Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

The computing continuum: From IoT to the cloud

Auday Al-Dulaimy [a,b,*], Matthijs Jansen [c], Bjarne Johansson [a], Animesh Trivedi [c], Alexandru Iosup [c], Mohammad Ashjaei [a], Antonino Galletta [d,*], Dragi Kimovski [e], Radu Prodan [e], Konstantinos Tserpes [f], George Kousiouris [f], Chris Giannakos [f], Ivona Brandic [g], Nawfal Ali [h], André B. Bondi [i], Alessandro V. Papadopoulos [a]

[a] *Mälardalen University, Sweden*
[b] *Dalarna University, Sweden*
[c] *Vrije Universiteit Amsterdam, The Netherlands*
[d] *University of Messina, Italy*
[e] *University of Klagenfurt, Austria*
[f] *Harokopio University of Athens, Greece*
[g] *Vienna University of Technology, Austria*
[h] *Monash University, Australia*
[i] *Software Performance and Scalability Consulting LLC., USA*

ARTICLE INFO

Keywords:
Computing continuum
Cloud computing
Fog computing
Edge computing
Mobile cloud computing
Multi-access edge computing
SDN
NFV
IoT
Use case
Reference architecture

ABSTRACT

In the era of the IoT revolution, applications are becoming ever more sophisticated and accompanied by diverse functional and non-functional requirements, including those related to computing resources and performance levels. Such requirements make the development and implementation of these applications complex and challenging. Computing models, such as cloud computing, can provide applications with on-demand computation and storage resources to meet their needs. Although cloud computing is a great enabler for IoT and endpoint devices, its limitations make it unsuitable to fulfill all design goals of novel applications and use cases. Instead of only relying on cloud computing, leveraging and integrating resources at different layers (like IoT, edge, and cloud) is necessary to form and utilize a computing continuum. The layers' integration in the computing continuum offers a wide range of innovative services, but it introduces new challenges (*e.g.*, monitoring performance and ensuring security) that need to be investigated. A better grasp and more profound understanding of the computing continuum can guide researchers and developers in tackling and overcoming such challenges. Thus, this paper provides a comprehensive and unified view of the computing continuum. The paper discusses computing models in general with a focus on cloud computing, the computing models that emerged beyond the cloud, and the communication technologies that enable computing in the continuum. In addition, two novel reference architectures are presented in this work: one for edge–cloud computing models and the other for edge–cloud communication technologies. We demonstrate real use cases from different application domains (like industry and science) to validate the proposed reference architectures, and we show how these use cases map onto the reference architectures. Finally, the paper highlights key points that express the authors' vision about efficiently enabling and utilizing the computing continuum in the future.

* Correspondig author.
E-mail addresses: auday.aldulaimy@mdu.se (A. Al-Dulaimy), angalletta@unime.it (A. Galletta).

https://doi.org/10.1016/j.iot.2024.101272
Received 3 January 2024; Received in revised form 18 May 2024; Accepted 27 June 2024
Available online 14 July 2024

1. Introduction

The Internet of Things (IoT) is a network of physical devices, called "*Things*", embedded with sensors. These "*Things*" are connected and exchange data with other devices to create an adaptive environment that can react to our needs. IoT breaks down barriers between the physical and digital worlds thanks to devices' ability to sense, communicate, analyze, and accordingly act. Such intelligence, supported by real-time data generated by IoT devices, revolutionizes and contributes to various aspects of our lives and paves the way for a connected, adaptive, and smarter future. The vast number of connected devices forms the endpoint or IoT layer, which generates data at regular intervals. Moreover, IoT layer devices can exchange data with systems at other layers, including the cloud computing layer, over the Internet.

The cloud computing model manages data in a centralized fashion in cloud data centers. However, this way of processing, analyzing, and storing data cannot meet diverse IoT applications' requirements. The remote locations of the cloud data centers require data transmission with IoT devices, which may result in network bottlenecks, intensive congestion, and latency issues. Thus, new computing models, like edge computing, emerged as an intermediate layer between IoT and cloud layers to overcome such issues and bring services to the proximity of IoT devices.

The IoT, edge, and cloud layers can be integrated to enable efficient and real-time data management for different applications and use cases. This is called the ***Computing Continuum***, in which computing is distributed across the layers of the continuum. Thus, high-performance, scalable infrastructure and high reliability of clouds, integrated with low-latency, privacy-preserving computation of edge, can be gained. The computing continuum, depicted in Fig. 1, consists of three main layers: endpoint, edge, and cloud layers. The endpoint or IoT layer includes objects or things surrounding us (*e.g.*, sensors, cameras, and mobiles). They are connected to the edge network and possibly with each other through wired or wireless connections. The edge layer combines edge nodes, which could be any device equipped with computing power. It is connected to cloud data centers. The cloud layer consists of large-scale data centers owned by cloud service providers. The cloud data center can be defined as a physical place for storing or processing data by utilizing the computing, storage, and networking infrastructures it hosts. Cloud data centers belonging to the same or different cloud service providers are usually connected directly or indirectly.

Considering the complexity of the computing continuum, there is a need to understand how to enable and utilize the layers and components efficiently, how new computing models can be integrated, and what potential directions are envisioned further to enhance functionality and performance. Given the absence of a universally accepted definition for the computing continuum, this paper takes a significant step forward by proposing a novel definition:***"The computing continuum is the paradigm that integrates, organizes, and considers the computing and network resources across different infrastructures (endpoint devices, edge nodes, cloud data centers, and the networks connecting them) for deploying workloads, instead of relying on a specific infrastructure"***. This definition serves as the cornerstone for exploring the computing continuum and its potential directions. Furthermore, we illustrate how existing and potential new elements of the computing continuum can be combined into new architectures, technologies, models, and concepts to meet the needs of new applications and services, primarily when these elements are hosted on diverse and possibly geographically distributed systems. To better convey this, we review the existing computing models, like edge and cloud computing. The computing continuum, which encompasses all these models, is discussed from the computation and communication perspectives, providing a template for running diverse applications in the computing continuum. We then design and create several novel use cases from different sectors (like industry, science, and health) to validate the reference architectures and to show how the unified reference architectures can be used to explore design trade-offs for application deployments in the computing continuum. In addition, the noteworthy trends related to the computing continuum are gathered and discussed. Moreover, this work lists various challenges that need to be studied to fully integrate the continuum layers and presents ideas for addressing such challenges. To conclude, this paper presents a comprehensive categorization and outlines the authors' vision of the future of computing. This vision is based on the diverse functional and non-functional requirements of the emerging applications, services, and use cases that will emerge in the future.

1.1. Motivation

As IoT applications generate a massive amount of data, sending all such data to the cloud for processing or storage is accompanied by challenges, e.g., latency. Depending on the system architecture, the amount of data involved, and the type of application, the associated computations could be done locally, close to the IoT layer, or centrally in the cloud. In contrast to cloud computing, several computing models (*e.g.,* edge computing) have emerged to offer the desired intelligence and necessary resources at the edge of the network to meet the requirements of novel applications and use cases. Models like edge computing can process the data locally, in a decentralized fashion, using heterogeneous resources. The advances of the emerging computation models give rise to a hierarchical architecture that can revolutionize the classical cloud computing architecture. The new architecture consists of three layers: the endpoints or IoT layer, which includes distributed end devices; the edge layer, which provides edge servers located at the edge of the network; and the cloud layer, which hosts central servers located in remote data centers. In addition to the computation models, the communication technologies have been customized to fulfill the requirements of connecting different nodes vertically (between different layers) and horizontally (within the same layer) to serve the edge–cloud applications efficiently. Most applications and use cases require optimizing all the aforementioned layers to converge toward a computing continuum. Orchestrating all layers enables the provision of reliable services under different dynamically varying conditions.

The computing continuum offers a wide range of innovative services [1] (*e.g.*, computation offloading, sites collaboration, and context awareness) that changed the concept of computing models. However, there are potential opportunities for improvement that

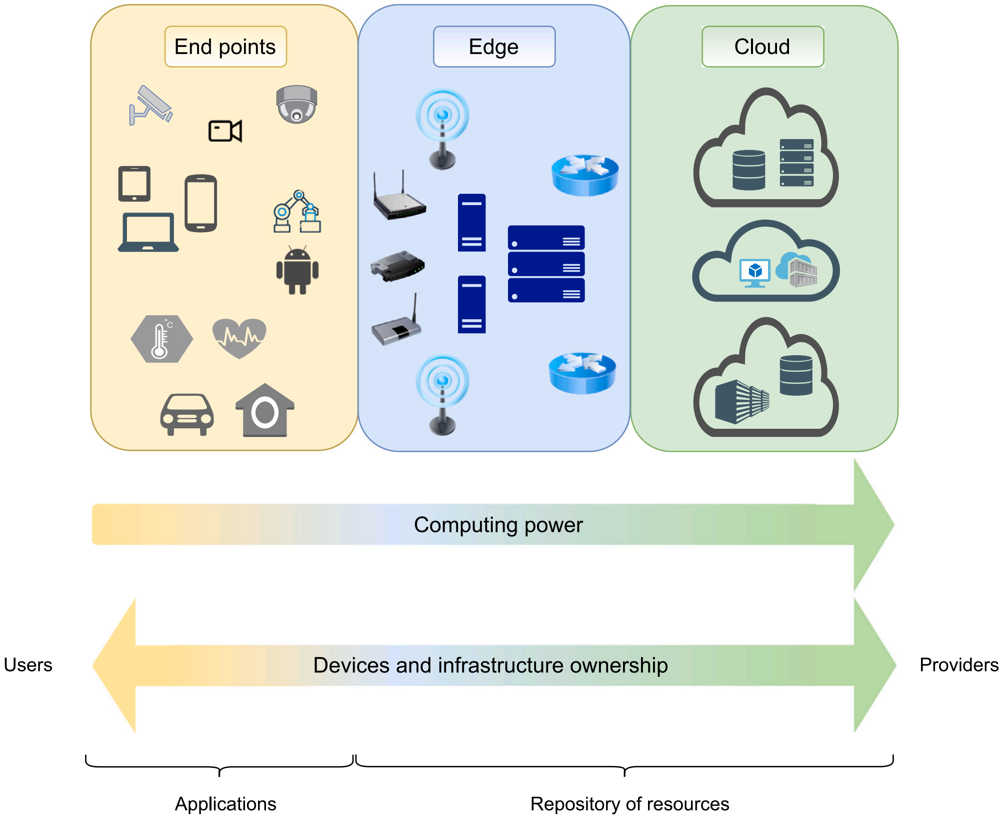


Fig. 1. The computing continuum.

lead to more efficient and elastic services, and challenges need to be investigated in the continuum. A better understanding of the computing continuum will guide researchers and developers in tackling and overcoming these challenges. Thus, this work presents a comprehensive overview covering the computation and communication aspects of the computing continuum. The work discusses the existing computing models and highlights their capabilities and limitations, proposes reference architectures, shows diverse real use cases and cloud-related trends, and finally illustrates the research directions on the computing continuum that should be explored.

1.2. Related work

Several works survey cloud computing and the computing models that emerged after it from different perspectives. A report in [2] prepared by the National Institute of Standards and Technology (NIST) provides a comprehensive overview of cloud computing with a reference architecture. Moreover, it discusses use cases in a cloud environment. The work in [3] describes the cloud computing architecture and its services and identifies several research challenges. In [4], the authors state their vision for enabling the computing continuum and support the vision by a use case. The authors in Ref. [5] summarize the main bottlenecks of cloud computing. They list some models that emerged after the cloud and discuss their current and future development opportunities. The survey in [6] discusses the changes the cloud infrastructure witnessed in the last decade, resulting in the need for various new computing architectures. It also addresses various challenges faced in the next generations of cloud systems. The work in [7] discusses the utility and cloud computing models. It also presents a comprehensive taxonomy of cloud computing research areas, divides them into sub-areas, and then lists each sub-areas challenges and future directions. Aiming to give a big picture of Future Generation Cloud Computing, the authors of [8] cover emerging trends, open challenges, and the research directions in cloud computing. The work in [9] discusses the practices in cloud engineering and its evolution and summarizes both technical challenges and research opportunities for the future of cloud computing from an engineering perspective. In [10], the authors present a survey of the edge and cloud computing models, compare the characteristics of these models, and discuss the orchestration of end–edge–cloud layers. Also, the authors discuss different topics in the three layers, such as computation offloading, caching, security, and privacy. Several potential research directions are presented as well. In [11], the authors present a review of edge and fog computing models. Besides, they cover several IoT use cases, discuss the topic of task scheduling in edge and fog computing, and show the role of software-defined networks (SDN) and network function virtualization (NFV) communication technologies in edge and fog computing. In addition, future directions are listed as open research challenges to be tackled by researchers. The work in [12] presents an overview of the edge–cloud computing model that assists the designs of Cyber–Physical Systems (CPS). It tackles several challenges related to

Table 1
Related works.

Related work	Targeted env.	Related trends	Use cases	Reference architecture		Challenges	Future directions
				Computation	Communication		
[2]	Cloud		✓	✓			
[3]	Cloud					✓	
[4]	Edge–Cloud		✓	✓			✓
[5]	Edge–Cloud		✓				✓
[6]	Edge–Cloud	✓					✓
[7]	Edge–Cloud	✓				✓	✓
[8]	Edge–Cloud	✓				✓	✓
[9]	Edge–Cloud	✓				✓	✓
[10]	IoT–Edge–Cloud					✓	✓
[11]	IoT–Edge–Cloud	✓	✓			✓	✓
[12]	IoT–Edge–Cloud	✓				✓	✓
[13]	IoT–Edge–Cloud	✓	✓			✓	✓
This work	IoT–Edge–Cloud	✓	✓	✓	✓	✓	✓

the QoS and summarizes possible future research directions on integrating edge–cloud computing with CPS. In [13], a survey that tackles fog computing architecture is presented. It discusses different techniques for utilizing computing resources at the fog layer and provides a use-case scenario to explain how to provision resources to an IoT application. The survey also lists several challenges and presents potential solutions for next-generation IoT applications from communication and computation perspectives.

These surveys are informative but do not delve deeply into the research issues in the computing continuum. Thus, the scope of our work is to cover the computation models that have emerged to date beyond cloud computing and to explain the communication technologies that support these models. We provide two novel reference architectures for the computing continuum and communication technologies. We present and discuss several use cases, including industrial, scientific, health, consumer, governance, and municipality domains. We relate these use cases to the proposed reference architectures and show how the orchestration of the computing continuum (in the three layers) can enhance the performance of the applications.

To our knowledge, no prior work in the literature has provided reference architectures for the computational model and the communication technologies in the computing continuum. Nor do prior works present use cases from different application domains while relating them to the reference architectures to validate them. Also, presenting several use cases from different sectors and relating them to the reference architectures to validate them are missed in the existing works. The contribution of the present work is summarized in Table 1, which shows the differences between our work and the related literature.

1.3. Contributions

Our work makes the following contributions:

- This work discusses the evolution of computing models toward the computing continuum paradigm. It focuses on cloud computing as the De Facto model that provides scalability, flexibility, cost-effectiveness, environmental sustainability, and a competitive advantage. The work provides elaborate illustrations of the cloud computing limitations that result in the need to propose new computing models to deal with such limitations.
- The work lists the computing models that form the computing continuum and the communication technologies that support these models. We explain the significance of aggregating all computing tiers and propose two novel reference architectures: a computing reference architecture to cover each edge–cloud computing model and a communication reference architecture to cover the edge–cloud communication technologies. We show how the reference architectures can be used to explore design trade-offs for application deployments in the computing continuum.
- We design and demonstrate several real use cases from different sectors. The use cases are novel, designed by the authors, and used to validate the proposed reference architectures by showing how their implementation layers are mapped into our description of the computing continuum. Besides the use cases, this work picks some of the cloud's current related trends, presents them, and states their importance to different sectors.
- The work highlights the challenges and limitations of different computing models and the computing continuum. This aims to provide insights into the limitations of these models and their impact on the overall computing systems.
- The work presents the authors' vision of the future of the computing continuum. Our vision states that applications and use cases will evolve toward more sophisticated use with more complex requirements, including computing resources and performance levels. Accordingly, the computing continuum needs to continue to evolve to meet such requirements. For this, we defined different research directions to be investigated for a more efficient and unified computing continuum while meeting applications' requirements.

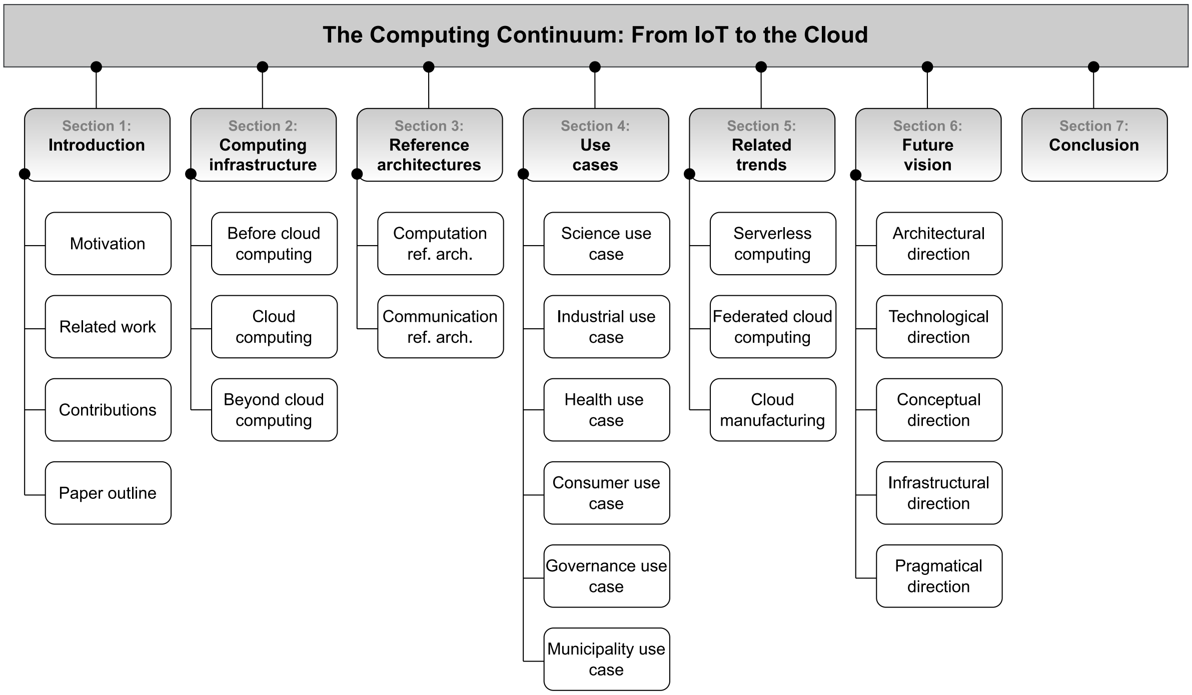


Fig. 2. The outline of the paper.

1.4. Organization of the paper

The remainder of this paper is organized as follows: Section 2 presents the evolution of computing paradigms, focusing on cloud computing, as it offers computing as a utility, which is a turning point in computing. Section 3 discusses the computing continuum and presents a computation reference architecture that considers this continuum. Also, the section maps different modern communication technologies to the computing continuum. Section 4 discusses several use cases from different sectors that employ the computing continuum. Section 5 shows innovative related trends spanning the computing continuum. Section 6 presents the authors' vision of the future of the edge–cloud continuum. Section 7 concludes the paper. Fig. 2 shows the outline of this paper.

2. Computing infrastructure: Before the cloud, the cloud, and beyond the cloud

Computing machines became common in the 1920s. Electronic computers replaced them in the 1940s and 1950s, making them smaller and faster. This led to the definition of the term "computing machine" being replaced with "computer", which was often preceded by "electronic" or "digital". Today, computers are everywhere and integrated into every aspect of our lives.

Computing passed by three main waves of evolution [14]: The first wave was when commodity-hardware-based servers replaced mainframes, while, at the same time, x86 architectures were becoming widespread in server space. The second wave was when computing infrastructure was virtualized to be shared by different applications. The third wave is the cloud computing era and its related trends. This section provides detailed descriptions of the way toward the computing continuum. It discusses cloud computing and the models before and beyond it.

2.1. Before cloud computing

Cloud computing has been built on top of several existing models and technologies. For example:

- *Mainframe*: Before mini-, micro-, and personal computers became widely available and cost-effective, applications were usually hosted on mainframes [15]. These provide access to data and processing power shared by batch and interactive workloads. In the early days of cloud computing, mainframes were sometimes used to host multiple Virtual Machines (VMs) supporting different services. Scaling up any service simply involves spinning up another VM if the mainframe has enough memory, disc space, and computing power to host it [16]. Users and programmers are often unaware of whether the hosting environment is cloud-based or mainframe-based, as the nature of the host is hidden from them. The key feature that distinguishes a mainframe from a cloud environment is that physical execution and resource allocation on the former are physically static, while both characteristics can be dynamically changed according to load and other operating conditions on the latter. Services and libraries may be supported on both. Unlike computations that are run on users' dedicated machines, those that take place in the cloud or on mainframes incur direct monetary costs that will only become visible to the system owner or account holder at the end of a billing cycle or, in the case of batch jobs on mainframes, at the end of job execution. Whether on mainframes or in the cloud, these costs are functions of memory occupancy, total CPU time, I/O activity, data storage size, and network bandwidth usage. The mainframes could be wholly owned or leased by a company or institution or made available to users by time-sharing bureaus. Either way, costs had to be attributed to enable amortization of the cost of acquiring and maintaining computer equipment and software. The cloud is no different, except that the access interface provides the illusion of infinite extensibility. Mainframes became available in the early 1960s [17], mainly to national laboratories and major scientific users.

- *Mini, micro, and personal Computers*. These are smaller general-purpose computers. The advent of mini-computers made it possible to place computing power in laboratories and other locations proximate to the users in a way the mainframe computers were not. The subsequent introduction of micro- and personal computers made it possible to think in terms of cheaply implementing distributed computations on machines that could be connected by local-area networks. The late 1960s and early 1970s witnessed the emergence of mini- and micro-computers [17], after Teletype Corporation introduced an electromechanical teleprinter Model 33 as a human interface. The teleprinter employed the newly standardized ASCII code for early mini- and micro-computers.
- *Cluster computing*: A cluster can be viewed as a processing system, mainly a composite of a set of computing nodes working together as a single unified computing resource with huge power. Moving towards cluster computing started in the late 1960s [18] when IBM connected large mainframes to provide more powerful and cost-effective systems.
- *Grid computing*: Grid computing is a style of computing where networked nodes function together to perform complicated tasks. Usually, such tasks need huge computing resources to be performed, such as analyzing or modeling big data [19]. Grid computing enables the sharing and dedicating of the distributed resources requested by the tasks dynamically at runtime. The resources are dedicated based on their availability, capability, cost, and users' QoS requirements [20]. IBM defined grid computing as [21]: *"A grid is a collection of distributed computing resources available over a local or wide area network that appear to an end user or application as one large virtual computing system. The vision is to create virtual dynamic organizations through secure, coordinated resource-sharing among individuals, institutions, and resources. Grid computing is an approach to distributed computing that spans not only locations but also organizations, machine architectures, and software boundaries to provide unlimited power, collaboration, and information access to everyone connected to a grid"*. In the late 1990s [22], the term "grid" was related to computing by describing a set of distributed resources connected over a Wide Area Network (WAN) to support a large-scale distributed application.

All these computing models utilized the parallel [23] and distributed [24] computing paradigms as a solution to perform different computing tasks.

2.2. Cloud computing

The cloud computing model delivers a wide range of services over the Internet via shared hardware resources that exist in data centers around the world. Conceptually, it can be viewed as three layers of observation. The resources represent the ***hardware layer*** of the cloud model. Cloud users do not interact with the hardware layer directly. Instead, they benefit from such resources via a ***virtualization layer***. Virtualization is the key concept of the cloud model that eliminates the details of the physical hardware and provides virtualized resources for ***user layer*** as VMs [25] or containers [26] forms. Many cloud computing definitions have been proposed over the past years as in [27–30]. However, no standard definition exists, but the most comprehensive definition, in our opinion, is the one presented by the National Institute of Standards and Technology (NIST). They defined cloud computing as [31]: "*A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources e.g. networks, servers, storage, applications, and services that can be rapidly provisioned and released with minimal management effort or service provider interaction*". They illustrated that the cloud model has essential characteristics, featured service models, and distinguished deployment models. More detail about the characteristics, deployment models, and service models can be found in [31].

However, all computing models, including cloud computing, have some limitations. Mainly, the limitations of cloud computing can be summarized as:

- **Connectivity**: To use cloud computing services, connectivity with remote data centers is a prerequisite. Most of the modernistic IoT applications and newfangled use cases require stable connections to meet their design goals. The functionality of such applications and use cases is negatively affected when the connection is temporarily unavailable and when it is under degraded conditions. Cloud data centers can offload computation power, but when the short reaction time plays the role among the other metrics to be considered, the decentralized fashion of cloud computing cannot solely be the best solution in serving some applications (*e.g.,* real-time applications).
- **Bandwidth issues**: To deliver its services, cloud computing deems sufficient bandwidth to exchange data with cloud data centers. The amount of data generated by the endpoints is huge, and when the number of such points is scaled, the amount of data is increased. Thus, the current cloud computing technologies will not suffice to handle the data streams.
- **Response time**: Transferring massive data will cause network bottlenecks leading to latency issues for applications and may cause a deterioration in computing performance. This would largely depend on the distance between the endpoints that generate data and the cloud data center. As the distance increases, the network cost rises, which is an overhead to consumers (*e.g.,* the small and medium scale manufacturers).

- **Centralized management fashion**: Cloud computing provides centralized operations, and its classical architecture does not support any localized services. This issue conflicts with the requirements of many applications and use cases.
- **Security aspects**: A major concern is the data security issues, which occur because of centralized data storage and sharing of resources for multiple users.
- **Application design aspects**: Cloud computing is best exploited by applications that have been designed with specific cloud-native architectural considerations (*i.e.*, scale up and down abilities, distributed operation, externalized state, service interfaces, application users multi-tenancy, ability to handle ephemeral resources and transient failures, etc.). However, this is still an ongoing challenge for various applications struggling to cope with the increased demands of this dynamic environment, lacking the ability to tackle the aforementioned issues [32].

By studying these limitations, and by reviewing some existing works in the literature [33–39], it could be concluded that the cloud computing model has many challenges to be tackled. The top challenges are Performance, Reliability, Availability, Scalability, Sustainability, Elasticity, Energy management, Virtualization management, Automated service provisioning, Traffic management, Data management, Latency, Location-awareness, Security, Privacy, Trust, Responsibility, Novel cloud architectures, and Application engineering for the cloud.

To overcome such limitations and challenges, novel computing models with new architectures are needed to extend the services from the cloud data centers to be closer to the users at the edge and/or core of the network. The models that emerged after cloud computing are listed, discussed, and compared in the next section.

2.3. Beyond cloud computing

Despite the increasing usage of cloud services, challenges limit their wide adoption, as explained in the previous section. To overcome these challenges, it is essential to modify the classical architecture of cloud computing to fulfill the requirements of different cloud-based applications and meet the rapid growth of the demands on cloud services. Thus, many models have been presented beyond cloud computing. In this paper, the emerging models are classified into two main categories: the *Computation Models* and the *Communication Technologies* available to implement them.

2.3.1. Computation models

The following computational models bring computing power closer to the systems or system components being controlled or enable computations to run where more computing power and data are needed than can be hosted on a remote device.

1. *Mobile Cloud Computing*: The term Mobile Cloud Computing (MCC) means running mobile applications on cloud computing resources. In this model, mobile devices act as clients that connect to the remote servers in the cloud via the Internet. Thus, MCC can be defined as [40]: *"A model for elastic augmentation of mobile device capabilities via ubiquitous wireless access to Cloud storage and computing resources, with context-aware dynamic adaption to changes in the operating environment"*. Mobile devices can run a broad range of applications, such as entertainment, business, and social media applications. These applications are becoming more complex and demanding of resources daily. Mobile devices are accompanied by problems such as resource limitation, low connectivity, and energy consumption. Researchers address and solve these problems through the cloud computing model. In mobile computing, thanks to the offloading concept, the computing power of cloud computing can be lent and used to execute and serve mobile applications by using remote and more powerful resources, instead of using the mobile device itself [41,42].
2. *Edge Computing*: Edge Computing (EC) is a decentralized model that aims to push the processing, storage, and networking capabilities to the network's edge. Recently, the number of devices (like smartphones and wireless devices) and sensors that are connected to the Internet has increased to a great extent. Such devices/sensors generate massive data, resulting in what is known as Big Data [43]. The architecture of the traditional cloud computing model was not designed to support the vast amount of data generated from IoT devices and sensors. Data from these devices and sensors has multiple attributes, such as Volume, Velocity, Variety, Veracity, and Value. Moving all data from the network edge to the cloud data centers for processing and/or storing may outstrip the network bandwidth capacity and result in excessive latency. As the amount of data increases exponentially, new problems, such as latency and low response time, need to be addressed. A possible approach to tackle such issues is to deal with the generated data near the devices that produce those data, i.e., the EC.
3. *Fog Computing*: Fog Computing (FC) is also a decentralized model whose purpose is moving cloud computing resources closer to the users to improve applications' overall efficiency and performance.
   EC and FC aim to move decision-making operations and program logic closer to the data sources and act as intermediate layers between Cloud data centers and devices/sensors that generate the data. But there are slight differences between them [44]: Fog nodes are extensions of the cloud and can exist in layers of hierarchy. Thus, balancing the load and transferring data between the nodes is possible. Fog nodes can be considered extended infrastructures in a hybrid cloud, while edge devices may or may not involve the cloud layer in the processing or storage services. In other words [45]: "*Both fog computing and edge computing involve pushing intelligence and processing capabilities down closer to where the data originates. The key difference between the two architectures is exactly where that intelligence and computing power is placed*".

Table 2
Comparison between existing computing models.

Characteristic	MCC	EC	FC	MEC	CC
Deployment	Distributed	Distributed	Distributed	Distributed	Centralized
Location awareness	Aware	Aware	Aware	Aware	Not aware
Physical resources	Limited	Limited	Limited	Limited	Unlimited
Distance to data source	Very close	Close	Close	Close	Far
Response time	Very fast	Very fast	Fast	Fast	Slow

MCC: Mobile cloud computing; EC: Edge computing; FC: Fog computing; MEC: Multi-access edge computing; CC: Cloud computing.

4. *Multi-Access Edge Computing*: This model, which was previously known as Mobile Edge Computing, is standardized in the European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG). Multi-Access Edge Computing, or MEC, combines IT services with cloud computing capabilities at the edge of the mobile network, aiming at reducing latency. Mobile subscribers can utilize this model to improve their users' experience as MEC offers effective network operation and efficient service delivery. According to ETSI, MEC now stands for "Multi-Access Edge Computing" to better reflect the growing interest in MEC apart from the mobile subscribers, as it enables a broader range of services [46]. Based on a virtualized platform, MEC is recognized as one of the key emerging technologies for 5G networks [47] together with Network Functions Virtualization (NFV) and Software-Defined Networking (SDN). MEC can enable various services to support new and innovative directions. Moreover, MEC can be considered a potential enabling technology for 6G networks, as it provides computing and caching capabilities required in the next generation networks [48]. However, MEC services fall into the following main categories [49]: (1) Consumer Oriented Services, *e.g.*, Gaming and Augmented Reality, (2) Network performance and QoE improvement, *e.g.*, Network Utilization Optimization and Intelligent Video Acceleration, and (3) Operator and third party Services, *e.g.*, Video Analytic and Connected/Autonomous Vehicles.
5. Other models: Other models emerged after the cloud. All these models fall under the umbrella of edge computing. For example:

   - *Cloudlet*: Cloudlet is a decentralized self-managed system that uses three layers (mobile device, cloudlet, and cloud) to bring the cloud services closer to being utilized by mobile devices. Cloudlet nodes exist in physical proximity to mobile devices. The nodes are accessible by the devices via high-speed wireless links. On the other hand, the nodes are connected to the cloud via high-speed access. The load of the mobile devices can be offloaded to local and more powerful resources in the cloudlet, which can be set up in common areas so that mobile devices can connect and act as clients to the cloudlet. In general, nodes at the cloudlet layer need to be powerful and very well connected via a high bandwidth network to provide low end-to-end latency [50].
   - *Dew computing*: This model allows operations to be performed on a personal local computer. This computer provides functionality and collaborates with cloud services [51].
   - *MIST computing*: This model pushes processing closer to the network edge. This can be done by involving the sensor and actuator devices in the processing, which results in (1) increasing subsystems' independence and (2) decreasing latency. However, managing the resulting network in MIST computing is not trivial. The only way to do this is via the resource-constrained devices at the network edge in a manner that resembles central management. This is not feasible [52].
   - *Osmotic computing*: Osmotic computing (OC) is an innovative resource management approach that automatically moves the computation from a Thing in IoT to the cloud based on the workload [53]. The idea behind OC is to decompose applications into microservices running in containers and deploy them opportunistically in Cloud/Edge and IoT systems. The reliability of services in OC is guaranteed by a Software Defined Membrane (SDMem) [54], an orchestrator that, starting from a service descriptor, monitors microservices and guarantees migrability and load balancing.

Table 2 highlights the similarities and differences of the discussed computing models.

2.3.2. *Communication technologies*

This section presents an overview of different communication technologies that can be used in the edge/cloud continuum (*i.e.*, between end node to edge and cloud). The communication technologies are commonly divided into the infrastructure, known as physical and data-link layers, and the application layers, where data exchange is handled. Table 3 provides an overview of the most common low-level communication technologies, while Table 4 presents common examples of application layer technologies. As shown in Table 3, each technology covers different requirements concerning determinism and reliability; thus, we identified those requirements in the table. For example, Ethernet technology for industrial systems is covered in the IEEE 802.1 standards, providing up to 10Gbit/s throughput. The determinism and reliability columns present different standards that support the increase of determinism and reliability. For example, IEEE 802.1Qbv, IEEE 802.1Qb, and IEEE 802.1Qca are all different standards under IEEE 802.1Q, known as Time-Sensitive Networking (TSN) standards, that increase real-time support in Ethernet networks. The latency column shows the lowest guaranteed latency possible in end-to-end communication using the concerned technology. In addition to the above specific communication standards, the Fifth Generation (5G) Wireless Communication System, and the next generation 6G, is becoming one of the wireless communication standards that can cover wide signal coverage with low latency. 5G can achieve

Table 3
Low layer technologies.

Tech.	Throughput	Range	Determinism	Reliability	Latency
Ethernet IEEE 802.1 family	10 Mbit/s–10 Gbit/s	100 m–40 km	IEEE 802.1Qbv - Enh. for Sched. Traffic IEEE 802.1Qbu - Frame Preemption IEEE 802.1Q - Quality of Service	IEEE 802.1Qca Path Ctrl and Rsrv. (PCR) IEC 62439-3 Clause 4 - PRP IEEE 802.3ad - Link Aggregation	≥100 μs
WLAN IEEE 802.11 family	1 Mbit/s–1,3 Gbit/s	10 m–100 m	IEEE 802.11e - Priority levels (QoS)	IEEE 802.11g IEEE 802.11n	≥20 ms
5G NR	10 Gbit/s	Multi-hop	3GPP specifications	3GPP specifications	≥3 ms

Table 4
Application level protocols.

Protocol	Model	QoS	Transport req.	Header	Field
AMQP	P2P	1–3	Reliable	8	Business applications
CoAP	CS PSl	–	Unreliable	4	Lightweight resource constrained devices
HTTP	CS	1	Reliable	>40	Wide range (browsers, devices, etc..)
MQTT	PSb	1–3	Reliable	2	Consumer devices and industry
OPC UA PubSub	PSb PSl	1–3 –	Reliable Unreliable	>30	Industry

system capacity growth of 1000 times and end-to-end latency reduction of 5 times compared to 4G and provides energy efficiency growth of at least 10 times and the area throughput growth of at least 25 times [55]. The next generation of 5G, known as 6G, is under design and development, which can be a game changer in time- and safety-critical industrial systems [56].

Table 4 presents the most common communication protocols and standards in industrial communication and edge/cloud continuum. The model column shows the communication model, for example, Peer-to-Peer (P2P), Client–Server (CS), Publish–Subscribe (PS) with a central broker (PSb), or brokerless (PSl). We use the same classification for QoS as MQTT: 1 — messages are delivered at most once, 2 — messages are delivered at least once, and level 3 means that messages are delivered exactly once. The transport requirement column specifies the protocol's requirements on the underlying transport protocol, reliable or unreliable. Reliable typically translates to ordered connection-oriented protocols, such as TCP, while unreliable can be realized with unreliable, connection-less protocols, such as UDP. The header column shows the overhead induced by the protocol header. Finally, the field column gives examples of fields/domains using the protocol.

Each communication technology is suitable for a specific topology. For instance, industrial Ethernet technologies, *e.g.*, TSN devices, are connected in the form of star, ring, or hybrid topologies.

Regarding communication management, the industrial networks within the factory LAN or connecting to the cloud require a management system to meet specific requirements, such as latency, reliability, and service continuity. These requirements can be considered offline during the network design and development and modified during run-time, known as run-time reconfiguration. There are several technologies which facilitate the network management, for example:

1. Software-Defined Networking (SDN): The architecture of traditional networks is a vertical architecture that consists of three core logical planes: (1) Control plane, (2) Data plane, and (3) Management plane. Such a vertical fashion adds complexity to the networks. Moreover, it makes it difficult to configure the networks according to predefined policies and to reconfigure them to respond to faults, load, and changes [57]. To simplify such complexity and make a network faster and easier to manage, SDN abstracts the vertical planes from applications and use cases, centralizes the network management, and enables a programmable network. This is done by an essential element of an SDN architecture, called the controller. SDN divides any network control problem into smaller tractable sub-problems to simplify network management. When forwarding a network, the control from network devices (switches and routers) is seized, and the controller logically enables centralized management and intelligence. In this way, the network infrastructure is abstracted from the applications. Redesigning networks with SDN offers flexibility and programmability across wide area networks (WAN) and data centers. Thus, it is widely adopted across data centers [57–59].
2. Network Functions Virtualization (NFV): Virtualizing the network combines software, hardware, and network functionalities into a manageable and economic domain known as a virtual network [60]. NFV architecture includes three components [61]: (1) NFV Infrastructure (NFVI): encompasses all networking hardware and software resources that represent the environment and are needed to execute the Virtualized Network Functions. (2) Virtualized Network Functions (VNF): represents the software that implements a network function (NF) running on NFVI. (3) NFV Management and Orchestration (NFV M&O): involves the management and orchestration of the NFVI and VNFs.
   NFV makes NF independent from the hardware and the other NFs. Implementing NFs is performed via software that can run anywhere in the network as required, without the need for installing new equipment [61].

SDN and NFV are mutually beneficial from each other, but they are different technologies and are not dependent on each other. NFV can be considered complementary to SDN. SDN is a technology that removes a network device's "brain" to facilitate network configuration/administration, while NFV is mainly for IT networking.

Table 5
Selection of reference architectures mapped to beyond cloud computing models.

Ref. architecture	MCC	EC	FC	MEC
Dinh et al. [71]	✓			
ECC [72]		✓		
IIC [73]		✓		
Intel, SAP [74]		✓		
OpenNebula [75]		✓		
Sittón-Candanedo et al. [76]		✓		
Qinglin et al. [69]		✓	✓	
Willner et al. [70]		✓	✓	
Mahmud et al. [77]			✓	
OpenFog Consortium [78]			✓	
Pop et al. [79]			✓	
ETSI [80]				✓
This work	✓	✓	✓	✓

MCC: Mobile cloud computing; EC: Edge computing; FC: Fog computing; MEC: Multi-access edge computing.

3. The designed reference architectures

The various computation and communication models (discussed in Section 2.3) tackle different concerns of different users and leverage different parts of the computing continuum. However, these models share many commonalities, as described in Table 2. This allows us to design a computation- and communication-focused reference architecture for the computing continuum, combining the characteristics of already existing models beyond cloud computing. In this section, we present the motivation and design of these two reference architectures.

3.1. Computation reference architecture

For our computation reference architecture, we consider the following models, previously discussed in Section 2.3: Mobile cloud computing, edge computing, fog computing, and multi-access edge computing. Although these models aim to solve different problems and thus have various designs, they overlap significantly.

We synthesize a list of characteristics shared by these models and other models beyond cloud computing [62]: (i) All five models support computation and storage offloading [63] from one or multiple endpoints to one or multiple endpoint, edge, or cloud devices [64]; (ii) the offload sources support either local processing or data preprocessing to reduce the amount of data sent over the network [65], while (iii) the offload targets support the management of data streams from multiple endpoints in parallel [66]; (iv) the computing devices involved have varying resource constraints [67] and (v) use a variety of networking technologies such as Ethernet, Wi-Fi, and mobile broadband [68]; (vi) a centralized controller in the cloud is used to manage the entire continuum deployment.

This overlap allows us to construct a unified reference architecture for the beyond cloud computing models, and we present its design in Fig. 3. Reference architectures for these specific models already exist and are compared to ours in Table 5. Most existing architectures focus on only a single model, while only prior work in [69] and in [70] includes multiple computation models for their architecture, namely edge and fog computing. Our computation reference architecture design is based on our previous work in [62].

The reference architecture consists of three distinct tiers of systems: Endpoint, edge, and cloud. This split is made from a data processing viewpoint: Data is generated by users on endpoints and processed either locally or remotely by offloading to resource-constrained edge devices close to the user or large clouds located far away. Below, we discuss the characteristics of each tier and the specific components in our architecture that make up systems in a tier.

3.1.1. Endpoint

Endpoint devices are typically operated by a single tenant, generate data via interactions with users or sensors, and are positioned at the far end of the network. Examples of endpoints are smartphones and IoT sensors. Endpoints are made up of the following four components in our reference architecture:

P1 Data Preprocessing: Some endpoint devices possess built-in data preprocessing capabilities to reduce the amount of data that needs processing. Video cameras, for example, can possess face detection capabilities [65].

P2 Application: User-defined logic that defines how to process or offload data generated on the endpoint. It can decide whether offloading to edge or cloud is viable, and if extra preprocessing is required, or if local processing is desirable, depending on application-level metrics.

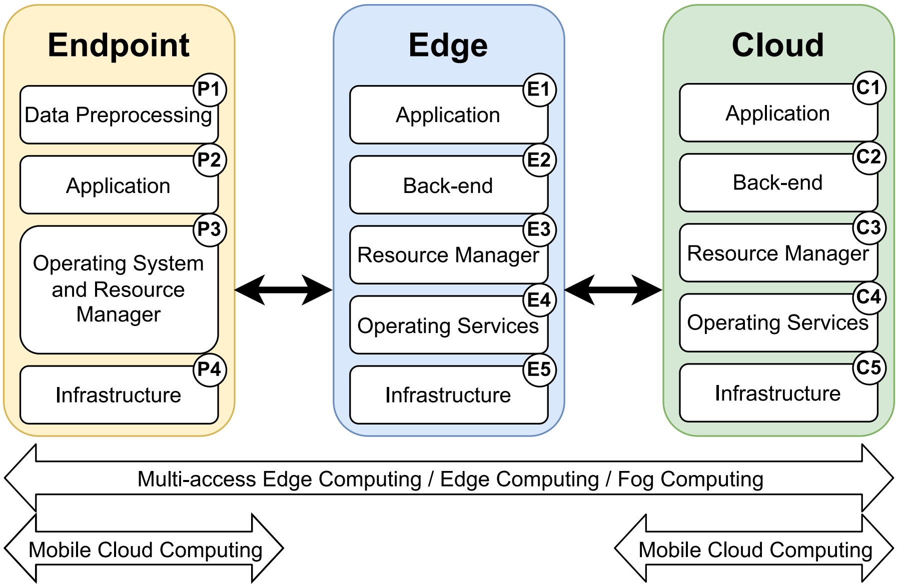


Fig. 3. The proposed computation reference architecture.

P3 Operating System and Resource Manager: The device management layer acts as an interface between the applications and the endpoint hardware. Endpoint operating systems, such as TinyOS and Android [81], are often optimized to fit the resource-limited endpoint devices and can support special endpoint hardware.

P4 Infrastructure: This includes all physical and virtual resources present on an endpoint device, from CPU and memory to virtual machines and containers. These resources can be directly interacted with by users, unlike the cloud, which a service provider operates.

3.1.2. Edge

The edge allows data to be processed in the field, close to the user, to meet applications' low-latency and privacy demands, which cannot be satisfied when offloading to centrally located clouds. We have provided a detailed analysis of what constitutes the edge in Sections 2.2 and 2.3. Most importantly, edge and cloud systems are more distributed than endpoint devices and possibly multi-tenant. To enable applications to use such a distributed environment, edge and cloud systems have an operating services component not present in endpoints. Overall, edge systems consist of the following components:

E1 Application: Architects of user applications on edge systems have to decide whether data offloaded by endpoints should be processed at the edge or should be forwarded to the cloud [64].

E2 Back-end: Provides support to applications that support resource-constrained devices. Back-ends often focus on applications from a single domain (*e.g.*, TensorFlow Lite for machine learning).

E3 Resource Manager: Manages the devices' physical and virtual resources and distributes these resources over edge applications from one or multiple tenants. Using a resource manager often dictates which computing models can be leveraged in the continuum.

E4 Operating Services: Provides support for applications running on the highly distributed, heterogeneous, and complex computing continuum. Systems can provide operating services for communication [82], metadata management [83], consensus [84], and more.

E5 Infrastructure: As with endpoints, this includes all physical and virtual resources present on the device. Unlike endpoints, however, edge systems can be managed by service providers instead of users. This can result in a system similar to cloud computing, where users exclusively use virtual resources while service providers manage the underlying physical resources.

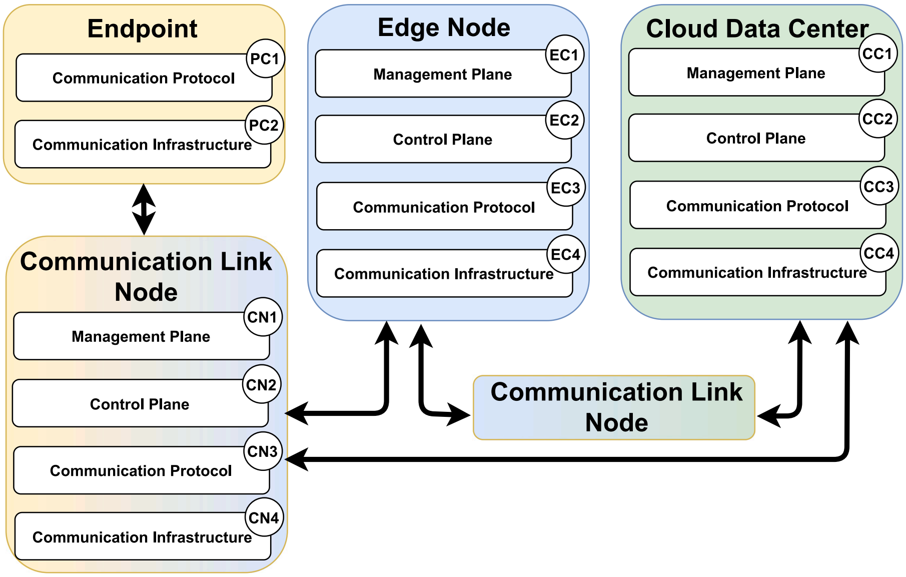

Endpoint
PC1 Communication Protocol
PC2 Communication Infrastructure
Edge Node
EC1 Management Plane
EC2 Control Plane
EC3 Communication Protocol
EC4 Communication Infrastructure
Cloud Data Center
CC1 Management Plane
CC2 Control Plane
CC3 Communication Protocol
CC4 Communication Infrastructure
Communication Link Node
CN1 Management Plane
CN2 Control Plane
CN3 Communication Protocol
CN4 Communication Infrastructure
Communication Link Node

Fig. 4. The proposed communication reference architecture.

Table 6
Selection of communication reference architectures.

Ref. arch.	Targeted env.	Scope
ISO/IEC [85]	IoT	Defining conceptual model for different components across IoT network links.
IEC [86]	IoT–Edge	Providing interoperability standards for the information exchange within and to power systems.
ETSI [87]	IoT–Edge	Specifying the network architecture of communications in transportation systems.
Intel [88]	IoT–Edge–Cloud	Accelerating the products, requirements, and feedback of Intel's partners.
This work	IoT–Edge–Cloud	Mapping different communication-related technologies in the computing continuum.

3.1.3. Cloud

The cloud is a parallel and distributed system consisting of virtualized nodes (possibly connected) owned by cloud providers and can be provisioned on demand to different consumers.

We have provided a detailed overview of cloud and cloud computing definitions in Section 2.2, and its differences to edge in Section 2.3. Although many differences exist between cloud and edge, their building blocks (**E1, E2, E3, E4, E5**) and (**C1, C2, C3, C4, C5**) are similar: Both provide physical and virtual resources to users, can leverage operating services and resource managers to abstract away the complexity of their distributed environment, and support applications with related back-ends for users to run their logic in. As such, the definitions for these individual components provided in the previous edge section also apply to the cloud, although with different example systems for each component for cloud compared to edge.

3.2. Communication reference architecture

This section proposes and discusses a communication reference architecture using a similar high-level abstraction approach as the computation reference architecture.

There are a few existing reference architectures for communication and networking. However, they focus on only a single layer or are designed for a specific scope or purpose. Our proposed communication reference architecture provides a recommended and technology-independent structure for all IoT applications. It maps different communication-related technologies in the IoT–edge–cloud continuum. In Table 6, we compared them with our architecture. Fig. 4 depicts the communication reference architecture, and in this section, we further describe every architecture component. We do not cover security since it is a vast topic on its own that encompasses all elements of the reference architecture, from the physical isolation of the network infrastructure to the end-to-end application-provided security measures.

3.2.1. Endpoint

An endpoint communicates with other endpoints, the edge, or the cloud. The communication demands vary significantly between domains and applications, from strict real-time constraints with short deadlines to more relaxed timing requirements. For example, saving power might be more important to some applications than having deterministic communication. An endpoint consists of two main communication components described below and identified in the reference architecture.

PC1 Communication Protocol: It provides communication means to the endpoint application and is an abstraction towards the underlying infrastructure. Different endpoints have different requirements, which reflect a suitable communication protocol. Table 3 provides an overview of common lower layer technologies (link layer), while Table 4 provides an overview of higher layer network technologies, *i.e.*, application layer protocols. Note that this endpoint component covers all network layers above the physical layer.

PC2 Communication Infrastructure: It provides the physical means to exchange data, known as Layer 1 in the OSI model.

3.2.2. Communication link node

A communication link node is part of the infrastructure, providing connectivity in the edge continuum, such as an Ethernet switch, router, Bluetooth extender, 5G base station, or virtualized communication nodes, such as VMs or containers. In the case of a point-to-point connection between devices, the communication link node can be as simple as a wire or the ether. The communication link node connects devices within and across layers in the edge continuum. One example of a communication link node is a router that provides endpoint interconnectivity and an Internet connection. Another example is a Bluetooth extender, which provides longer-range connectivity between two devices. Following, we describe the main components of a communication link node.

CN1 Management Plane: It is located on the network nodes in the case of traditional network management and handles the management of the device. On the other hand, communication link nodes capable of using SDN allow consolidation of the management plane centrally; in such case, CN1 is a thinner server, communicating with the central management plane (EC1, CC1).

CN2 Control Plane: It mainly controls the packet forwarding in the network. The forwarding paths are extracted by interpreting packets and their address information. The paths can also be configured in the control planes. In the case of traditional networking, i.e., without SDN technology, the control plane is located on the device. However, an SDN-capable communication link node allows remote and indirect control plane centralization (EC2, CC2). An SDN-capable communication link node gets the forwarding decisions from the central (remote) control plane.

CN3 Communication Protocol: This component provides a way to understand forwarding and bridging between the devices. For instance, a bridge, such as a WLAN Router, provides a translation between the network outside the LAN and the inside network infrastructure.

CN4 Communication Infrastructure: This component handles the packet forwarding, incorporating the data plane, and the physical media handling. A communication link node could support different infrastructures, including Ethernet or wireless connections.

3.2.3. Edge node

As in the computational reference architecture, the edge and cloud share the same high-level network components. The edge node can generally be seen as a node that can send and receive traffic in the network.

EC1 Management Plane: It is located inside the node in traditional networking, while it can reside in the controllers in case of using SDN technology. The main purpose, however, is to manage and configure traffic forwarding, routing, and management.

EC2 Control Plane: It provides a holistic control view of the network. The control plane typically uses protocols like OpenFlow to update and deliver forwarding decisions to SDN-capable communication nodes.

EC3 Communication Protocol: It provides edge node applications with communication to other edge nodes, endpoints, and cloud data centers.

EC4 Communication Infrastructure: It provides physical communication means like Ethernet or wireless.

3.2.4. Cloud data center

From the communication components perspective, the cloud data center can be seen as an edge node with higher computation capacity. Usually, the cloud data center is remote, meaning the communication infrastructure follows the Internet communication infrastructure. In this sense, the main components and their responsibilities are the same as in the edge node.

4. Proposed use cases to validate the reference architectures

This section discusses several recently designed and diffused use cases from different sectors that employ cloud–edge computing, including science, industrial, health, consumer, governance, and municipality use cases. We designed these use cases and then used them to validate the proposed reference architectures by highlighting specific components of the architectures and relating them to the use cases, focusing on mapping the components to the computing reference architecture. To validate the proposed computing reference architecture, we performed a numerical experimental proof-of-concept of the architecture, tested it, and evaluated its performance in [62].

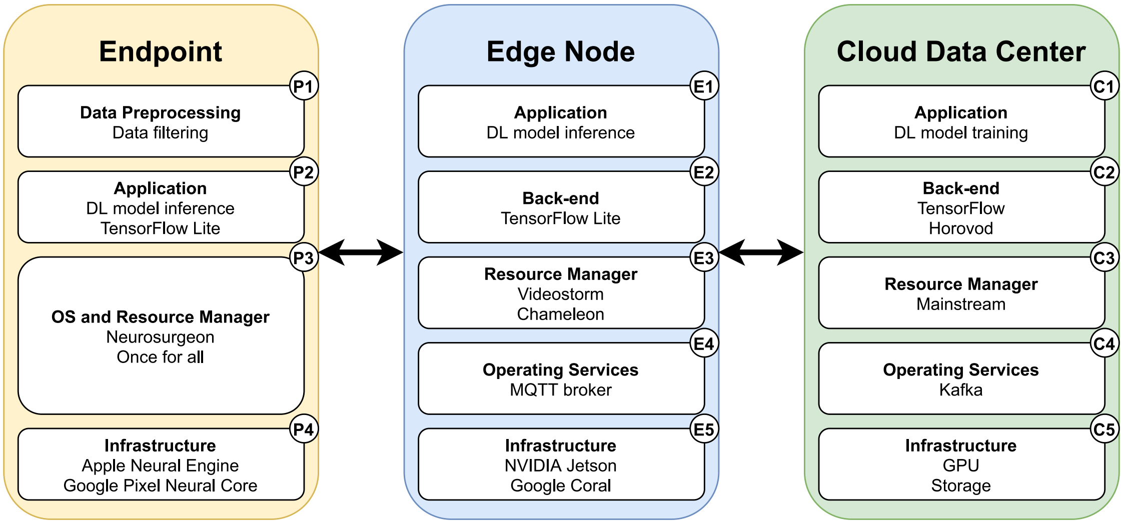


Fig. 5. Deep learning reference architecture.

4.1. Science use case

Machine learning (ML) applications use algorithms to process data, learn from it, and make predictions based on what has been learned. This includes Deep Learning (DL), a subset of Machine Learning (ML), where deep neural networks are trained on large amounts of data. DL can be used for a variety of purposes: recommendation systems [89], voice and video analysis [90], and data processing in general [91]. These applications are traditionally deployed in the cloud using cloud computing because of the high complexity of DL algorithms. However, this has three limitations [92,93]: (i) bandwidth: The amount of data generated by sensors that need to be processed by DL applications can quickly exceed available bandwidth to the cloud; (ii) latency: DL-based services for users at endpoints such as cognitive assistance [94] require very low latency, which the cloud cannot always offer because of high communication latency; (iii) privacy: The data processed by DL applications may contain sensitive information and may not be allowed to be moved to the cloud. To mitigate these limitations, DL applications can be offloaded from the cloud to (on-premise) edge nodes and endpoints.

In this section, we look at edge intelligence [95], how to facilitate DL in the edge continuum. We map state-of-the-art frameworks and systems to our reference architecture (Fig. 3).

4.1.1. Endpoint

Endpoint devices are resource-constrained by nature: They have too little computing and memory capacity to support cloud-native DL applications [96,97]. To enable DL at the endpoint level, recently many endpoint devices have been integrated with special DL hardware such as Apple's Neural Engine and Google Pixel's Neural Core for smartphones (P4 in Fig. 5) [98].

These limited resources make DL training on endpoints very difficult due to its use of large models and training data and is more often used for model inference and preprocessing (P2, E2). To enable model inference on endpoint hardware, specialized software libraries such as TensorFlow Lite [99] have been created with limited functionality to make them as lightweight as possible. Apart from software libraries, many cloud-native deep neural networks such as ResNet [100] also do not fit into the limited memory available at the edge. Neural networks either need to be compressed [101,102], or special networks for mobile devices such as MobileNet [103] need to be deployed. Despite these models being much simpler than their full-sized variants, they can achieve comparable accuracy [104].

Endpoint can also be used for preprocessing their generated data (P1). Preprocessing can reduce the amount of data sent to the edge and the amount of compute tasks on edge nodes, increasing efficiency and reducing energy usage. The work in [65] shows that, for object detection applications, it is possible to decrease end-to-end latency by 57 percent while achieving the same accuracy by filtering out blurry and similar camera frames.

These techniques and optimizations show that the location at which DL is done not only dictates the end-to-end latency, which itself is a trade-off between low-latency communication and resource-constrained computation [105] but also what DL algorithms and models can be used.

4.1.2. Cloud

Deep learning applications often require the use of large-scale models. To train these models, large amounts of data are required as well. While it is possible to shrink model sizes to allow model inference on edge and endpoint devices, reducing the training data size is much more complex. Model training is also a more compute-intensive task, requiring more powerful hardware. This makes the cloud perfect for deep learning due to its high computing, memory, and storage capacity. It can also leverage specialized accelerators for deep learning such as GPUs and TPUs [106] (C5 in Fig. 5). To allow cloud data centers to do both model inference and training (C1) leveraging powerful hardware such as accelerators, comprehensive software libraries such as TensorFlow [107] are used (C2).

4.1.3. Edge

The hardware resources of edge nodes are between the endpoint and cloud layers, using devices integrated with special DL hardware such as NVIDIA Jetson [108] and Google Coral [109] (E5). Edge nodes can be used for model inference similarly to endpoint and for model training, although they are limited compared to the cloud. However, unlike endpoints, edge nodes do not generate data themselves and rely on endpoints or cloud nodes to send them data for processing. This introduces a need for resource management on edge nodes to handle requests, possibly from multiple devices, as efficiently as possible on limited resources.

Compared to general resource managers often seen in the cloud domain such as Kubernetes [110], specialized resource managers for DL can leverage domain-specific knowledge to optimize application performance beyond when and where to schedule workload (P3, E3, C3). For example, VideoStorm [111] and Chameleon [112] propose a trade-off between latency, throughput, and model accuracy by using different (compressed) versions of the same model [113]. When an edge node is busy, applications can be forced to use smaller, less accurate models, which results in lower latency and higher throughput, while larger models can be used in times of low activity. On the other hand, Mainstream [114] proposes work sharing between concurrent transfer learning applications. With transfer learning, pre-trained DNNs are retrained to fit specific applications. The common computation can be shared if multiple applications use the same pre-trained DNN, reducing the overall system workload.

4.1.4. Hybrid optimization

Model inference does not have to be limited to a single device only. Each device in the edge continuum has its characteristics in terms of resource constraints and communication latency, and only by leveraging multiple devices can optimal performance be achieved [115]. Here, we give two examples. The authors in [116] propose*Once for all*, a deep learning network architecture that requires a model to be trained once, after which it can be adapted to fit a specific device's resources by shrinking the original model. This allows different versions of the same model to fit on endpoint and edge devices, simplifying the deployment process. The authors in [64] propose a framework for model partitioning between mobile devices and data centers. They can automatically partition DNNs and achieve an average end-to-end latency improvement of 3.1 times and an average energy usage reduction on mobile devices of 59.5 percent, showing the effectiveness of such techniques.

4.2. Industrial use case

Industrial IoT (IIoT) promises automation and real-time monitoring for smart manufacturing by leveraging the computing continuum. In general, IIoT applications are directly related to the infrastructure they run on, such as a manufacturing plant. This creates strict constraints on privacy, latency, and redundancy, among other things. The modern and competitive production environments led industrial and manufacturing companies to consider new solutions to increase revenue. Industry 4.0 relies on heterogeneous physical devices, "things", such as sensors and actuators in the production process. These Industrial IoT (IIoT) devices are integrated to simplify automation, improve communication, and increase production using smart machines without human intervention [117,118]. IIoT devices allow the collection of a large amount of data from the controlled process, which needs to be processed, analyzed, and stored. Naturally, this cannot be done locally in resource-limited embedded devices that collect the data. Instead, the data must be transferred to more powerful computing nodes at the edge or in the cloud. The choice of where to perform such operations depends on several aspects, including the geo-distribution of the data, the data rate, security and privacy issues, the required durability of the collected data, or simply the need for computation capabilities.

Computing models located in different layers, apart from the manufacturing plant, could be the key enabler in dealing with such challenges and solving them for the IIoT applications.

In this section, we investigate how Industrial IoT is currently realized, what infrastructure it is deployed on, and what can be done to improve the field. We map these findings onto our reference architecture in Fig. 3.

4.2.1. Endpoint

Endpoints of the IIoT can be used for data collection, data exchange, or control purposes [119]. Endpoints in industry and manufacturing could be resource-constrained smart sensors and actuators (P1, P4) that collect real-time data on the controlled processes' operations. These endpoints connected (PC1, PC2, CN3, CN4) to Programmable Logic Controllers (PLCs) are control systems for local control, but they do not support advanced processing (P2). These devices are typically connected through a backplane or using a field bus (PC1, PC2), *e.g.*, Profibus and DeviceNet, or wireless, *e.g.*, WirelessHART or 5G (P4), allowing for data communication and exchange among devices and towards the edge continuum. It is common for endpoint applications to run on Real-Time Operating Systems (RTOS) (P3), *e.g.*, VxWorks, ThreadX (Microsoft Azure), PikeOS, or FreeRTOS, that enable real-time operations and guarantees of the underlying industrial processes.

4.2.2. Cloud

On the cloud data center side, both general-purpose and industrial cloud platforms are utilized [117]. For example [117], AWS, IBM Cloud, Microsoft Azure, and SAP Cloud Platforms are among the leading solutions that are typically adopted, while Siemens MindSphere, Bosch IoT Suite, and PTC Cloud platforms are solutions tailored for industrial uses (C2). Depending on the applications, general-purpose platforms provide more flexible and integrated solutions. For example [117], SAP Cloud Platform can provide seamless integration between IoT applications, data analytics, and SAP Enterprise Resource Planning (ERP) system (C1). On the other hand, industrial cloud platforms allow for complete control over the computing resources and security while explicitly accounting for non-functional requirements of IIoT applications, such as throughput and latency. Data transferred (PC1, PC2, CN3, CN4, CC3, CC4) to the cloud layer can be processed or stored for online or offline analysis. Clouds offer much more compute and storage capabilities than edge or endpoint devices, *e.g.*, in the form of VMs or containers (C5). These resources are created, monitored, and run by Hypervisors like Xen or ACRN (C3). With these resources, cloud computing is capable of performing advanced analytics for tremendous amounts of data. Such kind of analysis supports decision-making processes in the plant [120]. In IIoT solutions, fault tolerance guarantees and strong message durability are the main priorities. Services like Kafka, Apache Storm, or Apache Spark are, therefore, the key technologies for handling the data coming to the cloud (C4).

There are some limitations in using cloud computing for IIoT (as discussed in detail in Section 2.2); they are: how to maintain connectivity with cloud data centers? How to ensure enough bandwidth to transfer data between the physical location of the plant and the cloud data center? How can network bottlenecks and latency caused by transferring massive amounts of data be avoided? How to handle data security concerns that occur because of the centralized data processing and storage?

4.2.3. Edge

Edge computing supports meeting the timing requirements of the different IIoT applications (E1). As the edge layer (EC3, EC4) is closer to the data source (PC1, PC2), it offers massive connections and prompt response time. In addition, it can provide automatic operation by running virtualized PLC or similar, together with maintenance and advanced security mechanisms. [121]. The edge layer contains computing devices, called Edge Gateways (EGs), *e.g.*, the Nerve fog node by TTTech, that operate as gateways between the industrial plant and the cloud (E5). The function of EGs is to perform data analytics in real time [122]. These resources can be managed by hypervisors, similar to the cloud, which can be modified to be used efficiently in industrial applications on edge computing, for example, ACRN hypervisor [123] and PikeOS [124] (E3). The Open Process Automation Standard uses the concept of Advanced Computing Platform, O-PAS description of edge computing capable of hosting virtual Distributed Control Nodes (DCN) [125]. Devices on the edge computing network can process and store data of IIoT applications. The networking model includes tools to ensure reliable and secure exchange of data in the industrial automation space [121], for example, Object Linking and Embedding (OLE) for Open Platform Communications Unified Architecture (OPC UA) (E4), which O-OAS also proposes as the communication model for increased interoperability.

Deterministic and time-efficient data management is crucial in real-time applications. Thus, back-end solutions have been proposed to manage this issue. Google presents different types of flexible and scalable databases (*e.g.*, Cloud Firestore and Realtime Database [126]) that support real-time data syncing for different applications (E2).

Although edge computing offers many benefits, cloud computing is fundamental in serving IIoT applications, as resources at the edge still have limited capacity compared to cloud resources.

4.2.4. Hybrid optimization

With hybrid optimizations, the resources at the cloud and edge layers can complement each other to meet the requirements of IIoT applications. Recently, many works have investigated this problem, taking the distance of cloud data centers and edge devices to data sources into account [127].

The authors of [128] present a system called RACE to execute real-time applications in the Cloud–Edge layers. RACE includes a cost-based optimizer that places the application on the cloud or edge based on the processing power of the resources. In [129], the authors describe the QoS-aware service allocation for the so-called Combined Fog–Cloud (CFC) architecture as an optimization problem to minimize the latency. Another work [130] introduces a layered Fog-to-Cloud (F2C) architecture suitable for executing IIoT applications.

These studies show that each layer has properties that support distinct roles or functions: the cloud layer supports high-performance data processing, while the edge (or fog) offers less network communication cost and faster response time. Thus, IIoT applications take advantage of both layers.

4.3. Health use case

In pandemic situations, the need for telehealthcare service becomes dramatically fundamental to reduce the movement of patients, thence reducing the risk of infection [131]. Telemonitoring is a quite recent topic. It allows doctors to evaluate a patient's health status and, in some cases, make decisions on treatments remotely. In healthcare, end-users include personnel and practitioners of healthcare centers and clinicians. Often, these persons are not very confident with information and communication technologies. Thus, they need easy-to-use software solutions and visualization tools to understand the patient's health status clearly. This is a very important task, especially in the telemedicine application scenario, where medical operators cannot directly contact their patients but rely on video communications and/or remote IoT devices for measuring bio-physical parameters. In telemedicine, an operator is responsible for evaluating the effective need for therapy or other health-related issues only by considering interviews and data collected at the patient's home. Recent breakthroughs in the Information and Communication Technology (ICT) field, such as Cloud Computing and IoT, can be adopted to achieve such a goal. However, transferring and processing sensitive personal data raise several security and privacy issues [132,133].

4.3.1. Endpoint

One or more diseases can affect patients (*i.e.*, COVID-19, diabetes, heart disease, etc.). For each of these pathologies, different vital parameters have to be monitored. Moreover, patients could also be bedridden, increasing the difficulty of the monitoring. For this reason, specific systems have to be used [134]. This crowded scenario requires different specific sensors for each parameter to monitor (P1). For instance, considering COVID-19 one of the most important parameters is the blood oxygenation measured using oximeters (such as finger pulse oximeters, smart bands, bracelets, or smartwatches); diabetes disease requires the constant monitoring of glucose done by using blood glucose monitoring systems (P2); heart disease instead is relatively more complex to monitor because more than a single parameter (such as heart rate, and blood pressure) have to be monitored [135]. These parameters can be monitored through sensors like health rate belts or integrated systems such as sphygmomanometers and smart watches (P4). These appliances, used to transmit collected data to the cloud data centers, usually implement proprietary communication protocols based on GATT (Generic ATTribute Profile) (P3) [136]. In parallel to the devices mentioned above, in recent years, micro-controller boards like ESP32 and Arduino (P4) have been used as medical devices [137]. The advantage of using these devices is their flexibility. That is, by using a general-purpose microcontroller, developers can implement their communication protocol, preprocess data on these devices, and transmit data to custom platforms [138]. The side effect of using general-purpose micro-controllers is the calibration process because they require costly devices.

4.3.2. Cloud

In 2017, "wanna cry", the first implementation of ransomware [139], prompted the affected hospital's IT managers to migrate their services to the cloud to reduce the risk of cyber attacks. Many telehealthcare platforms (C2) have been developed to allow data monitoring and sharing [140,141]. These platforms allow medical personnel to continuously check the health status of patients from remote locations and help medical doctors make the right decision about care [142]. In recent years, with the advent of blockchain, telemonitoring platforms are adding new features such as non-repudiation and trustworthiness [142–144]. Cloud computing-based healthcare systems mainly have two issues to be tackled, which could be solved in collaboration with the edge layer:

1. in case of cyber attacks on the cloud provider, data stored on the cloud could be publicly distributed [145] violating the patients' privacy;
2. disconnections of the hospital from the Internet make any or all services unavailable.

4.3.3. Edge

Edge computing refers to the enabling technologies allowing computation to be performed at the network's edge, reducing both latency and bandwidth for the computation and increasing the security of data [146]. In healthcare scenarios, edge computing devices (E5) are crucial because they allow the use of software and applications (E1) even when the hospital is disconnected from the Internet [147]. Edge devices, due to their limited computational capabilities, are not able to execute complex machine learning algorithms [148]. For this reason, in the last years within the scientific community, several lightweight machine learning algorithms (E2) for Edge devices have been proposed [149–151].

4.3.4. Hybrid optimization

Both Edge and Cloud solutions have benefits and drawbacks. Cloud-based solutions can cause privacy and service availability issues, while edge-based solutions cannot perform complex tasks on data. By combining both cloud and edge solutions, we can: i) reduce both privacy and security issues due to the storage of sensitive data on the Edge layer [152]; ii) analyze data in close to real-time by performing the computation on the Edge [153]; iii) allow the hospital to work even without an Internet connection [154]; iv) and perform complex computation and heavy machine learning algorithms on the Cloud [155].

4.4. Consumer use case

Cloud gaming supports gaming via streaming, that is, playing any game on any device without owning the physical hardware required to process it or needing a local copy of the game itself. However, the latency requirements are much stricter, the content is not buffered, and real-time input control processing is required. One common approach in cloud gaming networking is to perform the rendering and game logic on the cloud on remote CPUs/GPUs and decode the content locally. However, the main drawback of such an approach is the requirement for more bandwidth and processing time. In this respect, unstable and bandwidth-limited mobile networks might be unable to support real-time gaming video transmission, resulting in V-sync accumulating frames, skipping frames, or tearing. Therefore, communication technologies such as 5G and beyond are essential to deal with the problem at the RAN level. Unfortunately, even if frames are received at precisely 60 FPS with exactly 16.66667 ms between them, there will still be issues as the client screen's refresh rate will never match up perfectly to the rate the frame is received.

The need for ultra-low latency and mobility support can be tackled – in concept – by edge computing environments [156]. Such a proposal involves the utilization of local infrastructures as edge nodes and support streaming games within the same access network as the end devices directly from there. The advantage of this use case is that Windows games will be playable on platforms with limited access to AAA games (Linux, Android, IOS...). As part of this concept, key modules of the streaming solution need to be adapted to exploit the characteristics of the edge infrastructure and executed on edge nodes. A standalone game Edge module can be developed exploiting this framework. Low-end hardware can be used for high-end games using that module. To isolate and make the game execution independent, rendering, encoding, and Edge modules, as well as the management of the multiple users inputs, the solution can resort to sandboxing techniques to enable "*n*" users to play the same or different games each in their environments.

For the use case implementation, the following technological developments are necessary:

Table 7
A set of metrics for the game application.

Metric	Value
User Control Input (latency between edge and client)	≤ 30 ms
Edge Node Encoding Process Rate	= 60 fps
Streaming from Edge to Client	>8 Mbps
Jitter between Edge and Client	≤ 200 ms

Table 8
Latency vs. user experience vs game type.

Poor latency or lack of throughout symptoms	User experience	Game types
Lag	Gameplay stutters or does not respond to user input realistically	MMO, FPS, Any Real-time play
Time Jumps	Misaligning Client predictions with server state causes jumps and sudden changes in client game state	Shooters and other real-time play
Slow Loading	Delays in initiating sessions or moving to a new map while data is being downloaded	MMORPG, turn-based gaming, immersive graphical experiences
Failed Downloads	Download of a new game or new module to an existing game fails	Any game with a client
Buffering Video	Game viewing buffers or Video fails to start during event or replay	eSports (all online game types)

- Edge module: A server-side SDK developed as a set of functions that performs the following tasks: launching a game in a sandboxed environment, receiving input from the user, and sending back video and audio streams corresponding to the execution of the selected game. For the SDK to work, the path of the game to launch (previously installed in the server) and the IP and port where the client-side application is waiting for the connection will be required. The Edge module SDK will consist of two modules: a Sandbox system for the execution of the games (rendering) and a Game Session manager that will perform the encoding process of the video and audio output from the sandboxed game and the management of the userś inputs and sending them to the sandbox. This module also performs streaming.
- Client module: End-user software that captures the user input and receives and decodes the streams generated by the Edge module.

4.4.1. Endpoint

The Client application (P2) is the end-user software that connects remotely any device (P4) with the edge layer. The client is responsible for displaying the output (decoding of the streams) received from the Edge module (the content of the applications), capturing the user's "inputs" (keyboard, mouse, gamepad, controllers), and sending them to the Edge layer.

4.4.2. Cloud

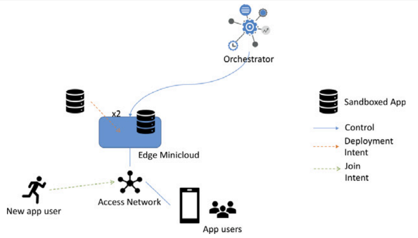
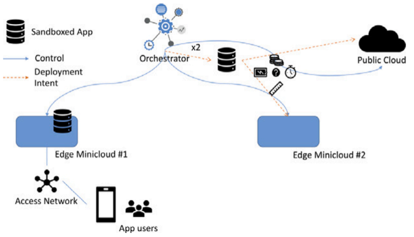
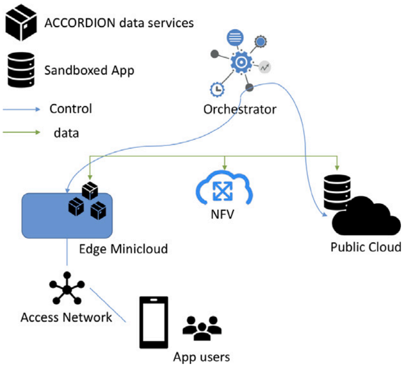
Cloud resources are not relevant in this use case. The edge layer can migrate to the cloud, as a last resort, *i.e.*, compromising the Quality of Experience instead of completely stopping the service. In this case, the main functionality of the central Cloud layer is to select appropriate edge locations and deploy the necessary components (C3).

4.4.3. Edge

The SDK (E2) at the edge layer will be integrated into the infrastructure's management framework (E3), allowing different instances of the edge module to be launched, each with a game (E1) in different virtual machines (E5) and different locations. Each user will be represented as a process (E1) in the active session on the virtual machine, and an intermediate software layer will be placed between the process and the operating system (E4) for virtualizing all the system resources and isolating the users from each other. Each session can have several processes opened simultaneously, although all of them share the system resources. Once the user connects, the game runs as a normal process on the active session, and the sandbox (E5) is executed between the process and the operating system, so all the communication between them is intercepted by the sandbox and can be modified without the application noticing it.

The SDK lying on the server side includes a set of functions that perform the following tasks: launching a game in a sandboxed environment, receiving input from the user, and sending back video and audio streams corresponding to the execution of the selected game. For the SDK to work, the path of the game to launch (previously installed in the server) and the IP and port where the client-side application (P2) is waiting for the connection are required. The SDK consists of two modules: a Sandbox system for executing the games (rendering) and a Game Session manager that will encode the video and audio output from the sandboxed game, manage the user's inputs, and send them to the sandbox. This module also performs streaming.

Table 9
Three scenarios manifesting potential implementations of the use case.

The first scenario depicts a situation whereby the orchestrator identifies an increase in the load at a certain edge mini-cloud and scales the sandboxed application to meet the demand. It is assumed that the sandboxed application components are programmed to form hierarchies and communicate with one another.	 Orchestrator x2 Edge Minicloud Sandboxed App Control Deployment Intent Join Intent New app user Access Network App users
The second scenario depicts a different situation: at some point, the orchestrator realizes that the current edge node can no longer support the demanded QoE. Therefore, it scales up, resorting to either the public cloud or another edge node depending on criteria such as faster deployment, lower latency, and lower costs.	 Sandboxed App Control Deployment Intent Orchestrator x2 Public Cloud Edge Minicloud #1 Edge Minicloud #2 Access Network App users
The third scenario implies using an application model that allows synergies between the application sandbox and a set of Virtualized Network Functions (VNF). The VNFs can intervene in the application flow and cache, encode, or apply data transformations to optimize the QoE. The orchestrator must be able to instrument the network and compute resources by querying the edge resources and the NFV slice manager to determine whether they can comply with the lightweight VNF requirements.	 ACCORDION data services Sandboxed App Control data Orchestrator NFV Edge Minicloud Public Cloud Access Network App users

4.4.4. *Hybrid optimization*

The cloud resources can be used to orchestrate the underlying Edge Modules, ensuring the deployment of cloud services (C3), like load-balancers and autoscalers. The cloud part can also monitor the overall QoS and migrate the Edge Module for service handover, proactively placing data and services in appropriate edge nodes. The use case can benefit from distributed orchestrators like [66,157,158].

In Table 7, we summarize a set of metrics for the game application as an example of the consumer use case, while in Table 8, we compare latency and user experience in different game types.

Table 9 provides scenarios that manifest potential use case implementations.

4.5. *Governance use case*

Smart governance has exposed a wide proliferation of compute-demanding applications that continually generate enormous amounts of data with a large variety, velocity, and volume [159]. With the increasing requirement for traffic safety, the advance of 5G, AI, and edge computing, completely new application areas are emerging to facilitate convergence of telecommunication (*e.g.,* 5G) and computational technologies (Edge, AI). An interesting use case of this is the intelligent traffic light that increases the security of participating parties, particularly vulnerable groups like pedestrians, bicycle drivers, or children. Injuries happen very often due to drivers' distractions or blind spots. With our use case, we address in particular yet unsolved problem that can be addressed in joint utilization of end devices, edge, and cloud technologies [160,161]. Fig. 6 illustrates our intelligent traffic light use case with all steps, involved components, and participants.

In Step 1, vulnerable road users (*e.g.,* pedestrians and cyclists) are captured in video frames collected by a deployed traffic camera (E1). Input video frames are analyzed online by the edge processing module in Step 2 (E2). Frames search for target users in the critical crosswalk or intersection area (*e.g.,* pedestrians).

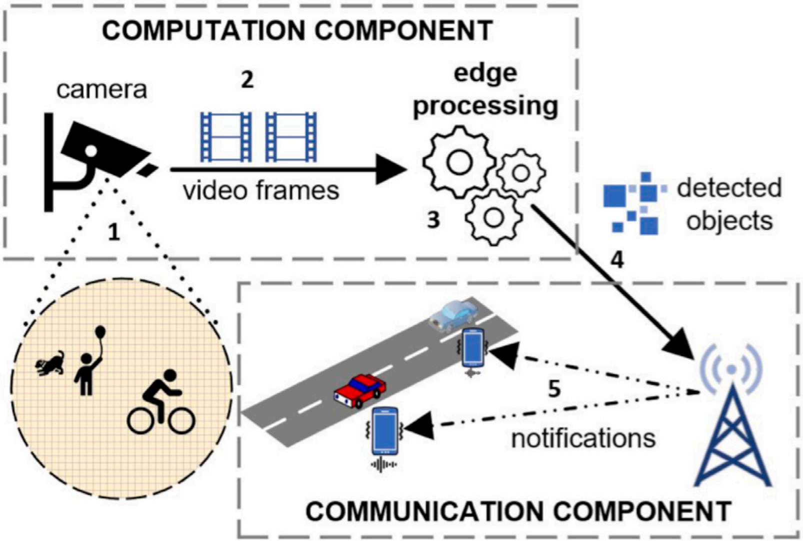


Fig. 6. Smart traffic light [160].

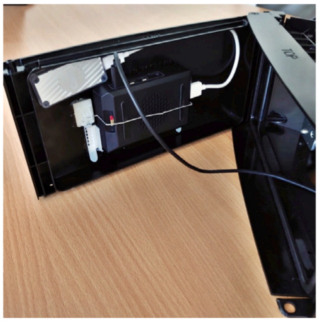
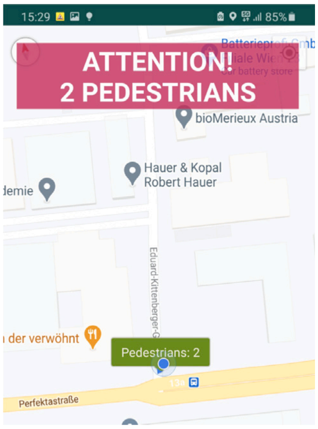


Fig. 7. Chamber of the smart traffic light with a Raspberry Pi [160].



ATTENTION!
2 PEDESTRIANS

Pedestrians: 2

Fig. 8. Object detection [160].

If the predefined objects (humans, pets, etc.) approaching this monitored area are detected (Step 3), a notification message is generated in Step 4 (E4). The message is then received by the end device (usually a smartphone) that has a pre-installed app installed, which shows visual and audio notifications (Step 5) (P1).

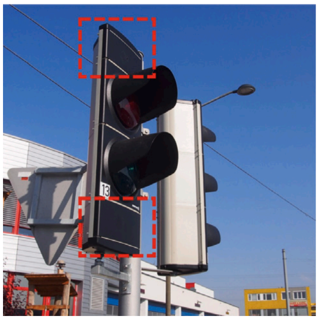


Fig. 9. Traffic light [160].

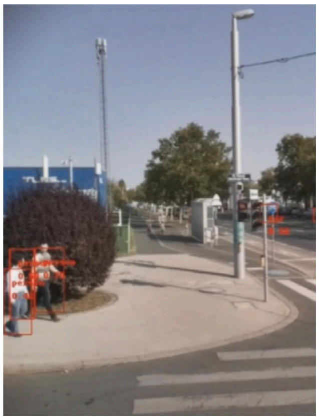


Fig. 10. Object detection [160].

4.5.1. Endpoint

As an endpoint, we use smartphones to visualize the alerts on Google Maps (P2). Fig. 8 depicts a sample alert that visually and acoustically alerts the drivers of the cars in case an object is detected in the blind spot. Since we are addressing critical situations, the alerts have to be visualized on the smartphone or linked dashboard display within specific time frames, and the communication protocol should have a minimal message overhead to keep the transfer time and latency low, offer guaranteed delivery of messages to users, and avoid unnecessary network flooding. Therefore, we decided on publish/subscribe (Pub/Sub) protocols that allow event-based notification and dynamic targeting of drivers close to a particular crossroad. In our scenario, accurately detecting pedestrians and cyclists requires aggregating sensor data at a single processing point. Therefore, a protocol designed for centralized processing is more desirable than a distributed protocol. Another desirable feature is the variety of different Quality of Service (QoS) that can be selected in demand, respecting different levels of provided latency. Once an object is detected, we notify all devices subscribed to a specific topic within a certain geographic area. Due to all restrictions and described demands, we selected MQTT [162] as the communication protocol (P3).

4.5.2. Edge

The Edge unit is necessary to process video frames in near-real-time. As shown in Fig. 7, we employ single-board Raspberry Pi (RPi) edge devices, to which we attach co-processors to improve the performance of neural network inference. Fig. 9 depicts concrete installation at a traffic light. As co-processors, we utilized Google's Coral Edge TPU accelerator and Intel's Neural Compute Stick 2 (NCS2) since both can be plugged in via USB and used for vision-based ML applications. Input video frames are constantly analyzed by the edge processing module in Step 2 (See Fig. 6), searching for target users in the critical crosswalk or intersection area. The software module on the Edge is developed using TensorFlow Lite, a version of the popular TensorFlow framework optimized for limited IoT devices, including RPi. We use pre-trained convolutional neural network (CNN) based object detection models, trained using the standard COCO dataset. Fig. 10 depicts detected objects.

4.5.3. Cloud

Cloud infrastructure is not shown in Fig. 9 but is necessary for long-term planning of traffic behavior in urban areas. Aggregated and anonymized data from the particular traffic light are collected and can be periodically sent to a city government to optimize traffic planning (C1). For example, it would be interesting to know how many people were in critical situations on a large scale.

For other decisions, it might be necessary to know at which time of the day/week/month most pedestrians are crossing the street to optimize routes of blue light organizations (*e.g.*, ambulance, police, fire, and search and rescue services). Aggregated data collected from various traffic lights can be used for diverse long-term strategic planning decisions.

4.5.4. Hybrid optimization

Hybrid optimization is necessary in case of staleness control [163]. Usually, we assume constant data distributions when we train our models. However, data distribution might change over time. A typical example would be when a traffic light changes environmental conditions due to the sun's path or the changed behavior of the traffic participants. In that case, models have to be retrained and redistributed over time. Another reason for hybrid optimization is the so-called cold start when a new traffic light is enrolled. To avoid cold starts, models from the repositories can be reused and re-adapted during the short time frame.

4.6. Municipality use case

Cloud and edge computing are appealing models that have the potential to overcome the data volume and communication latency-related issues [164]. However, the deployment of the municipality use case applications engenders new challenges. Therefore, we highlight in this section the role of cloud and edge computing in realizing the vision of such applications through a smart traffic management system focusing on parking occupancy, which is essential to properly manage the limited parking places in the city centers. The smart traffic use case is an example of traffic monitoring, guidance, and resource management systems [165]. Its primary focus is on applications with massive machine-type communications (*e.g.*, for sensor data) and enhanced mobile broadband (*e.g.*, for live footage), which are typical use scenarios for fog and edge computing. The parking occupancy in the smart traffic application is composed of the following core components:

1. Traffic sensors (cameras, inductive loop, or ultrasonic detectors), operated by the governance body, support real-time sensing and data transmission to a specially configured Edge device for initial processing.
2. Data aggregation and preprocessing is a virtual container instance that receives complete information from the sensors, such as frames from the camera or the number of detected vehicles by the inductive loop detector.
3. Parking occupation component uses the aggregated and preprocessed to determine the parking space's occupation status (i.e., free, occupied) represented by the defined geometry.
4. Licence plate recognition component is used to identify the vehicle's registration plate, which is later used to confirm if the vehicle is allowed to park in the given parking place.
5. Web component that informs the end-users about free parking spaces and provides further navigation information.

4.6.1. Endpoint

The endpoint is represented by traffic sensors that continuously gather information on the occupancy status of parking spaces. Considering the high price and range limitations of inductive loop or ultrasonic detectors, the most suitable approach for determining parking space occupancy is the utilization of traffic cameras (P1 and P2). However, different approaches are required at the end node to accommodate the requirements of the various sensors. When inductive loop or ultrasonic detectors are used, the data preprocessing is performed at the endpoint (P3 and P4). However, more computational resources are required when using traffic information from cameras, which are usually not available at the endpoint.

4.6.2. Cloud

The Cloud is utilized to visualize the available parking places and provide additional services, such as re-routing or driving navigation to the next available parking space or determining if the vehicle is lawfully parked. Multiple cloud services exist for visualization, including Amazon QuickSight and Microsoft Power Bi. However, Grafana, with the TrackMap plugin, can create a specifically tailored visualization service (C1). Related to driving navigation, services such as Google Maps Platform and OpenStreetMap API can be used in the Cloud (C2 and C4). Lastly, for more complex services, such as license plate recognition and confirmation of the lawfulness of the parked vehicle, specialized APIs and adapters have to be used, as many smart cities use different solutions for parking permissions.

4.6.3. Edge

The Edge is the essential layer that enables fast and reliable detection of available or occupied parking places over large geographical areas (E1, E2, and E3). Recent research work [166] shows that the Edge can perform image processing and machine learning inference efficiently. This allows for complex algorithms to be used with traffic cameras video streams, which, in combination with other types of traffic-specific sensors, can be used to identify available parking places across larger areas of the cities (E4 and E5). This system has higher added value, as license plate recognition can also be performed, which can be later used to compare with the centralized database in the Cloud.

Table 10
Summary of requirements for execution of the municipality use case components.

Requirement Functionality	Compute	Network	Energy
Data aggregation	Low (Endpoint)	<2 ms/>1 Gbps	Low
Parking occupation	Medium (Edge)	<12 ms/>1 Gbps	Low
Rerouting	High (Cloud)	<60 ms/>100 Mbps	Medium
Visualization	High (Cloud)	<100 ms/>100 Mbps	High

Table 11
Comparing the operations of different use cases across the layers of the computing continuum.

Use case	Endpoint	Edge	Cloud
Science	Data preprocessing	Small-scale model training	Large-scale model training
Industry	Data collection	Low-latency control	Offline analysis
Healthcare	Data collection	Data preprocessing	Offline analysis
Consumer	Visualization	Decision making and logic	Long-term planning
Governance	Alerts visualization	Data processing	Long-term analysis
Municipality	Data collection	Low-latency control	Visualization

4.6.4. Hybrid optimization

Smart traffic management systems can benefit from hybrid optimization. Although these applications are already highly distributed, they lack adaptability, as they require specific protocols for communication and highly specialized hardware resources. This implies that specific computing and communication infrastructure must be provisioned before the application can be deployed, thus hindering the possibility for utilization of public cloud provider and ad-hock resources. To solve this issue, proper resource provisioning and application scheduling systems have to be used, capable of managing resources that reside across different control domains, including public and private cloud and edge providers (C5).

In Table 10 we summarize the requirements of the municipality use case per functionality in terms of compute, network, and energy requirements.

4.7. Comparison

In Table 11, we summarize and highlight the various operations and functionalities of the designed use cases across distinct layers of the computing continuum.

The collaboration between the IoT, edge, and cloud layers results in an adaptive paradigm (the computing continuum) that is able to perform different operations to meet the requirements of diverse applications and use cases (usually expressed by metrics, *e.g.*, latency). Considering the system's current state and specific use case constraints, the layers are collaborating to offer an adaptive and cooperative system that considers predefined metrics. The computing continuum's adaptability is driven by the huge number of resources that are distributed across its layers, allowing the system to evolve different scenarios and run efficiently in various use cases.

5. The current related trends

Adopting cloud computing services has become indispensable for many institutions, enterprises, and manufacturers. Thus, new cloud-related trends in public cloud computing services have emerged to expand the range of services and enhance their quality. With such trends, the benefiting entities can simplify IT management, enhance their Return on Investment (ROI), and speed their paces toward digital transformation. Thus, new trends are expected to continue to emerge, to evolve with exciting new use cases. In this section, we picked some interesting cloud-related trends and discussed them.

5.1. Serverless computing

Serverless computing extends the classical cloud computing model by allowing developers to build and run applications without requiring server management or considering other operational aspects. Serverless computing allows developers to develop, deploy, and run their applications using cloud computing resources without the need to allocate and manage virtualized servers and resources. The responsibility of any operational aspects, such as fault tolerance or the elastic scaling of computing, storage, and communication resources to match varying application demands, is the responsibility of the cloud provider [167,168].

5.2. Federated cloud computing

Federated cloud computing refers to collaboration between different cloud providers. This collaboration allows for the sharing of computing power and other resources between cloud providers located in different geographical regions. This concept has many benefits, such as improved availability, mobility, cost-effectiveness, and power efficiency.

Federated cloud computing utilizes the concept of federated learning. Federated learning [169] (also known as collaborative learning) is the way to cooperate and train a single model by different agents or devices. The traditional way of training models raises many drawbacks and issues when the data size increases. Another important factor is data privacy. The centralized learning model requires all the participants to send their raw data to a central place to get trained. The process of sending raw data by several data generators (participants) to a central place to get trained triggers many questions, such as: What will happen if the number of data generators spikes to thousands or even millions? How to manage petabytes of data? How to protect the data?

However, the integration of cloud computing and federated learning comes with some challenges that can be summarized in the following points [170,171]: (1) Data Privacy: The main idea behind federated learning is how to protect the clients' data by training their data locally without a need to send the raw data to a central place, (2) Scalability: The current FL infrastructure assumes the existence of a central place that is responsible for averaging the clients' models and distributing the global model. The question is: What is the maximum limit for that central server? (3) Heterogeneity of clients and data: Clients might vary from mobile devices to hospital computers. If we go a bit wild, IoT micro-controllers can collect data using sensors and enough computation power to train the data locally. This diversity brings all the heterogeneity issues in distributed systems. (4) Clients Availability: a client might be selected to participate in updating the global model by uploading its local model to the central server. Some questions need answers concerning this point: Do we need to select the clients with high availability scores? And what metrics should be used to tell if a client is highly available in the federated learning world, and (5) Security — Intruders: It is possible for some clients with malicious intent to upload corrupted models to the central server, causing the global model to deviate from its optimal goal.

5.3. Cloud manufacturing

In the competitive and digitally transformed production environment, manufacturers face some challenges. The main two challenges are dealing with the large amount of data generated by the manufacturers' distributed resources and coordinating between manufacturers to share the usage of their data.

In the manufacturing industry, the production environment comprises the operations that support the production processes to create specific products. The production processes are typically carried out by distributed manufacturing resources set in different locations. Such industrial systems need to be shaped in a model that can manage all manufacturing services centrally. This model is called "Cloud manufacturing" (CMfg). CMfg expands the principles of cloud computing (Everything-as-a-Service, XaaS) to cover the manufacturing environment, resulting in new types of services that we can call Manufacturing-as-a-Service (MaaS).

The fundamental concept of CMfg is to encapsulate manufacturing resources, network capacity, and IoT features into services and then offer these services to different manufacturers upon request so they can improve their productions. In other words, with CMfg, the manufacturing resources are transformed into production services managed intelligently to keep the production at the top level [172]. CMfg aims at (1) transforming the traditional distributed manufacturing resources into manufacturing services and integrating them, and (2) Utilizing the cloud computing model to make such services available over the network.

Manufacturing companies can benefit greatly from adopting an XaaS model. By doing so, the companies eliminate the need for hardware maintenance and technical support from the IT team. Additionally, there is no requirement for software licensing and upgrade expenses, which can significantly reduce the company's outlay in the long term. This approach allows manufacturers to streamline their operations and optimize their budgets.

6. Future vision: On the development and evolution of the computing continuum

Computing has gone through several waves of evolution. This process does not show signs of stopping. This work presents several use cases and discusses different current computing trends. By investigating them from diverse perspectives, we showcase current and currently emerging applications and their complex requirements related to computing power, storage, and network. We also emphasized higher-level emerging needs, *e.g.*, related to sustainability and responsibility.

We envision the following:

1. The applications and use cases will evolve toward more sophisticated use with more complex requirements while supporting increasingly broader populations and organizations.
2. The computing continuum will continue to evolve radically at times to meet the requirements of novel services, applications, and emerging use cases.

The computing continuum currently provides the most promising approach among all computing paradigms and approaches to support such requirements. This is because the continuum promises to seamlessly use all the relevant resources and services across the entire infrastructure and all layers in the continuum, *e.g.*, end user layer (mobile or IoT devices), edge layer (edge/fog nodes), and cloud layer (data centers). This provides an excellent match with envisioned needs. When data is generated, it can be pre-processed or filtered at the end user layer or edge layer. Then, the process can be carried out further at the edge layer. The data can be

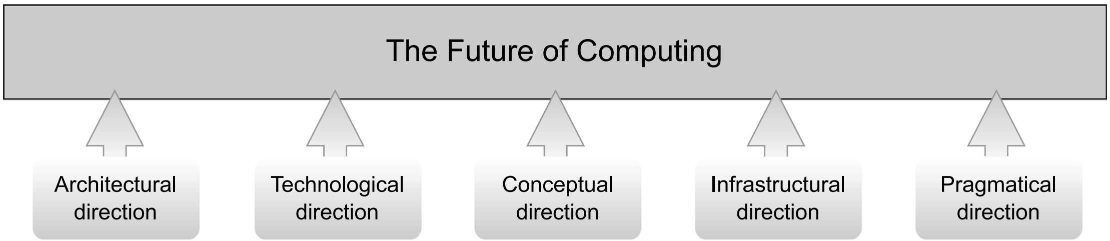


Fig. 11. The future of computing.

transferred to the cloud layer for complex processing and analysis. Connectivity, especially intra and between layers, is essential to the continuum. Therefore, communication and network technologies need to be utilized efficiently for better performance.

However, the computing continuum still faces some challenges related to Application design, Application placement decisions, Security and privacy concerns, Human-in-the-loop management systems, and Performance monitoring. Thus, more modern concerns require clear definitions and standardizations. Besides, different research directions on the computing continuum should be explored to meet the diverse requirements of the new applications and services. In this work, we discuss the following directions, based on the sophisticated functional and non-functional requirements of the wide range of applications and use cases expected to emerge (as summarized in Fig. 11).

6.1. Architectural direction

In this direction, we examine the potential for future research in the multi-layer architecture of the computing continuum. This architecture comprises a network of devices and systems that provide computing services to different applications. The direction suggests that by conducting further research on this architecture, it may be possible to enhance its functionality, leading to improved performance, efficiency, and user experience, and to develop a more robust and reliable computing architecture that can meet the growing demands of modern society.

- Improving geo-distributed data analytics approaches: Analyzing data on geographically distributed nodes (across endpoints, edge, and cloud data centers) is substantial these days and in the upcoming decade(s), especially after the emergence of Metaverse, second life, virtual worlds, and mirror worlds concepts. However, such analysis has multiple challenges, including providing low latency. Utilizing the integrated computing layers of the continuum when placing data is essential to overcome such limitations, considering the computing and storage limitations of the endpoints and edge nodes. A possible direction for improving data analytic approaches in the computing continuum is by reducing data analytics responses. This can be done by optimizing data placement via improving data locality and/or minimizing outliers in task execution. A critical issue to be considered here is communication and network congestion, especially in the endpoint and edge layers. Optimizing communication patterns can help deal with the probable network congestion [173].
- Re-thinking data management in decentralized ecosystems, across the entire continuum: In the continuum environment, distributed computing devices generate and process vast amounts of data at the edge of the network. Different users and applications may need to cooperate and share such data to achieve a common objective. Thus, new approaches are expected to manage data sharing efficiently in the layers of the continuum. There is much to investigate in abstractions, support for mobility and heterogeneity, failure models and managements, etc. [174].
- Expanding the layers of the computing continuum: The current computing continuum architecture may result in difficulties in migrating workloads between clouds. There is a need for more freedom in workload placement. This is to avoid stucking to a single service provider, and also to maintain the regulations of data storage and processing [175]. Adding another abstraction layer above the cloud can solve such problems. This layer can operate when cloud platforms are operating below it, so workloads and applications move and interface with different cloud platforms. With such architecture, named Sky computing or SuperCloud, different cloud providers can reconfigure their infrastructure on-the-fly to move services between their geo-distributed platforms.

6.2. Technological direction

This section discusses important points that we anticipate will lead to groundbreaking discoveries in the upcoming years. These points are related to the technologies used by the nodes host in the layers of the computing continuum, which will significantly improve the nodes' performance.

- Integrating advanced computer architecture into the continuum: In traditional computing architecture, known as von Neumann (vN) architecture, the processing unit and memory are physically separated but connected via shared buses. vN is simple to design and program, and most computers adopt it nowadays, but it comes with a drawback called the vN bottleneck.

This bottleneck results when transferring data between the processing unit and memory through the bandwidth-limited bus. Consequently, this causes performance degradation and energy waste. The novel Non-von Neumann (NvN) architecture is a promising solution to this problem [176]. Examples of the technologies in NvN computers:

  - Quantum computing is recognized by its ability to execute tasks quickly; thus, we think that quantum systems will act as future consolidated clouds. Thus, integrating quantum computing with the continuum will overcome many challenges, such as performance degradation and low latency. Thanks to quantum computing, Nanotechnology, and nanoscience may also be integrated, together with AI, into microscopic nodes [177].
  - Neuromorphic computing is a brain-inspired computing model. In neuromorphic computers, the processing unit and memory are controlled by an integrated environment composed of neurons and synapses. The applications running in such computers are defined by the structure of the neural network. Thanks to neuromorphic computing, some fundamental operational differences can be highlighted compared with the vN computers [178]: Collocated processing and memory, Highly parallel operations, Event-driven computation, Inherent scalability, and Stochasticity.

  Thus, we think computers based on NvN architecture can be injected into the layers of the computing continuum environment to support the other nodes.

- Considering optical computing for practical problems: Optical computing, also called photonic computing, is the technology where computing is performed entirely in the optical domain [179]. In general, computation is done in two stages: data transfer and data processing [180]. Optical computing promises to speed up the computation as data can be processed while it is transferred. However, it is in its infancy and needs development to be a mature paradigm.
- Adopting intelligent computing on a broader scale: With intelligent computing, computers will act as problem-solving and decision-making entities. Computers with brain-like intelligence need to be disclosed and adopted in the computing continuum layers. This can be done by developing novel theories for (1) the essential elements of human-like intelligence (silicon and carbon transistors) at the computers' macro-level and (2) the computation theory to generate uncertain results at the micro level instead of only deterministic results, as the creativity of intelligence is built on uncertainty [181].

6.3. Conceptual direction

The points in this section shape our view about some concepts that may contribute toward a more efficient computing continuum.

- Cognitive edge–cloud: To establish a self-managed and opportunistic collaboration between heterogeneous devices in the edge–cloud continuum, it looks promising to consider the cognitive devices in computing. Cognitive devices [182] are distributed in the edge–cloud and can make decisions autonomously using their restricted computation and storage resources based on information sensed from their environment. Cognitive edge–cloud computing can be built on several existing algorithms (*e.g.,* machine learning) and technologies (*e.g.,* sensing) to serve different use cases better. We expect cognitive devices to be smart and connect/disconnect to the system as required.
- Combining autonomic computing and AI/ML: Autonomic computing, also called self-adaptive systems, is a model that investigates the systems' ability to reach desirable behaviors by themselves, where behaviors could be self-configuration, self-optimization, self-protection, and self-healing [177]. We expect that intelligence-based autonomic computing will become a fundamental model to cope with the increasing scale of systems (such as digital twins and cyber–physical systems), as monitoring and administrating such systems is costly. Intelligence-based autonomic computing could be a promising solution to such challenges by self-adapting physical assets.
- Enabling application middleware and frameworks for continuum exploitation: Application middleware could be propped to easily support application logic to be written and distributed across the continuum. We think that dynamic discovery and configuration aspects should be supported to adapt to changing placement or run-time conditions. The ability to expand towards the continuum should be accompanied by relevant capabilities offered to applications to exploit these features directly. This exploitation should be performed abstractly, potentially combined with the self-* abilities to drive a seamless application evolution and adaptation in the computing continuum.
- Interactive apps: With the emergence and popularization of online gaming, augmented reality experiences for training and execution, and, more recently, digital twins, there is much room to improve and evolve the current generation of interactive applications. Incorporating field- (*e.g.*, from sensors) and user-data (*e.g.*, from gamers) will be a significant challenge, especially when such data need fast processing and immediate response to maintain the interactive conditions.
- Considering responsibility: The infrastructure of the computing continuum forms a predictable evolution but also a considerable open challenge. Although early ideas exist [183], much remains to be studied, designed, implemented, and deployed. Since the 2010s, societal factors such as responsibility, trust, and inclusion have become prominent concerns, especially for large-scale infrastructure serving many people and organizations.
- Considering society: The computing continuum needs to consider the human social dynamics in what we can call it *Social Computing*. Social computing refers to a new computing paradigm that addresses interdisciplinary applications, tools, and research topics, as well as the design and use of information and communication technologies that consider the social context. It can help analyze individual and organizational behavior [184]. The infrastructure in the continuum could be integrated with artificial social agents to generate social knowledge.
- Addressing sustainability: Sustainability is concerned with the (efficient) production and use of electricity, the emission of greenhouse gases, and other pollutants and factors that impact the climate.

6.4. Infrastructural direction

In addition to computing power, other infrastructure resources presented here are important for providing stable and efficient services in the continuum.

6.4.1. Communication

Novel communication frameworks are needed to connect the layers of the computing continuum and the heterogeneous devices within each layer so that multiple applications can be efficiently served. This section provides multiple points that could enhance communications in the computing continuum:

- Providing new communication capabilities: The emergence of a wide range of applications with high network demands will pose challenges that 5G technology may not handle. Future networks will play a crucial role in meeting the communication requirements of various applications and services. To fully understand such potential requirements, collaborating between the industry and research communities with a shared vision is indispensable to address the 6G era and even beyond [185]. What is expected from such a vision is to push beyond the technical limits of 5G and to come up with new capabilities that enable faster and more reliable data transfers.
- Exploring new technologies to revolutionize the Internet: All services offered by the computing continuum are done via the Internet. Thus, the Internet should be available everywhere to bring data directly into the edge and/or cloud. This requires a revolution in the internet infrastructures to change the ways of communication.
  A promising direction to achieve this goal is by, for example, utilizing Low Earth Orbit (LEO) or Geostationary Earth Orbit (GEO) satellites instead of using traditional communication media (*e.g.*, cables and radio). LEO satellites can be interconnected to provide global broadband that can be accessed directly anywhere on Earth. They aim to provide low latency and high bandwidth Internet access, while GEO satellites cover a specific geographic area on Earth.

6.4.2. Storage

Due to the big data generated from the IoT layer, storage becomes a fundamental resource in the computing continuum. There is a need for more efficient and fault-tolerant distributed storage systems that can provide fast access times, redundancy, and automatic backup features.

6.4.3. Power supply

Ensuring an uninterrupted power supply is necessary for providing stable services in the computing continuum. Different solutions could be adapted for providing reliable power solutions in the continuum, like the Green energy solutions that depend on renewable energy sources, *e.g.*, solar, and wind.

6.5. Pragmatical direction

In addition to the above four directions, a pragmatic direction also evolves from the vast resources and services already deployed and maintained as contemporary cloud, edge, and other infrastructure.

- A pragmatic evolution of current infrastructure: Linking to the already deployed infrastructure, especially services. A few very large players dominate the cloud services market, so their technological choices effectively are *De Facto* standards for many operational concerns. The market segmentation adopted by these tech giants, *e.g.*, serverless, pods, and larger resources at Amazon Web Services will not disappear soon and need to be considered in practical approaches for the foreseeable future.
- Similarly, various levels in the traditional stack tracing back to the 1980s and 1990s, have become De Facto standards and cannot be easily replaced. For example, operating systems, including the many variants of the popular Linux, impose interface and architectural limitations that will need to be considered for the next decade.

7. Conclusion

Recently, new use cases and applications have emerged with diverse and complex requirements. The cloud computing model offers on-demand provisioning of elastic computing resources to these applications to meet their requirements. Cloud computing has revolutionized the field of computing with its services, characteristics, and business model, but at the same time, it concurs with some challenges. New models have emerged, forming what is called a computing continuum, to solve such challenges and meet emerging application requirements and design goals. However, despite the emergence of new computing models, a number of challenges remain that need to be addressed and overcome in the future.

A better understanding of the computing continuum will help tackle and solve such challenges. Thus, this work presents a reference architecture for each edge–cloud computing model and the edge–cloud communication technologies. The reference architectures cover all models and technologies in the three distinct tiers of systems that form the reference architectures: endpoint, edge, and cloud. For more comprehension, we provide several real use cases from various sectors. These use cases help validate the architectures by highlighting specific components and demonstrating how they would be related and used in different applications and use cases. Finally, several potential research directions are envisioned from different perspectives to further enhance the functionality and performance of the computing continuum.

The comprehensive overview presented in this work is of great value to researchers and engineers who want to further investigate and advance the computing continuum. It is an essential resource for anyone looking to make meaningful contributions to the ongoing evolution of computing.

CRediT authorship contribution statement

Auday Al-Dulaimy: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Matthijs Jansen:** Writing – review & editing, Writing – original draft, Methodology. **Bjarne Johansson:** Writing – review & editing, Writing – original draft, Methodology. **Animesh Trivedi:** Writing – review & editing. **Alexandru Iosup:** Writing – review & editing, Conceptualization. **Mohammad Ashjaei:** Writing – review & editing, Writing – original draft. **Antonino Galletta:** Writing – review & editing, Writing – original draft. **Dragi Kimovski:** Writing – review & editing, Writing – original draft. **Radu Prodan:** Writing – review & editing. **Konstantinos Tserpes:** Writing – review & editing, Writing – original draft. **George Kousiouris:** Writing – review & editing, Writing – original draft. **Chris Giannakos:** Writing – review & editing, Writing – original draft. **Ivona Brandic:** Writing – review & editing, Writing – original draft. **Nawfal Ali:** Writing – original draft. **André B. Bondi:** Writing – review & editing, Writing – original draft. **Alessandro V. Papadopoulos:** Writing – review & editing, Writing – original draft, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

- This work is partially funded by the Swedish Knowledge Foundation (KKS) under the SACSys project, and by the Swedish Research Council (VR) under the PSI project.
- The work is partially supported by the Excellence in Production Research (XPRES), a government funded Strategic Research Area (SRA) within manufacturing engineering in Sweden.
- This work is partially supported by the Dutch Research Council (NWO) with grants OCENW.KLEIN.209 and GFP 6G FNS, and EU-funded projects MCSA-RISE Cloudstars and Horizon Graph-Massivizer.
- This work is partially supported by the Horizon Europe Graph-Massivizer project, grant agreement 101093202.
- This work is partially supported by the Italian Ministry of University and Research (MUR) under the GANDALF and STRUDEL projects.
- This research is funded in part by the Austrian Science Fund (FWF) through the following projects: Transprecise Edge Computing (Triton), Austrian Science Fund (FWF): 10.55776/P36870, Trustworthy and Sustainable Code Offloading (Themis), Austrian Science Fund (FWF): 10.55776/PAT1668223, and by the Austrian Research Promotion Agency (FFG) through the following project: Satellite-based Monitoring of Livestock in the Alpine Region (Virtual Shepherd), FFG Austrian Space Applications Programme ASAP 2022 #53079251.
- The authors would like to thank the SPEC RG Edge activity working group for their valuable contribution to this work.

References

[1] Henna Kokkonen, Lauri Lovén, Naser Hossein Motlagh, Juha Partala, Alfonso González-Gil, Ester Sola, Iñigo Angulo, Madhusanka Liyanage, Teemu Leppänen, Tri Nguyen, et al., Autonomy and intelligence in the computing continuum: Challenges, enablers, and future directions for orchestration, 2022, arXiv preprint arXiv:2205.01423.

[2] Michael Hogan, Fang Liu, Annie Sokol, Jin Tong, Nist cloud computing standards roadmap, NIST Special Publication 35 (2011) 6–11.

[3] Bhaskar Prasad Rimal, Eunmi Choi, Ian Lumb, A taxonomy and survey of cloud computing systems, in: 2009 Fifth International Joint Conference on INC, IMS and IDC, Ieee, 2009, pp. 44–51.

[4] Daniel Balouek-Thomert, Eduard Gibert Renart, Ali Reza Zamani, Anthony Simonet, Manish Parashar, Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows, Int. J. High Perform. Comput. Appl. 33 (6) (2019) 1159–1174.

[5] Yuezhi Zhou, Di Zhang, Naixue Xiong, Post-cloud computing paradigms: a survey and comparison, Tsinghua Sci. Technol. 22 (6) (2017) 714–732.

[6] Blesson Varghese, Rajkumar Buyya, Next generation cloud computing: New trends and research directions, Future Gener. Comput. Syst. 79 (2018) 849–861.

[7] Amir Taherkordi, Feroz Zahid, Yiannis Verginadis, Geir Horn, Future cloud systems design: challenges and research directions, IEEE Access 6 (2018) 74120–74150.

[8] Rajkumar Buyya, Satish Narayana Srirama, Giuliano Casale, Rodrigo Calheiros, Yogesh Simmhan, Blesson Varghese, Erol Gelenbe, Bahman Javadi, Luis Miguel Vaquero, Marco AS Netto, et al., A manifesto for future generation cloud computing: Research directions for the next decade, ACM Comput. Surv. (CSUR) 51 (5) (2018) 1–38.

[9] David Bermbach, Abhishek Chandra, Chandra Krintz, Aniruddha Gokhale, Aleksander Slominski, Lauritz Thamsen, Everton Cavalcante, Tian Guo, Ivona Brandic, Rich Wolski, On the future of cloud engineering, in: IEEE International Conference on Cloud Engineering, IC2E 2021, San Francisco, CA, USA, October 4-8, 2021, IEEE, 2021, pp. 264–275, http://dx.doi.org/10.1109/IC2E52221.2021.00044.

[10] Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, Tao Li, A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet, ACM Comput. Surv. 52 (6) (2019) 1–36.

[11] Mohammed Laroui, Boubakr Nour, Hassine Moungla, Moussa A. Cherif, Hossam Afifi, Mohsen Guizani, Edge and fog computing for IoT: A survey on current research activities & future directions, Comput. Commun. 180 (2021) 210–231.

[12] Kun Cao, Shiyan Hu, Yang Shi, Armando Walter Colombo, Stamatis Karnouskos, Xin Li, A survey on edge and edge-cloud computing assisted cyber-physical systems, IEEE Trans. Ind. Inform. 17 (11) (2021) 7806–7819.

[13] Abhishek Hazra, Pradeep Rana, Mainak Adhikari, Tarachand Amgoth, Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges, Comp. Sci. Rev. 48 (2023) 100549.

[14] CISCO, Types of data centers, 2022, https://www.cisco.com/c/en_in/solutions/data-center-virtualization/what-is-a-data%2Dcenter.html, (Accessed: 2022-03-17).

[15] Naresh Kumar Sehgal, P. Ch Bhatt, Cloud Computing: Concepts and Practices, Springer, 2018.

[16] Cara Erenben, Cloud computing: the economic imperative, eSchool News, Marist College (2009) 13–19, URL https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENPRESS/E090302E.pdf.

[17] The Computer History Museum (CHM), Timeline of computer history, 2022, https://www.computerhistory.org/timeline/computers/, (Accessed: 2022-07-11).

[18] Chee Shin Yeo, Rajkumar Buyya, Hossein Pourreza, Rasit Eskicioglu, Peter Graham, Frank Sommers, Cluster computing: High-performance, high-availability, and high-throughput processing on a network of computers, in: Handbook of Nature-Inspired and Innovative Computing, Springer, 2006, pp. 521–551.

[19] MicroSoft, What is grid computing?, 2022, https://azure.microsoft.com/en-us/overview/what-is-grid-computing/, (Accessed: 2022-03-03).

[20] Indu Gandotra, Pawanesh Abrol, Pooja Gupta, Rohit Uppal, Sandeep Singh, Cloud computing over cluster, grid computing: a comparative analysis, J. Grid Distributed Comput. 1 (1) (2011) 1–4.

[21] Liang-Jie Zhang, Qun Zhou, Jen-Yao Chung, Developing grid computing applications, IBM Report, IBM TJ Watson Research Center, New York, USA, 2003.

[22] Henri Casanova, Distributed computing research issues in grid computing, ACM SIGAct News 33 (3) (2002) 50–70.

[23] Blaise Barney, Donald Frederick, Introduction to parallel computing tutorial. Prepared by: Livermore computing: HPC at LLNL, 2022, https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial, (Accessed: 2022-04-25).

[24] Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2011.

[25] Auday Al-Dulaimy, Wassim Itani, Rached Zantout, Ahmed Zekri, Type-aware virtual machine management for energy efficient cloud data centers, Sustain. Comput.: Inform. Syst. 19 (2018) 185–203.

[26] Claus Pahl, Antonio Brogi, Jacopo Soldani, Pooyan Jamshidi, Cloud container technologies: a state-of-the-art review, IEEE Trans. Cloud Comput. 7 (3) (2017) 677–692.

[27] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, Cloud computing and grid computing 360-degree compared, in: 2008 Grid Computing Environments Workshop, Ieee, 2008, pp. 1–10.

[28] Daryl Plummer, Experts define cloud computing: Can we get a little definition in our definitions?, 2009, http://blogs.gartner.com/daryl_plummer/2009/01/27/experts-define-cloud-computing-can-we-get-a-little-definition-in-our-definitions/.

[29] The Open Commons Consortium (OCC), Cloud computing, 2009, https://www.occ-data.org/, (Accessed: 2022-03-28).

[30] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Fut. Gener. Comput. Syst. 25 (6) (2009) 599–616.

[31] Peter Mell, Tim Grance, et al., The NIST definition of cloud computing, 2011.

[32] Cristina Abad, Ian T. Foster, Nikolas Herbst, Alexandru Iosup, Serverless computing (dagstuhl seminar 21201), in: Cristina Abad, Ian T. Foster, Nikolas Herbst, Alexandru Iosup (Eds.), Dagstuhl Reports (ISSN: 2192-5283) 11 (4) (2021) 34–93, http://dx.doi.org/10.4230/DagRep.11.4.34, URL https://drops.dagstuhl.de/opus/volltexte/2021/14798.

[33] Qi Zhang, Lu Cheng, Raouf Boutaba, Cloud computing: state-of-the-art and research challenges, J. Int. Serv. Appl. 1 (1) (2010) 7–18.

[34] Frank Gens, New IDC IT cloud services survey: Top benefits and challenges, IDC Exchange (2009).

[35] John W. Rittinghouse, James F. Ransome, Cloud Computing: Implementation, Management, and Security, CRC Press, 2016.

[36] Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andrew Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al., Above the clouds: A berkeley view of cloud computing, Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28, (13) 2009, p. 2009.

[37] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan, The rise of "big data" on cloud computing: Review and open research issues, Inf. Syst. 47 (2015) 98–115.

[38] Auday Al-Dulaimy, Advanced virtual machine management with SLA constraints in cloud computing environments, (PhD thesis), Auday Al-Dulaimy, 2018.

[39] George Kousiouris, Dimosthenis Kyriazis, Functionalities, challenges and enablers for a generalized faas based architecture as the realizer of cloud/edge continuum interplay, in: Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER, SciTePress, (ISSN: 2184-5042) ISBN: 978-989-758-510-4, 2021, pp. 199–206, http://dx.doi.org/10.5220/0010412101990206.

[40] Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, Jie Yu, Di Niu, Bo Li, Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications, IEEE Wirel. Commun. 20 (3) (2013) 14–22.

[41] Mahadev Satyanarayanan, Fundamental challenges in mobile computing, in: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, 1996, pp. 1–7.

[42] Niroshinie Fernando, Seng W. Loke, Wenny Rahayu, Mobile cloud computing: A survey, Fut. Gener. Comput. Syst. 29 (1) (2013) 84–106.

[43] Cisco, Fog computing and the internet of things: extend the cloud to where the things are, White paper, 2015.

[44] Perry Lea, Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security, Packt Publishing Ltd, 2018.

[45] David Greenfield, Fog computing vs. Edge computing: What's the difference?, 2022, https://www.automationworld.com/products/data/blog/13315784/fog-computing-vs-edge-computing-whats-the-difference, (Accessed: 2022-03-22).

[46] ETSI, Multi-access-edge-computing, 2020, https://www.etsi.org/technologies/multi-access-edge-computing, (Accessed: 2022-03-24).

[47] 5G Infrastructure PPP Association, et al., 5G vision-the 5G infrastructure public private partnership: the next generation of communication networks and services, White Paper, February, 2015.

[48] Liqiang Zhao, Guorong Zhou, Gan Zheng, I. Chih-Lin, Xiaohu You, Lajos Hanzo, Open-source multi-access edge computing for 6g: Opportunities and challenges, IEEE Access 9 (2021) 158426–158439.

[49] Auday Al-Dulaimy, Yogesh Sharma, Michel Gokan Khan, Javid Taheri, Introduction to edge computing, Edge Comput.: Model. Technol. Appl. (2020) 1.

[50] Usman Shaukat, Ejaz Ahmed, Zahid Anwar, Feng Xia, Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges, J. Netw. Comput. Appl. 62 (2016) 18–40.

[51] Partha Pratim Ray, An introduction to dew computing: definition, concept and implications, IEEE Access 6 (2017) 723–737.

[52] Jürgo S. Preden, Kalle Tammemäe, Axel Jantsch, Mairo Leier, Andri Riid, Emine Calis, The benefits of self-awareness and attention in fog and mist computing, Computer 48 (7) (2015) 37–45.

[53] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, Rajiv Ranjan, Osmotic computing: A new paradigm for edge/cloud integration, IEEE Cloud Comput. 3 (6) (2016) 76–83, http://dx.doi.org/10.1109/MCC.2016.124.

[54] Massimo Villari, Antonino Galletta, Antonino Celesti, Lorenzo Carnevale, Maria Fazio, Osmotic computing: Software defined membranes meet private/federated blockchains, in: 2018 IEEE Symposium on Computers and Communications, ISCC, 2018, pp. 01292–01297, http://dx.doi.org/10.1109/ISCC.2018.8538546.

[55] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, Tie Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.

[56] Emilio Calvanese Strinati, Michael Peeters, Cesar Roda Neve, Manil Dev Gomony, Andreia Cathelin, Mauro Renato Boldi, Mark Ingels, Aritra Banerjee, Pascal Chevalier, Bartek Kozicki, Didier Belot, The hardware foundation of 6G: The NEW-6G approach, in: 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2022, pp. 423–428, http://dx.doi.org/10.1109/EuCNC/6GSummit54941.2022.9815700.

[57] Diego Kreutz, Fernando M.V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2014) 14–76.

[58] CISCO, Software-defined networking, 2022, https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html, (Accessed: 2022-04-02).

[59] Yosr Jarraya, Taous Madi, Mourad Debbabi, A survey and a layered taxonomy of software-defined networking, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1955–1980.

[60] A.U. Rehman, Rui L. Aguiar, Joao Paulo Barraca, Network functions virtualization: The long road to commercial deployments, IEEE Access 7 (2019) 60439–60464.

[61] ETSI, Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action, White paper, 2012, https://www.etsi.org/technologies/nfv, (Accessed: 2022-04-05).

[62] Matthijs Jansen, Auday Al-Dulaimy, Alessandro V. Papadopoulos, Animesh Trivedi, Alexandru Iosup, The SPEC-RG reference architecture for the compute continuum, in: Proceedings of the 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2023.

[63] Satyanarayanan, The emergence of edge computing, Computer 50 (2017).

[64] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, Lingjia Tang, Neurosurgeon: Collaborative intelligence between the cloud and mobile edge, ACM SIGARCH Comput. Archit. News 45 (1) (2017) 615–629.

[65] Wenlu Hu, Brandon Amos, Zhuo Chen, Kiryong Ha, Wolfgang Richter, Padmanabhan Pillai, Benjamin Gilbert, Jan Harkes, Mahadev Satyanarayanan, The case for offload shaping, in: Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, 2015, pp. 51–56.

[66] Ying Xiong, Yulin Sun, Li Xing, Ying Huang, Extend cloud to edge with KubeEdge, in: 2018 IEEE/ACM Symposium on Edge Computing, SEC, IEEE, 2018, pp. 373–377.

[67] Satyanarayanan, et al., The computing landscape of the 21st century, in: HotMobile, 2019.

[68] Bonati, et al., Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead, Comput. Netw. 182 (2020).

[69] Qi, Tao, A smart manufacturing service system based on edge computing, fog computing, and cloud computing, IEEE Access 7 (2019).

[70] Willner, Gowtham, Toward a reference architecture model for industrial edge computing, IEEE Commun. Stand. Mag. 4 (2020).

[71] Hoang, et al., A survey of mobile cloud computing: architecture, applications, and approaches, Wirel. Commun. Mob. Comput. 13 (2013).

[72] Edge Computing Consortium (ECC) and Alliance of Industrial Internet (AII), Edge computing reference architecture 2.0, Technical report, 2017.

[73] Tseng, et al., Introduction to edge computing in IIoT, Technical report, 2018.

[74] Intel and SAP, IoT joint reference architecture from intel and SAP, Technical report, 2018.

[75] OpenNebula, Edge cloud architecture - white paper, Technical report, 2021.

[76] Sittón-Candanedo, et al., A review of edge computing reference architectures and a new global edge proposal, FGCS 99 (2019).

[77] Mahmud, et al., Cloud-fog interoperability in IoT-enabled healthcare solutions, in: ICDCN, 2018.

[78] OpenFog Consortium, OpenFog reference architecture for fog computing, Technical report, 2017.

[79] Pop, et al., The FORA fog computing platform for industrial IoT, Inf. Syst. 98 (2021).

[80] ETSI, Multi-access edge computing (MEC); framework and reference architecture, ETSI, DGS MEC 3 (2019).

[81] Hill, et al., System architecture directions for networked sensors, in: ASPLOS, 2000.

[82] Light, Mosquitto: server and client implementation of the MQTT protocol, J. Open Source Softw. 2 (2017).

[83] Erwin, Erwin edge, 2021, https://www.erwin.com/products/, (Accessed: 2021-05-30).

[84] Hao, et al., EdgeCons: Achieving efficient consensus in edge computing networks, in: HotEdge, 2018.

[85] ISO/IEC, Internet of things (IoT) – reference architecture, 2018, https://www.iso.org/standard/65695.html, (Accessed: 2024-03-11).

[86] IEC, Power system management and associated information exchange. Part 1: Reference architecture, 2016, https://webstore.iec.ch/publication/26251, (Accessed: 2024-03-11).

[87] ETSI, Intelligent transport systems (ITS); vehicular communications; GeoNetworking. Part 3: Network architecture, 2014, https://www.etsi.org/deliver/etsi_en/302600_302699/30263603/01.02.01_60/en_30263603v010201p.pdf, (Accessed: 2024-03-11).

[88] Intel, The intel IoT platform. Architecture specification. Internet of things (IoT), White paper, 2019, https://d885pvmm0z6oe.cloudfront.net/hubs/intel_80616/assets/downloads/general/Architecture_Specification_Of_An_IOT_Platform.pdf, (Accessed: 2024-03-13).

[89] Xinxi Wang, Ye Wang, Improving content-based and hybrid music recommendation using deep learning, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 627–636.

[90] Hyan-Soo Bae, Ho-Jin Lee, Suk-Gyu Lee, Voice recognition based on adaptive MFCC and deep learning, in: 2016 IEEE 11th Conference on Industrial Electronics and Applications, ICIEA, IEEE, 2016, pp. 1542–1546.

[91] Xue-Wen Chen, Xiaotong Lin, Big data deep learning: challenges and perspectives, IEEE Access 2 (2014) 514–525.

[92] Weisong Shi, Schahram Dustdar, The promise of edge computing, Computer 49 (5) (2016) 78–81.

[93] Weisong Shi, George Pallis, Zhiwei Xu, Edge computing [scanning the issue], Proc. IEEE 107 (8) (2019) 1474–1481.

[94] Zhiming Hu, Maayan Shvo, Allan Jepson, Iqbal Mohomed, Interactive planning-based cognitive assistance on the edge, in: 3rd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 20), 2020.

[95] Xiaofei Wang, Yiwen Han, Victor C.M. Leung, Dusit Niyato, Xueqiang Yan, Xu Chen, Convergence of edge computing and deep learning: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 869–904.

[96] Paul Fremantle, A reference architecture for the internet of things, WSO2 White paper, 2015.

[97] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, Shiuhpyng Shieh, IoT security: ongoing challenges and research opportunities, in: 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, IEEE, 2014, pp. 230–234.

[98] Fritz AI, AI hardware for mobile ML, 2021, https://www.fritz.ai/mobile-ai-hardware, (Accessed: 2024-03-19).

[99] Tensorflow, TensorFlow lite, 2021, https://www.tensorflow.org/lite, (Accessed: 2021-05-15).

[100] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[101] Song Han, Jeff Pool, John Tran, William J. Dally, Learning both weights and connections for efficient neural networks, 2015, arXiv preprint arXiv:1506.02626.

[102] Matthieu Courbariaux, Yoshua Bengio, Jean-Pierre David, Binaryconnect: Training deep neural networks with binary weights during propagations, 2015, arXiv preprint arXiv:1511.00363.

[103] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

[104] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, 2016, arXiv preprint arXiv:1602.07360.

[105] En Li, Zhi Zhou, Xu Chen, Edge intelligence: On-demand deep learning model co-inference with device-edge synergy, in: Proceedings of the 2018 Workshop on Mobile Edge Communications, 2018, pp. 31–36.

[106] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al., {TVM}: An automated end-to-end optimizing compiler for deep learning, in: 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), 2018, pp. 578–594.

[107] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[108] NVIDIA, NVIDIA jetson: The AI platform for edge computing, 2021, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/, (Accessed: 2021-06-15).

[109] Google, Coral, 2020, https://coral.ai/, (Accessed: 2021-06-15).

[110] Kubernetes, Kubernetes, 2021, https://kubernetes.io/, (Accessed: 2021-06-08).

[111] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, Michael J. Freedman, Live video analytics at scale with approximation and delay-tolerance, in: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17), 2017, pp. 377–392.

[112] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, Ion Stoica, Chameleon: scalable adaptation of video analytics, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, 2018, pp. 253–266.

[113] Jiasi Chen, Xukan Ran, Deep learning with edge computing: A review, Proc. IEEE 107 (8) (2019) 1655–1674.

[114] Angela H. Jiang, Daniel L.-K. Wong, Christopher Canel, Lilia Tang, Ishan Misra, Michael Kaminsky, Michael A. Kozuch, Padmanabhan Pillai, David G. Andersen, Gregory R. Ganger, Mainstream: Dynamic stem-sharing for multi-tenant video processing, in: 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18), 2018, pp. 29–42.

[115] Alejandro Cartas, Martin Kocour, Aravindh Raman, Ilias Leontiadis, Jordi Luque, Nishanth Sastry, Jose Nuñez-Martinez, Diego Perino, Carlos Segura, A reality check on inference at mobile networks edge, in: Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking, 2019, pp. 54–59.

[116] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, Song Han, Once-for-all: Train one network and specialize it for efficient deployment, 2019, arXiv preprint arXiv:1908.09791.

[117] Khaled Al-Gumaei, Kornelia Schuba, Andrej Friesen, Sascha Heymann, Carsten Pieper, Florian Pethig, Sebastian Schriegel, A survey of internet of things and big data integrated solutions for industrie 4.0, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, ETFA, Vol. 1, IEEE, 2018, pp. 1417–1424.

[118] Mohammed Salman Shaik, Václav Struhár, Zeinab Bakhshi, Van-Lan Dao, Nitin Desai, Alessandro V. Papadopoulos, Thomas Nolte, Vasileios Karagiannis, Stefan Schulte, Alexandre Venito, et al., Enabling fog-based industrial robotics systems, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Vol. 1, IEEE, 2020, pp. 61–68.

[119] Anna Sukiasyan, Secure Data Exchange in IIoT (Ph.D. thesis), 2019.

[120] Miguel Saez, Steven Lengieza, Francisco Maturana, Kira Barton, Dawn Tilbury, A data transformation adapter for smart manufacturing systems with edge and cloud computing capabilities, in: 2018 IEEE International Conference on Electro/Information Technology, EIT, IEEE, 2018, pp. 0519–0524.

[121] Baotong Chen, Jiafu Wan, Antonio Celesti, Di Li, Haider Abbas, Qin Zhang, Edge computing in IoT-based manufacturing, IEEE Commun. Mag. 56 (9) (2018) 103–109.

[122] Inés Sittón-Candanedo, Ricardo S. Alonso, Sara Rodríguez-González, José Alberto García Coria, Fernando De La Prieta, Edge computing architectures in industry 4.0: A general survey and comparison, in: International Workshop on Soft Computing Models in Industrial and Environmental Applications, Springer, 2019, pp. 121–131.

[123] Linux Foundation project, ACRN, 2021, https://projectacrn.org/, (Accessed: 2021-06-22).

[124] SYSGO, Pikeos, 2021, https://www.sysgo.com/pikeos, (Accessed: 2021-06-22).

[125] O-PAF, O-PAS standard V1.0, 2021, https://publications.opengroup.org/p190, (Accessed: 2021-10-28).

[126] Google, Firebase database, 2021, https://firebase.google.com/docs/database/, (Accessed: 2021-06-22).

[127] Chao Li, Yushu Xue, Jing Wang, Weigong Zhang, Tao Li, Edge-oriented computing paradigms: A survey on architecture design and system management, ACM Comput. Surv. 51 (2) (2018) 1–34.

[128] Badrish Chandramouli, Joris Claessens, Suman Nath, Ivo Santos, Wenchao Zhou, RACE: Real-time applications over cloud-edge, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012, pp. 625–628.

[129] Vitor Barbosa C. Souza, Wilson Ramírez, Xavier Masip-Bruin, Eva Marín-Tordera, G. Ren, Ghazal Tashakor, Handling service allocation in combined fog-cloud scenarios, in: 2016 IEEE International Conference on Communications, ICC, IEEE, 2016, pp. 1–5.

[130] Xavi Masip-Bruin, Eva Marín-Tordera, Ghazal Tashakor, Admela Jukan, Guang-Jie Ren, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems, IEEE Wirel. Commun. 23 (5) (2016) 120–128.

[131] Joseph R. Barr, Daniela D'Auria, Fabio Persia, Telemedicine, homecare in the era of COVID-19 amp; beyond, in: 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), 2020, pp. 48–51, http://dx.doi.org/10.1109/AI4I49448.2020.00017.

[132] Mohammad S. Jalali, Adam Landman, William J. Gordon, Telemedicine, privacy, and information security in the age of COVID-19, J. Am. Med. Inf. Assoc. (ISSN: 1527-974X) 28 (3) (2020) 671–672, http://dx.doi.org/10.1093/jamia/ocaa310.

[133] Edimara Luciano, M. Adam Mahmood, Parand Mansouri Rad, Telemedicine adoption issues in the United States and Brazil: Perception of healthcare professionals, Health Inform. J. 26 (4) (2020) 2344–2361, http://dx.doi.org/10.1177/1460458220902957.

[134] Filipe Moreira, Demétrio Matos, Vítor Carvalho, Filomena Soares, Design of a biomedical kit for bedridden patients: a conceptual approach, in: IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1, 2019, pp. 6859–6865, http://dx.doi.org/10.1109/IECON.2019.8927741.

[135] O.Y. Tham, M.A. Markom, A.H. Abu Bakar, E.S. Mohd Muslim Tan, A.M. Markom, IoT health monitoring device of oxygen saturation (SpO2) and heart rate level, in: 2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering, ICITAMEE, 2020, pp. 128–133, http://dx.doi.org/10.1109/ICITAMEE50454.2020.9398455.

[136] Andrea Lacava, Valerio Zottola, Alessio Bonaldo, Francesca Cuomo, Stefano Basagni, Securing bluetooth low energy networking: An overview of security procedures and threats, Comput. Netw. (ISSN: 1389-1286) 211 (2022) 108953, http://dx.doi.org/10.1016/j.comnet.2022.108953, URL https://www.sciencedirect.com/science/article/pii/S1389128622001335.

[137] Sandeep Pirbhulal, Wanqing Wu, Subhas Chandra Mukhopadhyay, Guanglin Li, A medical-IoT based framework for ehealth care, in: 2018 International Symposium in Sensing and Instrumentation in IoT Era, ISSI, 2018, pp. 1–4, http://dx.doi.org/10.1109/ISSI.2018.8538031.

[138] Shenqi Jing, Ran Xiao, Tao Shan, Zhongmin Wang, Yun Liu, Application practice of smart hospital based on IoT cloud platform, in: 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), 2020, pp. 1–2, http://dx.doi.org/10.1109/ICCE-Taiwan49838.2020.9258204.

[139] E. Mishra, A. Bhatnagar, Wanna cry ransom ware: evaluating risk & implementing security measures, 2018, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058187719&partnerID=40&md5=5ff209d79556c45398922a319c651b56.

[140] Bindhu Raj L., R. Vandana, Santhosh Kumar B.J., Integrity based authentication and secure information transfer over cloud for hospital management system, in: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 139–144, http://dx.doi.org/10.1109/ICICCS48265.2020.9121079.

[141] Steven N. Baldassano, Shawniqua Williams Roberson, Ramani Balu, Brittany Scheid, John M. Bernabei, Jay Pathmanathan, Brian Oommen, Damien Leri, Javier Echauz, Michael Gelfand, Paulomi Kadakia Bhalla, Chloe E. Hill, Amanda Christini, Joost B. Wagenaar, Brian Litt, Iris: a modular platform for continuous monitoring and caretaker notification in the intensive care unit, IEEE Journal of Biomedical and Health Informatics 24 (8) (2020) 2389–2397, http://dx.doi.org/10.1109/JBHI.2020.2965858.

[142] Antonio Celesti, Armando Ruggeri, Maria Fazio, Antonino Galletta, Massimo Villari, Agata Romano, Blockchain-based healthcare workflow for tele-medical laboratory in federated hospital IoT clouds, Sensors (ISSN: 1424-8220) 20 (9) (2020) http://dx.doi.org/10.3390/s20092590, URL https://www.mdpi.com/1424-8220/20/9/2590.

[143] Fei Tang, Shuai Ma, Yong Xiang, Changlu Lin, An efficient authentication scheme for blockchain-based electronic health records, IEEE Access 7 (2019) 41678–41689, http://dx.doi.org/10.1109/ACCESS.2019.2904300.

[144] Leila Ismail, Huned Materwala, Sherali Zeadally, Lightweight blockchain for healthcare, IEEE Access 7 (2019) 149935–149951, http://dx.doi.org/10.1109/ACCESS.2019.2947613.

[145] Adil Hussain Seh, Mohammad Zarour, Mamdouh Alenezi, Amal Krishna Sarkar, Alka Agrawal, Rajeev Kumar, Raees Ahmad Khan, Healthcare data breaches: Insights and implications, Healthcare (ISSN: 2227-9032) 8 (2) (2020) http://dx.doi.org/10.3390/healthcare8020133, URL https://www.mdpi.com/2227-9032/8/2/133.

[146] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, Lanyu Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646, http://dx.doi.org/10.1109/JIOT.2016.2579198.

[147] Abdul Haseeb, Mihhail Matskin, Peep Küngas, Distributed web services discovery middleware for edges of internet, in: 2010 IEEE International Conference on Web Services, 2010, pp. 680–682, http://dx.doi.org/10.1109/ICWS.2010.87.

[148] Ananda M. Ghosh, Katarina Grolinger, Deep learning: Edge-cloud data analytics for IoT, in: 2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE, 2019, pp. 1–7, http://dx.doi.org/10.1109/CCECE.2019.8861806.

[149] Soumyalatha Naveen, Manjunath R. Kounte, Machine learning at resource constraint edge device using bonsai algorithm, in: 2020 Third International Conference on Advances in Electronics, Computers and Communications, ICAECC, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICAECC50550.2020.9339514.

[150] Ali Alnoman, Machine learning-based task clustering for enhanced virtual machine utilization in edge computing, in: 2020 IEEE Canadian Conference on Electrical and Computer Engineering, CCECE, 2020, pp. 1–4, http://dx.doi.org/10.1109/CCECE47787.2020.9255811.

[151] Tianyu Yang, Yulin Hu, M. Cenk Gursoy, Anke Schmeink, Rudolf Mathar, Deep reinforcement learning based resource allocation in low latency edge computing networks, in: 2018 15th International Symposium on Wireless Communication Systems, ISWCS, 2018, pp. 1–5, http://dx.doi.org/10.1109/ISWCS.2018.8491089.

[152] Antonio Celesti, Antonino Galletta, Maria Fazio, Massimo Villari, Towards hybrid multi-cloud storage systems: Understanding how to perform data transfer, Big Data Res. (ISSN: 2214-5796) 16 (2019) 1–17, http://dx.doi.org/10.1016/j.bdr.2019.02.002, URL https://www.sciencedirect.com/science/article/pii/S2214579618302004.

[153] Kuanishbay Sadatdiynov, Laizhong Cui, Lei Zhang, Joshua Zhexue Huang, Salman Salloum, Mohammad Sultan Mahmud, A review of optimization methods for computation offloading in edge computing networks, Digit. Commun. Netw. (ISSN: 2352-8648) (2022) http://dx.doi.org/10.1016/j.dcan.2022.03.003, URL https://www.sciencedirect.com/science/article/pii/S2352864822000244.

[154] Ahmed I. Awad, Mostafa M. Fouda, Marwa M. Khashaba, Ehab R. Mohamed, Khalid M. Hosny, Utilization of mobile edge computing on the internet of medical things: A survey, ICT Express (ISSN: 2405-9595) (2022) http://dx.doi.org/10.1016/j.icte.2022.05.006, URL https://www.sciencedirect.com/science/article/pii/S2405959522000753.

[155] Dinesh Soni, Neetesh Kumar, Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy, J. Netw. Comput. Appl. (ISSN: 1084-8045) 205 (2022) 103419, http://dx.doi.org/10.1016/j.jnca.2022.103419, URL https://www.sciencedirect.com/science/article/pii/S1084804522000765.

[156] Ioannis Korontanis, Konstantinos Tserpes, Maria Pateraki, Lorenzo Blasi, John Violos, Ferran Diego, Eduard Marin, Nicolas Kourtellis, Massimo Coppola, Emanuele Carlini, Zbyszek Ledwoń, Przemysław Tarkowski, Thomas Loven, Yago González Rozas, Mike Kentros, Michael Dodis, Patrizio Dazzi, Inter-operability and orchestration in heterogeneous cloud/edge resources: The ACCORDION vision, in: Proceedings of the 1st Workshop on Flexible Resource and Application Management on the Edge, FRAME '21, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450383844, 2020, pp. 9–14, http://dx.doi.org/10.1145/3452369.3463816.

[157] Mattia Fogli, Thomas Kudla, Bram Musters, Geert Pingen, Casper Van den Broek, Harrie Bastiaansen, Niranjan Suri, Sean Webb, Performance evaluation of kubernetes distributions (K8s, K3s, KubeEdge) in an adaptive and federated cloud infrastructure for disadvantaged tactical networks, in: 2021 International Conference on Military Communication and Information Systems, ICMCIS, IEEE, 2021, pp. 1–7.

[158] Wuyang Zhang, Sugang Li, Luyang Liu, Zhenhua Jia, Yanyong Zhang, Dipankar Raychaudhuri, Hetero-edge: Orchestration of real-time vision applications on heterogeneous edge clouds, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 1270–1278.

[159] Gabriela Viale Pereira, Peter Parycek, Enzo Falco, Reinout Kleinhans, Smart governance in the context of smart cities: A literature review, Inf. Polity 23 (2) (2018) 143–162.

[160] Ivan Lujic, Vincenzo de Maio, Klaus Pollhammer, Ivan Bodrozic, Josip Lasic, Ivona Brandic, Increasing traffic safety with real-time edge analytics and 5G, in: Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, 2021.

[161] Ivan Lujic, Vincenzo De Maio, Srikumar Venugopal, Ivona Brandic, SEA-LEAP: Self-adaptive and locality-aware edge analytics placement, IEEE Trans. Serv. Comput. (2021).

[162] Roger A. Light, Mosquitto: server and client implementation of the MQTT protocol, J. Open Source Softw. 2 (13) (2017) 265, http://dx.doi.org/10.21105/joss.00265.

[163] Atakan Aral, Melike Erol-Kantarci, Ivona Brandic, Staleness control for edge data analytics, Proc. ACM Meas. Anal. Comput. Syst. 4 (2) (2020) 38:1–38:24, http://dx.doi.org/10.1145/3392156.

[164] Dragi Kimovski, Dijana C. Bogatinoska, Narges Mehran, Aleksandar Karadimce, Natasa Paunkoska, Radu Prodan, Ninoslav Marina, Cloud—Edge offloading model for vehicular traffic analysis, in: 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2020, pp. 746–753.

[165] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, Liviu Iftode, Trafficview: A scalable traffic monitoring system, in: IEEE International Conference on Mobile Data Management, 2004. Proceedings. 2004, IEEE, 2004, pp. 13–26.

[166] Aya M. Kishk, Mahmoud Badawy, Hesham A. Ali, Ahmed I. Saleh, A new traffic congestion prediction strategy (TCPS) based on edge computing, Cluster Comput. (2021) 1–27.

[167] Samuel Kounev, Nikolas Herbst, Cristina L. Abad, Alexandru Iosup, Ian Foster, Prashant Shenoy, Omer Rana, Andrew A. Chien, Serverless computing: What it is, and what it is not? Commun. ACM 66 (9) (2023) 80–92.

[168] Panos Patros, Josef Spillner, Alessandro Vittorio Papadopoulos, Blesson Varghese, Omer Rana, Schahram Dustdar, Towards sustainable serverless computing, IEEE Internet Comput. (ISSN: 1089-7801,1941-0131) 25 (6) (2021) 42–50, http://dx.doi.org/10.1109/MIC.2021.3093105.

[169] Brendan McMahan, Daniel Ramage, Federated learning: Collaborative machine learning without centralized training data, 2017, Publication date: April 6.

[170] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, Virginia Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.

[171] Qiang Yang, Yang Liu, Tianjian Chen, Yongxin Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.

[172] Julia Siderska, Khambi Mubarok, Cloud manufacturing platform and architecture design, Multidiscip. Aspects Prod. Eng. 1 (2018).

[173] Qifan Pu, Ganesh Ananthanarayanan, Peter Bodik, Srikanth Kandula, Aditya Akella, Paramvir Bahl, Ion Stoica, Low latency geo-distributed data analytics, ACM SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 421–434.

[174] Animesh Trivedi, Lin Wang, Henri E. Bal, Alexandru Iosup, Sharing and caring of data at the edge, in: Irfan Ahmad, Ming Zhao (Eds.), 3rd USENIX Workshop on Hot Topics in Edge Computing, HotEdge 2020, June 25-26, 2020, USENIX Association, 2020, URL https://www.usenix.org/conference/hotedge20/presentation/trivedi.

[175] Zongheng Yang, Zhanghao Wu, Michael Luo, Wei-Lin Chiang, Romil Bhardwaj, Woosuk Kwon, Siyuan Zhuang, Frank Sifei Luan, Gautam Mittal, Scott Shenker, et al., {SkyPilot}: An intercloud broker for sky computing, in: 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), 2023, pp. 437–455.

[176] Dragi Kimovski, Nishant Saurabh, Matthijs Jansen, Atakan Aral, Auday Al-Dulaimy, André B. Bondi, Antonino Galletta, Alessandro V. Papadopoulos, Alexandru Iosup, Radu Prodan, Beyond von Neumann in the computing continuum: Architectures, applications, and future directions, IEEE Internet Comput. (2023).

[177] Sukhpal Singh Gill, Minxian Xu, Carlo Ottaviani, Panos Patros, Rami Bahsoon, Arash Shaghaghi, Muhammed Golec, Vlado Stankovski, Huaming Wu, Ajith Abraham, et al., AI for next generation computing: Emerging trends and future directions, Int. Things (2022) 100514.

[178] Catherine D. Schuman, Shruti R. Kulkarni, Maryam Parsa, J. Parker Mitchell, Prasanna Date, Bill Kay, Opportunities for neuromorphic computing algorithms and applications, Nat. Comput. Sci. 2 (1) (2022) 10–19.

[179] Amlan Ganguly, Sergi Abadal, Ishan Thakkar, Natalie Enright Jerger, Marc Riedel, Masoud Babaie, Rajeev Balasubramonian, Abu Sebastian, Sudeep Pasricha, Baris Taskin, Interconnects for dna, quantum, in-memory, and optical computing: Insights from a panel discussion, IEEE Micro 42 (3) (2022) 40–49.

[180] Auday Al-Dulaimy, Wassim Itani, Javid Taheri, Maha Shamseddine, Bwslicer: A bandwidth slicing framework for cloud data centers, Future Gener. Comput. Syst. 112 (2020) 767–784.

[181] Shiqiang Zhu, Ting Yu, Tao Xu, Hongyang Chen, Schahram Dustdar, Sylvain Gigan, Deniz Gunduz, Ekram Hossain, Yaochu Jin, Feng Lin, et al., Intelligent computing: The latest advances, challenges, and future, Intell. Comput. 2 (2023) 0006.

[182] Ana Juan Ferrer, Sören Becker, Florian Schmidt, Lauritz Thamsen, Odej Kao, Towards a cognitive compute continuum: An architecture for ad-hoc self-managed swarms, in: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, 2021, pp. 634–641.

[183] Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C.M. Moura, Aiko Pras, Cees de Laat, A responsible internet to increase trust in the digital world, J. Netw. Syst. Manag. 28 (4) (2020) 882–922, http://dx.doi.org/10.1007/s10922-020-09564-7.

[184] Wenji Mao, Fei-Yue Wang, Social computing in ISI, in: New Advances in Intelligence and Security Informatics, Academic Press, 2012, pp. 61–71.

[185] Ericsson, 6G – connecting a cyber-physical world, White paper, 2022, https://www.ericsson.com/en/reports-and-papers/white-papers/a-research-outlook-towards-6g, (Accessed: 2023-03-28).