2

Edge and Fog: A Survey, Use Cases, and Future Challenges

Cosmin Avasalcai, Ilir Murturi, and Schahram Dustdar

Distributed Systems Group, TU Wien, Vienna, Austria

2.1 Introduction

In the past couple of years, the cloud computing paradigm was at the center of the Internet of Things' (IoT) ever-growing network, where companies can move their control and computing capabilities, and store collected data in a medium with almost unlimited resources [1]. It was and continues to be the best solution to deploy demanding computational applications with the main focus on processing vast amounts of data. Data are generated from geo-distributed IoT devices, such as sensors, smartphones, laptops, and vehicles, just to name a few. However, today this paradigm is facing growing challenges in meeting the demanding constraints of new IoT applications.

With the rapid adoption of IoT devices, new use cases have emerged to improve our daily lives. Some of these new use cases are the smart city, smart home, smart grid, and smart manufacturing with the power of changing industries (i.e. healthcare, oil and gas, automotive, etc.) by improving the working environment and optimizing workflow. Since most of the use cases consist of multiple applications that require fast response time (i.e. real-time or near real-time) and improved privacy, most of the time the cloud fails to fulfill these requirements (i.e. network congestion and ensuring privacy).

To overcome these shortcomings, researchers have proposed two new paradigms, fog computing and edge computing, to enable more computational resources (i.e. storage, networking, and processing) closer to the edge of the network. Fog computing (FC) extends cloud capabilities closer to the end devices, such that a cloud-to-things continuum is obtained that decreases latency and network congestion while enforcing privacy by processing the data near the user [2]. On the same note, the edge computing vision is to migrate some computational

Fog Computing: Theory and Practice, First Edition.
Edited by Assad Abbas, Samee U. Khan, and Albert Y. Zomaya.

resources from the cloud to the heterogeneous devices placed at the edge of the network [3].

Embracing the vision of these paradigms and focusing on the deployment of multiple applications in close proximity of users, researchers have suggested new fog/edge devices. Among these devices, the most notable are mini servers, such as cloudlets [4], portable edge computers [5], and edge-cloud [6], which enable an application to work in harsh environments; mobile edge computing (MEC) [7] and mobile cloud computing [8] improve user experience and enable higher computational applications to be deployed on smartphones by offloading parts of the application on the device locally.

Many surveys are found in the literature that describe each paradigm in detail and its challenges [3, 9, 10]. However, there is no paper that compares the two; most of the time the terms *fog* and *edge* are both used to describe the same IoT network. Generally speaking, the visions of the two paradigms overlap, aiming to make available more computational resources at the edge of the network. Hence, the most significant difference is given by the naming convention used to describe them. The aim of this chapter is to offer a detailed description of the two aforementioned paradigms, discussing their differences and similarities. Furthermore, we discuss their future challenges and argue if the different naming convention is still required.

The remainder of the chapter is structured as follows: Section 2.2 defines the edge computing paradigm by describing its architectural features. Next, Section 2.3 presents in detail the fog computing paradigm and describes two use cases by emphasizing the key features of this architecture. Section 2.4 describes several illustrative use cases for both edge and fog computing. Section 2.5 discusses the challenges that these paradigms must conquer to be fully adopted in our society. Finally, Section 2.6 presents our final remarks on the comparison between fog and edge computing.

2.2 Edge Computing

As we explore new IoT applications and use cases, the consideration of proximity between edge nodes and the end-users is becoming increasingly obvious. The physical distance between the edge and the user affects highly end-to-end latency, privacy, network, and availability. Recently, this leads to a new paradigm allowing computation to be performed in close proximity of user and IoT devices (i.e. sensors and actuators). Edge computing [11] is a new paradigm aiming to provide storage and computing resources and act as an additional layer, composed of edge devices, between the end-user IoT device and the cloud layer. In edge computing, we define “edge” as any computing and network resources along the path

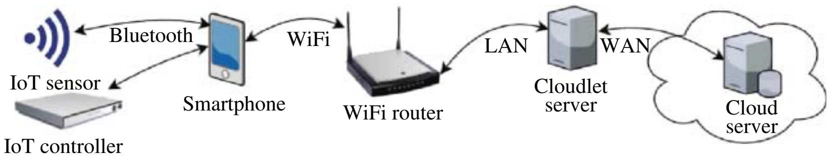

Figure 2.1 Edge computing solution using an IoT and edge devices [12].

between the initial source of data and destination storage of data (fog nodes, cloud data centers).

Edge computing is ever stronger when converging with IoT, offering novel techniques for IoT systems. Multiple definitions of edge computing are found in the literature; however, the most relevant is presented in [3]. Authors define edge computing as a paradigm that enables technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services. The proposed paradigm is a relatively new concept and due to the same nature, the term "edge computing" in literature may refer to all other architectures, such as MEC, cloudlet computing (CC), or fog computing (FC). However, we vision edge computing as a bridge between IoT things and the nearest physical edge device that aims to facilitate the deployment of the new emerging IoT applications in users' devices, such as mobile devices (see Figure 2.1).

Authors [12] in Figure 2.1, present the main idea of the edge computing paradigm by adding another device in the form of an edge device. Such a device can be referred to as a personal computer, laptop, tablet, smartphone, or another device capable of locally processing the data generated by IoT devices. Furthermore, depending on device capabilities, it may offer different functionalities, such as the capability of storing data for a limited time. In addition, an edge device can react to emergency events by communicating with the IoT devices and can aid other devices like cloudlet, MEC server, and cloud data center, by preprocessing and filtering the raw data generated by the sensors. In such scenarios, the edge computing paradigm offers processing near to the source of data and reduces the amount of transmitted data. Instead of transmitting data to the cloud or fog node, the edge device, as the nearest device to the source of the data, will do computation and response to the user device without moving data to the fog or cloud.

Edge computing is considered a key enabler for scenarios where centralized cloud-based platforms are considered impractical. Processing data near to the logical extremes of a network – at the edge of the network – reduces significantly the latency and bandwidth cost. By shortening the distance that data has to travel, this paradigm could address concerns also in energy consumption, security, and privacy [13]. However, the rapid adoption of IoT devices, resulting in millions of interconnected devices, are challenges that Edge Computing must overcome.

2.2.1 Edge Computing Architecture

The details of edge computing architecture could be different in different contexts. For example, Chen et al. [14] propose an edge computing architecture for IoT-based manufacturing. The role of edge computing is analyzed with respect to edge equipment, network communication, information fusion, and cooperative mechanism with cloud computing. Zhang et al. [15] proposes the edge-based architecture and implementation for a smart home. The system consists of three distinct layers: the sensor layer, the edge layer, and the cloud layer.

Generally speaking, Wei Yu et al. divides the structure of edge computing into three aspects: the front-end, near-end, and far-end (see Figure 2.2). A detailed description is given below:

- The **front end** consists of heterogeneous end devices (e.g. smartphones, sensors, actuators), which are deployed at the front end of the edge computing structure. This layer provides real-time responsiveness and local authority for the end-users. Nevertheless, the front-end environment provides more interaction by keeping the heaviest traffic and processing closest to the end-user devices. However, due to the limited resource capabilities provided by end devices, it is clear that not all requirements can be met by this layer. Thus, in such situations, the end devices must forward the resource requirements to the more powerful devices, such as fog node or cloud computing data centers.
- The **near end** will support most of the traffic flows in the networks. This layer provides more powerful devices, which means that most of the data processing and storage will be migrated to the near-end environment. Additionally, many tasks like caching, device management, and privacy protection can be deployed at this layer. By increasing the distance between the source of data and its processing destination (e.g. fog node) it also increases the latency due to

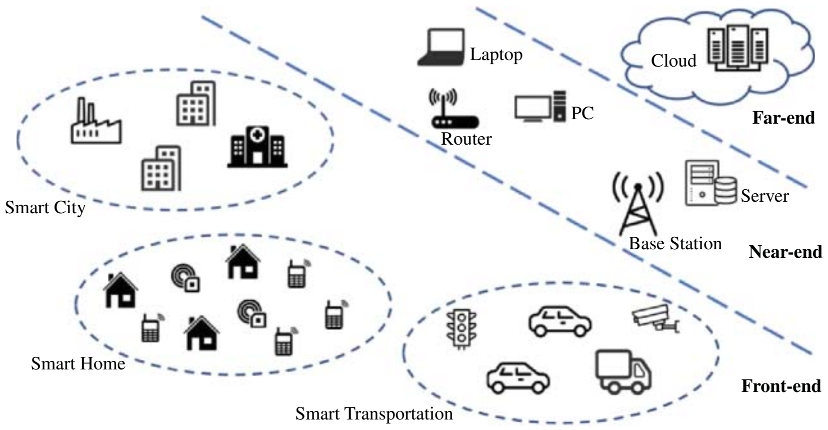

Figure 2.2 An overview of edge computing architecture [16]. (*See color plate section for the color representation of this figure*)

the round-trip journey. However, the latency is very low since the devices are one hop away from the source where the data is produced and consumed.

- The **far end** environment is cloud servers that are deployed farther away from the end devices. This layer provides devices with high processing capabilities and more data storage. Additionally, it can be configured to provide levels of performance, security, and control for both users and service providers. Nevertheless, the far-end layer enables any user to access unimaginable computing power where thousands of servers may be orchestrated to focus on the task, such as in [17, 18]. However, one must note that the transmission latency is increased in the networks by increasing the distance that data has to travel.

2.3 Fog Computing

Fog computing is a platform introduced by Cisco [10] with the purpose of extending the cloud capabilities closer to the edge of the network. Multiple definitions of fog computing are found in the literature; however, the most relevant is presented in [19]. According to them, fog computing is a geographically distributed computing architecture connected to multiple heterogeneous devices at the edge of the network, but at the same time not exclusively seamlessly backed by cloud services. Hence, we envision fog as a bridge between the cloud and the edge of the network that aims to facilitate the deployment of the newly emerging IoT applications (see Figure 2.3).

A fog device is a highly virtualized IoT node that provides computing, storage, and network services between edge devices and cloud [2]. Such characteristics are found in the cloud as well; thus, a fog device can be characterized as a mini-cloud

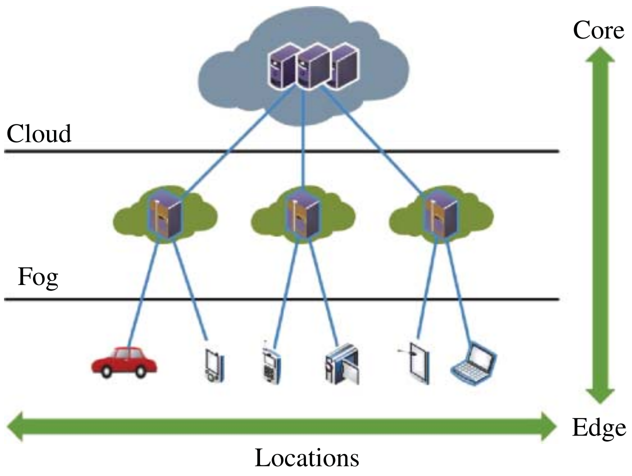

Figure 2.3 Fog computing a bridge between cloud and edge [20].

which utilizes its own resources in combination with data collected from the edge devices and the vast computational resources that cloud offers.

By migrating computational resources closer to the end devices, fog offers the possibility to customers to develop and deploy new latency-sensitive applications directly on devices like routers, switches, small data centers, and access points. Depending on the application, this can impose stringent requirements, which refers to fast response time and predictable latency (i.e. smart connected vehicles, augmented reality), location awareness (e.g., sensor networks to monitor the environment) and large-scale distributed systems (smart traffic light, smart grids). As a result, the cloud cannot fulfill all of these demands by itself.

To fill the technological gap in the current state of the art where the cloud computing paradigm is at the center, the fog collaborates with the cloud to form a more scalable and stable system across all edge devices suitable for IoT applications. From this union, the developer benefits the most since he can decide where is the most beneficial, fog or cloud, to deploy a function of his application. For example, taking advantage of the fog node capabilities, we can process and filter streams of collected data coming from heterogeneous devices, located in different areas, taking real-time decisions and lowering the communication network to the cloud. Thus, fog computing introduces effective ways of overcoming many limitations that cloud is facing [1]. These limitations are:

1. **Latency constraints.** The fog shares the same fundamental characteristics as the cloud in being able to perform different computational tasks closer to the end-user, making it ideal for latency-sensitive applications for which their requirements are too stringent for deployment in the cloud.
2. **Network bandwidth constraints.** Since the fog offers the possibility of performing data processing tasks closer to the edge of the network, lowering in the process the amount of raw data sent to the cloud, it is the perfect device to apply data analytics to obtain fast responses and send to the cloud for storage purposes only filtered data.
3. **Resource constrained devices.** Fog computing can perform computational tasks for constrained edge devices like smartphones and sensors. By offloading parts of the application from such constrained devices to nearby fog nodes, the energy consumption and life-cycle cost are decreased.
4. **Increased availability.** Another important aspect of fog computing represents the possibility of operating autonomously without reliable network connectivity to the cloud, increasing the availability of an application.
5. **Better security and privacy.** A fog device can process the private data locally without sending it to the cloud for further processing, ensuring better privacy and offering total control of collected data to the user. Furthermore, such devices can increase security as well, being able to perform a wide range of security functions, manage and update the security credential of constrained devices and monitor the security status of nearby devices.

The previous section introduces the edge computing paradigm as a solution to the cloud computing inefficiency for data processing when the data is produced and consumed at the edge of the network. Fog computing focuses more on the infrastructure side by providing more powerful fog devices (i.e. fog node may be high computation device, access points, etc.) while edge computing focuses more toward the "things" side (i.e. smartphone, smartwatch, gateway, etc.). The key difference between the edge computing and fog computing is where the computation is placed. While fog computing pushes processing into the lowest level of the network, edge computing pushes computation into devices, such as smartphones or devices with computation capabilities.

2.3.1 Fog Computing Architecture

Fog computing architecture is composed of highly dispersed heterogeneous devices with the intent of enabling deployment of IoT applications that require storage, computation, and networking resources distributed at different geographical locations [21]. Multiple high-level fog architectures have been proposed in the literature [22–24] that describe a three-layer architecture containing (1) the smart devices and sensor layer which collects data and send it forward to layer two for further processing, (2) the fog layer applies computational resources to analyze the received data and prepares it for the cloud, and (3) the cloud layer, which performs high intensive analysis tasks.

Bonomi et al. [10] present a fog software architecture (see Figure 2.4) consisting of the following key objectives:

- *Heterogeneous physical resources*. Fog nodes are heterogeneous devices deployed on different components, such as edge routers, access points, and high-end servers. Each component has a different set of characteristics (i.e. RAM and storage) that enables a new set of functionalities. This platform can run on multiple OSes and software applications, deriving a wide range of hardware and software capabilities.
- *Fog abstraction layer*. The fog abstraction layer consists of multiple generic application programming interfaces (APIs) enabling monitoring and controlling available physical resources like CPU, memory, energy, and network. This layer has the role of making accessible the uniform and programmable interface for seamless resource management and control. Furthermore, using generic APIs, it supports virtualization by monitoring and managing multiple hypervisors and OSes on a single machine with the purpose of improving resource utilization. Using virtualization enables the possibility of having multitenancy by supporting security, privacy, and isolation policies to ensure the isolation of different tenants on the same machine.
- *Fog service orchestration layer*. The fog service orchestration layer has a distributed functionality and provides dynamic and policy-based management

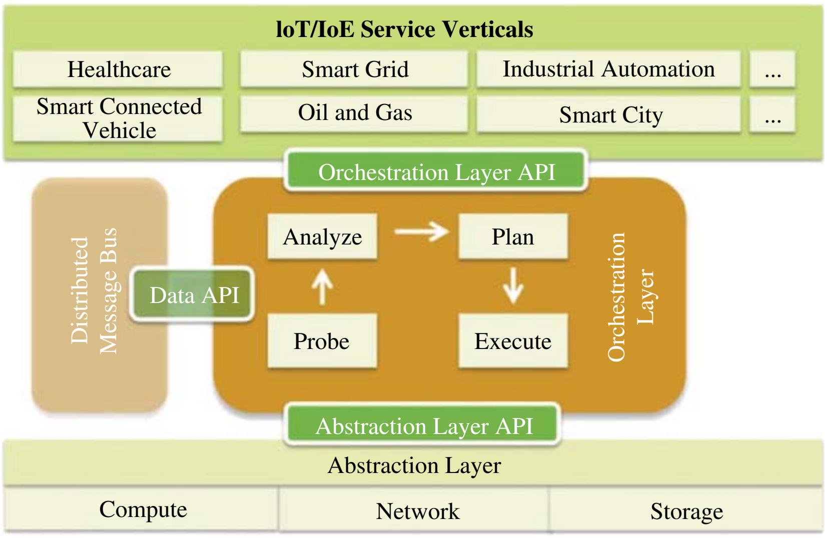

Figure 2.4 Fog computing architecture [10]. (*See color plate section for the color representation of this figure*)

of fog services. This layer has to manage a diverse number of fog nodes capabilities; thus, a set of new technologies and components are introduced to aid this process. One of these components is a software agent called foglet capable of performing the orchestration functionality by analyzing the deployed services on the current fog node and its physical health. Other components are a distributed database that stores policies and resource metadata, a scalable communication bus to send control messages for resource management, and a distributed policy engine that has a single global view and can perform local changes on each fog node.

2.4 Fog and Edge Illustrative Use Cases

In this section, we present two illustrative use cases for both the fog and edge paradigms with the purpose of showing key features, helping us to further understand the concept and applicability in real-world applications.

2.4.1 Edge Computing Use Cases

The rapid growth of big data frameworks and applications such as smart cities, smart vehicles, healthcare, and manufacturing has pushed edge computing among the major topics in academia and industry. With the increasing demand for high availability in such systems, system requirements tend to increase over time. The stringent requirements of IoT systems have recently suggested the architectural placement of a computing entity closer to the network edge. This architectural shifting has many benefits, such as process optimization and interaction speed. For example, if a wearable ECG sensor were to use the cloud instead of the edge it will consistently send all data up to the centralized cloud. As a consequence, in such a scenario it will cause high communication latency and unreliable availability between the sensor and the centralized cloud. In real-time, safety-critical IoT use-cases, devices must comply to stringent constraints to avoid any fatal events. In this scenario, the latency delay introduced by sending the data to the cloud and back is inadmissible. Thus, in case of a critical event detected by the sensor, a local decision must be taken by the edge device, rather than sending data to the cloud.

Edge computing is well suited for IoT deployments where both storing and processing data can be leveraged locally. For example, consider a smart home where the sensory information is stored on the edge device. Simply by doing encryption and storing sensory information locally, edge computing shifts many security concerns from the centralized cloud to its edge devices. In addition, IoT applications consume less bandwidth, and they work even when the connection to the cloud is affected. Furthermore, edge devices may assist in scheduling the operational time of each appliance, minimizing the electricity cost in a smart home [25]. This strategy considers user preferences on each appliance determined from appliance-level energy consumption. All such examples can benefit from the edge computing paradigm and to demonstrate the role of this paradigm in different scenarios, we describe in this section two possible use cases in healthcare [12] and a smart home [3, 15].

2.4.1.1 A Wearable ECG Sensor

This scenario consists of a wearable ECG sensor attached to the human body through a smartwatch and a smartphone that acts as an edge device, as presented in Figure 2.5. As for the communication, Bluetooth is used to connect the ECG

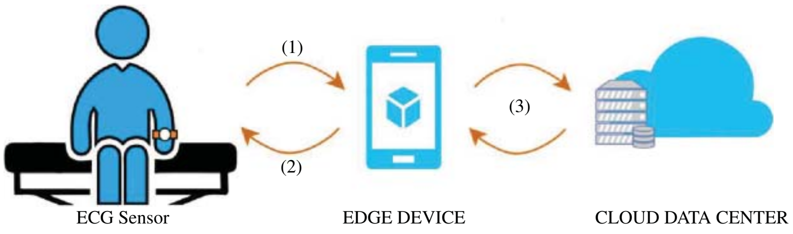

Figure 2.5 A wearable ECG sensor.

sensors with the edge device, while WiFi is used to connect the smartphone to fog devices and cloud.

Generally, users prefer smartwatch devices that provide monitoring heart functions while they continue normal physical activities. Due to the limited battery life and storage capacity of the smartwatch, we assume that the data produced by this device is around 1 KB per second and it is stored in the smartphone. Based on this assumption, daily produced data by a wearable device is around 86 MB per day and 2.6 GB monthly. One must note that smartphones have limited battery life and storage capacity. Hence, the smartphone at some point has to transfer the gathered data to another device that provides more storage capacity.

Referring to Figure 2.5, one can witness that data streaming is realized between the wearable device and the smartphone. Both devices remain connected to each other during the operation time. In case of any critical event, the wearable device interacts with the edge device and notifies the user for any situations. The process (1) start with getting real-time values from a wearable device to the smartphone. The smartphone application checks (2) periodically the wearable device to see if the connection between them is active. In addition, the smartphone may run out of free disk space and one can configure the application for daily synchronization (3) with another storage capability device, or with a central cloud storage or even with a fog node.

Since the wearable device and the edge device has limited resource capabilities, one must consider the energy consumption of both devices. In such system architecture, the first recommended approach is to decide what data to transmit to the cloud, what to store locally, and the last is to develop better monitoring algorithms. In the other words, when designing such systems, the critical point is to consider the energy consumption, which is affected by three main functions that are realized between devices, such as (1) communication, (2) storage, and (3) processing requirements. Hence, developers have to code software with highly efficient streaming algorithms, storing essential monitoring information, and avoiding continuously data transfers with the central cloud.

2.4.1.2 Smart Home

The smart home or smart apartment is an intelligent home network capable of sensing the home's occupants actions, and assisting them by providing appropriate services. In the following scenario, we will describe an example to illustrate a situation under development at the smart home, where an edge device is considered as a mediator between the IoT things deployed in a home environment. The smart home provides resources to the residents that are deployed in rooms. Each room has smart doors, smart windows, sensors (i.e. temperature, humidity, proximity, fire alarm, smoke detector, etc.), radio-frequency identification (RFID) tags, and readers.

The traditional implementation of the presented use case situation requires collecting data to a back-end cloud-based where system stores, processes, and replies to both real-time user queries. The configuration must be done on the cloud server, and each device sends the information to a central server for further processing of the data. Each of the devices contains a unique identification number and a lot of information saved in the cloud. Even if two devices reside near each other, the retrieval of the data must be done through communication with the cloud. A similar implementation of a system for monitoring environmental conditions by using a wireless sensor network (WSN) is given in [26]. However, just adding a WiFi connection to the current network-enabled devices and connecting it to the cloud is not enough for a smart home. In addition, in a smart home environment, besides the connected device, it must support communication with non-IP based resources, cheap wireless sensors, and controllers deployed in rooms, pipes, and even floor and walls.

Deploying a huge number of things in a smart home environment results in an impressive amount of produced data. One must consider that the data produced has to be transported to the processing units, assuring privacy and providing high availability. Since personal data must be consumed in the home, an architecture based only on the cloud computing paradigm is not suited for a smart home. In contrast, edge computing is perfect for building a smart home where data reside on an edge device running edge operating system (edgeOS). As a result, all deployed edge devices can be connected and managed easily and data can be processed locally by an edge device.

Figure 2.6 shows the structure of a variant of edgeOS in the smart home environment. EdgeOS provides a communication layer that supports multiple communication methods, such as WiFi, Bluetooth, ZigBee, or a cellular network. By using one of the methods, edgeOS collects data from the deployed things and mobile devices. In a smart home, most of the things will send data periodically to the edge device, respectively to the edgeOS. Collected data from different things need to be fused and massaged in the data abstraction layer. It is desirable that human interaction with edge devices is minimized. Hence, the edge device should consume/process all the data and interact with users in a proactive fashion. Additionally, the data abstraction layer will serve as a public interface for all things connected to edge devices where it enables the applicability of operations on the things.

Finally, on top of the data abstraction layer is the service management layer. This layer is responsible to guarantee a reliable system including differentiation (i.e. critical services must have higher priority compared to a normal service), extensibility (i.e. new things can be added dynamically), and isolation (i.e. if something crashes or is not responding, the user should be able to control things without crashing the whole edgeOS).

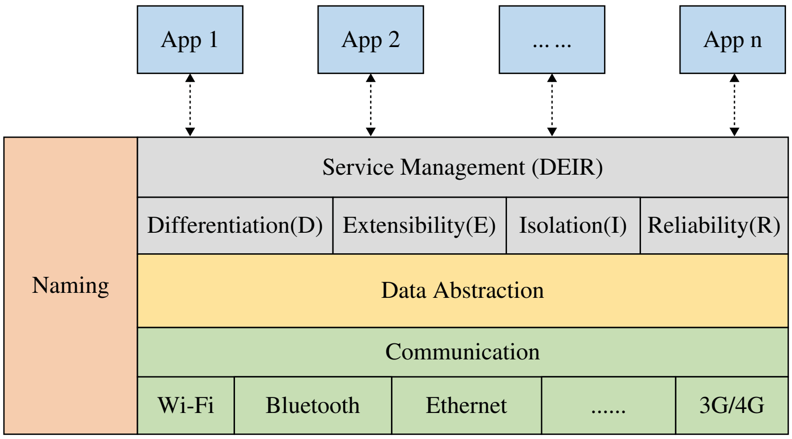

Figure 2.6 Structure of edgeOS in the smart home environment [3].

2.4.2 Fog Computing Use Cases

The new fog computing provides an improved quality of service (QoS), low latency and ensures that specific latency-sensitive applications meet their requirements. There are many areas like the healthcare, oil and gas, automotive, and gaming industries that can benefit from adopting this new paradigm. For example, by performing predictive maintenance the downtime of manufacturing machines can be reduced, optimizing the workflow in a manufacturing plant, or fog computing can simply monitor the structural integrity of buildings, ensuring the safety of workers and clients. However, by implementing such architecture not only businesses can profit. At the same time, life in the city as we know it today can be improved. Multiple day-to-day activities can be optimized to yield better living comfort. For example, consider the following scenario: we can improve congestion on the highway by using smart traffic congestion systems, optimize energy by creating smart grids, and lower the fuel consumption and waiting time in traffic by using a smart traffic light system. All such examples can benefit from this paradigm and, to demonstrate the role of fog in different scenarios, we describe in this section two possible use cases in a smart city, i.e. a smart traffic light system [10] and a smart pipeline monitoring system [27].

2.4.2.1 Smart Traffic Light System

In a smart traffic light system scenario, the objective is to lower the congestion in the city and optimize traffic flow. The immediate outcome of adopting this approach is the protection of the environment by lowering CO_2 emissions and reducing fuel consumption. Enabling an optimization like this requires the implementation of a hierarchical approach that enables real-time and near real-time operations, as well as analysis of data over long periods of time.

Each intersection in the city represents a component of our system where a smart traffic light application is deployed. The application is in charge of analyzing the collected data from local sensors and CCTV cameras and performs three major tasks: (1) compute the distance of every approaching vehicle in all directions and adapt the traffic light accordingly; (2) monitor pedestrians and cyclists to prevent any accidents, and (3) collect relevant data to help improve the overall system performance. Note that these functionalities require fast response time in case of (1) and (2), the exception being the last functionality (3), which only sends data to a higher layer for further investigation, without waiting for a response.

Another important component of our use case is the global node that creates a control function for each intersection. The key role for a global node is to collect all data from each smart traffic light and determine different commands, such that a steady flow of traffic is maintained. Notice that compared with the time requirements for the tasks deployed at an intersection, the functionality here requires a near real-time response.

The aforementioned hierarchical architecture of our traffic light system benefits from the advantages introduced by the fog computing paradigm. An immediate advantage over the cloud computing paradigm is its capabilities of orchestrating a wide range of distributed devices placed at each intersection. At the same time, it enables devices capable of analyzing data and performing fast response-time actions. Our system can be designed as a four-layer architecture, composed by the sensor layer, a fog device layer present locally at each intersection, another fog layer composed of the global node and the cloud layer. An overview of this architecture is presented in Figure 2.7.

2.4.2.2 Smart Pipeline Monitoring System

The smart pipeline monitoring system is an application deployed in the concept of smart cities, with the scope of monitoring the integrity of pipelines and preventing any serious economic and ecologic consequences. As an illustration, consider the case in which a pipeline that transports extracted oil from an offshore platform has failed, and the repercussion of failure has a big impact on the environment.

A pipeline system has an important role in our lives, being an essential infrastructure used to transport gas and liquids. It spreads throughout the entire city and provides us with basic needs like drinkable water. However, the integrity of a pipeline diminishes due to aging and sudden environmental changes. In the end, the risk of failure rises as corrosion and leaks appear.

To prevent such threats, a pipeline monitoring system has the capabilities of detecting any serious threats, reducing the overall failure by predicting three types of emergencies: (1) local disturbances (leakage, corrosion), (2) significant perturbations (damaged pipelines or approaching fire), and (3) city emergency situations. Since the infrastructure covers an entire city, our use case requires an architecture that supports geo-distribution of devices and low/predictable latency.

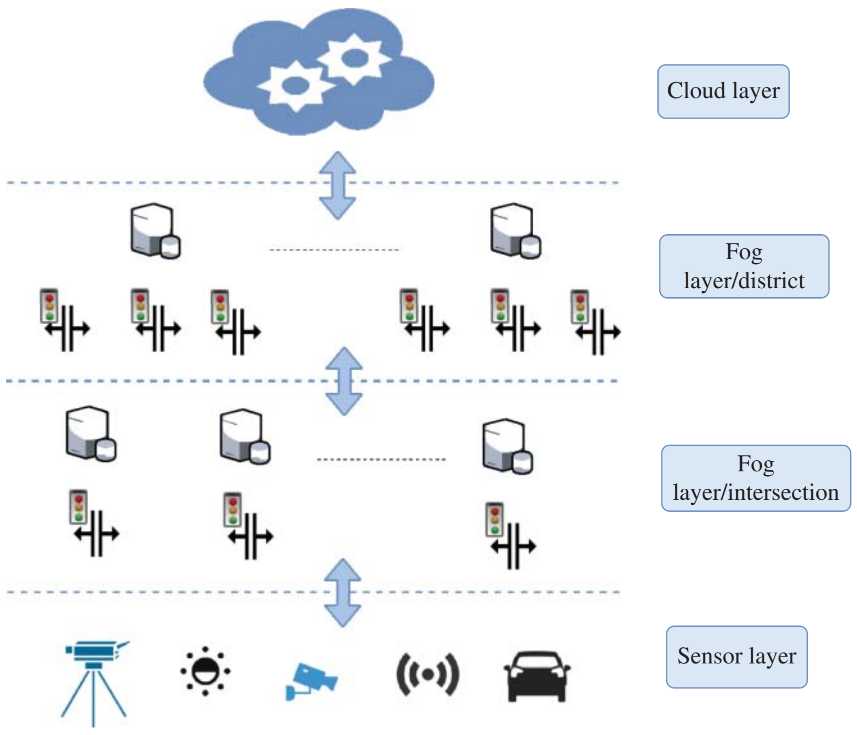

Figure 2.7 Smart Traffic light system.

An immediate solution is a four-layer fog computing architecture, described in Figure 2.8.

As in the case of a smart traffic light system, the fog architecture requires distinct layers to enable different time-scale responses. Depending on how wide the monitored area is, four or more layers can be used. The first layer represents the cloud and has the purpose of offering a global perspective of the entire system. In this layer, hard computational analysis is performed to prevent and respond to citywide disasters. Since the layer performs analysis on historical data, it does not require real-time or near real-time responses. Following, the second layer represents fog devices that are responsible to prevent any failure, at a smaller scale than cloud, in every neighborhood. The key purpose behind this layer is to identify potential hazardous events, based on the collected measurements from multiple devices. In this situation, near real-time responses are required for better prediction. Next, the third layer is composed of local fog nodes that identify potential threats based on the data received from the sensors and act by outputting control signals when it is necessary for fast recovery from a found threat. Furthermore, these devices

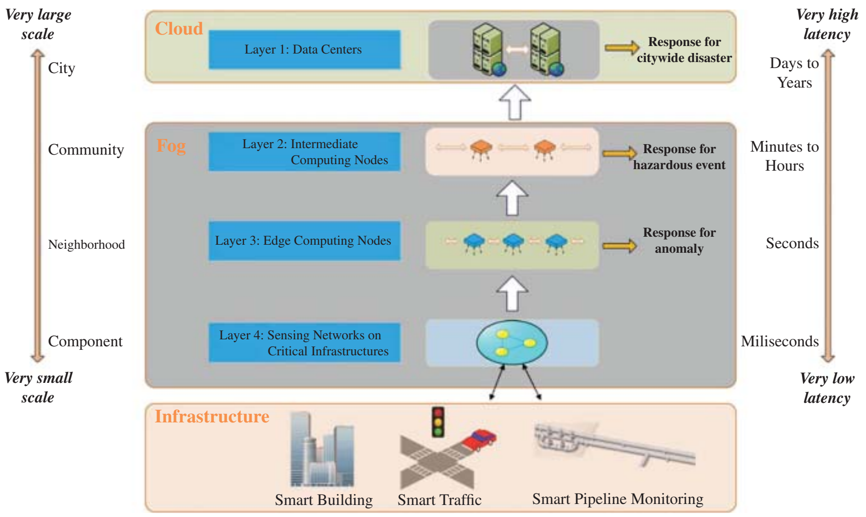

Figure 2.8 Smart pipeline monitoring system architecture.

process the raw data and prepare it for the layer above. Finally, the last layer is represented by sensors that generate measurements for the aforementioned layers in the architecture.

2.5 Future Challenges

Fog and edge computing vision introduce multiple advantages by migrating some computational resources at the edge of the network. The underlining of these paradigms is to create an IoT network environment covered with a vast amount of interconnected distributed heterogeneous devices having the purpose to deploy and manage demanding applications closer to the user. Yet, it is a nontrivial task to design platforms where all these required characteristics are met.

In this section, we identify and discuss the challenges that these paradigms must conquer in order to fulfill their full potential. We group these challenges in three main areas, i.e. resource management, security, and privacy, and network management.

2.5.1 Resource Management

Moving computational resources from the cloud closer to the end nodes stand in the center of the fog and edge paradigm. Therefore, novel resource management to fully utilize the available resources and process applications in close proximity of the user is imperative to the successful adoption of these systems. Since IoT devices are resource-constrained devices, applying resource management techniques at the edge will allow edge nodes to optimize their resource utilization (e.g. energy-aware smart devices that increase their battery levels by of loading computation to other nearby nodes), improve data privacy, and enable devices to collaborate and share resources to process IoT applications.

A taxonomy of resource management at the edge, based on the current state-of-the-art research in this area, is presented in [28]. According to this classification, a total of five different categories are identified considering the objective of the technique.

The first category refers to resource estimation and represents one of the fundamental requirements in resource management, i.e. the capability of estimating how many resources a certain task requires. This is important for handling the uncertainties found in an IoT network and providing at the same time a satisfactory QoS for deployed IoT applications. The second category is represented by resource discovery and aims to aid the user to discover available resources already deployed at the edge. Resource discovery complements resource estimation by keeping the pool of available computational resources updated.

Once the system can estimate and discover resources, a third category appears having the purpose of allocating IoT applications in close proximity to the users. This technique, called resource allocation, utilizes the knowledge of available resources to map parts of the applications at different edge devices such that its requirements are met. There are two different perspectives of the allocation: (1) it represents the initial deployment to the edge of the network, deciding where to map the application; and (2) it serves as a migration technique by self-adapting when a node has failed. Moreover, one challenge arises when sharing resources between distributed edge devices, i.e. a close collaboration between nodes enforced by security and privacy is required. Solving this challenge creates the fourth category, i.e. resource sharing.

Finally, the last technique is called resource optimization and is obtained by combining the aforementioned resource management approaches. The main objective is to optimize the usage of available resources at the edge according to the IoT application constraints. Usually, the developer creates the QoS requirement of his application before deploying it to the edge.

2.5.2 Security and Privacy

Adopting the vision of fog and edge computing, more applications that today reside in the cloud are moved to the edge of the network. By deploying and connecting IoT devices, we can transform our homes in a more digitalized environment that adapts automatically, based on our behavior. However, with such benefits arise a set of privacy and security issues that we must address. For example, one can easily study the behavior of a family by simply accessing the generated data from sensors deployed in the house. Hence, ensuring data privacy and security remains a crucial factor in the evolution of edge and fog paradigms.

To evaluate the security and privacy enforced in systems based on fog and edge devices, the designer can use the confidentiality, integrity, and availability (CIA) triad model, representing the most critical characteristics of a system [29]. While any breach of the confidentiality and integrity components yields a data privacy issue, the availability component refers to the property of the nodes to share their resources when required. Since fog and edge represents an extension of the cloud, such systems inherit not only the computational resources but also the security and privacy challenges. Besides these challenges, due to the deployment of devices at the edge of the network more security challenges appear. Yi et al. identify the most important security issues of fog computing as authentication, access control, intrusion attack, and privacy [9].

Considering the dynamic structure of an IoT network, authentication is an important key feature of fog and edge computing and was identified. as the main security issue in fog computing [20]. The authentication serves as the connectivity mechanism that allows to securely accept new nodes into the IoT network. By providing means to identify each device and establish its credentials, a trust is created between the new added node and network. The current security solutions proposed for cloud computing may have to be updated for fog/edge computing to account for threats that do not exist in its controlled environment [21]. One solution to securely authenticate edge devices is presented in [30].

A comprehensive study of security threats for edge paradigms (i.e. fog and edge computing, and MEC, among others) was presented in [31], where the importance of security is motivated for the overall system and each individual component. An edge ecosystem consists of different edge nodes and communication components, ranging from wireless to sensors and Internet-connected mobile devices, distributed in a multilayer fog architecture. While each individual component has its own security issues, new different security challenges appear by combining and creating an edge ecosystem. By reviewing the scope and nature of potential security attacks, the authors propose a threat model that analyzes possible security risks (see Table 2.1).

For this model, the authors in [31] discover all important components of edge paradigms and describe all attacks that can occur against them. As depicted from Table 2.1., we can observe that five different targets i.e. network infrastructure, service infrastructure composed of edge data center and core infrastructure, virtualization infrastructure and user devices [31] are identified. The network infrastructure represents the various communication networks that connect edge devices which an adversary can attack using one of the following: denial

Table 2.1 Threat model for fog and edge computing [21].

Fog components / **Security issues**	**Network infrastructure**	**Service infrastructure (edge data center)**	**Service infrastructure (core infrastructure)**	**Virtualization infrastructure**	**User devices**
DoS	✓			✓	
Man-in-the-middle	✓				
Rogue component (i.e. data center, gateway, or infrastructure)	✓	✓	✓		
Physical damage		✓			
Privacy leakage		✓	✓	✓	
Privilege escalation		✓		✓	
Service or VM manipulation		✓	✓	✓	✓
Misuse of resources				✓	
Injection of information					✓

of service (DoS), man-in-the-middle attacks, and rogue datacenter. An example of a man-in-the-middle attack on an IoT network is presented in [32]. On the one hand, an adversary could attack the service infrastructure, at the edge of the network, by using physical damage, rogue component privacy leakage, privilege escalation, and service or virtual machine (VM) manipulation. On the other hand, the core infrastructure is more secure being prone to attacks like rouge component, privacy leakage, and VM manipulation [31]. Finally, the virtualization infrastructure is exposed to attacks, such as DoS, privacy leakage, privilege escalation, service or VM migration, and misuse of resources; while user devices are susceptible to attacks like VM manipulation and injection of information.

Privacy, defined as the protection of private data, ensures that a malicious adversary cannot obtain sensitive information while data is in transit [33]. At the moment, privacy is most vulnerable since the data of end users is sent directly to the cloud. From this point of view, edge and fog paradigms enforce privacy by moving the computation closer to the user. In doing so, data can be processed locally and the user can control what third parties are accessing his private data based on a defined role-based access control policy. However, some privacy challenges remain open, such as (i) the awareness of privacy in the community where, for example, almost 80% of WiFi user still use their default passwords for their routers and (ii) the lack of efficient tools for security and privacy for constrained devices [3].

2.5.3 Network Management

The network management plays the most important role in both edge and fog paradigms since it represents the means of connecting all smart devices at the edge and ultimately providing available resources by deploying more nodes. Since the nature of an IoT network consists of heterogeneous devices, which are highly dispersed across large areas, an engaging task is to manage and maintain connectivity. Newly emerging technologies like software-defined networks (SDNs) and network function virtualization (NFV) are seen as a possible solution that may have a significant impact in implementing and maintaining the network increasing the scalability and reducing cost [19].

Considering the volatile nature of the network, providing a seamless connectivity mechanism is critical since both mobile and stationary devices coexist in the network. Therefore, another aspect of network management is related to connectivity. This mechanism must be able to provide the possibility of connecting/disconnecting easily from the network such that the uncertainty introduced by mobile devices is accommodated. Moreover, providing this encourages an increased deployment of smart devices by users and manufacturers alike, without extra cost or expert knowledge.

An effort in this direction is made by the I3: the intelligent IoT integrator, developed by USC [34], having the purpose of creating a marketplace where users can share their private data with application developers and receive incentives for it. There are two main advantages of designing the marketplace like this: first, the users are encouraged to deploy more edge devices, which in return extends the IoT network with more resources that app developers can use; and second, there is a pool of data that developers can utilize to improve their IoT applications.

2.6 Conclusion

The never-ending increase in interconnected IoT devices and the stringent requirements of new IoT applications has posed severe challenges to the current cloud computing state-of-the-art architecture, such as network congestion and privacy of data. As a result, researchers have proposed a new solution to tackle these challenges by migrating some computational resources closer to the user. The approach taken in this solution made the cloud more efficient by extending its computational capabilities at the end of the network, solving its challenges in the process.

Continuing to improve this solution, multiple paradigms appeared, having as their underlying vision the same goal of deploying more resources at the edge of the network. Besides their common vision, some paradigms were influenced by their considered use case, e.g. MEC paradigm enables constrained devices like smartphones to offload parts of the applications to save resources. However, two of the most popular paradigms (i.e. fog and edge computing) are widely used in research today.

These two paradigms were designed to enable processing IoT applications at the endpoints of the network, sharing more similarities than others. Other than the naming convention, the difference at the beginning for the two, i.e. fog computing extends the cloud creating a cloud-to-things continuum and edge computing places the application directly on the edge devices, was represented by the location where computations are performed. Since in the past couple of years there were tremendous advances for edge devices, this difference between the two has disappeared, both fog and edge aiming to deploy applications as close as possible to the edge of the network. Considering the similarities they share, we argue that there is no difference between their purpose of them.

Acknowledgment

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). This publication was partially supported by the TUW Research Cluster Smart CT.

References

1 Chiang, M. and Zhang, T. (2016). Fog and IoT: an overview of research opportunities. *IEEE Internet of Things Journal* 3 (6): 854–864.

2 Bonomi, F., Milito, R., Zhu, J., and Addepali, S. (2012). *Fog computing and its role in the Internet of Things, 1st ACM Mobile Cloud Computing Workshop*, 13–15.

3 Shi, W., Cao, J., Zhang, Q. et al. (2016). Edge computing: vision and challenges. *IEEE Internet of Things Journal* 3 (5): 637–646.

4 Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing* 8 (4): 14–23. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2009.82 http:// http://ieeexplore.ieee.org/document/5280678.

5 Rausch, T., Avasalcai, C., and Dustdar, S. (2018). Portable energy-aware cluster-based edge computers. In: *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 260–272.

6 Elias, A.R., Golubovic, N., Krintz, C., and Wolski, R. (2017). 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)*, 247–258.

7 M. T. Beck, M. Werner, S. Feld, and S. Schimper, Mobile edge computing: a taxonomy. Citeseer.

8 Fernando, N., Loke, S.W., and Rahayu, W. (2013). Mobile cloud computing: a survey. *Future Generation Computer Systems* 29 (1): 84–106, including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X12001318.

9 Yi, S., Li, C., and Li, Q. (2015). A survey of fog computing: concepts, applications and issues. In: *Proceedings of the 2015 Workshop on Mobile Big Data*, 37–42. ACM.

10 Bonomi, F., Milito, R., Natarajan, P., and Zhu, J. (2014). *Fog Computing: A Platform for Internet of Things and Analytics*, 169–186. Cham: Springer International Publishing [Online]. Available: https://doi.org/10.1007/978-3-319-05029-4 7.

11 Shi, W. and Dustdar, S. (2016). The promise of edge computing. *Computer* 49 (5): 78–81.

12 Gusev, M. and Dustdar, S. (2018). Going back to the roots|the evolution of edge computing, an IoT perspective. *IEEE Internet Computing* 22 (2): 5–15.

13 Pate, J. and Adegbija, T. (2018). Amelia: an application of the Internet of Things for aviation safety, in 15th. In: *IEEE Annual on Consumer Communications & Networking Conference (CCNC), 2018*, 1–6. IEEE.

14 Chen, B., Wan, J., Celesti, A. et al. (2018). Edge computing in IoT-based manufacturing. *IEEE Communications Magazine* 56 (9): 103–109.

15 Zhang, S., Li, W., Wu, Y. et al. Enabling edge intelligence for activity recognition in smart homes. In: *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, vol. 2018, 228–236. IEEE.

16 Yu, W., Liang, F., He, X. et al. (2018). A survey on the edge computing for the Internet of Things. *IEEE Access* 6: 6900–6919.

17 Chen, Z., Xu, G., Mahalingam, V. et al. (2016). A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research* 3: 10–23.

18 Xu, X., Sheng, Q.Z., Zhang, L.-J. et al. (2015). From big data to big service. *Computer* 48 (7): 80–83.

19 Yi, S., Hao, Z., Qin, Z., and Li, Q. (2015). Fog computing: platform and applications. In: *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb) (HOTWEB)*, vol. 00, 73–78. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/HotWeb.2015.22.

20 Stojmenovic, I. and Wen, S. (2014). The fog computing paradigm: scenarios and security issues. In: *2014 Federated Conference on Computer Science and Information Systems*, 1–8.

21 Osanaiye, O., Chen, S., Yan, Z. et al. (2017). From cloud to fog computing: a review and a conceptual live VM migration framework. *IEEE Access* 5: 8284–8300.

22 Dastjerdi, A.V. and Buyya, R. (2016). Fog computing: helping the Internet of Things realize its potential. *Computer* 49 (8): 112–116.

23 Sarkar, S., Chatterjee, S., and Misra, S. (2018). Assessment of the suitability of fog computing in the context of Internet of Things. *IEEE Transactions on Cloud Computing* 6 (1): 46–59.

24 Shi, Y., Ding, G., Wang, H. et al. (2015). The fog computing service for healthcare. In: *2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, 1–5.

25 Xia, C., Li, W., Chang, X. et al. (2018). Edge-based energy management for smart homes. In: *2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, 849–856. IEEE.

26 Bajrami, X. and Murturi, I. (2018). An efficient approach to monitoring environmental conditions using a wireless sensor network and nodemcu. *e & i Elektrotechnik und Informationstechnik* 135 (3): 294–301. [Online]. Available: https://doi.org/10.1007/s00502-018-0612-9.

27 Tang, B., Chen, Z., Hefferman, G. et al. (2017). Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial Informatics* 13 (5): 2140–2150.

28 Tocze, K. and Nadjm-Tehrani, S. (2018). A taxonomy for management and optimization of multiple resources in edge computing. *Wireless Communications and Mobile Computing* 2018, Art. No: 7476203: 1–23.

29 Farooq, M.U., Waseem, M., Khairi, A., and Mazhar, S. (2015). A critical analysis on the security concerns of Internet of Things (IoT). *International Journal of Computer Applications* 111 (7).

30 Puthal, D., Obaidat, M.S., Nanda, P. et al. (2018). Secure and sustainable load balancing of edge data centers in fog computing. *IEEE Communications Magazine* 56 (5): 60–65.

31 Roman, R., Lopez, J., and Mambo, M. (2018). Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78: 680–698. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16305635.

32 Wang, Y., Uehara, T., and Sasaki, R. (2015). Fog computing: issues and challenges in security and forensics. In: *2015 IEEE 39th Annual Computer Software and Applications Conference*, vol. 3, 53–59.

33 Zhou, M., Zhang, R., Xie, W. et al. (2010). Security and privacy in cloud computing: a survey. In: *2010 Sixth International Conference on Semantics, Knowledge and Grids*, 105–112.

34 University of Southern California, I3: The intelligent IoT integrator (i3), https://i3.usc.edu.