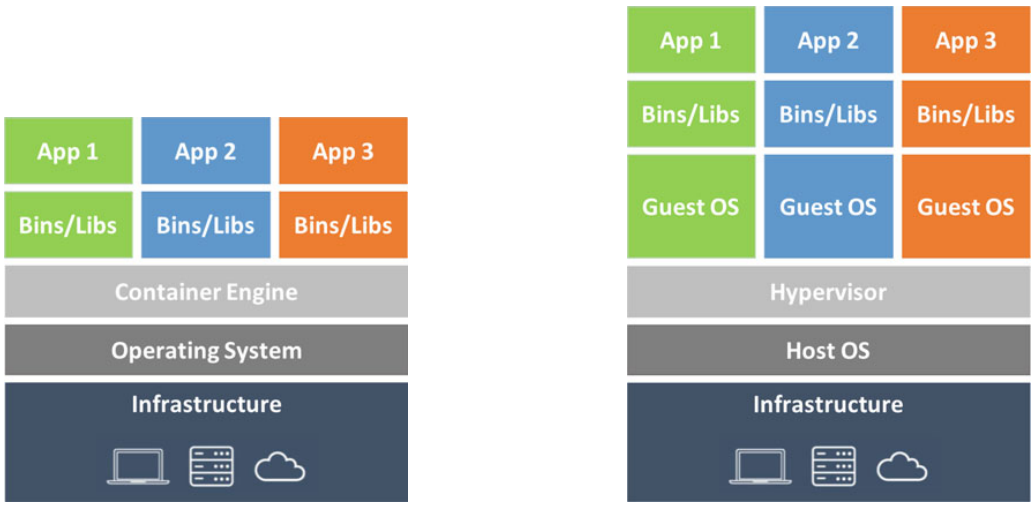
Virtualisation refers to an abstraction of an Operating System (OS), computing resources, storage device, and/or network devices. Especially in computing, the term *virtualisation* often implies the reference to the creation of a Virtual Machine (VM) managed by a *hypervisor*, a middleware responsible for providing an abstraction or emulation layer from the hardware. The hardware running the hypervisor is called the host whereas all emulated VMs running inside them are referred to as *guests*. In practice, there exists two types of hypervisors illustrated in Figure 5b:

	Edge Computing Architecture Layer			
Characteristics	**IoT**	**Edge**	**Fog**	**Cloud**
Deployment	Distributed	Distributed	Distributed	Centralised
Components	Physical devices	Edge Nodes	Fog Nodes	Virtual resources
Location awareness	Aware	Aware	Aware	Aware
Computational Limits	Limited	Limited	Limited	Unlimited
Storage Limits	Very limited	Limited	Limited	Unlimited
Data	Source	Process	Process	Process
Distance to data source	The source	The nearest	Near	Far
Response time	No response time	The fastest	Fast	Slow
Nodes count	The largest	Very large	Large	Small

Table 1: Main characteristics and functionality within Edge Computing Architectures [15].

1. *Type-1 (Native) hypervisors* are running directly on hardware (hence often referred to as bare metal hypervisors). Xen [16], VMware ESXI [17] or Hyper-V [18] are examples of such hypervisors.
2. *Type-2 (Hosted) hypervisors* require a host operating system whose capabilities are used in order to perform virtualisation operations and emulations. Examples of such hypervisors includes KVM [19] or VirtualBox [20].

With the advent of Docker [21] in 2013, another virtualisation model became popular: *container*-based virtualisation, which does not emulate an entire computer. Instead, the host operating system is providing most features to the container software in order to isolate processes from other processes and containers as depicted in Figure 5a. It is indeed a built-in feature of the Linux kernel to provide such isolation capabilities to isolate processes. Other operating systems may provide similar mechanisms, such as FreeBSD's jails [22]. This allows to design lightweight images - a code-based file that includes all libraries and dependencies, which makes this technology particularly suitable for the development of Edge Computing architectures. Because there is no full emulation of hardware, software running in containers has to be compliant with the host system's kernel and CPU architecture. Furthermore, since containers are so small, there are usually hundreds of them loosely coupled together, which is why *container orchestration frameworks* such as Red Hat OpenShift [23] and Kubernetes [24] (depicted in the next section) are used to provision and manage them.



(a) Container Environments [7].

(b) Type-1 (Bare-Metal) or Type-2 (Hosted) Hypervisors [7].

Figure 5: Main types of virtualisation frameworks.

Virtualisation is of course considered in other components of large-scale computing infrastructures relevant for Edge Computing. At the level of the network backbone, network virtualisation comes under the form of *Software-Defined Networks* (SDN), a manageable and cost-effective approach to enable dynamic, programmatically efficient network configuration in order to improve network performance and monitoring. This makes this type of architecture suitable for the high-bandwidth and dynamic nature of the applications that run on top of Edge Computing devices. SDN decouples the forwarding process of network packets (or data plane) inherent to the physical network, from the routing and control process. Often composed by one or more controllers, the

control plane is generally referred to as "brain of the network", where most of the crucial decisions are made as illustrated in Figure 6. Openflow [25] and OpenvSwitch [26] are a well-known Open-Source implementation frameworks in SDN network virtualisation.

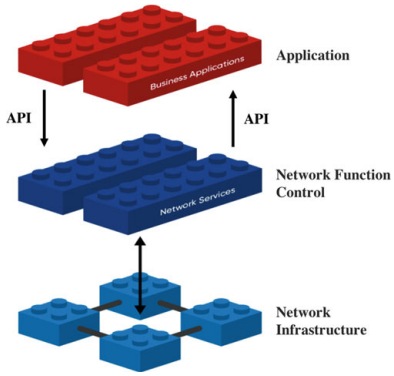


Figure 6: A high-level architecture overview of SDN [7].

4.3. Resource Management and Edge Orchestration

Resource management is crucial in terms of allocating computing resources, CPU, memory, storage, and network on standard traditional HPC settings. Especially for Edge Computing, the energy resource is very important. For example, a mobile user does not want to recharge his or her smartphone often, and also some sensors should not need to be charged frequently. In particular, for the Edge analytics, developers should know how much power is needed for the specific application and how data could be pulled/pushed from the device. Eventually, this kind of problem falls into the optimisation problem of scheduling and workload placement. In general, there are some important requirements for Edge platforms management, summarised in Table 2.

Requirement	Description
Scalability	Ability to address a large number of Edge devices of different type and capabilities with appropriate deployment and communicating protocol.
Security	Privacy preserving for security tokens and support for integrity checks within the infrastructure
Heterogeneity	Support for a high degree of heterogeneity within hardware/software
Volatility	Support for volatile availability and mobile hardware/software components
Data Protection	GDPR Compliance, ensure all data is kept locally and on-the-fly encrypted
Infrastructure Performance	Very low latency, lightweight publish-subscribe network protocol as MQTT [27]. High performance containerised resources with fast (Zero-touch) provisioning allowing easy system upgrades
Application Portability	Unified architecture view via MEC compliance enabling Function as a Service (FaaS) capabilities
Data Analytics	Supports for Data Management and Data Analytics Pipeline Engine.

Table 2: General requirement for Edge devices management and orchestration.

The orchestration of Edge Computing resources is thus quite challenging and still a work in progress. We can cite several frameworks for which an adaptation to the management of Edge devices is on-going:

- Generic container orchestration middlewares such as OpenShift [23] (Red Hat) or Kubernetes [24] (Google). Figure 7 depicts the generic working model architecture. In particular, Kubernetes features the following characteristics:
  - It is automatically bin-packing the containers based on the containers' resources and other constraints. For example, in Edge Computing, the computing resource of the Edge might host multiple users and applications. It is important to schedule them in an optimised way to use the Edge resources to minimise the energy consumption and achieve lower latency.
  - Kubernetes has a feature of self-healing, which is a significant factor for Edge Computing. For example, when there is a sudden failure or nodes are killed, users do not want to get notified; instead, the service and application should be recovered/migrated to other resources.
  - Another feature of Kubernetes is load balancing, which is based on the IP address and a single DNS name for a set of containers.

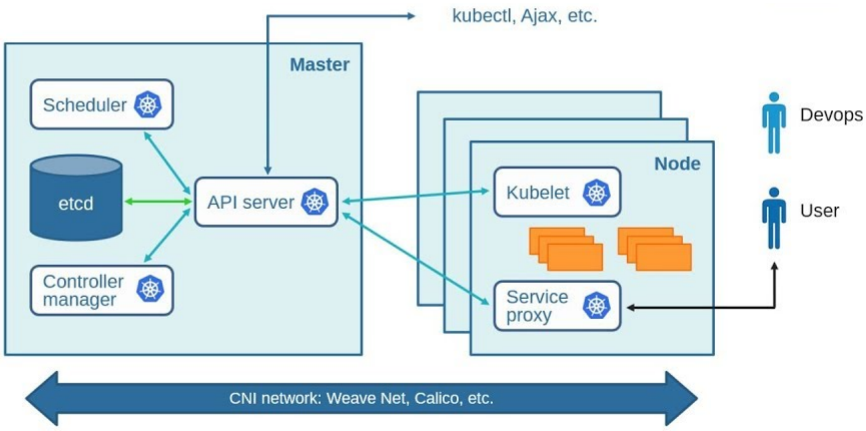


Figure 7: A high-level architecture overview of Kubernetes [7].

Among other initiatives, the KubeEdge [28] project is dedicated to making an open platform, which is built upon Kubernetes and provides fundamental infrastructure support for network, application deployment and metadata synchronization between Cloud and Edge architectures. To enable fast and lightweight communications between all the Edge devices (*i.e.*, from low power single board computers to full-capable servers and HPC compute nodes), the MQTT [27] protocol implemented over the Mosquitto [29] message broker service is used.

- ONAP (Open Network Automation Platform) [30], a comprehensive platform for orchestration, management, and automation of network and Edge Computing services for network operators, cloud providers, and enterprises. More tailored to MEC management and allowing to orchestrate physical and virtual network functions synchronously. The ONAP project provides a unified operating framework for vendor-agnostic, policy-driven service design, implementation, analytics and lifecycle management for large-scale workloads and services. A high-level overview of the ONAP platform architecture is proposed in Figure 8.

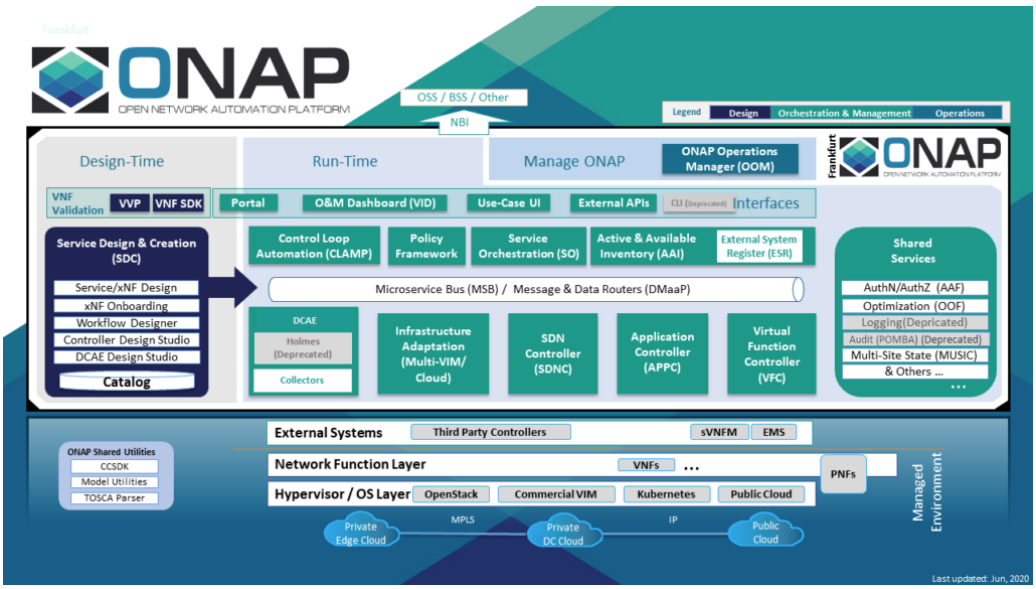


Figure 8: A high-level architecture overview of the Open Network Automation Platform (ONAP) [30].

4.4. Developing Platform enabling Data Analytics for Edge Computing

Edge Analytics refers to processing the data of the crucial application and service entirely or partially. For example, in healthcare, monitoring the older adults at home using the ECG (ElectroCardioGram) or EEG (ElectroEncephaloGram), it is better to perform the data processing at home, and only if something abnormal happens send out the data to the hospital or the doctor. Thus, the data flow between home and hospital is minimised, which reduces the data failure between two points. Not only the healthcare system can leverage this

model, nowadays, but electronic devices which are connected to IoT networks can also process the data and exchange the essential information. For example, such devices can be drones, robots, cameras, and sensors, etc.

As mentioned in Table 2, data analytics capabilities are required within the development tools and platforms, enabling an Edge Computing infrastructure. If the Edge appliances cannot process the data, then it has to send the data to the cloud. In this context, there are several open-source tools available as software platforms supporting the data analytics pipeline engine for Edge Computing.

- TensorFlow [31], a framework that supports Machine Learning (ML)/AI computations. It also supports a wide range of architectures, including, CPUs, GPUs, embedded and mobile systems. TensorFlow supports different programming languages, such as C/C++, Python, and Java. It even supports heterogeneous capability (CPU+GPU) computation.
- OpenCV [32], a software platform that supports computer vision computations. It has more than 250 optimised algorithms that can detect and recognise, *i.e.*, faces, identify objects, classify human actions in videos, etc. OpenCV also provides an API for Java, C/C++, and Python.
- Apache Edgent [33], an analytic tool that runs on the Edge device based on the Apache incubator project. It allows to store fewer data on the Edge device and limits the amount of data to be transmitted to the analytic server. Apache Edgent supports the Java API, and it makes Edge systems to be more autonomous.
- TensorFlow Lite [34], an open-source and designed for on-device inference, used for the AI/ML/ Deep Learning (DL) applications. In particular, it is used for image classification, object detection, and text classification.
- Apache MXNet [35], supporting a distributed computing platform with up to eight programming language bindings. Libraries in MXNet enables use-cases in computer vision and natural language problems. It is also has a rich eco-system that supports the other AI/ML/DL libraries for Edge Computing, *i.e.*, MXFusion [36], Keras-MXNet [37], and InsightFace [38].

5. Latest Trends in Edge Computing

Edge Computing will have to deal with ongoing developments in AI/ML/DL to do computational analysis and data processing. To do such complex arithmetic calculations, powerful embedded devices are required to perform these calculations are needed. There are lots of embedded hardware solutions available nowadays to support these requirements. Here we explain selected embedded architectures that will make Edge Computing more efficient and optimised.

5.1. Nvidia Jetson

Nvidia has introduced the Jetson series of embedded architectures to support ML calculations for embedded applications. Jetson TK1 was first introduced in 2014. Nvidia has released many series in embedded architecture since then. Nvidia also introduced the tensor cores to support the ML calculations for embedded architecture, which can be seen in Table 3. Figure 9 shows the Nvidia Jetson modules of Nano and Xavier NX. The Jetson Volta architecture has a Tensor Processor Unit (TPU), which can do computation on a large volume of data with low precision (which can be low as 8-bit precision). This kind of hardware functionality is quite useful for AI/ML/DL computations.

	Nvidia Jetson			
	Nano	**TX2 Series**	**Xavier NX**	**AGX Xavier**
Architecture	Maxwell	Pascal	Volta	Volta
CUDA cores	128	256	384	512
CPU	Quad-core ARM Cortex-A57 MPCore processor	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM Cortex-A57 MPCore	6-core NVIDIA Carmel ARMv8.2 64-bit CPU 6 MB L2 + 4 MB L3	8-core Carmel ARM v8.2 64-bit CPU, 8 MB L2 + 4 MB L3
Memory	4 GB 64-bit LPDDR4, 1600 MHz 25.6 GB/s	8 GB 128-bit LPDDR4 Memory 1866 MHx - 59.7 GB/s	8 GB 128-bit LPDDR4x @ 1600 MHz 51.2 GB/s	32 GB 256-Bit LPDDR4x 136.5 GB/s
Storage	16 GB eMMC 5.1	32 GB eMMC 5.1	16 GB eMMC 5.1	32 GB eMMC 5.1
Tensor cores	n/a	n/a	48	64

Table 3: Latest Nvidia Jetson Embedded Architecture Comparison.

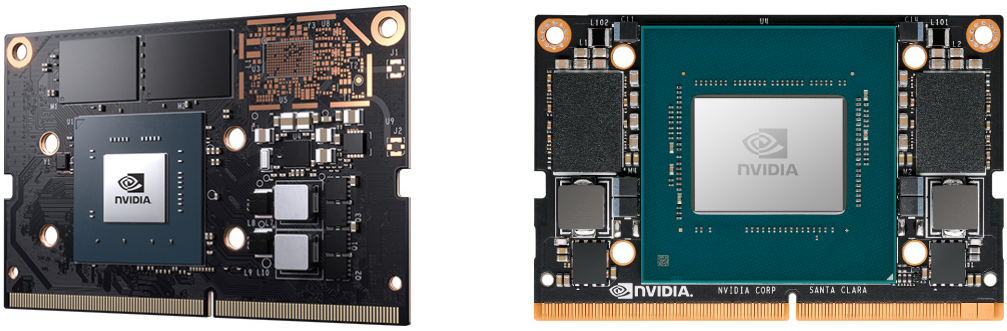


Figure 9: Nvidia Jetson Nano (left); Nvidia Jetson Xavier NX (right) [39].

5.2. ASUS Tinker Board

ASUS has introduced the ASUS Tinker Board in early 2017. This early embedded architecture can run in 32-bit mode, but the latest ones can run on 64-bit. And they are direct competitors to the Raspberry PI series. Table 4 shows the ASUS Tinker's latest embedded architecture, and Figure 10 shows the architecture model outline. Tinker Edge T has a Google Edge TPU as a coprocessor. The Google Edge TPU is based on an Application-Specific Integrated Circuit (ASIC). This means the TPU can do the specific application, *i.e.*, a digital voice recorder or a high-efficiency Bitcoin miner. On the other hand, Tinker Edge R has an AI-specific accelerator as a coprocessor.

	Asus			
	Tinker Board	**Tinker Board s**	**Tinker Edge T**	**Tinker Edge R**
Architecture	ARMv7-A (32-bit)		ARMv8 (64-bit)	
CPU	Quad core 1.8 GHz ARM Cortex-A17 (up to 2.6 GHz turbo clock speed)		Quad core 1.5 GHz ARM Cortex-A53	Hexa core. 2x Cortex-A72 cores up to 1.8 GHz,4x Cortex-A53cores @ 1.4 GHz
GPU	600 MHz Mali-T760 MP4 GPU		GC7000 Lite 3D GPU	800 MHz Mali-T860 MP4 GPU
Memory	2 GB dual channel LPDDR3		1 GB LPDR4	4 GB dual channel LPDR4 for system, 2 GBLP DDR3 for NPU
Co-processors	n/a		Google Edge TPU 4 TOPS of performance	NPU 3 TOPS of performance

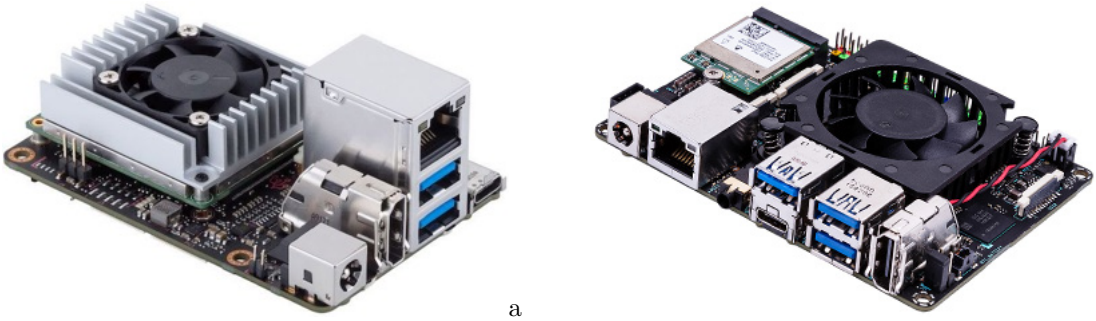
Table 4: Asus Tinker Board Architecture Comparison.

5.3. Raspberry PI

Raspberry PI is also an emerging embedded architecture in the Edge Computing domain. It has the latest ARM Cortex-A7 CPU and VideoCore GPU. This VideoCore GPU is based on Digital Signal Processing (DSP), which means it can efficiently process multimedia applications with low power consumption. Table 5 and Figure 11 show the latest Raspberry PI architecture and outline.

	Raspberry	
	RPI 3 Model B+	**RPI 4 Model B**
Archictecture	ARMv8-A (64/32-bit)	
CPU	4 Cortex-A53 1.4 GHz	4 Cortex-A72 1.5 GHz
GPU	Broadcom VideoCore IV @ 250 MHz	Broadcom VideoCore VI @ 500 MHz
Storage	8 GB	1, 2, 4 or 8 GiB

Table 5: Raspberry PI Architecture Comparison.



a

Figure 10: ASUS Tinker T (left); ASUS Tinker R (right) [40].

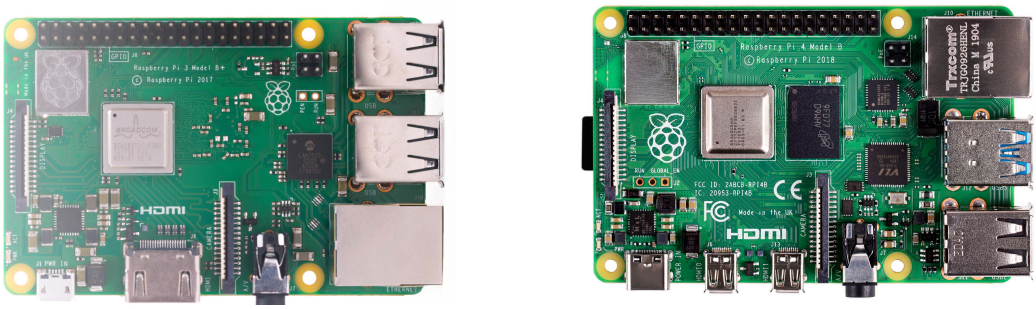


Figure 11: Raspberry 3 B+(left); Raspberry 4 B (right) [41].

5.4. *Kalray MPPA*

Another impressive embedded architecture is Kalray, which has many CPU cores, unlike other embedded architecture. Kalray named their embedded architecture "Massively Parallel Processor Array" (MPPA). Kalray 3rd generation MPPA architecture is called Coolidge (MPPA3-80 Coolidge), based on FinFET technology with 16 nm size. It has 80 high-performance AI accelerated and fully programmable cores, connected with a 600 GB/sec Network-on-Chip and advanced PCIe Gen4 and 200G Ethernet [42]. According to Kalray, the architecture does not have GPU cores comparable with other embedded architectures, yet KONIC200 has a different module, which can support Vision & AI, Storage, SmartNIC and 5G functionality. This can be seen in Figure 12. Table 6 shows the computational performance of AI/ML software. Here, TOPS refers to "Tera Operations Per Second", which defines the AI computational performance (INT8) on the given architecture. KONIC200 reports good AI performance even though it does not have a GPU, concluding that Kaltay MPPA is probably suitable for Edge Computing.

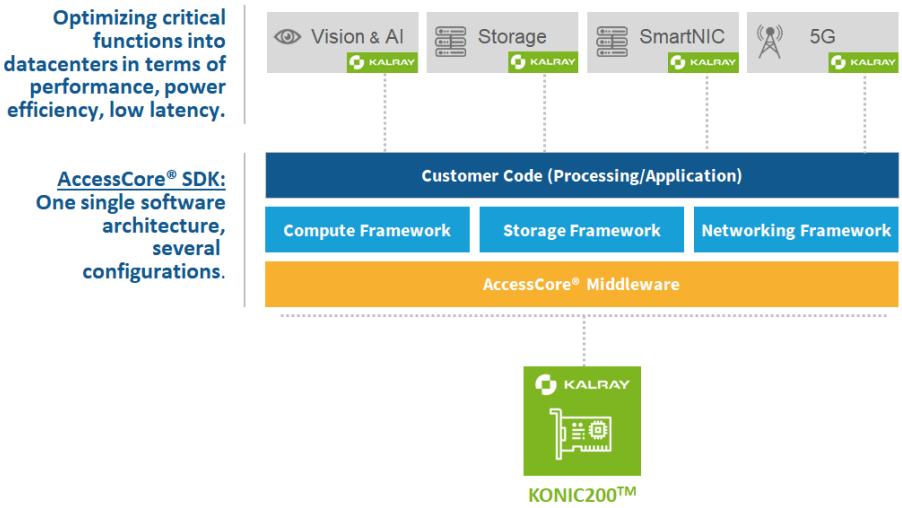


Figure 12: Modular Software in KONIC200 [42].

CNN Model	KONIC200 FH/LP (1 x MPPA (Coolidge) 80 cores)	KONIC200 - HP (2 x MPPA (Coolidge) 160 Cores)
TOPS	25 TOPS (8-bit)	50 TOPS (8-bit)
GoogLeNet	3025 fps	5445 fps
Faster-RCNN (VGG16)	302 fps	537 fps
Yolo v3	310 fps	564 fps

Table 6: AI and Compute Acceleration in KONIC200 [42].

5.5. Example Applications

Several compute-intensive applications are using the above mentioned advanced embedded architectures for Edge Computing. The following list shows a few relevant test-cases:

- Embedded architecture can process cryptographic functions to fix the privacy and security issues in the smart grid [43].
- A simple test has been done on TK1 at the edge for the MapReduce application. It seems the cluster of TK1 at the Edge shows the same throughput as one single traditional x86/64 Intel server while saving significant energy. Similarly, KMeans [44] also tested in three nodes of TK1 against one Intel server. The throughput is similar to each other, but 68% of energy is saved in TK1 compared to Intel [45].
- In recent years, there has been a lot of ongoing development for embedded architecture software development to use the latest embedded architecture. This enables Edge Computing to do real-time computations very quickly and efficiently. Figure 13 shows the available software in Nvidia Jetson, it can support, *i.e.*, DL and computer vision.
- Kalray supports acceleration of CNN, Computer Vision and Math apps up to 1.1 TFLOPS & 25 TOPS. It is easily programmable in C/C++/Open standards. And accommodates complex data flow in parallel and sequential modes [42].

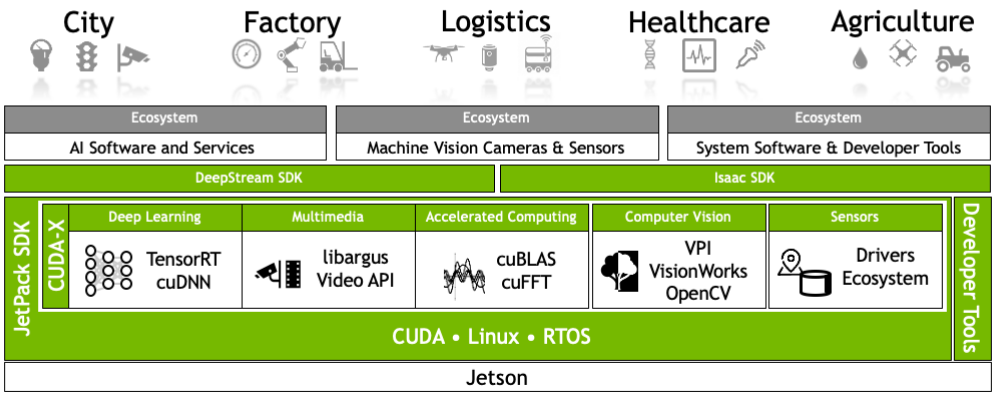


Figure 13: Jetson Software for AI Edge Devices [46].

6. In Reality: Edge vs. Supercomputers/Cloud Computing

As mentioned earlier in Section 5, there is an ongoing development of embedded architectures and Edge Computing based architectures. Moreover, Edge Computing is mainly deployed for AI/ML/DL computations. Nevertheless, almost all of these AI/ML/DL computations are very demanding computationally, based on mathematical and probabilistic modelling. For example, DL has three stages, which are 1) training, 2) evaluation and 3) prediction. In particular, the training part of DL requires significant computational power generally exceeding the capacities of traditional resources from Edge embedded architectures.

In addition, while ML is typically used for language translations, language recognition, autonomous vehicles, computer vision, text generation, and robots, these methodologies require a massive volume of data that needs to be processed at the training stage of DL. Presently, Edge Computing can not handle this massive volume of data (*i.e.*, videos, images, audio files, and text documents) for the training step in DL. Usually, it is best handled by a remote HPC clusters or supercomputers. Once the neural network is trained, it can be deployed at the Edge for prediction (*i.e.*, predicting videos, images, and audio files). In this case, even though Edge

Computing is evolving in terms of architecture and software, it still depends on the supercomputer/local cluster (or the equivalent Cloud Computing resources) for the data and computational intensive calculations.

In all cases, the following list depicts a few of the challenges that Edge Computing faces in this configuration:

- *Heterogeneous Data.* Often IoT devices produce different types of data, such as images, text, videos, and sound. Processing mixed data types need a special algorithm and requires more computational power. For example, multimodel deep learning is used for heterogeneous data (to process video and audio). On the other hand, it is not easy to implement this at the Edge device; eventually, this requires a Cloud Computing platform or cluster/supercomputer [47].

- *Distributed Computation.* IoT devices produce a vast volume of data at the Edge. Recently researchers have come up with the concept of an edge-based distributed learning algorithm [48], which is sharing the computation at the Edge and the Cloud, specifically doing less intensive computation at the Edge and high intensive computation at the cloud. This way, the workload is shared between the Edge and the Cloud. Presently, the edge-based distributed learning algorithm is used for fraud detection and market analysis. However, the accuracy and efficiency of the distributed model approach is still an open research challenge [49].

7. Selected Use Cases of Edge Computing enabled Applications

7.1. Medical Applications

- *Overview*: HPC at the Edge for medical imaging merges HPC/AI and medical sensing technology in order to provide precision medicine through the use of real-time advanced monitoring and analysis of a patient's medical data to detect early pathologies while lowering the risk of privacy breaches by keeping the data on site. This granular, yet massive amount of patient data can be analysed at the Edge, transformed, and then only pertinent data is sent to the cloud such as alerts or data stripped of information that could lead to the patient's privacy being compromised. Medical Imaging at the Edge using HPC/AI removes the latency and dependence on Cloud Computing resources, as well as reduces the patient's digital footprint by limiting how many systems have access to data. AI used in medical imaging provides tools that augment the clinician's intelligence in a way where they are able to provide better care at reduced costs [50]. Figure 14 illustrates the digital development in healthcare and how Edge Computing is being used in healthcare.

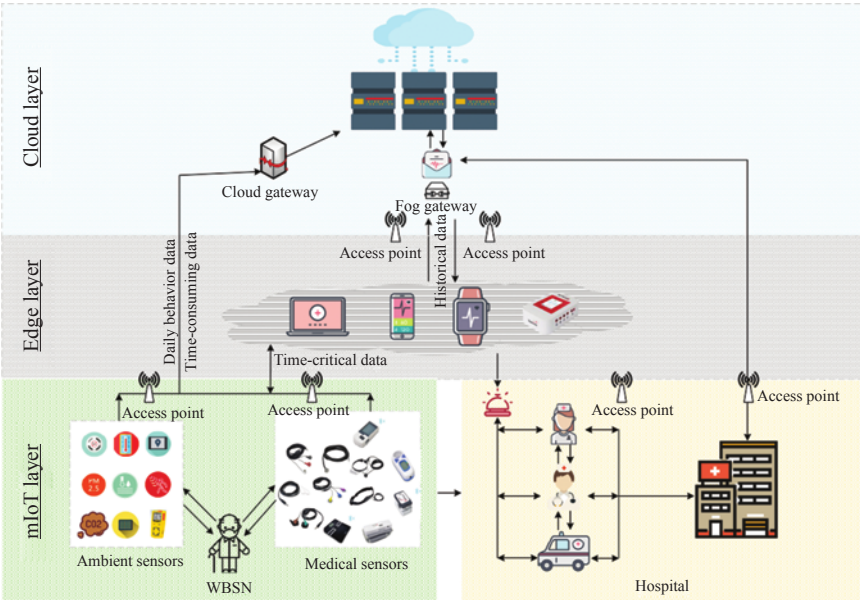


Figure 14: Edge Computing in Healthcare [51].

- *Examples*:

  Case 1: CT/MRI Scanning technology Centre for Clinical Data Science partnering with GE Healthcare, NVIDIA, Nuance, and DASA to build HPC/AI technology that is integrated into CT, MRI, and Workstation machines which utilise AI that is trained on vast data sets of diagnostic medical data [52].

Case 2: Handheld diabetic retinopathy diagnostic camera Through NVIDIA's virtual accelerator inception program, Taiwanese medical firm MiiS has built a highly portable handheld diagnostic tool that combines NVIDIA Jetson TX2, a high-resolution camera, Edge Computing architecture and GPU-powered AI algorithms to perform instant screening of diabetic retinopathy [53].

- *Security Concerns*: The EU has a strict legal framework in place to ensure consumer protection and to protect personal data and privacy, minimising risks to confidentiality and integrity of data [54]. Moreover, the EU GDPR sets certain rules and regulations for how the data is being processed and handled in EU [55]. In addition, there are specific rules that pertain to sectors such as healthcare that will continue to apply to AI and medical devices: Creation of comprehensive documentation and record keeping of data sets used for training and testing, programming and training methodologies such as the processes and techniques used to build, test, and validate AI systems especially those used to ensure the system is not biased in a way that could lead to prohibited discrimination arising from the usage of AI [56], auditing abilities of how a patient's medical data is being used as well as who is accessing it.

7.2. Smart City

- *Overview*: In the future cities will have sensors that will collect various data, for example, in transportation, medical health, and urban security. Moreover, urbanisation is rapidly increasing. According to the UN, it is estimated that, by 2050, over 6 billion people will be living in the cities [57]. In the future, to have sustainable development in the town, a smart city is an excellent solution. This might help to solve the problems that may arise in food supply, medical care, transportation, culture and entertainment in the cities. These sensors will usually generate a large volume of data, and this data should be processed quickly. Sending these data to the cloud will need faster data movement (latency and data traffic in the network), and privacy. Therefore, these generated data should be processed closer to where it is produced. In general, Edge devices have limited computing and storage, so it is also necessary to integrate multiple computing models. A few cases of Edge Computing used in a smart city are listed below.

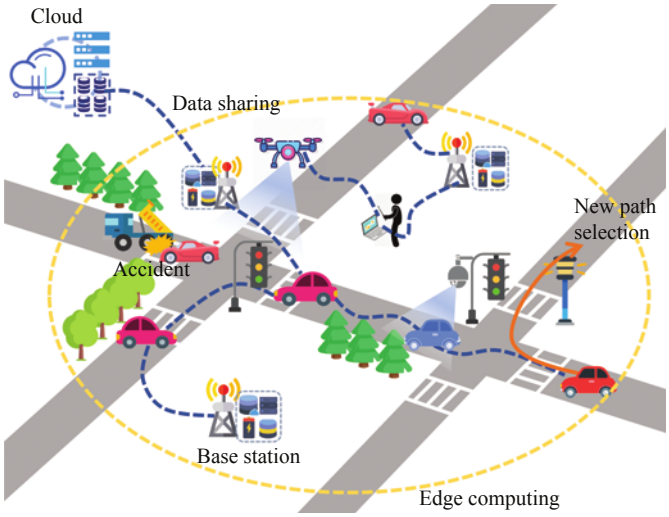


Figure 15: Example of Edge Computing in the Smart City [57].

- *Examples*:

Case 1: Closed-Circuit Televisions CCTV are nowadays typically installed in almost all private and government premises. These CCTVs will capture the movements of the objects. This will ensure the safety protocol in the given premises. For this reason, the data collected by the CCTVs should be processed quickly. To enable this, CCTVs should be connected to the Edge device through the LAN connection. Edge devices can also use the latest technology called special image processing chip to process videos more efficiently [58]. Figure 15 shows an example of a smart city.

Case 2: Smart Home A smart home is controlled by lots of sensors in lighting, kitchen, television, and surveillance, and its technology is also rapidly developing with the help of IoT [59]. Again, to make it more efficient (minimisation of the latency) and effective (privacy of home data), the generated data should be processed quickly where it is generated. Edge Computing will be playing an essential role in facilitating that objective.

- *Security Concerns*: while the user adoption for IoT technologies violating the boundaries of private areas is surprisingly high, as testify for instance by the success of voice assistant like Amazon Echo/Alexa or Google Home, this raises several security concerns for spying intelligence which remains an open challenge.

7.3. Industrial/Manufacturing Applications

- *Overview*: Industry 4.0 combines Edge HPC with AI in industrial automation environments. It aims towards waste reduction, work reduction, and worry reduction in the work space. It is used for connecting machines-to-machines and machines-to-people in a way where on-demand production environments, equipment, and workers can quickly and intelligently react to dynamically changing factory floor/environmental conditions [60]. Certain industrial applications may need to react quickly to real-time changing environmental conditions which may be uncovered in data too voluminous to be sent to the cloud, such as image recognition data that guides a robotic arm to interact with an object on a moving assembly line or creates alerts if dangerous conditions arise. Moving data offsite for analysis may also incur transmission latencies, that exceed the reactions times required for industrial applications, such as being able to shutdown an assembly line if a foreign object interferes with the industrial process. This is all assuming that the industrial site is even able to acquire a high-speed network connection due to geographic constraints.

- *Examples*:

  Case 1: Welding Quality Assurance : Rexroth, a Bosch company, has developed a Weld Spot ML analytics software which operates on a smart Edge that is located close to welding controllers. Until now, destructive tests were the only reliable way to test if a spot weld was done according to quality specifications. Weld Spot analytics software contains an AI engine which employs ML algorithms to detect anomalies and provides this information to welding engineers through a UI that displays a variety of data and analyses [61].

  Case 2: Worker Ergonomics safety A novel real-time spatio temporal Pyramid Graph Convolutional Network trained on video of warehouse workers performing typical repetitive duties and their associated movements was integrated with a traditional ergonomic risk index to assess the potential of musculoskeletal disorders in the warehouse [62]. This system can send alerts and warnings based on video analysis of actions that are above a certain risk threshold [63].

- *Security Concerns*: Limiting the amount of data sent to the cloud for processing also limits your attack surface from malicious actors. There is a much lower risk of your data being intercepted, tampered with, and stolen when it stays on-premise. Networking automated machinery poses a threat where compromised machines can be manipulated either directly or by tainting real-time and training data that your algorithms are using to control manufacturing processes [64].

7.4. Smart Grid and Public Safety

- *Overview*: Electricity is one of the primary sources for humans to conduct most of the activities in daily life. In recent years, special emphasis has been placed on how electricity is produced and distributed to facilitate better economic, technical, and environmental reports. In particular, how it is generated, distributed, and controlled, and monitored through digital instruments. The smart grid is a term that refers to how the whole electricity production and distribution are controlled by the smart digital instruments (for example, sensors) and embedded systems. Figure 16 shows an example of Edge Computing in the smart grid. Over the past years, surveillance security has been playing an important role in our daily life, for example, ATM centre. Most of the surveillance security is based on the visual feed, where this feed needs to be analysed quickly using AI/ML/DL for better security reasons without taking much time with accuracy. And also, sometimes, there is some high risk of data being manipulated or leaked over the network. The following cases show how Edge Computing will improve or tackle this problem.

- *Examples*:

  Case 1: Smart Grids Such infrastructures will benefit in numerous ways by adopting Edge Computing. For example, Edge Computing will make electricity to be bi-directional. This methodology allows the customer to be both consumers and producers of energy. This means that it will enable the customers to produce renewable energy (*i.e.*, solar power and biofuels) and sell back their excessive production to other consumers in the accessible market created by the Edge computing-enabled smart grid [43].

  Case 2: Visual Detection During the process of visual detection, there could be a threat that might come from the firmware, direct physical access, and visual layer-based attacks. Sometimes these threats can be identified or not. In order to eliminate these threats, Edge Computing offers many solutions through AI/ML. For example, in face spoofing attacks, one can use the online frame forgery detection technique. Furthermore, in many cases, at the Edge, it would be inefficient to process the high-resolution video frames. There has been an algorithm defined in AI/ML to overcome this problem, for example, the super-pixel-based technique [65].

- *Security Concerns*: the automatic tracking of citizen movements and usage facilitated by Edge Computing architectures raises a serious concern with regards privacy preserving rights, which have to analysed in the context of a degraded security climate due to the increase in terrorists attacks hitting in the last decades EU countries among many others. For sure the ongoing developments within the Edge Computing paradigm could be of great help to protect critical infrastructures as smart grids, or sustain global decisions improving the global security of our continents while preserving our privacy.

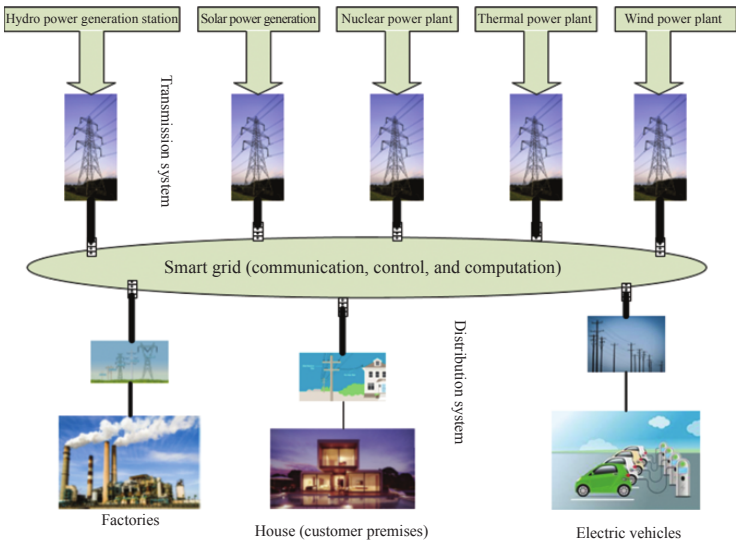


Figure 16: Example of Edge Computing in the Smart Grid [43].

8. Present Challenges

Even though Edge Computing is promising, there are still a few more areas that need to be well defined and documented. This will lead to further development in Edge Computing and its uses. The following list presents the areas where Edge Computing faces some challenges.

- **Naming:** Usually, Edge devices run many applications; each architecture has its method and structure. It is more convenient to have a naming scheme for programming, identification of things, and data communication in Edge Computing. But unfortunately, there is no standardised naming mechanism available for Edge Computing. This makes it very difficult for any programmer to understand the various communication and network protocols in Edge Computing [15].

- **Programmability:** In general, the Edge Computing architecture is heterogeneous. This makes the runtime and programming language different from standard architectures. Eventually, programming for Edge Computing becomes difficult. Whereas in the Cloud, the end-users usually can deploy their code written in a specific language. Moreover, Cloud Computing is known for its transparency [15].

- **Edge Device Management:** Managing Edge device is not an easy task, since it involves complex functionality, mainly, scalability, security, heterogeneity, and infrastructure performance. For example, Edge Orchestration should have the functionality of self-healing; this will minimise human intervention. Moreover, mobility is also a challenging topic; often, Edge devices move (for example, vehicles and cell phones). In this scenario, Edge devices shifting or collecting data from different locations, this will make a challenge in processing the data. Furthermore, Table 2 shows more of the critical functionality of the resource management and Edge Orchestration.

9. Summary

This technical report gives an overview of the Edge Computing paradigm and its applications, provides a comparison between Edge and Cloud Computing, and also points out the importance of this novel computing model to sustain the digital developments ongoing within our society. The key characteristics of Edge Computing architectures are discussed, including a brief survey of the orchestration middleware available together with the tools enabling the management, the effective deployment and the integration of data analytics capabilities within this novel distributed computing infrastructure. The latest trends at the heart of hardware developments for Edge Computing platforms are analysed, with concrete examples on the way Artificial Intelligence techniques and associated algorithms are tackled in this context. This technical report further explains why Edge Computing still depends on cloud technology or HPC supercomputers and lists a few critical challenges still opened in Edge Computing implementation. Finally, four categories of real-world applications affecting our daily life are proposed. They only illustrate the concrete benefit and potential impact this novel paradigm can bring to improve our digital society for the coming decades.

References

1. A. Tekin, A. Tuncer Durak, C. Piechurski, D. Kaliszan, F. Aylin Sungur, F. Robertsen, and P. Gschwandtner. State-of-the-art and trends for computing and network solutions for hpc and ai, prace technical report. *PRACE Technical Report, Dec 2020*, 2020.

2. A. Johansson, C. Piechurski, D. Pleiter, and K. Wadwka. Data management services and storage infrastructures. *PRACE Technical Report, Dec 2020*, 2020.

3. Cisco. Cisco Annual Internet Report (20182023) White Paper. *White Paper*, 2020.

4. Cisco. Global Cloud Index: Forecast and Methodology, 2014–2019. *White Paper*, 2014.

5. Cisco Edge-to-Enterprise IoT Analytics for Electric Utilities Solution Overview. URL `https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/big-data/solution-overview-c22-740248.html`.

6. Matthew Finnegan. Boeing 787s to create half a terabyte of data per flight, says virgin atlantic. *Computerworld UK*, 6, 2013.

7. Jie Cao, Quan Zhang, and Weisong Shi. *Edge Computing: A Primer*. Springer, 2018.

8. Koustabh Dolui and Soumya Kanti Datta. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2017.

9. Robert B Bohn, John Messina, Fang Liu, Jin Tong, and Jian Mao. Nist cloud computing reference architecture. In *2011 IEEE World Congress on Services*, pages 594–596. IEEE, 2011.

10. Types of Cloud Computing Structures. URL `https://www.uniprint.net/en/7-types-cloud-computing-structures/`.

11. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7): 1645–1660, 2013.

12. Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36, 2010.

13. Kevin Ashton et al. That "internet of things" thing. *RFID journal*, 22(7):97–114, 2009.

14. Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, and Rajkumar Buyya. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *Journal of Systems and Software*, 154:22–36, 2019.

15. Auday Al-Dulaimy, Yogesh Sharma, Michel Gokan Khan, and Javid Taheri. Introduction to edge computing. In *Edge Computing: Models, technologies and applications*, pages 3–25, 2020.

16. Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37(5):164177, October 2003. ISSN 0163-5980. doi: 10.1145/1165389.945462. URL `https://doi.org/10.1145/1165389.945462`.

17. Vmware esxi. URL `https://www.vmware.com/products/esxi-and-esx.html`.

18. Anthony Velte and Toby Velte. *Microsoft Virtualization with Hyper-V*. McGraw-Hill, Inc., USA, 1 edition, 2009. ISBN 0071614036.

19. Sun Microsystems Inc. The k virtual machine (kvm). White paper, 1999. URL `http://java.sun.com/products/cldc/wp/`.

20. Virtualbox. URL `https://www.virtualbox.org/`.

21. Docker. URL `https://www.docker.com/`.

22. Matteo Riondato. Freebsd handbook: Jails. URL `https://www.freebsd.org/doc/handbook/jails.html`.

23. S. Pousty and K. Miller. *Getting Started with OpenShift: A Guide for Impatient Beginners*. O'Reilly Media, 2014. ISBN 9781491904725. URL `https://books.google.fr/books?id=K6aSAwAAQBAJ`.

24. Kubernetes, . URL `https://cloud.google.com/kubernetes-engine`.

25. Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):6974, March 2008. ISSN 0146-4833. doi: 10.1145/1355734.1355746. URL https://doi.org/10.1145/1355734.1355746.

26. Open vswitch (ovs). URL https://www.openvswitch.org/.

27. Message queuing telemetry transport (mqtt), iso/iec 20922. URL https://mqtt.org/.

28. Kubeedge, . URL https://kubeedge.io/.

29. Eclipse mosquitto: An open source mqtt broker. URL https://mosquitto.org/.

30. Open network automation platform (onap). URL https://www.onap.org/.

31. Tensor flow, . URL https://www.tensorflow.org/.

32. Opencv. URL https://opencv.org/.

33. Edgent incubator, . URL http://edgent.incubator.apache.org/docs/edgent-getting-started.html.

34. Tensorflow lite, . URL https://www.tensorflow.org/lite/models.

35. Apache mxnet, . URL https://mxnet.apache.org/versions/1.7.0/.

36. Mxfusion. URL https://mxfusion.readthedocs.io/en/master/index.html.

37. Keras-mxnet. URL https://github.com/awslabs/keras-apache-mxnet.

38. Insightface. URL https://github.com/deepinsight/insightface.

39. Autonomous machines. URL https://developer.nvidia.com/embedded-computing.

40. Asus tinker board series. URL https://tinker-board.asus.com/product-series.html.

41. Raspberry pi products. URL https://www.raspberrypi.org/products/.

42. Programmable accelerator cards for data centers. URL https://www.kalrayinc.com/download/konic-80-200/.

43. Alem Fitwi, Zekun Yang, Yu Chen, and Xuheng Lin. Smart grids enabled by edge computing. In *Edge Computing: Models, technologies and applications*, pages 381–408, 2020.

44. Kmeans. URL https://github.com/NVIDIA/kmeans.

45. Dumitrel Loghin, Lavanya Ramapantulu, Oana Barbu, and Yong Meng Teo. A time–energy performance analysis of mapreduce on heterogeneous systems with gpus. *Performance Evaluation*, 91:255–269, 2015.

46. Jetson software. URL https://developer.nvidia.com/embedded/develop/softwarE.

47. MG Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *arXiv preprint arXiv:1908.00080*, 2019.

48. Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

49. Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, 2(1):1–11, 2013.

50. C Van Himbeeck. *Edge AI In Health Applications With A Focus On Hearing.* 2020.

51. Yucen Nan, Wei Li, Shuiguang Deng, and Albert Y. Zomaya. Smart healthcare systems enabled by edge computing. In *Edge Computing: Models, technologies and applications*, pages 337–356, 2020.

52. Clinical data science. URL ttps://www.ccds.io/partnerships/.

53. Nvidia blog. URL https://blogs.nvidia.com/blog/2020/05/24/medimaging-integrated-solution-edge-ai/.

54. Epf presentaion, . URL https://www.eu-patient.eu/globalassets/policy/data-protection/data-protection-guide-for-patients-organisations.pdf.

55. Bob Duncan. Can eu general data protection regulation compliance be achieved when using cloud computing? In *Cloud Computing 2018: The Ninth International Conference on Cloud Computing, GRIDs, and Virtualization*, volume 35. IARIA, 2018.

56. EU commission white paper, . URL https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf.

57. Wuhui Chen, Zhen Zhang, and Baichuan Liu. Smart cities enabled by edge computing. In *Edge Computing: Models, technologies and applications*, pages 315–337, 2020.

58. Amira Hadj Fredj and Jihene Malek. Real time ultrasound image denoising using nvidia cuda. In *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 136–140. IEEE, 2016.

59. Biljana L Risteska Stojkoska and Kire V Trivodaliev. A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454–1464, 2017.

60. Jay Lee, Hossein Davari, Jaskaran Singh, and Vibhor Pandhare. Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters*, 18:20–23, 2018.

61. rexroth a bosch company. URL https://www.boschrexroth.com/en/us/products/product-groups/welding-technology/weld-spot-analytics/index.

62. Matteo Rubagotti, Tasbolat Taunyazov, Bukeikhan Omarali, and Almas Shintemirov. Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control. *IEEE Robotics and Automation Letters*, 4(3):2746–2753, 2019.

63. Washington edu news. URL https://www.washington.edu/news/2019/08/19/ergonomics-machine-learning/.

64. Policy department EU parliament. URL https://www.europarl.europa.eu/RegData/etudes/STUD/2016/570007/IPOL_STU(2016)570007_EN.pdf.

65. Deeraj Nagothu, Ronghua Xu, Seyed Yahya Nikouei, Xuan Zhao, and Yu Chen. Smart surveillance for public safety enabled by edge computing. In *Edge Computing: Models, technologies and applications*, pages 409–433, 2020.

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EUs Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 823767.