

Review

Edge-Oriented Computing: A Survey on Research and Use Cases

Nour Alhuda Sulieman ^{1,*}, Lorenzo Ricciardi Celsi ², Wei Li ³, Albert Zomaya ³ and Massimo Villari ¹

¹ Dipartimento di Scienze Matematiche e Informatiche, Scienze Fisiche e Scienze Della Terra, Università di Messina, Piazza Pugliatti 1, 98122 Messina, Italy; mvillari@unime.it

² ELIS Innovation Hub, via Sandro Sandri 45-81, 00159 Roma, Italy; l.ricciardicelsi@elis.org

³ Centre for Distributed and High Performance Computing, University of Sydney, Sydney, NSW 2006, Australia; weiwilson.li@sydney.edu.au (W.L.); albert.zomaya@sydney.edu.au (A.Z.)

* Correspondence: nosulieman@unime.it

Abstract: Edge computing is a distributed computing paradigm such that client data are processed at the periphery of the network, as close as possible to the originating source. Since the 21st century has come to be known as the century of data due to the rapid increase in the quantity of exchanged data worldwide (especially in smart city applications such as autonomous vehicles), collecting and processing such data from sensors and Internet of Things devices operating in real time from remote locations and inhospitable operating environments almost anywhere in the world is a relevant emerging need. Indeed, edge computing is reshaping information technology and business computing. In this respect, the paper is aimed at providing a comprehensive overview of what edge computing is as well as the most relevant edge use cases, tradeoffs, and implementation considerations. In particular, this review article is focused on highlighting (i) the most recent trends relative to edge computing emerging in the research field and (ii) the main businesses that are taking operations at the edge as well as the most used edge computing platforms (both proprietary and open source). First, the paper summarizes the concept of edge computing and compares it with cloud computing. After that, we discuss the challenges of optimal server placement, data security in edge networks, hybrid edge-cloud computing, simulation platforms for edge computing, and state-of-the-art improved edge networks. Finally, we explain the edge computing applications to 5G/6G networks and industrial internet of things. Several studies review a set of attractive edge features, system architectures, and edge application platforms that impact different industry sectors. The experimental results achieved in the cited works are reported in order to prove how edge computing improves the efficiency of Internet of Things networks. On the other hand, the work highlights possible vulnerabilities and open issues emerging in the context of edge computing architectures, thus proposing future directions to be investigated.

Keywords: edge computing; cloud computing; industrial internet of things; mobile edge computing; hybrid edge-cloud



Citation: Sulieman, N.A.; Ricciardi Celsi, L.; Li, W.; Zomaya, A.; Villari, M. Edge-Oriented Computing: A Survey on Research and Use Cases. *Energies* **2022**, *15*, 452. <https://doi.org/10.3390/en15020452>

Academic Editors: Ricciardi Celsi and Gwanggil Jeon

Received: 29 November 2021

Accepted: 31 December 2021

Published: 10 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many technologies have recently been proposed to face the daily challenges of having smarter platforms. Numerous studies have been conducted, and different applications have been launched to provide better infrastructure and interconnected networks to satisfy customers' requirements in the market for having on-demand, reliable, and secure smart services.

It is clear that big data analysis plays a critical role in terms of improving the provided quality of these applications and services, starting from processing a vast amount of data in order to take actions when they are needed or to create patterns that can be useful in the future for prediction purposes. Usually, analysis solutions are implemented with a *centralized cloud data center* where data are saved and processed. However, cloud solutions

are continuously facing challenges due to a high need for new features that could solve existing technical limitations and network-overloading problems of the centralized cloud data center, in addition to the critical necessity of responding to some events in real time and taking actions immediately without waiting for the transmission delay of such a network.

Edge computing can be considered as one of the candidate technologies that could tackle these problems. Namely, edge computing is a distributed computing paradigm that allows computation and data storage to be brought as close as possible to the relevant data sources. Edge computing technology has been proposed to satisfy the requirements of low latency and to reduce bandwidth consumption [1]. It was nominated in 2018 as one of the best technologies that will lead businesses in the aforementioned future, especially businesses linked to internet of things (IoT) networks where a fast response is one of the main targets of the application. Moreover, the continuous pulling and pushing of data from/to the installed IoT devices pose different problems related to the latency and reliability purposes of V2I and I2V applications [2–6]. Edge computing technology has been used successfully in some practical applications to solve some technical challenges, considering some essential issues that are needed to be solved in some applications, and to face the incredible increase in data congestion which is generated by the geo-distributed network of IoT devices [7–13]. The shift to this paradigm is expected to improve response times and save bandwidth significantly. Often, edge computing is mistakenly confused with IoT. It must be clarified that edge computing is a topology-sensitive and location-sensitive paradigm of distributed computing, while IoT must be regarded only as a use case instantiation of the edge computing paradigm.

The main contribution of this survey paper is as follows.

Section 2 discusses the reference architecture for edge computing networks.

Section 3 summarizes the motivation of our paper and how it differs from other existing surveys.

Section 4 proposes a comprehensive overview of the recent literature based on the main technical challenges that are arising in the edge computing domain, namely focusing on:

- Optimal placement of servers in mobile edge computing (Section 4.1);
- Security of data transferred through edge networks (Section 4.2);
- Distributed edge computing platforms and the relevance of hybrid edge-cloud computing (Section 4.3);
- The performance of edge computing evaluated through simulation platforms (Section 4.4);
- Improving the performance of edge networks (Section 4.5);
- Edge computing over 5G/6G networks (Section 4.6);
- Edge computing for industrial internet of things (IIoT, Section 4.7).

Section 5 discusses and summarizes the main findings reported in Section 4.

Section 6 discusses the main businesses that are currently taking operations on the edge, together with the most used edge computing platforms (both proprietary and open source).

Finally, Section 7 concludes the paper.

2. Reference Architecture

The reference architecture for edge computing is a federated network structure capable of extending cloud services to the edge of the network through the introduction of edge devices between terminal devices and cloud computing. Such architecture is generally divided into:

- Terminal layer;
- Edge layer;
- Computing layer.

The terminal layer is composed of all devices that are connected to the edge network, including mobile terminals and IoT devices such as sensors, smartphones, smart cars, and cameras. These devices may play the role of both data consumers and data providers.

The boundary layer or edge layer is located at the edge of the network and is composed of the edge nodes that are distributed between terminal devices and clouds, including base stations, access points, routers, switches, and gateways.

Eventually, the cloud computing layer embeds the most powerful data processing center in the considered architecture, consisting of high-performance servers and storage devices, with advanced computing and storage capabilities: these are in charge of data analysis, regular maintenance, and decision support to businesses.

Accordingly, the reference architecture is sketched in Figure 1, as inspired by the edge computing white paper 3.0 released in December 2018 [14].

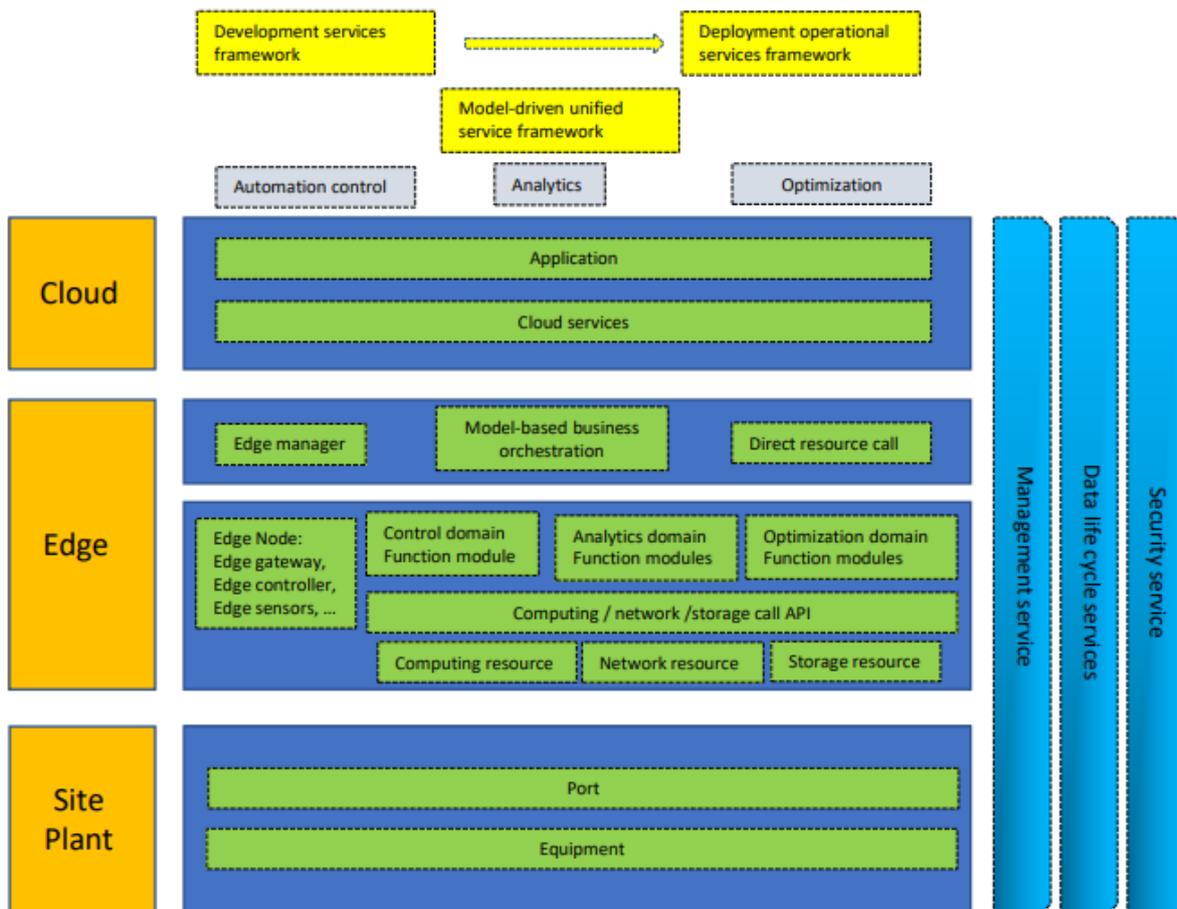


Figure 1. Edge computing reference architecture [14].

Edge computing has advantages over cloud computing in terms of fast data processing and analysis in real-time: the rapid growth of data volume and the pressure of network bandwidth are instead the main disadvantages of cloud computing [15]. In this respect, edge computing being closer to the data source, data storage and computing tasks can be carried out in the edge computing node, which reduces the intermediate data transmission process. Edge computing normally exploits proximity to users and provides users with high-performing intelligent services, thus improving the data transmission performance, ensuring real-time processing and reducing delay time.

The main difference that results as a consequence of this, therefore, lies in that edge computing is helpful for small-scale intelligent analysis in real-time and with low network bandwidth pressure. In contrast, cloud computing is aimed at large-scale centralized processing, despite exhibiting much higher network bandwidth pressure.

3. Motivation

Recent years have been characterized by the emergence of the edge computing paradigm. Many companies have announced that they have started to apply the new technology in their businesses or are planning to use it soon due to its advantages of enhancing business performance and meeting customers' requirements. On the other hand, many academic papers, articles, and surveys have been published either to propose new infrastructures using edge computing for existing systems or to prove the feasibility, the productivity, and the high performance of some practical experiments based on such technology. Moreover, multiple edge platforms are currently used in the market, and many companies have started to shift their businesses to the edge by using more powerful servers, gateways, and other solutions.

Until 2014, edge computing was not the most attractive topic for IT researchers and the technological world. However, as more efforts have been made towards exploiting this new technology, the trend of published papers in this field has increased as suggested in Figure 2.

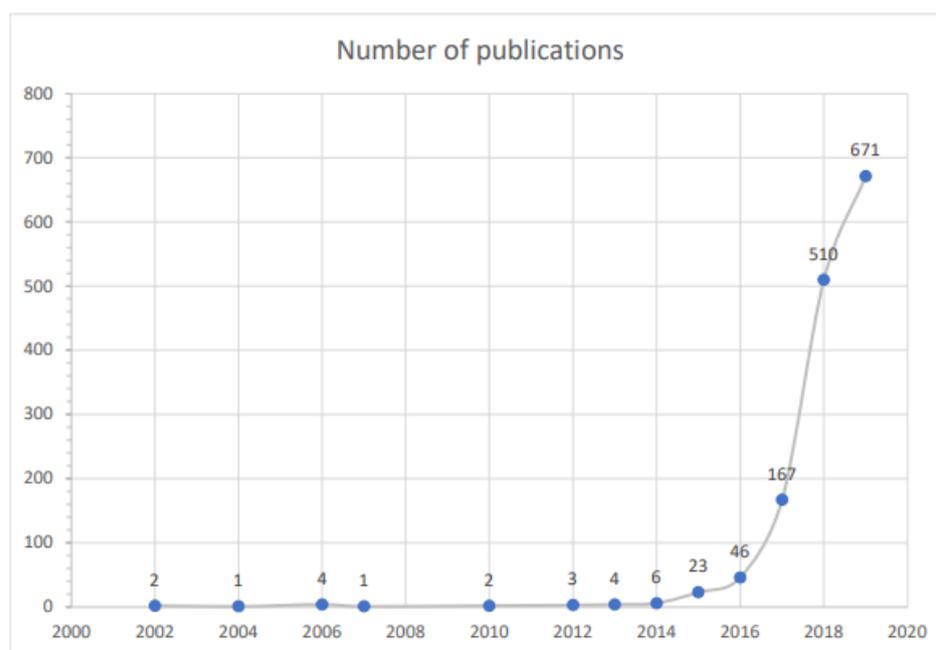


Figure 2. Number of scientific publications on the edge computing topic versus time [16].

Accordingly, the relevance of edge computing platforms has been emerging in several application scenarios, as shown by the pie chart in Figure 3 [17].

This review article provides a comprehensive report of the most relevant problems and use cases and discusses new possible infrastructures to be applied with respect to edge computing technology.

In particular, the main emerging paradigms are outlined, highlighting the advantage of edge computing with respect to the traditional cloud computing scenario.

As anticipated in Section 2, edge computing enables a relevant decrease in the latency between processing data in cloud centers and exploiting them at end user locations, balancing network traffic, avoiding network bottleneck, and reducing the response time for time-sensitive applications and real-time analysis insights. In particular, the authors found that a comprehensive survey was still not available with respect to the recent works focused on the topics of optimal server placement, data security, hybrid edge-cloud, simulation platforms, approaches for improving edge networks, and the relevant applications of edge computing to 5G/6G networks as well as to the domain of IIoT. Hence, this is the reason behind the authors' choice to structure Section 4, as already detailed in the Introduction.

Moreover, highlighting the recent use of edge computing in some companies such as Apple, Dell, Amazon, and many others, besides showing how the existing platforms in the market use edge technology, is also important, with the aim of exploring the business-related benefits which can be achieved using edge technology. This is the main reason behind the authors' choice to write a dedicated section (namely, Section 5).

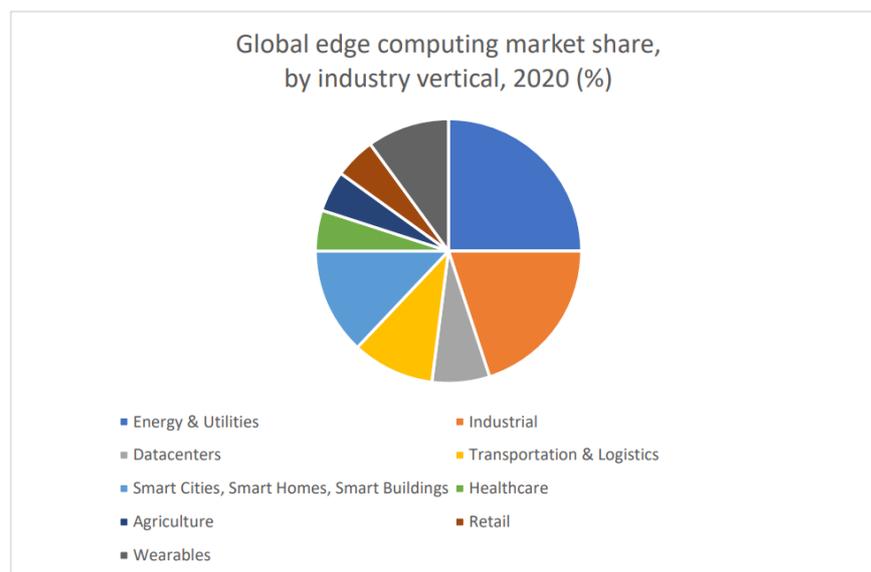


Figure 3. Global edge computing market share by industry vertical, as of 2020 [17].

4. Edge Computing in Recent Research

A summary of the recent edge contribution in research, business, and technological platforms is given in the following subsections. It emerges that multiple edge platforms are currently used in the market, and many companies have started to shift their business to the edge by using more powerful servers, gateways, and other solutions. Tables 1 and 2 sum up the most cited contributions among the works reported in the following subsections.

4.1. Optimal Placement of Servers in Mobile Edge Computing

Research has recently started giving attention to the challenge of performing *mobile edge computing* [18] far from the main central system architecture, as well as of mitigating latency to messages and transmission response times through the network.

Li and Wang in [19] study the problem of edge servers' placement and propose an advanced algorithm in which the locations of the edge servers are optimized in such a way as to improve the utilization of the installed servers, in addition to reducing the number of servers which work in the idle mode. In order to solve this problem, the authors assume that the mobile edge network behaves as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} represents the set of edge nodes with the potential locations of edge servers, and \mathcal{E} accounts for the communications among them. Although the energy consumption of edge servers is due to several sources such as CPU, hard drive memory, and other hardware elements, CPU is the most consuming resource based on the studies, which is why the authors consider CPU usage as the target cost function when it comes to saving energy through the mobile edge network.

Another perspective is given in [20,21], where the authors adjust the traditional particle swarm optimization (PSO) algorithm to make it applicable in discrete optimization problems such as the above-mentioned mobile edge network scenario. They challenge the conventional concept according to which each edge node should have a corresponding edge server, and they look for an optimal solution increasing the profiteering of servers and at the same time decreasing their total number. The proposed solution defines a *one-to-many*

assignment relationship between an installed server in the network, on the one hand, and edge nodes, on the other hand, and finds the best positioning solution that decreases the energy consumption of servers' CPUs while satisfying the following assumptions:

- CPU power consumption is based on the value of the server's power in both idle and full states of utilization;
- Each node should be assigned to one server, and each server could serve one or more nodes;
- Latency time should be always less than or equal to a pre-defined threshold.

The proposed algorithm for positioning edge servers has been tested on a real dataset from Shanghai Telecom where 3233 edge nodes are considered. Tests are considered two cases: in the first experiment, the number of edge nodes is fixed, while the distance threshold between nodes and servers varies; in the second experiment, the threshold is fixed, and the number of edge nodes is increased. The proposed approach shows better results when compared to traditional algorithms, yielding up to 12% less power consumption in the first experiment and up to 13% less power consumption in the second one, thus proving that the available resources of edge servers can be exploited dynamically rather than statically assigning one node to only one server. Numerical analyses are also reported in [20], demonstrating the average number of edge servers for the edge server placement with respect to the edge server coverage.

In [22], Xiao et al. proposed a heuristic algorithm that predicts a strategic place for edge servers based on resource requirement forecasting. The authors manage to determine a suitable number of possible server locations for a specific data source (mobile, bus station, PC, gaming stations, etc.) by predicting the next destination of this data source, in addition to using a data naming mechanism between a data source and the possible linked server. Markov chain modeling is used in this respect, together with an approach for detecting the optimal server location based on the following three parameters: migration time between two servers, processing time in the new server, and required time to transmit the result to the data source. A simple Kolmogorov equation is applied for the presented Markov chain modeling in this paper, while the optimization problem of a set of servers to be placed with the requirements of each server needs is formalized through different formulas in [22]. A data naming mechanism has been used as a connection method between data sources and edge servers in order to exchange the important mapping information (needed CPU power for the task, location of the data source, time, capacity, etc.) for the overall goal of enhancing edge computing by selecting the best service provider linked to the user's requests and available resources.

In [23], an efficient analytical model for self-adaptive service migration in edge computing networks is proposed. The aim of the authors is to keep the service location, which is provided by edge nodes, as close as possible to the end user place through dynamic adaptation targeted at maximizing the efficiency of the network performance. The mean absolute percentage error is evaluated as a key performance indicator while iteratively performing the following sequence of actions.

- Monitor: this action saves useful information, allowing one to define which edge node in the network is able to provide the requested service.
- Analyze and Plan: this action allows for deciding whether the service should be migrated to another node or not, based on the distance between user mobility and the edge node that hosts the service.
- Execute: this action launches the migration process.

By using basic probability and renewal theory arguments, different closed-form expressions for the average cost per unit time experienced by a mobile device or by the system are derived and listed in [23]. The resulting optimization model, defined as the "regional planning pattern", incorporates: a distance model, evaluating the distance between each device and each service provider; a time model, which describes the events of interest (device mobility) that could affect the distance model; a cost model, accumulating the

costs of a specific device each time it is evaluated to have a non-optimal distance; and a control model, according to which all non-optimal distances of devices are reported to a centralized planning component. Such a regional planning pattern is expected to have positive effects in the field since it presents less data traffic throughout any edge computing network arranged according to a fully distributed architecture.

4.2. Security of Data Transferred through Edge Networks

Another relevant emerging research topic is related to the security of data transferred over edge networks.

In [24], the security of the data shared across different domains using edge computing technology is investigated. The authors consider each edge node and its related equipment as a domain, where the centralized cloud connects all the domains of edge nodes with each other through the backhaul network. Four main components are considered in this model: the central cloud, which is the shared pool of all computational resources; edge servers, which are placed closer to end users' locations; the data owner, who can encrypt a message to all the users; and the data user, who receives messages from the owner. Entities in the proposed paradigm follow the traditional system architecture of edge computing, noting that both the data user and data owner are located in the first level (edge devices). To ensure the security of shared data between different edge domains, the RSA algorithm and CP-ABE are used. For example, the data owner of a specific edge domain (A) defines its privacy policy and attribute list with other domains and creates public and private keys. It encrypts and sends its public key with the request to edge domain (A) that it wants to share data with domain (B). Edge domain (A) validates the source information and sends the sharing request and the related public key to the cloud service provider. The latter finds the target domain (B) and forwards it the request (A wants to share data with B), in addition to the public key of (A). When (B) receives the request, it also uses its own attributes to generate security keys. It uses symmetric encryption to encrypt the previous attributes, and it uses the public key of (A) to encrypt this symmetric key. In this way, domain (A) can use its key to decrypt the attribute list of (B) and obtain its public key in order to be able to decrypt the shared data. As a result, only domains that satisfy the privacy policy can access the data and decrypt the shared messages, which proves the efficiency of the proposed model. However, this scheme could be a bit complicated if we consider the encryption, decryption, and authentication steps, in addition to the communication costs.

As part of the increasing focus on security issues in edge computing paradigms, a survey has been conducted in [25] to highlight the main security vulnerabilities and possible attacks while using virtualization technologies in the edge/fog computing paradigm. The authors analyze Unikernel, real-time operating systems, and containerization technologies and approaches while they were deployed on real edge computing use cases (smart cities, smart home, e-health, etc.), and then they depict the level of security impacts caused by different possible attacks. In their analysis, they defined five categories of vulnerabilities: vulnerabilities directly linked to the docker; vulnerabilities inside images; vulnerabilities of the Linux kernel; insecure production system configurations; and vulnerabilities in the image distribution, verification, decompression, and storage process. Many possible attacks may occur when docker technology is applied such as remote code execution, MAC flooding, DoS on images, data leakage, ARP spoofing, Zip-bomb-like attacks, changing image behavior, viruses, privilege escalation, container escape attacks, and others. On the other hand, several attacks have been detected when using Unikernel such as privilege escalation, attacks related to hypervisor vulnerabilities, and DoS attacks. The results proposed in this work show that Unikernel has a security isolation advantage over containers, while the latter have setup facility as a distinguishing advantage. Real-time operating systems have good performance in many applications, but they suffer from update difficulties, and they can be adopted only in such cases on constrained devices. One major point regards the level of security impact caused by the previously mentioned possible attacks. In other words, taking an e-health system as an example, attacks perpetrated against a specific system

could be the reason for a person's death, if compared to other systems where the same type of attack has a lower impact. As a result of the analyses, strategic guidelines have been proposed at the end of the work in order to reduce the risks of existing vulnerabilities. Unikernel could provide less attack risk in cloud gaming applications and better code integrity, while in smart cities, real-time operating systems could be a more feasible option.

In [26], Alrowaily and Lu introduce the main security concepts that should be taken into consideration in the safeguarding of edge computing networks. Keeping users' personal data safe and secure is one of the most important challenges emerging with the high growth of edge computing networks since all components communicate with each other and exchange significant amounts of data. This paper summarizes what should be considered in order to manage edge network privacy with respect to the following concepts: pseudonyms, unobservability, unlinkability, and anonymity. However, with the ever-rising amount of transferred data through the network, there is always a need to check that the functionality concepts which were mentioned before are working in the proper way. Therefore, four evaluating components are presented in this paper, and by ensuring their investigation, network security is guaranteed: (1) confidentiality, (2) integrity, (3) availability, and (4) access control and authentication. A thorough discussion is provided with respect to using these four components with the aim of reducing the relevant risks that may occur. For instance, multiple security algorithms are proven to yield superior results in many IoT applications that extensively rely on an edge computing networks.

4.3. Distributed Edge Computing Platforms and the Relevance of Hybrid Edge-Cloud Computing

An innovative computing platform by the name of DNR (Distributed Node Red) has been proposed in [27] as an extension to the well-known Node Red open source tool, which can provide a visual data flow for IoT applications. Three versions of this platform have been already tested, each one being used to solve a specific challenge. In more detail, Node Red is a visual tool where developers can evolve their applications by dragging and dropping entities and wiring them one to another. However, Node Red usually deals with single processes and does not support distributed applications such as edge computing networks. DNR-v1 already tried to solve this problem by enabling the execution of nodes on any device within the network (edge CPU, cloud server, etc.). This was managed by adding the concept of device id to the Node Red paradigm. Hence, in DNR-v1, each node in the flow is assigned to a unique device id representing an identification for the device the node will be deployed on. Launching the ability of allowing nodes to be executed on different devices cast a new challenge related to break-up cases in the data flow. For this reason, the two concepts of wire-in and wire-out were added. Nodes that cannot be run on the assigned device id are replaced with remote wires using a publish/subscribe mechanism. Wire-in nodes use the subscribe technique to receive data from another node while wire-out nodes receive data from another node and publish it in a communication broker so that the data are received by other parties. Then, DNR-v2 was released, with two main targets: enabling more complex and larger geo-distributed applications to be applied using this platform and executing nodes on multiple devices in parallel. In order to meet the previous requirements, a primary restriction parameter is added to each node in the flow in which not only is the type of device where the process should be implemented defined, but it is also allowed to choose the device based on other parameters such as memory size, physical location, CPU power, and others. Two other notions have been added to face the previous challenges: the first notion is the wire cardinality to solve the problem of parallel execution by allowing hosting instances of the same node by different devices, and the second notion is the wire fragmentation which is useful when one or more instances are accessible while all data sources contact the same one. Later, DNR-v3 introduced a communication layer to the previous platform, which is able to coordinate the connection with external software components to achieve the application objectives. This is implemented by adding new coordination nodes to the existing wires in the paradigm in addition to placing a centralized coordinator capable of receiving all control messages and passing them to the coordination

nodes, these latter taking the responsibility of the execution process to either pull or push data from other software components, and updating the status of coordination nodes after finishing the process execution. DNR experiments devised new solutions for having an exogenous platform in order to increase the computation efficiency through edge networks. Recent technological openness imposes a harder effort in the research field in order to obtain the best integrated technological solutions in order to increase the application efficiency. It is clear from the evolution of the DNR platform that the distributed platform of edge computing is an attractive topic in research, especially in that it can be integrated with containerization technology, which relevantly simplifies the development and execution of distributed applications.

In [28], a theoretical model is proposed with the aim of deciding when an edge-only or hybrid edge-cloud set up is to be used and also when it is better to rely on traditional cloud architecture. The proposed model is mainly based on the following parameters: selectivity ratio, computation-communication ratio, and cloud processing speed-up. As intuition suggests, the paper proves that when the cloud speed-up parameter is low and the computation-to-communication ratio is small, edge-only systems are faster than cloud-only ones. When comparing hybrid edge-cloud computing with cloud-only systems, it emerges that the hybrid setup is faster when the relevant hybrid edge-cloud speed-up parameter is greater than one. The authors analyze the performance of the previous three system architectures on a specific framework where two MapReduce functions are run on a hybrid system architecture, the former on the edge and the latter in the cloud. The performance is then tested on seven real applications, three of which do not support the hybrid execution. AWS is chosen as the relevant cloud-computing technology for this experiment, while two low-power devices are used to represent edge components. Only one application exposes a faster execution time when bandwidth is low in cloud computing compared to edge-only systems, while for large bandwidth, all the other applications have faster processing time on edge-only systems compared to cloud-only systems. The cloud-only setup is always faster when the application has low-data movement, whereas the hybrid edge-cloud system is always faster for large inputs and a small intermediate data size. All in all, the proposed model provides a useful framework for deciding whether to execute tasks using the traditional scenario of cloud computing or move the workload to edge-only or hybrid edge-cloud.

In [29], the main challenges in terms of application development relying on edge computing technology are investigated, and an efficient solution is proposed using containerization technology. Implementing an application on edge networks needs the satisfaction of some specific hardware requirements in terms of network connectivity, CPU power, and capacity that are significantly higher than in the case of applications running on normal desktop computers. In particular, the dependency of edge computing applications on obtaining real-time data from other vendors' applications poses a new time consumption challenge in terms of data suppliers when the latter have problems in their data sources. Additionally, there emerges the repetitive process after any change is made in the application, which consists of restarting the life cycle of the containerization process from scratch and sending the new version of the code to edge devices. The authors propose a remote debugging method which creates a docker container with all required programming libraries and packages being included in the remote edge node. This scenario enables application updates at the edge device without any need to create new containers each time a change happens, thanks to the presence of a secure copy of the application inside the remote debugging container. This proposal was tested in a production area for a CNC (computer numeric control) machine, and it showed better performance in terms of time consumption in addition to security proof while moving data among containers. Data migration happened by using the Docker Compose tool. The proposed approach showed positive results in C++ applications, and it is expected to yield the same favorable effect for applications in other programming languages, thus affecting the efficiency of the increasing number of edge computing applications. The advantages of edge computing technology

increasingly prove its effectiveness and worthiness in many applications and show the capability to answer users' requests fluently and locally using the installed servers that are placed physically closer to end users. These advantages make edge technology one of the best choices for some critical applications which need real-time actions such as tasks with hard deadlines in energy and manufacturing plants.

Table 1. Overview of research works cited in Sections 4.1–4.4.

Research Topic	Description	Conclusion
Edge Servers' Placement [19]	It provides optimized solutions to reduce the number of servers	Dynamic server assignment provides improved performance in place of current static solutions
Edge Servers' Placement [22]	Heuristic algorithm selecting servers based on the resource requirements forecasting	An interesting approach to choosing the best service provider
New Analytical Mode (Regional Planning Pattern) [23]	Self adaptive-service migration in edge network	Leads to less data traffic
Security of Edge Data [24]	It divides edge network into multiple domains and uses symmetric encryption for sharing data between these domains	High communication costs in addition to complicated encryption, decryption, and authentication processes
Possible Vulnerabilities Using Virtualization [25]	Survey mentioning the main security vulnerabilities and possible attacks in edge/fog computing paradigm	Results show that Unikernel has a security isolation advantage over containers; container has set up facility as a distinguish advantage; RTOS suffers from update difficulty, and it could be adopted only in such cases on constrained devices
Security Concepts to be Considered in Safeguarding Edge Network [26]	Network security is guaranteed by always checking 4 components: confidentiality, integrity, availability and access control, and authentication	By considering privacy challenges and security vulnerabilities of the network early on, output performance can be improved
DNR: New Computing Platform [27]	It is an extension to Node Red open source tool	It enables Node Red to be applied on distributed edge network and executes nodes in parallel; more complex and higher geo-distributed applications and better computation efficiency
Challenges of Application Development Using Edge Computing [29]	A solution is proposed using containerization technology by creating remote debugging method	It is expected to provide better performance for applications regardless their programming languages since it is tested with an application written in C++
Offloading Computational Model [30]	Works that compare response time when offloading to cloud only, to edge only, and to both of them	Testing results show that the best performance can be achieved by the combination between edge and cloud technologies, not using one of them
Running Neural Network on Edge-computing Platform with VPU [31]	It shows how it works when using devices from different operating systems and having different powers	It is expected in the future to have a data deluge transferring through the backhaul network, while the power of mobile edge computing topology is able to deal with this new situation of IoT network
SDEC-Based Open Edge Computing New System Architecture [32]	It utilizes the available resources in the best way	It improves the performance and reduces energy consumption; a scenario which targets the software models is one of the main challenges

Moreover, relative to the field of medical emergency services, an application of video-based heart rate detection on an edge computing platform is proposed in [30], consisting of four entities: a smartphone (Huawei Honor 8), a base station (according to the open source project of the 5G SoftRAN cellular system), an edge server, and a cloud server. The application test starts from sending facial videos from a user's smart phone to the edge server where data preprocessing happens, while other features are sent to the cloud for more complex analysis where cloud servers take this responsibility. Eventually, the combined result returns a reliable heart rate measurement to the smartphone. Experimental results emphasized that the response time of this hybrid edge-cloud architecture is 20% and

40% less than in the case of using edge technology and cloud technology only, respectively. This is a sign that despite many of the current expectations that edge computing will replace cloud technology, practical experiments are continuously showing that the best performance can be achieved through their combination.

Another interesting practical experiment is proposed in [31]. This experiment runs a neural network on an edge computing platform with the support of VPU (vision processing unit); it was shown how it works when using devices from different operating systems and with different powers. An RPi3B was used as an edge device. It has a 1.2 GHz 64bit quad-core Cortex-A53 (ARMv8) CPU, four USB 2.0 ports via an on-board five-port USB hub, and 1 GB low-power DDR2 SDRAM, and it draws a maximum of 6.7 W at peak load. However, an NCS (Neural Compute Stick) was used as the deep learning device of the experiment powered by the same powerful and performance features of VPU. Ubuntu 16.04 is installed on a physical x86 64 system, and the Debian Stretch runs on a Raspberry Pi 3 Model B. The Neural Compute SDK comes with C++ and Python (2.7/3.5) APIs. The model has been tested on the Google MobileNets neural network, and the results showed that the RPi3B is able to effectively recognize objects in real time in combination with the embedded deep learning device. On the other hand, another test has been done to compare the performance of RPi3B with the Ubuntu system when they work separately. The results showed that running a separated Ubuntu gives 9.3 frames per second with a single NCS attached, while the RPi3B gives 5.7 FPS. With two sticks, Ubuntu gives 6.6 frames per second, while the RPi3B produces 3.5 frames per second. This positive result sheds light on the ability of edge computing to be used in many critical applications such as self-driving cars or any of the recent robotics applications which do not require human engagement. In the foreseen future, we can expect to have a data deluge transferring through the backhaul network. In this respect, a mobile edge computing network topology with its local computation and storage power is able to deal with such a new situation in IoT networks.

4.4. Simulation Platforms for Evaluating the Performance of Edge Computing

The Castnet framework has been proposed in [33] as a new simulation platform able to evaluate real edge solutions and explore new possible scenarios. The proposed framework aims to basically provide a high level of abstraction through the independent queues that are responding to the requests coming from each entity in the network. This improves response time with respect to the used shared queue by other existing simulators such as iFogSim and ns-3. iFogSim is considered as an extension of the CloudSim framework, and it provides modeling support for an edge infrastructure; it is designed to be used as a single event queue, which makes it inefficient for larger scale simulations. Likewise, ns-3 lacks native abstractions to describe any application-level functionality which is to be carried out along the network paths. Instead, Castnet is a lightweight, extendable, fast, and easy platform to use for edge computing simulations. The high level of abstraction lets Castnet support all entities in an edge computing scenario with different edge infrastructure deployment models and characteristics (latency, computation, and storage). The Castnet abstraction is an incorporation of five levels which form the simulator input:

- Local computational power;
- Edge infrastructure;
- Edge functionality;
- Edge nodes placement;
- Links between nodes in addition to a workload engine able to define the pattern of the incoming request.

Each created request through the network is linked with a timestamp and handled to elements with a processing time that is less than or similar to its timestamp: this distribution helps exploit the limited resources of the network in the best way. The Castnet framework has been implemented in C/C++ using boost and nlohmann json external libraries, while Python has been used for generating the synthetic workload. The first evaluation test of

Castnet was simple; it considered an edge local server as a proxy, receiving requests and forwarding them to another node/server for processing. The results of this test showed similar results between the three simulators Castnet, iFogSim, and ns-3. As a second validation experiment, the authors tested the functionality of the edge caching capacity and its effect on the response time; the test was done on three levels: non-configured cache and 50% and 100% cache capacity. The obtained results showed the following measures for responding time in milliseconds for the three simulators Castnet, iFogSim, and ns-3 in sequence: 5.00, 5.01, and 5.43. As the maximum taken time to handle the request, 95.05, 112.02, and 110.9 were registered for the three simulators, consequently, while for the average needed time, the following results were registered in sequence: 13.63, 17.80, and 16.32.

Another relevant comparison in [33] is related to the number of code lines which is needed to implement an edge application in Castnet compared to iFogSim and ns-3 : 188, 1199, and 714, respectively. The scalability evaluation also has been compared between Castnet and ns-3 while increasing the number of edge nodes from 10 through 100 to 500; iFogSim was not considered in this test because it needs a much longer time. The comparison results proved that ns-3 has a 17.31–44.83% longer time. From all previous experiments, we can confirm that Castnet outperforms the other two simulators even with changing conditions. This poses this new platform as an easy, flexible, and extendable platform for edge computing applications. However, even with all the benefits and flexibility features in the proposed Castnet infrastructure, it is still has some limitations. For instance, it provides a means to address tasks and operations that should be done on the edge, but it is not able to do the automatic division of the application logic between the edge and cloud. Recent research has started to focus on the ability of obtaining benefits from the available resources of edge nodes in the best way in order to achieve the desired insights of the IoT application.

In [34], the authors propose the SaRa model as a probabilistic model to approximate the reliability of edge resources in a volunteer cloud (e.g., cuCloud). It is considered to be one of the recent volunteer solutions of cloud computing where no data center exists, while it relies on multiple dedicated centers. The proposed model assigned a reliability factor to each node in order to represent the trustiness level of this node in providing high quality services. This factor is calculated from two outcomes: the success of VM in satisfying QoS and the failure in terms unsatisfied QoS. The observed results were classified into different classes based on many features such as latency and priority in order to be used later when defining trusted and untrusted nodes. A Google cluster usage trace has been used to test the approach with 100 physical machines, where their highest failure rate was in the range of [5–117] patterns during a life cycle of 29 days. The SaRa model only used the first 26 days of trace data, while the failure rate of the last three days was calculated to validate the estimation accuracy between each compared method. The correlation coefficient has been used to compare the traditional methods (Traditional Reputation, RD Reputation [35], and RBCC [36]) with the newly proposed SaRa model based on the obtained reliability values of the first 26 days by SaRa and the failure rate of the last three days. By considering four different classes of the failure rate, SaRa shows in most of the cases better values of the correlation coefficient compared to the other approaches. The most important achieved feature by this model in distinguishing trusted from untrusted nodes has a great impact on the platform itself since it gives high flexibility to its nodes in joining and leaving the model. However, some other crucial issues should be addressed in order to improve the model performance, such as fault tolerance and the distrust volunteer nodes.

Table 2. Overview of research works cited in Sections 4.4–4.7.

Research Topic	Description	Conclusion
Castnet New Simulation Platform [33]	It provides a high level of abstractions through independent queues responding to the requests	It has some limitations since it is not able to do the automatic division of the application logic between edge and cloud
New Resource Management Technique [37]	It distributes tasks between edge nodes and facilitates privacy constraints	High efficiency in finding the best candidate nodes which minimize end-to-end latency and meet privacy requirement of the application objectives
Theoretical Comparison Model [28]	It decides when processing time in edge-only/hybrid edge-cloud is faster than in cloud architecture	Hybrid mode is always faster for large input
New Data Processing Scheme [38]	Data processing is divided between edge devices and fog nodes to decrease trained data samples and communication cost	It outperforms the centralized one by decreasing latency, reducing labeling and communication costs and reaping possible privacy benefits; however, new efforts should be oriented to study how viable security risks could affect the overall performance of the system if the number of edge devices increases
5G+ Edge Computing [39]	It shows the possibility of integrating IoT, 5G, and cloud computing by conducting a research test-bed and then evaluating the system performance by running mission-critical time sensitive application on an edge cloud platform	Native control loops can work successfully on edge cloud infrastructure without losing the provided benefit by edge or cloud nodes or without affecting system stability; some improvements related to scheduling, synchronizations, and passing messages will have a notable increase in the targeted applications
SaRa Probabilistic Model [34]	It approximates the reliability of edge resources in cuCloud	SaRa shows better values of correlation coefficient compared to the other approaches; it distinguishes trusted from untrusted nodes in cuCloud which gives high flexibility to its nodes in joining and leaving the model
MEC-based Web AR New Solution [40]	It processes the incoming requests and deploys web AR applications; Docker technology was used to dynamically schedule the diversity of web AR applications	It provides better performance; however, web Ar applications still need to balance between benefits and costs, besides acquiring higher efficiency in sustainability issues
New Architecture for Edge Computing [41]	WiFi routers which are located close to edge devices organize how edge nodes collaborate in catching and processing participatory sensing data	It improves the performance and eliminates the possibility of network bottleneck; this causes decreasing the network traffic with obtaining more accurate sensing results

4.5. Advances for the Performance Improvement of Edge Networks

With the rapid integration between IoT and the human physical world, the SDEC-based open edge computing system architecture has been proposed in [32] as an innovative approach to increase the possibility of having a novel edge-computing framework able to utilize the available resources in the best way. The proposed approach uses pooling, virtualization, and abstraction technologies to split software and hardware layers in the traditional edge-computing scenario, which helps cache the complexity of the hardware layer and manage all edge resources and services by the software layer. SDEC is divided into five main parts: SDED, SDESto, SDECR, SDESer, and the SDEC controller.

- SDED virtualizes and abstracts edge physical devices, so it can be considered as the digital copy of the physical devices in the edge network.
- SDESto is a very important model in the proposed infrastructure because it helps manage edge storage capacity and use it in the best manner. This happens by abstracting and pooling edge storage in a virtualized pool, and then fast dynamic mapping can be done between edge storage and application requirements.

- SDECR is responsible for classifying and abstracting all computing devices in the edge network (CPU, RAM, AI chip, etc.), which eases finding the device that matches the computing requirements of a specific application.
- SDEC controller is a central managing and controlling unit where an intelligent managing of the edge resources happens and guarantees edge system flexibility.
- SDESer is the software copy of the edge services' functionality; it abstracts all services in a virtualized pool and enables sharing the physical hardware by different edge services.

Finally, we can summarize SDEC-based open edge computing system architecture into four main parts: the edge devices, the local network that connects edge devices with each other and with the SDEC platform, the SDEC platform, and the smart edge application which is directed to end users. This approach provides an extendable, automatic, and more flexible platform compared to the traditional edge computing ones, and it enables the management of all edge resources in the best way in order to improve the performance and reduce energy consumption. However, the need for an updated scenario targeting the software models and updating their characteristics automatically is one of the main challenges of this approach since the manual updates for these software definition models are almost impossible due to the number and type of objects involved in the SDEC paradigm.

An innovative scheme has been proposed in [38], where the tasks for processing data are divided between edge devices and fog nodes, aiming to decrease the trained data samples and communication cost. This scheme creates the initial training model by applying federated learning in the middleware fog devices using a specific number of data samples. Afterwards, the centralized fog node publishes the model to all edge devices where active learning is applied locally and separately using the maximal entropy function. As a result, new models are sent back to the fog node from the edge, noting that the number of these models is equal to the number of edge devices. Then, weights aggregation is done by choosing the best training model or by taking their average, which is considered as the input for the next round. This new method retains users' personal privacy for further data analysis processes by applying the active learning locally on the edge and decreasing communication costs thanks to federation learning, which is usually applied on the fog centralized node. A CNN (convolutional neural network) is used as a model for the above-mentioned training purposes, while all experimental methods are implemented in Python on the MNIST dataset, which counts 6000 handwritten images representing the training dataset and 1000 handwritten images representing the test dataset. Their experiments were applied on two levels: the first concerns the overall performance when applying active learning on edge devices compared to choosing data randomly, whereas the second experiment is aimed at demonstrating the overall performance while increasing the number of training data. The obtained results show that the proposed approach outperforms the centralized one by decreasing latency and reducing labeling and communication costs, possibly yielding privacy benefits. With the important potential benefits of this proposal, new efforts should be oriented to increase the number of edge devices used within the network and to study how the viable security risks could affect the overall performance of the system.

In the context of facing bandwidth and latency challenges in the existing implementations of web augmented reality (AR), a MEC-based web AR solution has been proposed in [40]. System components can be summarized as the following: the MEC platform is used to process the incoming requests and deploy web AR applications, while an abstract service layer for the web AR with AR cache forms the running environment of the web AR. However, the Docker technology has been used to dynamically schedule the diversity of web AR applications. Experiments were done by using a MI MIX 2 mobile phone 10 times, where frame sizes were 250×250 pixels, weighing on average 25.04 KB. The proposed architecture was compared with two existing solutions: cloud computing solutions and pure front-end solutions. The comparisons regarded three levels: latency, FPS, and power consumption. The obtained results show better performance in the new approach compared to other

conventional solutions. However, web AR applications still need to balance benefits and costs, in addition to acquiring a higher efficiency in terms of sustainability issues.

In [41], the authors proposed an innovative architecture for edge computing in which WiFi routers are located close to edge devices, and they organize how edge nodes collaborate in catching and processing participatory sensing data. The presented system architecture in this work consists of a cloud-based main coordination server which sends the requirements and task directives to WiFi routers geographically distributed within the network. Each router distributes the execution of a set of micro-services between the nodes which are located in its area. At the end, edge devices execute the assigned tasks by downloading the needed execution packages from the cloud repository and later return the output to the router. An indoor navigation map use case has been used to evaluate the previous system architecture in which navigation output results are provided in pairs in a form (name of room's occupant, room number). As a result, a participatory sensing task is created to collect this information, which is basically a scenario of image recognition. Five main phases have been considered in the previous approach to execute the micro-service:

- Taking photos;
- Image content recognition;
- Verification of recognized data;
- Recognition model update;
- Distribution to participants' devices.

They evaluated the energy consumption carried out by the new architecture for multiple micro-services, in which energy consumption was registered as 949 μAh for taking photos, 80 μAh for image pre-processing, 23 μAh for data recognition, and 522 μAh for verifying photos. Evaluation results of this system architecture show that it is able to improve the performance of the network and to eliminate the possibility of network bottleneck when huge amounts of data are transferred or when we have mobile devices with a limited battery. This causes a reduction in network traffic, while achieving more accurate sensing results. From the initial results of this proposal, we think that it is going to be a promising solution, especially when dealing with rich data such as photos and videos.

Moreover, in [37], the authors propose and test a resource management technique which has two main objectives: task distribution between edge nodes, which helps save energy and ease privacy constraints by enabling each device in the network to follow its own privacy roles and its availability. Based on this proposal, the main complete objective of the IoT application is divided into many sub-tasks where each sub-task can be executed on a different node. This increases the feasibility and practicality of the application. In order to cover the dependencies between tasks, the application is represented as a directed acyclic graph, where the nodes are the application tasks, while the edges represent the dependencies between them. However, the resource management technical platform follows an auction house functionality and has two main models: the policy module and the bidding policy module. When the application is ready for execution, bidding nodes are created where each node has its own available resources. Bidding nodes connect to the main deployment node with a communication link named the dispatcher. The responsibility of the policy module can be summarized in mapping the objective of a specific sub-task with offers provided by bidding nodes, while keeping the end-to-end latency as short as possible in choosing the best offers. Additionally, the bidding policy module works to choose the best candidate offer based on the privacy requirements built by the node itself. Their approach is tested on a montage graph from the real world and run on a single core Intel i5 2.3 GHz, while changing the range of divided sub-tasks from 24 to 50 and the range of resource constraints between 1 and 10. Testing results show efficiency at finding the best candidate nodes that minimize end-to-end latency and meet the privacy requirement of application objectives.

4.6. Edge Computing over 5G/6G Networks

In [39], the authors attempt to show the possibility of integrating IoT networks, 5G, and cloud computing by conducting a research test-bed, and then they evaluate the system performance by running a mission-critical time-sensitive application on an edge cloud platform. A 5G wireless network is one of the most important components of the system; it was implemented using a LuMaMi test-bed, while many computing nodes were connected and joined by this network. Four different types of nodes were used to carry out the computation tasks in this network: a Raspberry Pi 3Bs for representing the plant (mechanical device which is continuously performing a specific task), an Intel Core i7 desktop for representing edge nodes, an Intel Core i7 VM for representing ERDC, and an Intel Xeon VM for representing AWS. The two VMs of AWS and ERDC were connected to the subnet over a VPN, which facilitated the direct access between computing nodes. However, Calvin was used to represent the cloud platform. For evaluating the system performance, the ball and beam was the control process of the test-bed research, and it was considered as a time-sensitive mission-critical application. They tested system characteristics, overall performance, and the controller's ability to continuously work successfully. Evaluation results show that the native control loops which have been applied on the cloud platform can work successfully on the edge cloud infrastructure with the possibility of changing the controller location without losing the benefit provided by edge or cloud nodes or without affecting system stability. Some improvements related to scheduling, synchronizations, and passing messages will have a notable increase in the targeted applications by having high adaptive, user-friendly, low jitter, and low latency applications.

In [42], an innovative architecture aimed at providing on-demand services in terms of communication, computation, and caching in 3D space anytime and anywhere is proposed, relying on tight integration between conventional ground base stations and flying nodes. This is particularly relevant for 6G, since the sixth generation standard will exploit terrestrial and non-terrestrial (e.g., satellite and aerial) platforms jointly, with the aim of improving radio access capability and unlocking the capability of offering cloud services in 3D space on demand, namely through the incorporation of mobile edge computing (MEC) functions on non-terrestrial platforms (e.g., aerial vehicles and low-orbit satellites). This will allow the extension of MEC support to devices and network elements in the sky, thus forging a space-borne MEC that enables artificial-intelligence-driven, customized, and distributed on-demand services. In such a way, the end users eventually experience the impression of being surrounded by a distributed computer, capable of fulfilling all requests with nearly zero latency. For this to happen, it is of paramount importance that resources are jointly orchestrated by relying on artificial-intelligence-based algorithms, as well as by exploiting virtualized network functions that are dynamically deployed in a distributed way across terrestrial and non-terrestrial nodes.

4.7. Edge Computing for Industrial Internet of Things

Edge computing for industrial internet of things (IIoT) has the potential to transform the energy industry, through its ability to process large amounts of information in real time, and ultimately improve the safety and efficiency of operations.

The growth of IoT devices has multiplied by millions the amount of data that can and must be processed by enterprises in their data transformation process. In order to make this process more efficient and effective, edge computing allows for complementing the processing capabilities of centralized cloud infrastructures with machine learning and artificial intelligence algorithms being processed at the edge, i.e., at the node which the relevant data have originated from and which is closer to users or devices. For oil and gas, power utilities, and other energy players across the value chain, edge computing can enhance production capabilities, improve processes, extend asset life, and create numerous opportunities for the deployment of additional capabilities.

IIoT is a crucial research field induced by the internet of things (IoT). IIoT links all types of industrial equipment through the network, establishing data acquisition, exchange,

and analysis systems, as well as optimizing processes and services, in such a way as to reduce cost and enhance productivity. The combination of edge computing in IIoT can drastically reduce the decision-making latency, avoid wasting bandwidth resources, and, to some extent, protect privacy.

In more detail, IIoT edge computing is particularly relevant in the following contexts.

- Oil and gas distribution infrastructures: in this domain, one day of downtime due to a failure can cost more than 20 million dollars, and the average occurrence of such downtime events is five times per year for large operators. In this respect, IIoT edge computing makes it possible to analyze the data in real time to avoid problems in advance, or else to identify their causes much more quickly. All this is achieved with the highest possible level of security.
- Electrical substations: especially in medium to low voltage scenarios, the concept of the smart transformer is arising, which, in addition to being connected, enables real-time dynamic regulation of the power supply to the different lines, which, currently, elements such as electrical chargers or batteries are depending on. In this respect, IIoT edge computing provides adjustments in real time, thus preventing failures and avoiding unnecessary displacements as well as generating new services that can increase the return on investment of the entire value chain.
- Consumption points: 2020 was an unprecedented year in terms of energy self-consumption. In Spain alone, 596 megawatts were installed, 30% more than in 2019. Through IIoT edge computing, and accompanied by sensors that can measure production or storage conditions or smart actuators (relays) that can monitor and control consumption, energy savings can be raised by double-digit figures.
- Renewables: IIoT edge computing offers the possibility to design and implement on-edge systems for predictive maintenance of photovoltaic plants, in particular allowing for (i) the maximization of energy production while flexibly and efficiently managing the plant throughout its lifecycle and in compliance with security and safety standards, and (ii) the exploitation of insightful analytics tools based on big data and multi-sensor platforms—for further detail in this respect, the interested reader is referred to the solution proposed in [43].

IoT applications have been revealing the potential advantages in shifting a decisive part of the logic from the cloud to the edge over the last few years. Shi et al. [44] wrote a survey paper about the vision and possible advantages of distributing logic across the network but also mentioned the relevant challenges that need to be faced.

It is necessary to point out that edge computing brings additional computation to the edge as a consequence, which can create some overhead [45]. Yet, the only practical case study measuring the exact overhead caused by the edge computing paradigm, and to which extent it influences the battery life of IoT devices, is provided in [46], which analyzes the impact of edge computing on the energy consumption of IoT devices.

In [47], the authors discuss the vital role that MEC will play at enabling IIoT applications, and in [48], a typical application scenario is described, specifically discussing the benefits introduced by MEC in automotive use cases.

In [49], a thorough discussion of the virtualization opportunities at the network edge is provided, proposing a container-based NFV platform that runs and orchestrates lightweight container VNFs, with three application IIoT examples. In addition, in [13], a 5G-enabled software-defined vehicular network is proposed as a relevant example of IIoT combining software-defined networking with MEC.

More generally, from the cited works, it emerges that the edge computing reference architecture in IIoT is divided into the cloud layer and edge layer, and the edge layer can also be subdivided into near-edge, mid-edge, and far-edge. For the advanced technologies of edge computing in IIoT, the most challenging tasks are to be regarded in terms of routing, task scheduling, data storage and analytics, security, and standardization. Relative to IIoT, the combination of edge computing with blockchain, machine learning, SDN, and 5G will become an obvious trend.

5. Discussion

This Section is dedicated to summarizing the main insights emerging from the review of the literature provided in Section 4. It is clear that the paradigm of mobile edge computing is emerging as an ETSI-defined network architecture concept, allowing cloud computing capabilities and an IT service environment to shift to the edge of cellular networks and, more generally, to the edge of any network. The idea behind this is that by running applications and performing the related processing tasks closer to the end user, network congestion is drastically reduced and applications perform significantly better. In this context, the optimal placement of servers allows for obtaining a dramatic reduction in power consumption and the improved management of data traffic, even up to a dozen of percentage points.

As concerns the security of edge computing, this is one of the hottest research issues. Network edge data imply personal privacy, and the distributed architecture of edge computing increases the dimension of attack vectors: even though edge computing clients are smarter, they are more vulnerable. Hence, there are emerging challenges and dedicated solutions in terms of (i) lightweight data encryption and fine-grained data sharing based on multiple authorized parties; (ii) security management in multi-source heterogeneous data propagation; (iii) security management for large-scale interconnected services and resource constrained terminals, especially at the superposition of mobile and internet networks when it comes to coping with the resource limitations of storage, computing, and battery capacity of edge terminals; and (iv) privacy protection when deploying IoT services.

In addition to the advantages due to the distributed nature of the edge computing paradigm, there are relevant benefits, even to adopting hybrid edge-cloud computing since there is almost always the need for at least some of the edge data to be pushed to a centralized or cloud-based system for overall monitoring and management. This way, even the most autonomous edge systems can be monitored for correct operation and have means to be adapted dynamically, either by human- or machine-based intervention. In this respect, a two-way data flow is required to support a hybrid edge-cloud architecture seamlessly.

Several platforms, such as Castnet and SaRa, are being proposed as simulators and can therefore be used for benchmarking purposes.

Meanwhile, innovative schemes and architectures (such as SDEC and others) are currently being studied in order to push the performance of edge networks to the limit, even exploiting the potentialities of augmented reality and artificial intelligence, as well as to employ them in the 5G/6G domain.

Finally, the most promising application scenario for edge computing is certainly that of industrial internet of things, where mobile edge and hybrid edge-cloud architecture are proving successful with respect to oil and gas distribution use cases, as well as electrical consumption and renewable generation.

6. Businesses Taking Operations at the Edge

Many companies have started to expand their business by bringing edge technology to the market in order to increase reliability, quality, and speed. Dell [50] designed multiple micro modular data center solutions (MDSs) to increase their business capabilities; more specifically, 4-series micro modular data centers increase the computation power of the edge since they support 17U storage and computing in addition to their ability to work in different weather conditions (either extremely cold or hot) without affecting the quality of the provided services. On the other hand, Dell EMC DSS 8440 is a newly designed micro data center, which aims to accelerate the process of machine learning algorithms executed on the edge for providing faster results. Additionally, Dell EMC DSS 7000 is a storage server able to meet and handle the increasing storage requirements of businesses because it is able to store 10.8 petabytes on every 42U rack, and this makes it a good choice for huge storing environments.

Moreover, and with the continuous diffusion of IoT applications, it is highly needed to have fast, predictable, and simple IoT implementation solutions. Hewlett Packard

Enterprise, in this respect, devised a new edge solution [51] named HPE Edgeline EL8000 Converged Edge System, capable of speeding up data transfer through the backhaul network since it is supported by a 10G SFP+ data plane, in addition to its capabilities in terms of providing a higher level of flexibility to IoT applications through its RJ45/SFP+ connections in the transferring network. This multiplex server proves its high performance and quality for video analytics, IoT, and data streaming. However, its main component (HPE Edgeline Chassis Manager) carries out monitoring and alarming tasks to illuminate the off-time of the installed servers. It is considered as a multiple-purpose system thanks to its scaling flexibility in changing and meeting the considered use case requirements. HPE Edgeline Converged Edge Systems feature high storage capability due to the eight existing drives where data can be stored in addition to their ability to provide high efficiency even in a high density workload. HPE edge systems offer NEBS certifications, and this improves the quality of the provided services in terms of reliability, efficiency, security, or user-confidence.

The XR2 server is one of the latest digital solutions of Dell [52] where many technologies are integrated to face data explosion challenges on the edge. Faster performance, bigger storage, and higher capacity can be achieved using this server thanks to adding GPU support and to activating the parallel computation power. It is 20% less deep than the traditionally used servers, which makes it a good flexible and powerful solution taking advantage of the latest Intel® Cascade Lake processors, $8 \times 2.5'$ processor chassis, and four supportive NVMe drives. It is able to take advantage of many existing infrastructures on the edge, and it is also different from the conventional AC power solutions due to the XR2 server rugged server DC power. This new technological solution has been tested in different environments where fire, water, humidity, and other conditions exist, and it proved that it is able to face the massive data workload that needs to be processed and analyzed in real time on the edge regardless of the occurring conditions. The main goal of this product is to guarantee that the system works well, whatever the environment it is in. This digital solution can be considered as one of the best existing ones, especially when there is a high workload in the location and outside the data center. It guarantees that the system works in all environments and with high performance. In this case, customers do not need to worry about the system operations on the edge.

Apple, in turn, has started to use their own ARM chipset [53] instead of the current Intel one in order to increase the performance and decrease the costs. As they created their own OS instead of relying on Microsoft to optimize the operating system, they will be independent from Intel by using a custom ARM chipset. Although the currently used processors by Apple in their devices such as the iPad Pro are 22% faster than the existing ones on the market, it is expected that the upcoming new chipset is going to be more powerful and able to provide better graphic and processing performance. With its smaller size, it is expected to have thinner and lighter devices launched by Apple in which this chipset is installed, leading to higher flexibility. During Apple's annual Worldwide Developers Conference (WWDC) [54], they mentioned that this shifting to their own chip will provide new features, and it will enable more powerful applications run by Apple devices, in addition to having higher control over the software and hardware and a longer life for batteries. Many difficulties are expected to be faced until the final launching, but this step is one of the articulated stops in Apple history for many reasons, the most important among them being the achievement of full independence from Intel.

Voice processing is one of the most implemented operations in edge applications especially in smart homes. The Alexa edge device [55] enables controlling edge home devices easily by speaking. It forms an efficient power approach since it helps to define the right time to activate the usage of edge devices: this has a great effect on mobile battery-operated devices such as smart phones. Alexa uses a DSP ("digital signal processor") to define alarming words pronounced by customers to execute some instructions. When the processor detects one of these words, such as Alexa, it activates all system functionality (language processing, audio recording, etc.) instead of recording voices all the time and sending

all recorded data to the cloud. This saves energy and optimizes the usage of network capabilities and resources. Edge devices also run many voice tracking algorithms in order to separate the origin voice from the noise, resulting in a highly accurate interpretation for the incoming voice commands. It sends texts to the cloud instead of voice, which improves the response time. In addition, its offline work ability increases the quality of the provided services. The Alexa edge device maintains customer privacy thanks to its local voice recognition service, which avoids sending all recorded data to the cloud. Starting from the previously mentioned use cases and taking into consideration the constant need of having real-time and local response, the increasingly important role and the upcoming wide spread of edge technology in all business fields in the near future become clear.

In the industry, Microsoft is considered as one of the most thorough platforms of edge computing. The Azure IoT edge launched by Microsoft combines the container modules with a cloud computing management interface. However, at the end of March 2020 [56], Microsoft started to preview a new advanced solution by activating three new services: Edge Zones, Private Edge Zones, and Edge Zones with Carrier. In the industrial field, Azure zones are connected through a private 5G network, in this way creating a small local area closer to end users and providing cloud services inside the porter network. The available computation amount of data and connectivity will be different based on the handled service, but for developers, it is going to be the experience of traditional Azure. Microsoft's newly designed solutions will guarantee security and deference services, and they will help decrease the time cost of the process since they are trying to distribute the workload automatically. In the future, Microsoft aims to integrate more of their APIs into the carriers where much of the information is their own (location, QoS, latency), in addition to integrating 5G slicing into the platform.

Multiple tools are also offered by Amazon to facilitate working with edge computing, such as activating responding to events using a serverless lambda function [57]. The lambda function is also known as the function-as-a-service platform where developing, executing, and managing functions are available for the applications locally and closer to end users in serverless mode. Using the lambda function with other AWS allows developers to build customized, high performance edge applications since the user needs to upload its code in the lambda function, which will carry out the responsibility of running it globally close to the application's end users, thus resulting in a lower latency and faster response. Moreover, it increases the powerful level of edge applications due to the automatic running and scaling of the code, and it eliminates the need for managing the underlying infrastructure. The Amazon lambda function responds to the events generated by the Amazon CloudFront content delivery network (CDN) and runs the application code in the closest AWS region to the place where one is running one's code from; it also offers testing lambda function code locally with the CloudWatch feature. As a result, AWS is able to provide higher quality running, scaling, and testing for edge applications' codes, in addition to obtaining insights into edge applications faster.

FogHorn is a lightning edge platform aiming at increasing the intelligence for IoT deployments on the edge. It shows an unrivaled fast response and low latency [58] due to the migration of complex event processor (CEP) engines for data analysis with the capability of machine learning algorithms in a single software. This software offers recognizing events and activates closed-loop control for real-time data analysis and processing. It performs complex pattern recognition on the pulled data from sensors and addresses all the steps of machine learning algorithms (cleaning, filtering, normalization, standardization, etc.) in order to increase its performance in the next experiments with other future events. Shifting data analysis and machine learning power to the edge increases the intelligence level of the services that are provided, in addition to activating a new class of applications and use cases that can be deployed on the edge for monitoring, maintenance, optimization, and AI processes [59].

All edge applications in the industry should consider management, security, and compliance risks, in particular with the current revolution of IoT and with the distribution

of millions of sensors and cameras in our cities. As a result, a management platform is proposed by IBM [60] to face previous challenges. The autonomous and remote management capability of this platform helps reduce cost in addition to its ability to manage all edge end points from a single place with RedHat OpenShift. IBM edge manager represents the central administrative point which manages all edge devices by considering the four main elements of identification, agreement, execution, and verification of the management actions, in this way simplifying the traditional management scenario. The IBM edge manager application can provide direct services to the edge end devices, decrease response time, and distribute the workload over edge end points, thereby avoiding the bottleneck challenge which usually happens in the central cloud infrastructure.

This application gives the ability to detect the criteria of what should be deployed on each device. By considering these criteria with the management policy, it guarantees that the right program works on the right node and at the right time. Dell edge computing platforms provide superior results with high performance in edge applications thanks to their newly launched technological solutions, either micro data servers or IoT gateways, which offer a secure, flexible, manageable, scalable, and open infrastructure [61], allowing the customer to easily control and react to events in the physical world. Their provided infrastructure helps deal perfectly with increasing business demands, especially in data analysis and machine learning applications, in addition to its high efficiency of understanding most of the operations that are happening on the edge due to the capabilities it has in streaming data analytics. The administrator-attended time in edge applications can also be reduced by up to 99.1% [62] per server by using Dell technological solutions.

Moreover, Hewlett Packard Enterprise (HPE) has invested a lot in the sector of edge computing and smart services. They aim to have a single IT system that fits all edge harsh environments in order to guarantee edge capabilities in responding to end users' requests in all conditions [63]. Their target is to engage end users much more in the provided smart services by acquiring more data from them in order to speed up their business on the edge. For example, the HPE Edgeline EL8000 delivers superior density, weight, power, computation, cost, and cooling. Their infrastructure has very good computation, storage, aggregation, filtering, and analysis of data, including video streaming. It also features an ability to translate industrial protocols [64].

Rigado can be considered as a very effective solution for edge applications due to the secure connection it offers by using Cascade routers in addition to performance monitoring. It uses Ubuntu Snaps for managing the deployed applications on the edge in addition to pre-configured sensors and devices. The Rigado Cascade Edge-as-a-Service solution is applicable in commercial buildings and spaces such as smart offices, connected retail, and intelligent logistics, and it forms a promising solution for developing solutions for connected hospitality quickly and in a reliable and secure way [65]. On the other hand, edge computing has been considered very seriously by Cisco, which offers access points and routers for security and management responsibilities on the edge, in addition to its continuous progress in providing solutions which are able to accelerate the translation of data insights into decision making. Its trusted platform has very high quality for the provided services, low latency response, and relevant capabilities in terms of breaking down the complexity of network overlays [66]. This will result in improving the performance of an application life cycle management and in enhancing the business user experience.

ClearBlade also offers a core middleware platform for IoT on the edge [67] and provides reliable, secure, and fast IoT solutions. It enables the deployment of any IoT solution easily and regardless of whether it is simple or complicated and without being affected by the used internet protocol. Its ability to continue working in offline mode and to provide real-time data analysis services is a very attractive feature for edge time-sensitive applications, in addition to many other features such as data filtering and streaming, powerful integration capabilities, sync and state management, and security and adaptive deployment. All the previous reasons make this platform a very good candidate for edge computing applications. However, developing, deploying, managing, and automating edge cloud

platforms are all guaranteed with Saguna MEC solutions. It supports a 5G connection over the existing 4G one in order to provide reliable and low latency communication [68]. It is a fully virtualized platform which relies on Intel® Xeon® processors and Linux-based* operating systems, in this way providing an open environment for running third-party MEC applications. Its MEC server has broadband delivery, application registration, traffic steering, radio insights, and DNS caching features, in addition to a MEC gateway which operates as a virtual machine (VM) running on a server to ensure user mobility and to preserve core network functionality. The testing results of this solution showed an improvement in the user experience, reduction in the cost of data delivery and extension to new markets such as IoT, and the possibility of offering new revenue.

To sum up, Figures 4 and 5 sketch more clearly how relevant businesses have been taking operations at the edge, as described above.

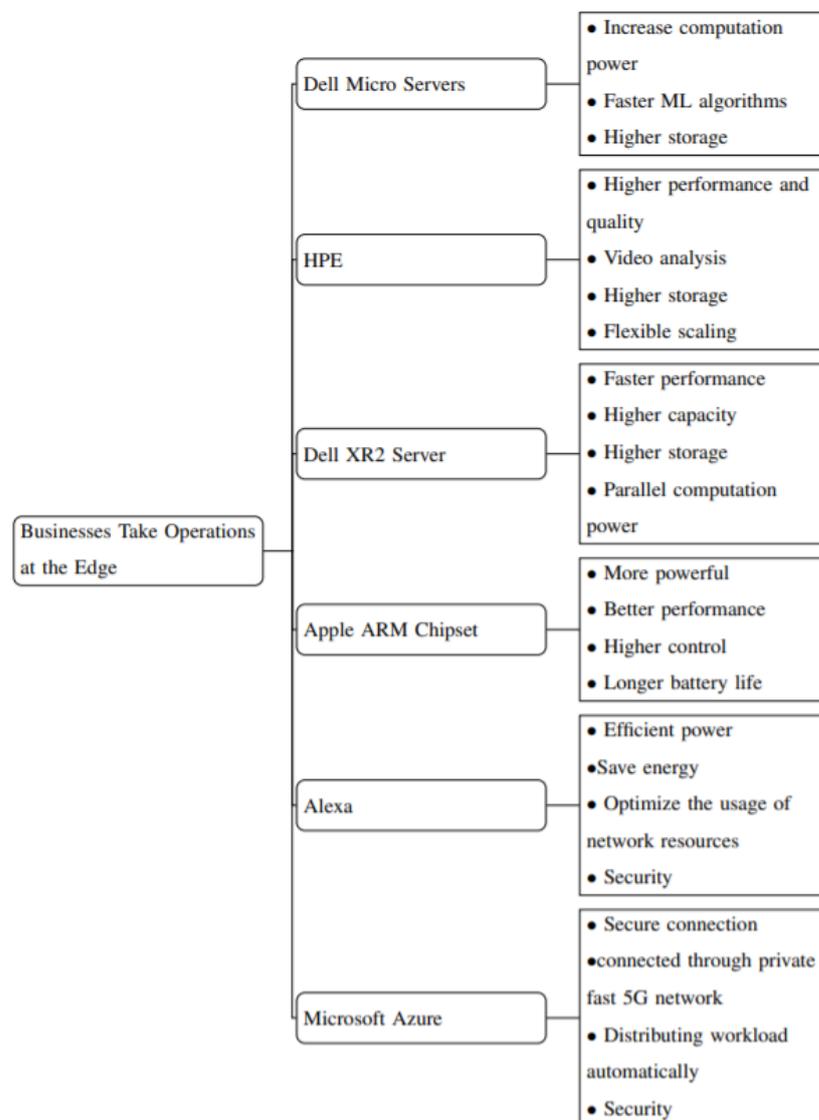


Figure 4. Businesses taking operations at the edge: Amazon, Apple, Dell, Hewlett Packard, Microsoft [50–56].

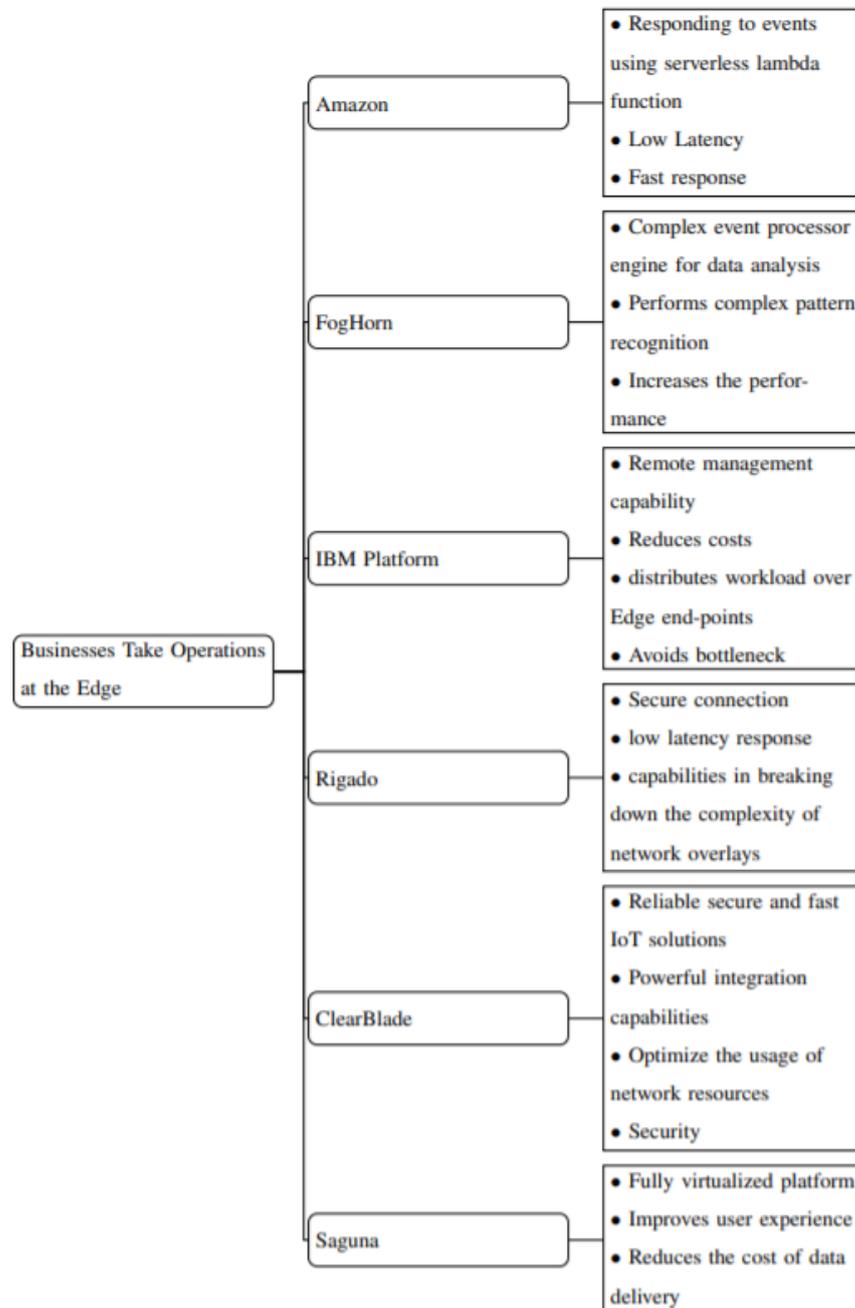


Figure 5. Businesses taking operations at the edge: Amazon, ClearBlade, FogHorn, IBM, Rigado, Saguna [57–60,65,67,68].

In this subsection, we provide a brief review of the the main most used open source edge computing platforms. Previously, in Section 6, only proprietary edge computing platforms were discussed (as shown in Figures 4 and 5).

Towards the serverless computational platforms, many new system architectures have been proposed in which developers are concerned about coding tasks without thinking if the server is actually running the code [69].

- **OpenFaas:** a framework that provides serverless services on top of container platforms [70]. Due to the possibility of packaging the code in a Docker/OCI image, it provides developers with a high flexibility for writing their code in any programming language and for any operating system. OpenFaas is portable to any cloud, and it

allows developers to easily access the functionality elements thanks to user-friendly interfaces. The main components of OpenFaas are an API Gateway which is considered as a route for the provided functions, in addition to its role in scaling services, and a Watchdog function for handling function monitoring, noting that each created container has its own Watchdog function.

- **Knative:** created by Google to run serverless services on Kubernetes. The latter is an open source system for deploying, managing, and scaling centralized applications. It uses master-worker architecture, where the master node detects all management APIs, while running functionalities happen on the worker nodes separately. Later, all running results are reported back to the Kubernetes master [71]. Three main components are provided by Knative: build, serving, and events [72]. The build component allows developers to define the way of compiling, packaging, and building their code. Serving allows code deploying as a container image running onto the Kubernetes worker node, while eventing handles an event-driven model, and it determines the objects of each event.
- **Kubeless:** an open source platform built on top of Kubernetes [73]. It also allows for implementing sophisticated code with Faas on top of Kubernetes. There are three fundamental components: functions to be executed, considering that each function is linked with some events; triggers, which form the bridge between procedures and function execution; and the required running environment of the handling code. Kubeless reduces the operational costs and separates the infrastructure from the application code that is running on it; this makes managing the provided services much easier.
- **Fission:** provides the possibility of creating Faas applications by provisioning definition files and without creating any container. The controller is a key module in Fission architecture; it is responsible for changing user operations into Kubernetes operations. It also supports two executors: PoolManager and NewDeploy. PoolManager is a pool of general and functional containers. However, it is a single run-time instance for a given function; therefore, it is not suitable for serving massive traffic. NewDeploy solves this problem since it supports numerous function instances [74].
- **Nuclio:** an open source serverless platform that aims to minimize the workload heaviness and automatically deploy data-science-based applications. It speeds up running functions since its own nuclio processor ecosystem basically consists of event-source listeners for fetching events and data sources, a run-time engine which initializes the suitable running environment, data bindings providing the possibility of obtaining benefits from external necessary files, and the controller, which is the component responsible for managing and monitoring all processor components and features [75].

7. Conclusions and Future Work

Edge computing is a promising technology useful for facing many daily challenges of smart services and real-time sensitive applications. In this paper, we present a comprehensive survey discussing edge computing technology from the recent research topics to demonstrate how numerous companies are currently trying to shift their businesses to the edge in order to increase their business performance and meet their customers' expectations as successfully as possible. Namely, this review article is structured along two main different lines: (1) edge technology in research and (2) edge computing in the business sector with a specific focus on the main technological platforms. In terms of recent research results, the authors found that a comprehensive survey was still not available relative to the topics of optimal server placement, data security, hybrid edge-cloud, simulation platforms, approaches for improving edge networks, and the relevant applications of edge computing to 5G/6G networks, as well as to the domain of industrial internet of things. Hence, the authors report the main achievements with respect to the above-mentioned research topics, representing the current direction along which innovative solutions are being developed and deployed.

In the near future, with the continuous development of the internet, 5G/6G networks, and industrial internet of things applications, edge computing will play a very important role and promote the development of various industries. In particular, its success will critically depend on its capability to offer decision support to business and to exploit the virtues of hybrid edge-cloud architectures, which are capable of coping with both small-scale intelligent analysis in real-time and large-scale centralized processing.

Author Contributions: Conceptualization, N.A.S., L.R.C., W.L., A.Z. and M.V.; methodology, N.A.S., L.R.C., W.L., A.Z. and M.V.; investigation, N.A.S. and L.R.C.; writing—original draft preparation, N.A.S. and L.R.C.; writing—review and editing, N.A.S. and L.R.C.; visualization, N.A.S. and L.R.C.; supervision, L.R.C.; project administration, L.R.C.; funding acquisition, L.R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by ELIS Innovation Hub within a Joint Research Project with Enel Green Power S.p.A.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Sapienza, M.; Guardo, E.; Cavallo, M.; La Torre, G.; Leombruno, G.; Tomarchio, O. Solving critical events through mobile edge computing: An approach for smart cities. In Proceedings of the 2016 IEEE International Conference on Smart Computing (SMARTCOMP), St Louis, MO, USA, 18–20 May 2016; pp. 1–5.
- Cao, H. Developing an Analytics Everywhere Framework for the Internet of Things in Smart City Applications. 2019. Available online: https://www.researchgate.net/publication/337851792_Developing_an_Analytics_Everywhere_Framework_for_the_Internet_of_Things_in_Smart_City_Applications (accessed on 8 June 2021).
- Gagliardi, G.; Casavola, A.; Lupia, M.; Cario, G.; Tedesco, F.; Scudo, F.L.; Augimeri, A. A smart city adaptive lighting system. In Proceedings of the 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, Spain, 23–26 April 2018; pp. 258–263.
- Kakderi, C.; Komninos, N.; Tsarchopoulos, P. Smart cities and cloud computing: Lessons from the STORM CLOUDS experiment. *J. Smart Cities* **2019**, *2*, 4–13. [\[CrossRef\]](#)
- Khan, L.U.; Yaqoob, I.; Tran, N.H.; Kazmi, S.A.; Dang, T.N.; Hong, C.S. Edge-Computing-Enabled Smart Cities: A Comprehensive Survey. *IEEE Internet Things J.* **2020**, *7*, 10200–10232. [\[CrossRef\]](#)
- Liu, Y.; Yang, C.; Jiang, L.; Xie, S.; Zhang, Y. Intelligent edge computing for IoT-based energy management in smart cities. *IEEE Netw.* **2019**, *33*, 111–117. [\[CrossRef\]](#)
- Nastic, S.; Rausch, T.; Scekic, O.; Dustdar, S.; Gusev, M.; Koteska, B.; Prodan, R. A serverless real-time data analytics platform for edge computing. *IEEE Internet Comput.* **2017**, *2*, 64–71. [\[CrossRef\]](#)
- Taleb, T.; Dutta, S.; Ksentini, A.; Iqbal, M.; Flinck, H. Mobile edge computing potential in making cities smarter. *IEEE Commun. Mag.* **2017**, *55*, 38–43. [\[CrossRef\]](#)
- Wang, X.; Han, Y.; Leung, V.C.; Niyato, D.; Yan, X.; Chen, X. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [\[CrossRef\]](#)
- Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [\[CrossRef\]](#)
- Yousefpour, A.; Devic, S.; Nguyen, B.Q.; Kreidieh, A.; Liao, A.; Bayen, A.M.; Jue, J.P. Guardians of the Deep Fog: Failure-Resilient DNN Inference from Edge to Cloud. In Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things, New York, NY, USA, 10–13 November 2019; pp. 25–31.
- Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the edge computing for the Internet of Things. *IEEE Access* **2017**, *6*, 6900–6919. [\[CrossRef\]](#)
- Huang, X.; Yu, R.; Kang, J.; He, Y.; Zhang, Y. Exploring mobile edge computing for 5G-enabled software defined vehicular networks. *IEEE Wirel. Commun.* **2017**, *24*, 55–63. [\[CrossRef\]](#)
- Carlini, S. The Drivers and Benefits of Edge Computing. Schneider Electric—Data Center Science Center. 2016; Volume 8. Available online: <https://resources.enterprisetalk.com/ebook/49459-Schneider-Electric-EN-1.pdf> (accessed on 10 July 2021).
- Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.H.; Konwinski, A.; Zaharia, M. *Above the Clouds: A Berkeley View of Cloud Computing*; EECS Department, University of California: Berkeley, CA, USA, 2009.
- Sakhdari, J.; Izadpanah, S.; Zolfaghari, B.; Rahati-Quchani, M.; Shadi, M.; Abrishami, S.; Rasoolzadegan, A. Edge Computing: A Systematic Mapping Study. *arXiv* **2021**, arXiv:2102.02720.

17. GVR Report CoverEdge Computing Market Size, Share & Trends Report Edge Computing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services, Edge-managed Platforms), By Application, By Industry Vertical, By Region, And Segment Forecasts, 2021–2028. Available online: <https://www.grandviewresearch.com/industry-analysis/edge-computing-market/> (accessed on 5 July 2021).
18. Bilal, K.; Khalid, O.; Erbad, A.; Khan, S.U. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Comput. Netw.* **2018**, *130*, 94–120. [CrossRef]
19. Li, Y.; Wang, S. An energy-aware edge server placement algorithm in mobile edge computing. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 66–73.
20. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2017**, *5*, 450–465. [CrossRef]
21. Maia, A.M.; Ghamri-Doudane, Y.; Vieira, D.; de Castro, M.F. Optimized placement of scalable iot services in edge computing. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Washington, DC, USA, 8–12 April 2019; pp. 189–197.
22. Xiao, K.; Gao, Z.; Wang, Q.; Yang, Y. A heuristic algorithm based on resource requirements forecasting for server placement in edge computing. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 354–355.
23. Personè, V.D.N.; Grassi, V. Architectural issues for self-adaptive service migration management in mobile edge computing scenarios. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 27–29.
24. Fan, K.; Pan, Q.; Wang, J.; Liu, T.; Li, H.; Yang, Y. Cross-domain based data sharing scheme in cooperative edge computing. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 87–92.
25. Caprolu, M.; Di Pietro, R.; Lombardi, F.; Raponi, S. Edge computing perspectives: Architectures, technologies, and open security issues. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 116–123.
26. Alrowaily, M.; Lu, Z. Secure edge computing in iot systems: Review and case studies. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 440–444.
27. Giang, N.K.; Lea, R.; Blackstock, M.; Leung, V.C. Fog at the edge: Experiences building an edge computing platform. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 9–16.
28. Loghin, D.; Ramapantulu, L.; Teo, Y.M. Towards analyzing the performance of hybrid edge-cloud processing. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 87–94.
29. Ozcan, M.O.; Odaci, F.; Ari, I. Remote Debugging for Containerized Applications in Edge Computing Environments. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 30–32.
30. Li, X.; Ding, R.; Liu, X.; Yan, W.; Xu, J.; Gao, H.; Zheng, X. Comec: Computation offloading for video-based heart rate detection app in mobile edge computing. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 11–13 December 2018; pp. 1038–1039. Available online: <https://ieeexplore.ieee.org/document/8672279> (accessed on 10 August 2021).
31. Hochstetler, J.; Padidela, R.; Chen, Q.; Yang, Q.; Fu, S. Embedded deep learning for vehicular edge computing. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 341–343.
32. Hu, P.; Chen, W. Software-defined edge computing (SDEC): Principles, open system architecture and challenges. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 8–16. Available online: <https://ieeexplore.ieee.org/document/9060313> (accessed on 12 August 2021).
33. Daga, H.; Yoon, H.; Bhardwaj, K.; Gupta, H.; Gavrilovska, A. From Back-of-the-envelope to Informed Estimation of Edge Computing Benefits in Minutes Using Castnet. In Proceedings of the 2019 IEEE International Conference on Fog Computing (ICFC), Prague, Czech Republic, 24–26 June 2019; pp. 165–174.
34. Alsenani, Y.; Crosby, G.; Velasco, T. SaRa: A stochastic model to estimate reliability of edge resources in volunteer cloud. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 121–124.
35. Wang, X.; Yeo, C.S.; Buyya, R.; Su, J. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Gener. Comput. Syst.* **2011**, *27*, 1124–1134. [CrossRef]
36. Celestini, A.; Lafuente, A.L.; Mayer, P.; Sebastio, S.; Tiezzi, F. Reputation-based cooperation in the clouds. In *International Conference on Trust Management*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 213–220.
37. Avasalcai, C.; Tsigkanos, C.; Dustdar, S. Decentralized resource auctioning for latency-sensitive edge computing. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 72–76.

38. Qian, J.; Gochhayat, S.P.; Hansen, L.K. Distributed active learning strategies on edge computing. In Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 21–23 June 2019; pp. 221–226. Available online: <https://ieeexplore.ieee.org/abstract/document/8854053> (accessed on 15 July 2021).
39. Skarin, P.; Tärneberg, W.; Årzen, K.E.; Kihl, M. Towards mission-critical control at the edge and over 5G. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 50–57.
40. Ren, P.; Qiao, X.; Chen, J.; Dustdar, S. Mobile edge computing—A booster for the practical provisioning approach of web-based augmented reality. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Bellevue, WA, USA, 25–27 October 2018; pp. 349–350.
41. Song, Z.; Cheng, J.; Chauhan, A.; Tilevich, E. Pushing Participatory Sensing Further to the Edge. In Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 8–13 July 2019; pp. 24–26.
42. Strinati, E.C.; Barbarossa, S.; Choi, T.; Pietrabissa, A.; Giuseppi, A.; De Santis, E.; Kim, I. 6G in the sky: On-demand intelligence at the edge of 3D networks. *arXiv* **2020**, arXiv:2010.09463.
43. Arena, E.; Corsini, A.; Ferulano, R.; Iuvara, D.A.; Miele, E.S.; Ricciardi Celsi, L.; Villari, M. Anomaly Detection in Photovoltaic Production Factories via Monte Carlo Pre-Processed Principal Component Analysis. *Energies* **2021**, *14*, 3951. [[CrossRef](#)]
44. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [[CrossRef](#)]
45. Harrison, D.C.; Burmester, D.; Seah, W.K.; Rayudu, R. Busting myths of energy models for wireless sensor networks. *IEEE Internet Things J.* **2016**, *52*, 1412–1414. [[CrossRef](#)]
46. Mocnej, J.; Miškuf, M.; Papcun, P.; Zolotová, I. Impact of Edge Computing Paradigm on Energy Consumption in IoT. *IFAC PapersOnLine* **2018**, *51*, 162–167. [[CrossRef](#)]
47. Porambage, P.; Okwuibe, J.; Liyanage, M.; Ylianttila, M.; Taleb, T. Survey on multi-access edge computing for Internet of Things realization. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2961–2991. [[CrossRef](#)]
48. Giust, F.; Sciancalepore, V.; Sabella, D.; Filippou, M.C.; Mangiante, S.; Featherstone, W.; Munaretto, D. Multi-access edge computing: The driver behind the wheel of 5G-connected cars. *IEEE Commun. Stand. Mag.* **2018**, *2*, 66–73. [[CrossRef](#)]
49. Baktir, A.C.; Ozgovde, A.; Ersoy, C. How can edge computing benefit from software-defined networking: A survey use cases and future directions. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2359–2391. [[CrossRef](#)]
50. Dell EMC Modular Data Center Solutions. Available online: <https://www.delltechnologies.com/en-us/solutions/modular-data-centers.htm> (accessed on 15 July 2021).
51. HPE Converged Edge System. Available online: <https://buy.hpe.com/us/en/servers/edgeline-systems/edgeline-systems/edgeline-converged-edge-systems/hpe-edgeline-el8000-converged-edge-system/p/1011622898> (accessed on 6 July 2021).
52. Dell technologies. Available online: <https://www.delltechnologies.com/it-it/oem/oem-xr2-rugged-server.htm> (accessed on 18 July 2021).
53. Making macOS Run Well on ARM Processors Isn't the Hard Part. Available online: <https://www.theverge.com/2020/4/24/21233661/mac-os-arm-processor-transition-apps-developers-catalyst-wwdc> (accessed on 16 July 2021).
54. Apple Mac Computers Make Jump to Its Own Chips. Available online: <https://www.bbc.com/news/technology-53142989> (accessed on 20 July 2021).
55. Alexa, Edge Computing, and the Future of Enterprise IT. Available online: <https://www.ctera.com/company/blog/alex-edge-computing-and-the-future-of-enterprise-it/> (accessed on 6 August 2021).
56. Azure Edge Zones: Microsoft's Plan to Dominate Edge Computing and 5G. Available online: <https://www.datacenterknowledge.com/microsoft/azure-edge-zones-microsoft-s-plan-dominate-edge-computing-and-5g> (accessed on 25 July 2021).
57. Lambda Function. Available online: [https://aws.amazon.com/lambda/edge/?nc1=h\[_\]ls](https://aws.amazon.com/lambda/edge/?nc1=h[_]ls) (accessed on 13 July 2021).
58. Lighting Edge Platform. Available online: <https://www.foghorn.io/edge-ai-platform/> (accessed on 5 July 2021).
59. FogHorn Extends Industrial Edge Intelligence Leadership with \$25M in Series C Funding. Available online: <https://www.globenewswire.com/news-release/2020/02/25/1989987/0/en/FogHorn-Extends-Industrial-Edge-Intelligence-Leadership-with-25M-in-Series-C-Funding.html> (accessed on 18 June 2021).
60. IBM Edge Application Manager. Available online: <https://www.ibm.com/cloud/edge-application-manager/> (accessed on 16 August 2021).
61. Edge Solutions. Available online: <https://www.delltechnologies.com/en-us/solutions/surveillance/partner-validation.html> (accessed on 5 July 2021).
62. Dell Technologies Unveils Expanded Portfolio Targeting Edge Computing. Available online: <https://www.hpcwire.com/off-the-wire/dell-technologies-unveils-expanded-portfolio-aims-to-help-customers-realize-competitive-insights/> (accessed on 18 July 2021).
63. Edge Systems. Available online: <https://www.hpe.com/uk/en/servers/edgeline-systems.html> (accessed on 16 July 2021).
64. Intelligent Edge. Available online: <https://www.hpe.com/us/en/solutions/edge.html#:~:text=Edge%20systems-,HPE%20Edgeline%20Converged%20Edge%20Systems,data%2C%20and%20translating%20industrial%20protocols> (accessed on 18 July 2021).
65. Rigado Cascade Solution. Available online: <https://www.rigado.com/> (accessed on 28 July 2021).
66. Edge Computing Solutions. Available online: <https://www.cisco.com/c/en/us/solutions/service-provider/edge-computing.html> (accessed on 25 July 2021).
67. ClearBlade IoT Edge Platform. Available online: <https://www.clearblade.com/clearblade-edge-platform/> (accessed on 22 July 2021).

68. MEC Mobile Edge Computing. Available online: <https://www.saguna.net/solution/mec-for-mobile-operators/> (accessed on 15 July 2021).
69. Gand, F.; Fronza, I.; El Ioini, N.; Barzegar, H.R.; Pahl, C. Serverless Container Cluster Management for Lightweight Edge Clouds. 2020; pp. 302–311. Available online: <https://www.scitepress.org/Papers/2020/93795/93795.pdf> (accessed on 25 July 2021).
70. Understanding open source serverless platforms: Design considerations and performance. In Proceedings of the 5th International Workshop on Serverless Computing, Davis, CA, USA, 9–13 December 2019; pp. 37–42.
71. Bila, N.; Dettori, P.; Kanso, A.; Watanabe, Y.; Youssef, A. Leveraging the serverless architecture for securing linux containers. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 5–8 June 2017; pp. 401–404.
72. Knative Components & Architecture: Kubernetes Based Serverless Application Framework. Available online: <https://www.vamsitalkstech.com/architecture/knative-kubernetes-based-serverless-application-framework/> (accessed on 10 July 2021).
73. Balla, D.; Maliosz, M.; Simon, C. Open Source FaaS Performance Aspects. In Proceedings of the 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), Milan, Italy, 7–9 July 2020; pp. 358–364.
74. Fission Executor. Available online: <https://docs.fission.io/docs/concepts/components/core/executor/> (accessed on 14 July 2021).
75. Nuclio Processor. Available online: <https://nuclio.io/docs/latest/concepts/architecture/> (accessed on 16 July 2021).