Edge Computing: Use Cases and Research Challenges

Cosmin Avasalcai and Schahram Dustdar

Abstract

The continuum increase of connected devices and the rise of new emergent applications with fast response times, higher privacy, and security, push the horizon to a new industrial revolution. As a result, the impact of optimizing production and product transactions manifests a fierce necessity of developing new concepts like the Industry 4.0. The combination of traditional manufacturing and industrial practices with the increasingly large-scale machine-to-machine and the Internet of Things deployments, helps manufacturers and consumers to better communication and monitoring, along with new levels of analysis, providing a truly productive future. Edge computing represents an integral part of Industry 4.0, having the purpose of enabling computational resources closer to the edge of the network. In this chapter, we describe in detail this paradigm by looking at its advantages and disadvantages as well as some representative use cases. Finally, we present and discuss the research challenges found in edge computing, mostly focusing on resource management.

Keywords

Edge Computing • Industry 4.0 • Internet of Things

C. Avasalcai
Technology, Siemens AG, Vienna, Austria
e-mail: cosmin.avasalcai@siemens.com

S. Dustdar (✉)
Distributed Systems Group, TU Wien, Vienna, Austria
e-mail: dustdar@dsg.tuwien.ac.at

B. Vogel-Heuser and M. Wimmer (eds.), *Digital Transformation*,
https://doi.org/10.1007/978-3-662-65004-2_5

1 Introduction

The Internet of Things (IoT) devices have seen tremendous technological improvements in their capabilities over the last couple of years. A trend that contributes to the appearance of new use cases such as smart city, smart manufacturing, and smart home, with the power of transforming our daily lives and work environment. However, with the increasing adoption of connected devices, the amount of generated data grows making the current cloud-centric solutions face challenges in meeting the stringent requirements of IoT applications; applications that require low latency as well as better privacy and security.

As a solution to these challenges, researchers proposed a new paradigm, i.e., edge computing, to extend the cloud capabilities closer to the edge of the network. Edge computing enables more computational resources (i.e., processing power, memory, and storage) in the proximity of IoT devices, allowing to process data closer to its origin [27]. Benefits that can transform and optimize the workflow of many different industries like Automotive, Healthcare, and Manufacturing.

For example, by employing edge computing to improve manufacturing efficiencies, smart manufacturing aims to merge the digital (IT) and analog (OT) worlds (building connectivity and orchestration to enable flexibility in physical processes to address a dynamic and global market). Additionally, it seeks to respond in a short time to meet changing demands and conditions in the factory, in the supply network, and to fully integrate manufacturing systems—a key focus of the Industrial 4.0. Clearly, a test of such responsiveness is the capability of customized mass production. Moreover, the manufacturing sector is being fundamentally reshaped by the unstoppable progress of the 4th Industrial Revolution, powered by the IoT and edge computing. Therefore, Industry 4.0 can be seen as the initiators of the smart manufacturing era.

Industry 4.0, like so many new technologies, is not a hot topic as many belief; it is more a rebirth of an older concept that is utilizing newly developed technology. Industry 4.0 is essentially a revision to smart manufacturing that makes use of the latest technological inventions and innovations [13]. Industry 4.0 was coined by the German government initiative and it aims to safeguard a sustainable competitive advantage for the manufacturing base, focusing on connecting the IT and OT using edge devices. The technology identifies itself as the industry that characterizes this century.

In this chapter, we focus on describing edge computing by examining its advantages and disadvantages. Additionally, to further understand the benefits and the challenges of this paradigm, we present an example use case scenario in which edge computing is adopted to build a smart factory. Finally, we examine the research challenges associated with the adoption of edge computing from the point of view of resource management, network communication, and security and privacy issues.

The remainder of the chapter is structured as follows: Sect. 2 defines the edge computing paradigm by describing its architectural features. Next, Sect. 3 describes a smart factory illustrative use case for edge computing, while in Sect. 4 we discuss the challenges that must be overcome to fully integrate edge computing in our society. Finally, Sect. 5 presents our final remarks on edge computing and its challenges.

2 Edge Computing

Edge computing facilitates the operation of computing, storage, and networking services closer to the edge of the network [15] creating a bridge, by adding an additional layer of nodes, between IoT devices (i.e., sensors and actuators) and cloud [26]. The edge layer consists of distributed edge devices with different capabilities, e.g., cloudlets [25], portable edge computers [23], and edge-cloud [11], enabling the deployment of applications in remote locations. An edge device is characterized by (i) heterogeneity, (ii) mobility, and (iii) limited computational resources. Figure 1 presents an overview of an extended cloud-centric architecture, with the addition of edge computing.

Multiple definitions of edge computing are found in the research literature, however, in our opinion, the most relevant is presented in [27]. The authors define edge computing as an enabler for technologies to process data near end-users, i.e., on downstream data for cloud services and upstream data for IoT services. Considering the growing adoption of IoT devices and the stringent requirements of emerging IoT applications, it is clear that processing data closer to the end-users is important. Due to its nature, edge computing has many characteristics [20] described below:

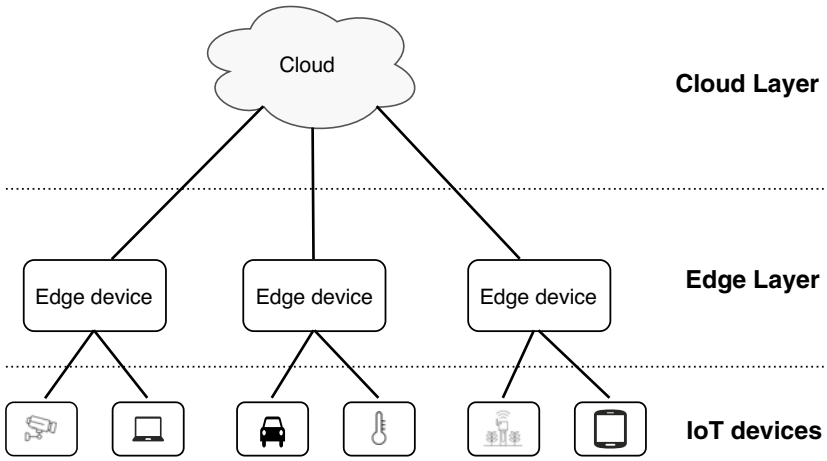

Fig. 1 Edge computing: a bridge between Cloud and IoT devices

1. **Proximity** Since computational resources are available in the proximity of end-users, both cloud and IoT devices can benefit from allocating applications to the edge nodes.
2. **Low latency** The placement of computational resources near the end-user enables deployed applications to provide responses in a short amount of time. A characteristic that aids the cloud in meeting the stringent requirements of latency-sensitive IoT applications.
3. **Increase availability** Maintaining the deployed application operational, even in the absence of a stable connection to the cloud, represents another important characteristic of edge computing. Since there is an extra layer, where deployed application may reside, the applications can work properly independent of the connections to other upper layers (i.e., cloud or any layer which contains more powerful devices).
4. **Device mobility** Edge computing supports device mobility, meaning that enables new IoT devices to connect and use the nearby edge nodes. However, when the IoT devices leave that location, the IoT device interrupts the connection with the old edge device and a new connection forms with the new closest edge node. A behavior that introduces uncertainty into the network; uncertainty that must be considered when deploying and maintaining an application at the edge.
5. **Device heterogeneity** Edge computing consists of distributed edge nodes, communication technologies ensuring the connection between these devices, and different infrastructures. Heterogeneity comes from each edge element, e.g., there may be a variety of differences between edge devices like software, hardware, and technology. Combining all these differences results in the appearance of an interoperability problem. As a result, device heterogeneity represents a challenge and must be considered when deploying an application.
6. **Context-Awareness** Since devices may enter and leave the network at any time without offering any information, context data enables the application coordinator to recover from a bad state, introduced by device mobility, by understanding the environment where the application is deployed. Context data consists of knowledge of device location, environmental characteristics (e.g., temperature sensor, video, and images, etc.), and network information.

Edge computing capabilities shine when converging with IoT and cloud creating novel techniques for IoT systems. Edge computing allows customers to develop and deploy new IoT applications on edge devices, taking advantage of lower latency and increased privacy and security, processing data at the edge without the need of transferring it to a remote location like a cloud. As a result, we consider edge computing as an extension of cloud, helping cloud to meet the stringent requirements of IoT applications, e.g., smart connected vehicles or augmented reality which requires low latency and fast response times, sensors networks that requires location awareness, and smart grids which require large-scale distributed systems.

Many devices can fill the role of an edge device, ranging from resource-constrained devices like smartphones or single-board computers to server-class data centers. Due to

this diversity, in the research literature similar paradigms were proposed such as mobile edge computing (MEC) [5] and fog computing [6], that have the same objective—to move computational resources closer to the edge of the network [10]. MEC considers that an edge device is a micro data server placed at a telecommunication relay station and aid resource-constrained devices (i.e., a smartphone) to compute high computational microservices. In contrast, fog computing consists of distributed highly virtualized fog nodes, i.e., cloudlets, that shares the same characteristics as the cloud. We observe that in both cases, the underlying principle is the same, i.e., to extend the cloud and allow the deployment of IoT applications closer to the end-user.

In conclusion, edge computing fills the technological gap found in cloud-centric IoT systems by collaborating with the cloud to create a more scalable and reliable system where IoT applications may be deployed. From this collaboration, new possibilities to deploy the application appears, letting the developer choose if an application should be placed in the cloud or on the edge layer, depending on the application's requirements. An action that can be done manually by the developer or automatic using resource management techniques.

As we can observe, edge computing transformed the current cloud-centric architecture, bringing many advantages to ensure the correct deployment of emergent applications. However, adopting this paradigm in any industry is not a trivial task. Edge computing brings many challenges that makes the development, deployment, and management of IoT applications more difficult—we transition from a target architecture where an application is deployed in a central location, i.e., in the cloud, to a distributed system where a single edge device may not be capable to host an entire application.

To take advantage of the distributed available resources found in an edge computing architecture, we must change the application model and develop novel resource management techniques. For the former, the application model must change from a monolithic to a microservice-based architecture—the developer must divide the application's functionality into multiple microservices [2]. Developing the new application model represents a challenge in itself since the developer must correctly define different requirements for each microservice as well as creating the application's communication flow. For the latter, the current resource management techniques used to deploy an application to the cloud cannot ensure the correct deployment in an edge computing architecture. Deploying an application in an edge computing architecture is not a trivial task since edge devices are heterogeneous, mobile, spatially distributed, and prone to failure. As a result, we require novel resource management techniques to find a deployment strategy for our application and manage the deployed application at the edge of the network. The management of deployed applications, at runtime, is a difficult challenge by itself—we require techniques to recover an application from a faulty state when the target edge architecture has changed (i.e., edge nodes have failed or left the network).

3 Use Cases

Edge computing, in collaboration with cloud computing, can transform the current functionalities of industries by enabling the deployment of emergent IoT applications. For example, the current city infrastructure evolves into a smart city infrastructure by adopting the edge computing architecture providing advantages to people as well as companies; it can create new work environment boosting productivity with smart buildings, can improve the living conditions of a family by creating a smart home environment, and can create smart factories by bringing together the IT and OT resulting in production optimizations and cost reductions. In this section, we focus on the smart factory scenario where edge computing is adopted, presenting and discussing the inherent advantages of edge computing and the research challenges that appear in this context.

3.1 Smart Manufacturing Scenario

Typically, an industrial factory is isolated from the Internet, keeping the IT and OT separated from each other by using private industrial networks. However, this approach was designed for static networks with a limited number of nodes—a scenario that is not true anymore for the modern industrial factories. These factories have seen an increase in the number of participating nodes and require support for dynamic changes and online reconfigurations [16]. Industry 4.0 and the adoption of edge computing aims at connecting the cloud computing to the manufacturing systems using the Internet; an approach that brings many advantages for industrial factories like, better connectivity, interoperability, and scalability [14].

To better understand the impact of edge computing in a smart factory environment we discuss the following IoT-based manufacturing scenario presented in [8]. The proposed edge computing architecture consists of three different layers, an edge layer, the IoT devices, and the cloud layer. However, in such an architecture the engineer must consider the impact of individual domains when developing it; four different domains are identified, i.e., the device domain, the network domain, the data domain, and the application domain (see Fig. 2).

We present each individual domain below:

1. **The device domain** A domain located either on the IoT devices layer or in the proximity of such devices like sensors, actuators, and robots. The main purpose of the device domain is to provide flexible communication, by using different standardized communication protocols; it enables the capabilities of edge nodes to collect and process data received from IoT devices, based on which it can optimize the control of the industrial machinery.
2. **The network domain** A domain that creates a bridge between the IoT devices and edge nodes. Moreover, it enables a separation between network transmission and the control of industrial machinery with the help of software-defined networking (SDN). Finally, since we are in a environment where applications require low latency, in the deployment

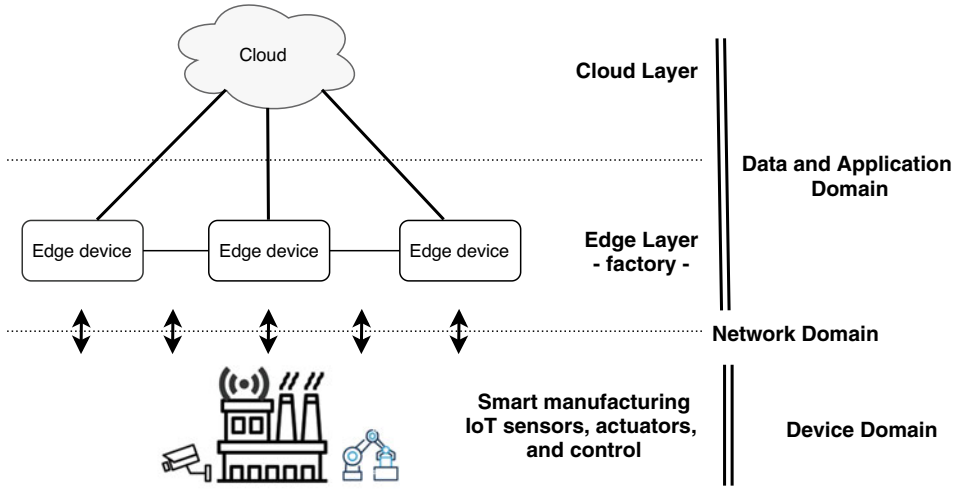

Fig. 2 Smart manufacturing edge computing architecture

of an IoT application we need to control not only the execution of a microservice on a host node but also the transmission of messages between dependent microservices; a solution for the scheduling of messages in the network is offered by the time-sensitive network (TSN).

3. **The data domain** The edge layer is where the data domain resides. In this layer, microservices can be deployed to perform different actions, e.g., take decisions based on the received sensor data or pre-process sensor data for storing in the cloud.
4. **The application domain** This domain offers the possibility to migrate applications from the cloud on the edge devices. A practice that improves the overall manufacturing capabilities of our smart factory and reduces the operational costs.

Considering the edge computing architecture presented in Fig. 2, we continue by presenting two real-world industrial use-cases [9], i.e., (i) accessing and using machine data at runtime and (ii) deploying machine software updates.

The first use case aims at accessing and using the machine data at runtime, gathered from sensors installed on the shop floor. These sensors generate several megabytes of data that provide valuable information regarding the state of the process and each machine – information that can be further used for applications, like predictive maintenance and throughput optimization. However, the current isolated industrial infrastructure lacks the resources to handle this vast amount of generated data—a problem that is solved by adopting edge computing. In this case, edge computing allows the implementation of additional functionalities, enabling data analysis techniques to be performed on-site and transferring vast amounts of data to the cloud for further processing.

The second use case aims at deploying software updates to machines found on the shop floor with a small or no downtime required. To deploy any updates to a industrial factory, many months of prior preparation are required to minimize as much as possible the required downtime. Typically, to install these updates on the physical programmable logic controllers (PLCs), an operator has to do this manually [9]. However, because the production process must stop during this procedure, the operator must perform the updates in a timely manner—an approach that is prone to introduce more errors, resulting in further disruptions. These challenges can be mitigated by using an edge device to host a virtual PLC—PLCs that can be updated automatically from the cloud.

One could say that the aforementioned use cases tackle only the first two challenges discussed briefly in Sect. 2, i.e., developing and deploying of applications. Indeed, in the second use case, we present only the deployment of application updates—applications that are already operational on the edge architecture. However, in reality, in the second use case, the management of the deployed applications is a fundamental part of the use case. As previously mentioned, deploying an application on the target edge computing architecture is not enough—we must ensure the correct functionality of the deployed application throughout its entire life cycle. Therefore, the management of applications is an integral part of any use case that targets application deployment.

In conclusion, adopting edge computing in a manufacturing scenario can bring many advantages and help converge the IT and OT domains to achieve better control and optimization of the manufacturing environment. Furthermore, we can add to a industrial factory many more industrial IoT applications that will improve the overall workflow of a factory. However, many challenges must be solved before edge computing can be adopted safely.

Considering the smart factory scenario, we have identified three research challenges, i.e., security, network communication, and resource management. First, we need to ensure that each edge device is secure enough to withstand any security attack initiated from different sources, e.g., external agents, third-party service providers, and malicious operators [9]. Besides security, ensuring that edge devices can communicate properly represents the second challenge that we must address. Finally, the last and most important challenge is to migrate the applications from cloud-centric architecture to edge computing architecture. As mentioned in Sect. 2, providing resource management in an edge computing architecture is not a trivial task since available resources are distributed among edge devices. We discuss in detail the three challenges in the next section, where we present an introduction to each as well as their research challenges.

4 Research Challenges

One of the main challenges is the integration of OT with IT for the existing factories in the world—a slow and difficult process that may represent a challenge for the next decades. In Sect. 3, we highlight that most of the existing factories use industrial networks that

are isolated from the outside world. When transforming a typical industrial factory into a smart factory we must integrate the existing legacy systems into the edge computing architecture—a legacy system is part of the OT and represents the backbone of the factory, e.g., supervisory control and data acquisition (SCADA) is used to build the industrial network that connects multiple industrial PCs and PLCs [7]. More details on how to achieve the IT and OT convergence is presented in [22], while more specific use cases that target predictive maintenance and security are presented in details in [1, 28] respectively. In the use-cases presented in Sect. 3, we can observe that the integration already took place by consolidating the functionality of PLCs, industrial PCs, and gateways into an edge node. However, many challenges must be addressed before adopting edge computing.

In this section, we provide an overview of the three most important challenges that must be solved when adopting the edge computing paradigm, identified in the previous section, i.e., (i) resource management, (ii) security and privacy, and (iii) network management. In a typical Industry 4.0 edge computing architecture, there is a forth challenge, i.e., data management, that will not be discussed in this chapter but plays an equal role in the successful transformation of industrial factories.

4.1 Resource Management

With the introduction of the edge computing paradigm, the demand for novel resource management techniques, to deploy and maintain an application on the IoT network, has increased. Resource management stays at the core of this paradigm as a prime technique to efficiently utilize the available computational resources distributed near the end user. Therefore, resource management plays an important role in the successful adoption of edge computing.

Resource management does not refer only to the deployment of an application but represents an optimal combination of different groups to deploy and manage the application at the edge such that it satisfies all application's requirements [30]. There are four different groups, i.e., resource discovery, resource allocation, resource migration, and resource sharing; each with its own goal.

Resource Discovery Techniques for discovering resources in a distributed network are well covered in the research literature. However, these techniques cannot be leveraged at the edge of the network, due to device heterogeneity, modern workloads (e.g., machine learning applications), and rapidity to discover the available resources [32]. Resource discovery aims at keeping the pool of available resources updated, by discovering resources already deployed at the edge. Hence, seamless discovery and removal of nodes without introducing extra latency and communication overhead is desired. An example of a resource discovery technique deployed in a smart city scenario is presented in [21]. In this paper, the authors propose an edge-to-edge metadata replication framework that uses the kademlia as a com-

munication protocol and elastic search to enable nodes to store and search data in a fast manner. To conclude, the resource discovery plays a more critical role in a smart city scenario, where we target volatile edge computing architectures defined by high uncertainty. In contrast, in a smart factory, there is less uncertainty and typically the participating nodes are know – rendering the need for a resource discovery technique less important.

Resource Sharing By employing resource sharing techniques, we ensure that edge nodes are willing to share resources and collaborate to achieve a common goal. A very important aspect that stays at the core of resource management; without collaboration between nodes, the nodes cannot host applications at the edge of the network. Resource sharing is using incentive techniques to ensure collaboration between nodes. As a consequence, an edge node is motivated to share as many available resources as possible, since sharing more resources will give in return more incentives. Similar to resource discovery, resource sharing is less important in the context of Industry 4.0, where edge nodes are sharing resources willingly without the need of an incentive—all edge nodes belong to the owner.

Resource Allocation After we knows the available resources shared between edge nodes and the microservices' resource requirements as well as the application's objectives, the deployment of the application can start. Resource allocation refers to the process of mapping microservices to edge nodes such that the resource requirements of each microservice and the overall application's quality of service (QoS) are satisfied. Therefore, we employ resource allocation techniques to find a deployment strategy to efficiently use the available resources found in the target edge computing architecture. From the definition of edge computing (Sect. 2), we identify two possibilities to use resource allocation, i.e., microservice offloading and microservice allocation. For the former, the aim is to help the user devices (e.g., a smartphone) to execute applications by offloading high computational microservices to nearby edge nodes; a technique that helps these resource-constrained devices to optimize the utilization of resources. The later has the purpose of extending cloud capabilities in satisfying the application's requirements by allocating a part of the microservices on edge devices where the collected data can be processed.

Resource allocations techniques play an important role in Industry 4.0 since edge computing brings together both IT and OT by acting as a bridge between the two. The consolidation of legacy systems into edge nodes makes the deployment of latency-sensitive applications more challenging. In this setting, on a single edge node, both control applications and latency-sensitive application must co-exist. In our case, a latency-sensitive application represents a non-critical application—such an application may miss a deadline without having high repercussions. Therefore, during the deployment stage, we need to make sure that any new application added to an edge node does not has an impact on the already deployed control applications. To achieve this, we must combine resource allocation techniques to devise a microservice allocation to edge nodes and a scheduling techniques to schedule both control and latency sensitive applications on the local edge node such that the control

applications correct functionality is preserved. For example, we can use the decentralized resource allocation technique, presented in [3], to find a satisfiable deployment strategy for a latency-sensitive application. This technique allows each edge node to decide what microservices to host, by using a set of different decision strategies, considering the current available resources and hosted application. Therefore, we can use the scheduling technique presented in [4]—a scheduling technique that ensures the correct functionality of control application when new latency-sensitive application are added to the host node—as a decision strategy to determine if a certain microservice can be mapped on a node.

Resource Migration Considering the dynamic nature of edge computing architectures, the deployment of an application does not ensure the correct functionality over the entire application's lifespan; the network topology used at deployment time is not static, i.e., it is more likely to change multiple times before the application finishes its execution. As a result, we need to develop resource migration mechanisms to recover the application from a bad state—a state where the application's requirements are not satisfied. However, performing a migration of microservices between edge nodes requires more communication overhead. For example, migrating a 35 MB microservice to another node takes 35 s, assuming a 1 MB communication link. We can mitigate the communication overhead by devising a migration technique based on microservice replication—we simply change the communication link to another available replica of that particular microservice, instead of moving a microservice from a failed node to another. In contrast to the first approach, using replication lowers the communication overhead but required more available resources in the target edge computing architecture—each microservice replica will consume a part of the total available resource even when the replicas are not used. As a result, the developer of an edge system must consider the advantages and disadvantages of the two approaches and choose the one that fits best with the system's needs.

In conclusion, creating a resource management technique, to ensure the optimal utilization of the available computational resources shared between multiple resource-constrained devices, represents the first step in the adoption of edge computing. Moreover, it is important to mention that in the context of smart manufacturing and Industry 4.0, some of the groups are not so critical or not important at all. For example, resource sharing and resource discovery are easier to adopt since in a manufacturing context there is a controlled environment, where all nodes belong to the same administrative entity. As a result, there is no need for incentives mechanism or techniques for resource discovery, because the network does not have so many uncertainties compared to a smart city scenario.

4.2 Network Management

Edge computing consists of interconnected devices distributed at the edge of the network. Thus, network management has an important role to fill in the overall network architecture,

ensuring that edge devices can communicate to transfer process data from a microservice to another dependent microservice found on a different node. By communicating, nodes can collaborate and share resources to host different deployed applications. Besides the communication path, the network management must provide extra functionality i.e., (i) assists the resource management technique in finding satisfiable deployment strategies, (ii) ensures seamless connectivity for new devices, and (iii) provide a deterministic network for control applications.

Assisting Resource Management When deploying latency-sensitive IoT applications, one of the objectives is to satisfy the end-to-end delay (e2e delay) of the application's communication flow. The e2e delay is the result of the combination of the worst-case execution time (WCET) of a microservice on a node and the communication latency between two dependent microservices and their host nodes. This is particularly important when we deploy applications on an edge computing architecture, since different microservices may reside on different nodes—a mapping that has a high impact on the overall e2e delay of the deployed applications. Therefore, we must introduce two new mechanisms to (i) monitor the latency in the network, at runtime, without introducing extra communication overhead and (ii) find the WCET of microservices. It is important to mention that both latency and WCET are dependent on the microservices' location on the target edge architecture.

Seamless Network Connectivity Considering the uncertainty found in an edge architecture, providing a seamless connectivity mechanism is imperative in a network where both stationary and mobile devices exist. These mechanisms should ensure that new devices can join the network automatically as well as leave it; a job that must be performed by the device requiring from the user a limited technical background. A characteristic that increases the adoption of new edge devices, extending the capabilities of a network in a smart city scenario. An example of a self-organizing approach to seamlessly add new nodes to the system [19]. In this setting, the edge nodes are capable to self-organize either in a hierarchical or a peer-to-peer manner such that the objectives of the system are satisfied, e.g., fault tolerance or proximity awareness. In the context of Industry 4.0, this challenge may not be an issue for the current industrial factories where there is a more or less static environment. However, in the future, this will play an important role, since in a smart manufacturing floor a robot may leave one network and join another to start performing different activities.

Deterministic Network For most IoT applications deployed in a smart city scenario, satisfying a QoS in terms of availability is enough for ensuring great service. Usually, in these cases, the time when a message arrives is not important. However, for control applications that have hard deadlines to meet, knowing the time of arrival for all messages sent between microservices is critical. These types of applications can be found in smart factories, where the objective is to control robots and other factory equipment. As a result, a new communication technology that creates a deterministic network is proposed called a time-sensitive

network (TSN) [17]. In TSN there are three types of messages, i.e., time-triggered (TT), audio-video bridging (AVB), and best effort (BE), where the biggest priority is assigned to the TT messages used by control applications. It is interesting, that in a deterministic network it is possible to schedule the messages as well, offering the possibility to ensure a communication flow for the application where the delays are known.

Many other new emerging technologies have been proposed in the research literature, such as software-defined networks, network function virtualization, and network slicing to implement the network, increasing the scalability while reducing the cost [29].

4.3 Security and Privacy

Edge computing enables the digitalization of our environment and everyday lives. However, with an increase in digitalization, we expose our private life to malicious users, as such engineers must develop and enforce new privacy and security techniques on the edge network. Not only that each device has its privacy and security challenges, but these devices also inherit them from the cloud. For example, a burglar can monitor the activity of a home by accessing the edge devices placed in that home, from which it can learn the behavior of the family and when the house is empty, giving the possibility to plan a heist accordingly. As a consequence, upon the adoption of edge devices, enforcing privacy and security is a crucial task.

To identify the privacy and security issues that an edge architecture faces, an engineer can apply the confidentiality, integrity, and availability (CIA) triad model [12], which represents the most three important rules that each architecture must abide. In this model, the privacy is evaluated using the confidentiality and integrity components, while for the security the engineer can use the availability component. Furthermore, according to to [34], four challenges must be overcome when placing devices in the proximity of end-users, i.e., authentication, access control, intrusion attack, and privacy.

Security If resource management is one of the main challenges when adopting the edge computing paradigm from the perspective of hosting applications in the edge ecosystem, security represents the greatest challenge when creating the edge computing ecosystem. There are multiple reasons for this identified in [24]. First, the ecosystem inherits all security issues each component has, hence, it is not enough to protect each component of the system but to create a security mechanism that protects the ecosystem and takes into account the collaboration between all components. This task is not trivial since the mechanism must coordinate with the local security techniques deployed on the other components.

Second, edge devices are resource-constrained devices meaning that applying the security mechanisms deployed in the cloud is not feasible. New techniques must be developed that are not centralized and are autonomous because there are possible scenarios where there is

no central control system and at the same time the technique must have a small impact on the edge ecosystem.

Finally, the impact of a successful security attack on an edge architecture has big repercussions for the industries where edge computing is applied since all information about the user will be vulnerable to malicious users. An edge ecosystem represents a big challenge for security since it has multiple layers of technologies that must be protected, resulting in a larger attack surface. In conclusion, if security mechanisms are not developed to protect the architecture, the benefits of adopting edge computing can be outweighed by malicious attacks. This is especially true for Industry 4.0, where ensuring the security of an industrial factory is critical—managing to exploit the security vulnerabilities found in smart manufacturing could result in increased damage to the factory system or even produce catastrophic events. For example, in 2010 there was a malware called Stuxnet that targeted industrial process systems [18]. This malware managed to use the IT resources to manipulate specific processes by monitoring certain variables and change the control commands without being noticed by the system operator. This action leads to material damages, i.e., the product does not meet the required specifications and even permanent damages to the production system.

An extensive study of the security threat model for edge computing paradigm is presented in [24]. The authors identify five different attack points, i.e., network infrastructure, edge node, core infrastructure, virtualization infrastructure, and user devices, each representing the components of an edge ecosystem. For every identified component, a set of possible attacks is discussed. For example, a man in the middle attack, in which an attacker can take control of a part of the network from where eavesdropping or traffic injection attacks can be launched (an example is presented in [33]), is a vulnerability of the network infrastructure component. In contrast, some attacks can target multiple components, e.g., a rogue component can be easily inserted in the network since edge computing by definition consists of many interconnected devices owned by different administrative entities; a rogue component attack can target both the network and core infrastructures as well as the edge node component.

Privacy Privacy represents the process of protecting the user's private data from malicious adversary while in transit [35]. With the current cloud-centric architecture privacy is most vulnerable since all data from the IoT devices is sent to the cloud for further processing. A problem that is diminished by the edge computing paradigm which enables the processing of data closer to its origin. However, the problems of privacy remains since data is still sent between devices and new privacy challenges appear inherited from the edge paradigm such as (i) privacy concerns due to user awareness of privacy rules, e.g., there are more than 80% of WiFi users still use their default password and (ii) the absence of privacy tools for resource-constrained devices [27].

Privacy is not only a concern found in smart city scenarios but also for Industry 4.0 where personal data does not refer to a person but the smart factory as a whole. Next, we present a set of privacy concerns that may apply to Industry 4.0. There are many privacy concerns

regarding an edge computing ecosystem that considers the entire data path starting from collecting the data by edge devices, processing it locally on the edge or in the cloud, and disseminating it, if necessary, back to the edge. According to [31], there are five different privacy concerns, i.e., (i) data collection and identification ensuring that data is not only processed locally but it follows the user preferences and legal or administrative frameworks (e.g., the EU General Data Protection Regulation), (ii) aggregation and inference refers to the process of combining data and connecting it to users to whom it belongs, (iii) secondary use, insecurity, and exclusion concern the manipulation and storage of collected data, (iv) decisions and boundaries protect the user from invasive acts from the deployed applications, and (v) appropriation and distortion which targets the protection of user's privacy after the data are collected.

5 Conclusion

With the introduction of edge computing, a shift in the industries has appears where the overall desire is to migrate the execution of latency-sensitive applications from the cloud closer to the edge of the network. The core idea behind edge computing is to enable more computational resources in the proximity of end-users, facilitating the deployment of IoT applications that have stringent requirements like low latency and increased security and privacy; requirements that are harder to satisfy with the current cloud-centric approach. Hence, industries that adopt this paradigm can observe many benefits for their users and production as well. For example, in the automotive domain, with the development of a smart factory, the convergence of IT and OT is possible, resulting in a new architecture that integrates all devices across the entire network, from cloud to devices found on the factory floor; a change that allows for optimization and a reduction of operational costs.

Edge computing has many benefits, however, it introduces a series of challenges that must be solved upon its adoption. We group these challenges into three categories, i.e., resource management, network management, and security and privacy. Each category focus on a particular part of the edge ecosystem, i.e., resource management refers to challenges found when hosting an application on the edge, considering both the deployment and maintenance aspects, while network management looks at the connectivity challenges having the purpose of ensuring seamless integration of new devices; finally, the last category focuses on ensuring the security and privacy of the users.

In conclusion, in this chapter, we aim at introducing the edge computing paradigm and discuss its advantages and disadvantages. Furthermore, to increase the understanding of this paradigm, we present multiple use cases to exemplify the deployment of a latency-sensitive application in this ecosystem. Finally, we identify and debate the most important challenges that must be overcome before the adoption of edge computing.

Acknowledgements The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA–Fog Computing for Robotics and Industrial Automation.

References

1. Al-Hawawreh, M., Sitnikova, E.: Developing a security testbed for industrial internet of things. IEEE Internet of Things Journal **8**(7), 5558–5573 (2021). https://doi.org/10.1109/JIOT.2020.3032093
2. Avasalcai, C., Murturi, I., Dustdar, S.: Edge and Fog: A Survey, Use Cases, and Future Challenges, chap. 2, pp. 43–65. John Wiley & Sons, Ltd (2020). https://doi.org/10.1002/9781119551713.ch2, https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119551713.ch2
3. Avasalcai, C., Tsigkanos, C., Dustdar, S.: Decentralized resource auctioning for latency-sensitive edge computing. In: IEEE International Conference on Edge Computing (EDGE). IEEE (2019)
4. Barzegaran, M., Karagiannis, V., Avasalcai, C., Pop, P., Schulte, S., Dustdar, S.: Towards extensibility-aware scheduling of industrial applications on fog nodes. In: IEEE International Conference on Edge Computing (EDGE). pp. 67–75. IEEE (2020)
5. Beck, M.T., Werner, M., Feld, S., Schimper, S.: Mobile edge computing: A taxonomy. Citeseer
6. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. 1st ACM Mobile Cloud Computing Workshop pp. 13–15 (2012)
7. Boyer, S.A.: Scada: Supervisory Control And Data Acquisition. International Society of Automation, Research Triangle Park, NC, USA, 4th edn. (2009)
8. Chen, B., Wan, J., Celesti, A., Li, D., Abbas, H., Zhang, Q.: Edge computing in iot-based manufacturing. IEEE Communications Magazine **56**(9), 103–109 (2018)
9. Denzler, P., Ruh, J., Kadar, M., Avasalcai, C., Kastner, W.: Towards consolidating industrial use cases on a common fog computing platform. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 172–179 (2020). https://doi.org/10.1109/ETFA46521.2020.9211885
10. Dustdar, S., Avasalcai, C., Murturi, I.: Invited paper: Edge and fog computing: Vision and research challenges. In: 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE). pp. 96–9609 (April 2019). https://doi.org/10.1109/SOSE.2019.00023
11. Elias, A.R., Golubovic, N., Krintz, C., Wolski, R.: Where's the bear?—automating wildlife image processing using iot and edge cloud systems. In: 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI). pp. 247–258 (April 2017)
12. Farooq, M.U., Waseem, M., Khairi, A., Mazhar, S.: A critical analysis on the security concerns of internet of things (iot). International Journal of Computer Applications **111**(7) (2015)
13. Gilchrist, A.: Introducing industry 4.0. In: Industry 4.0, pp. 195–215. Springer (2016)
14. Givehchi, O., Landsdorf, K., Simoens, P., Colombo, A.W.: Interoperability for industrial cyber-physical systems: An approach for legacy systems. IEEE Transactions on Industrial Informatics **13**(6), 3370–3378 (2017)
15. Gusev, M., Dustdar, S.: Going back to the roots-the evolution of edge computing, an iot perspective. IEEE Internet Computing **22**(2), 5–15 (2018)
16. Gutiérrez, M., Ademaj, A., Steiner, W., Dobrin, R., Punnekkat, S.: Self-configuration of IEEE 802.1 TSN networks. In: IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–8 (2017)

17. Gutiérrez, M., Ademaj, A., Steiner, W., Dobrin, R., Punnekkat, S.: Self-configuration of IEEE 802.1 TSN networks. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA pp. 1–8 (2018)
18. for Information Security (BSI), T.F.O.: Improving it security. shorturl.at/zQZ18 (2021 (accessed March 14, 2021))
19. Karagiannis, V., Schulte, S.: Distributed algorithms based on proximity for self-organizing fog computing systems. Pervasive and Mobile Computing **71**, 101316 (2021)
20. Khan, W.Z., Ahmed, E., Hakak, S., Yaqoob, I., Ahmed, A.: Edge computing: A survey. Future Generation Computer Systems **97**, 219–235 (2019). https://doi.org/10.1016/j.future.2019.02.050, http://www.sciencedirect.com/science/article/pii/S0167739X18319903
21. Murturi, I., Avasalcai, C., Tsigkanos, C., Dustdar, S.: Edge-to-edge resource discovery using metadata replication. In: IEEE 3rd International Conference on Fog and Edge Computing (ICFEC). pp. 1–6. IEEE (2019). https://doi.org/10.1109/CFEC.2019.8733149
22. Pop, P., Zarrin, B., Barzegaran, M., Schulte, S., Punnekkat, S., Ruh, J., Steiner, W.: The fora fog computing platform for industrial iot. Information Systems **98**, 101727 (2021). https://doi.org/10.1016/j.is.2021.101727, https://www.sciencedirect.com/science/article/pii/S0306437921000053
23. Rausch, T., Avasalcai, C., Dustdar, S.: Portable energy-aware cluster-based edge computers. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). pp. 260–272 (Oct 2018). https://doi.org/10.1109/SEC.2018.00026
24. Roman, R., Lopez, J., Mambo, M.: Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems **78**, 680–698 (2018). https://doi.org/10.1016/j.future.2016.11.009, http://www.sciencedirect.com/science/article/pii/S0167739X16305635
25. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The Case for VM-Based Cloudlets in Mobile Computing. IEEE Pervasive Computing **8**(4), 14–23 (Oct 2009). https://doi.org/10.1109/MPRV.2009.82, http://ieeexplore.ieee.org/document/5280678/
26. Shi, W., Dustdar, S.: The promise of edge computing. Computer **49**(5), 78–81 (May 2016). https://doi.org/10.1109/MC.2016.145
27. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: Vision and challenges. IEEE Internet of Things Journal **3**(5), 637–646 (2016)
28. Strauß, P., Schmitz, M., Wöstmann, R., Deuse, J.: Enabling of predictive maintenance in the brownfield through low-cost sensors, an iiot-architecture and machine learning. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 1474–1483 (2018). https://doi.org/10.1109/BigData.2018.8622076
29. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. IEEE Communications Surveys Tutorials **19**(3), 1657–1681 (thirdquarter 2017). https://doi.org/10.1109/COMST.2017.2705720
30. Toczé, K., Nadjm-Tehrani, S.: A taxonomy for management and optimization of multiple resources in edge computing. CoRR **abs/1801.05610** (2018), http://arxiv.org/abs/1801.05610
31. Tsigkanos, C., Avasalcai, C., Dustdar, S.: Architectural considerations for privacy on the edge. IEEE Internet Computing **23**(4), 76–83 (2019)
32. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., Nikolopoulos, D.S.: Challenges and opportunities in edge computing. In: 2016 IEEE International Conference on Smart Cloud (SmartCloud). pp. 20–26 (2016)
33. Wang, Y., Uehara, T., Sasaki, R.: Fog computing: Issues and challenges in security and forensics. In: 2015 IEEE 39th Annual Computer Software and Applications Conference. vol. 3, pp. 53–59 (July 2015). https://doi.org/10.1109/COMPSAC.2015.173

34. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 workshop on mobile big data. pp. 37–42. ACM (2015)
35. Zhou, M., Zhang, R., Xie, W., Qian, W., Zhou, A.: Security and privacy in cloud computing: A survey. In: 2010 Sixth International Conference on Semantics, Knowledge and Grids. pp. 105–112 (Nov 2010). https://doi.org/10.1109/SKG.2010.19