2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)

Edge and Fog Computing: Vision and Research Challenges

Schahram Dustdar
Distributed Systems Group
TU Wien
Vienna, Austria
dustdar@dsg.tuwien.ac.at

Cosmin Avasalcai
Distributed Systems Group
TU Wien
Vienna, Austria
c.avasalcai@dsg.tuwien.ac.at

Ilir Murturi
Distributed Systems Group
TU Wien
Vienna, Austria
i.murturi@dsg.tuwien.ac.at

***Abstract*—Recently, the wide adoption of Internet of Things (IoT) devices has introduced new challenges that the current cloud-centric approach must overcome. The high-latency obtained from sending daily massive volumes of generated data to the cloud, for further processing, is insufficient to meet the stringent requirements of emerging IoT applications. As a consequence, researchers have introduced new paradigms, like edge and fog computing, with the purpose of extending cloud capabilities closer to the edge of the network. This extension of the cloud enables IoT applications to be deployed in the proximity of sensors, adding new benefits like fast response time and better security and privacy. In this paper, we discuss in detail both paradigms based on their individual characteristics and use cases. Furthermore, we explain what future challenges, i.e., resource management, security and privacy, and network management, researches must solve to enable the adoption in society. Finally, we present our vision regarding a smart city scenario in which users have the possibility of customizing their environment as they desire, by seamlessly downloading applications on a personal edge network.**

***Index Terms*—Edge Computing, Fog Computing, Internet of Things**

I. Introduction

Cloud computing as one of the biggest advances over the last decade in technology has been seen as a key component for the development, deployment, and execution of IoT platforms where companies can move their control, computing capabilities, and store collected data in a medium with almost ”unlimited” resources [1]. It remains as one of the well-accepted solution for the deployment of demanding computational applications focusing mainly on the processing of large amounts of data. Data is generated from geographically distributed IoT devices such as sensors, smartphones, laptops, and vehicles. Today, however, this paradigm faces increasing challenges in meeting the demanding limitations of new IoT applications.

The success of the Internet of Things and the widespread availability of mobile devices featuring sensing capabilities provide new means for the development of new applications that affect our everyday 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 & Gas, Automotive, etc.) by improving working environments and optimizing workflows. Due to the dynamic nature that prevails in these environments many applications require fast response time and increased privacy clouds often fail to fulfill.

To overcome these shortcomings, researchers, both from academia and industry, proposed two new paradigms, called Fog Computing and Edge Computing, which bring the computational resources (i.e., storage, networking and processing) closer to the edge of the network. Fog Computing brings cloud capabilities closer to the end devices such that a cloud to things continuum is obtained [2]. As a consequence, this reduces the reliance on cloud-based environments while decreases latency and network congestion. Furthermore, it enforces privacy by processing the data near the user. Similarly, the vision of Edge Computing is to move some computational resources from the cloud to the resource constraint devices located at the logical extremes of a network [3]. Hence, we vision the edge computing as a bridge between IoT things and the nearest edge device (i.e., smartphones, etc.) to the user.

Researchers have proposed new fog/edge devices in order to embrace the vision of these paradigms and focus on the deployment of multiple applications in close proximity to users. The most noteworthy of these devices 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 [7] and Mobile Cloud Computing [8] improve user experience and enable the deployment of higher computational applications on smartphones by offloading different parts of the application on the device locally.

During the literature review, we found many surveys that describes each paradigm in details and their challenges [9]–[11]. However, there is no paper comparing fog and edge paradigm while both terms are most often used to describe the same IoT network. In general, the vision of the two paradigms overlaps in order to make more computation resources available at the edge of the network. Hence, the biggest difference is given by the naming convention used to describe them. The aim of this paper is to offer a detailed description of the two aforementioned paradigms, discussing their differences and similarities. Furthermore, we present their challenges and discuss the different naming convention is still required. Finally, we present an IoT platform which combines both, edge and fog devices with the purpose of enabling

978-1-7281-1442-2/19/$31.00 ©2019 IEEE
DOI 10.1109/SOSE.2019.00023

seamless deployment of IoT devices by the user.

The remainder of the paper is structured as follows: In Section II we present in detail the Fog Computing paradigm and describe one illustrative use case by emphasizing the key features of this architecture. Next, Section III defines the Edge Computing paradigm by describing its architecture features. In Section IV we discuss challenges faced by these paradigms, while in the process to be fully adopted in society. Section V presents our vision of the future of smart cities. Finally, Section VI outlines on the comparison between Fog and Edge Computing.

II. Fog Computing

Fog Computing is a computing paradigm introduced by Bonomi et al. [10] with the purpose of extending the cloud capabilities closer to the edge of the network. Several definitions have been proposed to formally define fog computing, e.g., Yi et al. [12]. According to the authors, Fog Computing is a geographically distributed computing architecture connected to multiple heterogeneous devices which allows the provision of resources and services at the edge of the network without depending on cloud services. Hence, we vision the fog paradigm as a bridge between the cloud and the edge of the network to facilitate the deployment of new IoT applications (see Figure 1).

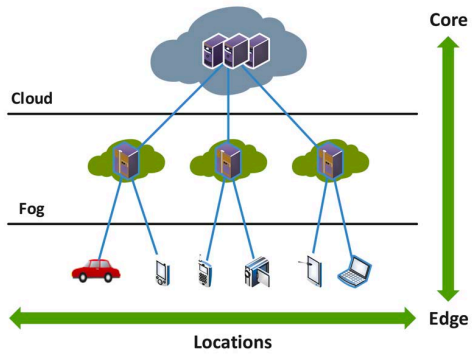

Fig. 1. Fog computing a bridge between Cloud and Edge [13]

A fog device is mainly characterized by its highly virtualized resources that provide computation, storage, and network services between edge devices and cloud [2]. Generally speaking, a fog device can be characterized as a mini cloud that uses its own resources in combination with data collected from the edge devices and unlimited computer resources offered by the cloud. However, even similar features are also found in the cloud environment it should be noted that in contrast to cloud computing, fog nodes provide limited resources.

Fog computing offers the opportunity to develop and deploy new latency-sensitive applications on devices with computation capabilities such as network devices, mini data centers, servers. Depending on the application context, this can enforce strict requirements for rapid response time and predictable latency (i.e., smart connected vehicles, augmented reality), location awareness (e.g., sensor networks to monitor the environment conditions) and large-scale distributed systems (smart traffic light, smart grids). As a result, for latency-sensitive applications, long propagation latency (WAN) and real-time requirements for mobile scenarios make cloud environments incapable to fulfill the user expectations. Hence, fog computing mitigates the operation of end devices with cloud computing data centers.

The current state of the art does not meet the stringent requirements for latency-sensitive applications [1]. In order to create a scalable and stable system between edge devices and cloud environments suitable for IoT applications, a cloud-fog interaction is therefore introduced. Such an approach, where the cloud and fog collaborate for achieving a specific user goal and preserving user experience gives an opportunity to the developers to decide where to deploy and compute a function of the application. For example, using the capabilities of the fog node, we can process and filter data streams from heterogeneous devices in different areas, make real-time decisions and reduce the communication network to the cloud. Fog Computing, therefore, introduces effective ways to overcome many of the limitations facing the cloud [1]. These limitations are:

1) **Latency Constraints**. Fog nodes supports the same basic features that clouds can perform, i.e., different computing tasks closer to the end user, where the latency - sensitive application benefits most from its stringent requirements.
2) **Network Bandwidth constraints**. Fog paradigm offers the ability to carry out data processing tasks closer to the network edge. As a consequence, the amount of raw data sent to the cloud is reduced. Therefore, performing data analysis in fog devices reduce the latency in response while filtered data is sent to the cloud for long-term storage.
3) **Resource constrained devices**. For resource constrained edge devices such as smartphones and sensors, fog computing can perform computational tasks. The energy consumption and life-cycle costs are reduced by discharging parts of the application from such restricted devices to nearby fog nodes.
4) **Increased availability**. Fog Computing supports autonomous operation without depending on the cloud's network connectivity. As a result, an application is increasingly available and reliable.
5) **Better security and privacy**. In the fog paradigm users have the ability to control the collected data. Fog devices reduce the need to transfer private data toward the cloud while such sensitive data stays and it is processed locally. Hence, fog devices increase security as well, being able to perform a wide range of security functions, manage and update the security credentials of constrained devices and monitor the security status of nearby devices. In addition, data integrity and privacy are ensured even more when data have to travel shorter network distance to reach the computation node.

A. Fog Computing Architecture

The Fog Computing architecture consists of highly dispersed heterogeneous devices designed to enable the deployment of IoT applications requiring storage, computing and networking resources distributed at various geographical locations [14]. Several high-level fog architectures have been proposed in the literature [15]–[17] which describe a three-layer architecture containing (i) the physical layer also known as the smart devices and sensors layer collects data and send it forward to the nearest layer for further processing, (ii) in the fog layer, the received data is computed, responded to the user and prepared for the cloud (iii) the cloud layer stores data for long-term and performs high intensive analysis tasks.

Bonomi et al. [10] present a fog software architecture (see Figure 2) consisting of the following key objectives:

- **Heterogeneous physical resources**. In the fog paradigm, we refer to the fog nodes as heterogeneous devices such as network devices (i.e., routers, access points), data centers or even high-end servers. This layer is composed with devices with different hardware capabilities (i.e., CPU, RAM, and storage) and may provide a set of functions specific to the device. The platform is available for multiple operating systems and software applications, resulting in a wide range of hardware and software capabilities.
- **Fog abstraction layer**. This layer consists of generic application programming interfaces (APIs) which enables managing the physical resources of fog device. We refer to the management of resources in terms of monitoring and controlling the available physical resources such as a CPU, RAM, storage, energy, and network. The role of this layer is to make a uniform and programmable interface accesible for the seamless management and control of resources. In addition, this layer also supports virtualization and allows multiple hypervisors and operating systems to be managed on a single machine using generic application programming interfaces (APIs). Moreover, the use of virtualization enables multi-tenancy to ensure the isolation of different tenants on the same machine by supporting security, privacy and isolation policies.
- **Fog service orchestration layer**. This layer has a distributed functionality and provides dynamic and policy-based management of fog services. In addition, fog service orchestration layer is responsible for managing a variety of fog node capabilities, thereby a set of new components that help this process are introduced. A well-known software agent component called foglet is capable to manage the capabilities of the fog node in such a way that it can monitor physical health of the device and can orchestrate functionality by analyzing current deployed services. There are other components such as a distributed database which is responsible to store policies and resource meta-data, a scalable communication bus to send control messages for resource management, and a distributed policy engine with a single global view that can change each fog node locally.

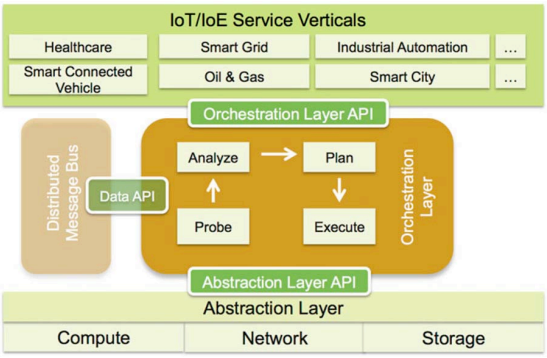

Fig. 2. Fog Computing architecture [10]

B. Illustrative use case

The Fog Computing paradigm has improved the user experience by increasing the Quality of Service (QoS), providing low latency and ensuring that specific applications which are sensitive to latency meet their strict requirements. Many areas such as Healthcare, Energy, Automotive, and Gaming Industry can benefit from this new paradigm. As an example, predictive maintenance can reduce the downtime of production machines, optimize the workflow in a production plant or simply monitor the structural integrity of buildings that ensure the safety of workers and customers. However, the benefits provided by the fog computing paradigm are not limited only for businesses purposes. At the same time, as the development of the smart cities continue, the life in the city as we know it today can be improved even further. Daily activities can be optimized to improve the comfort of living. For example, consider the following scenario, by using smart traffic systems we can avoid road congestion, while smart traffic light system can even help more to manage congested roads, reduce fuel consumption and minimize the waiting time. In order to demonstrate the role of fog paradigm in different scenarios, in this section we describe a smart traffic light system [10].

1) Smart Traffic Light System: The objective of an intelligent traffic light system scenario is to reduce city congestion and optimize traffic flow. The immediate result of this approach is environmental protection by reducing harmful emissions (i.e., nitrogen oxides, carbon monoxide, sulfur dioxide) and reducing fuel consumption. Hence, such an optimization requires the implementation of a hierarchical approach which supports both real-time and near real-time operations, as well as providing an environment which supports performing high computation tasks like big data analysis.

Each intersection in the city represents a component of our system where a smart traffic light application is deployed. The application is responsible for the analysis of data collected from local sensors and CCTV cameras and carries out three main tasks, (i) the traffic light is adjusted based on the distance of each approached vehicle from different directions

(ii) pedestrians and cyclists are monitored in order to prevent any accidents and (iii) relevant data is collected to improve the overall performance of the system. One can clearly notice that the functionalities provided in case of (i) and (ii) require rapid response time, while the last functionality (iii) sends data to a higher layer for further analyzing without waiting for a response.

The global node that creates a control function for each intersection is another important component of the presented use case. The key role of a global node is to collect all data from each smart traffic light and determine various commands in order to maintain steady traffic flow. Note that the functionality here requires an almost real-time response compared to the time requirements for the tasks deployed at an intersection.

The Fog Computing paradigm enables implementation of our traffic light system where stringent requirements are fulfilled. As one can notice an immediate advantage over the centralized architectures is the ability to coordinate a wide range of distributed devices at each intersection. In the meantime, fog devices can use their computational resources to analyze data and carry out quick response time actions. Our system can be designed as a four-layer architecture, composed by the sensor layer, a fog device layer at each intersection, another fog layer composed of the global node and the cloud layer. An overview of this architecture is presented in Figure 3.

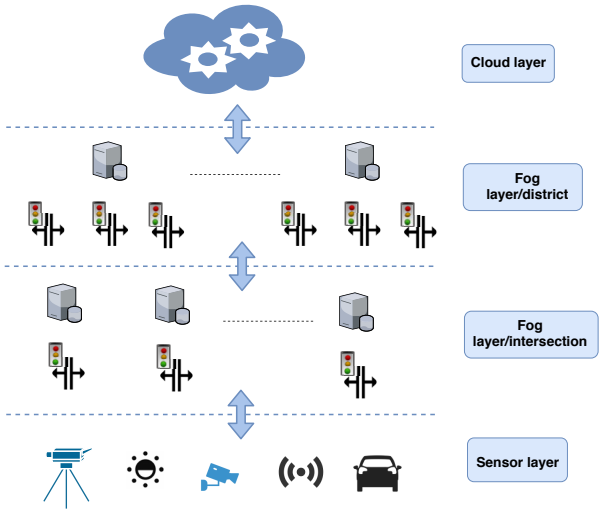

Fig. 3. Smart Traffic Light System.

III. Edge Computing

Edge Computing [3] is a new paradigm with the underlying vision of migrating computational resources from the cloud closer to the edge of the network. Multiple definitions of edge computing are found in the literature, [11] states that edge computing enables computations to be performed at the edge, offering benefits for both cloud and IoT services alike. Deploying parts of IoT applications on such edge devices, not only reduce network congestion and bandwidth waste that the current cloud-centric state of the art faces, but new IoT applications with more stringent requirements suchlike fast response time, better data privacy, and increased availability can be deployed. As we can see, by lowering the physical distance between the applications and user, these requirements can be successfully satisfied.

The proposed paradigm is a relatively new concept, hence the term ”edge computing” in literature may refer to all other architectures as well such as Mobile Edge Computing, or Fog Computing. From all of them, the edge computing concept is interchangeable with fog computing [18]. The key difference being the location into the IoT network where processing of data is performed. In the case of fog computing, the data is processed as close as possible to the end user devices, while edge computing pushes the limits even further by allowing personal devices like smartphones or laptops to process some data locally.

Considering edge vision and its similarities to fog, we envision Edge Computing the lowest level in an IoT network where new emerging IoT applications can be deployed in user devices like smartphones (see Figure 4).

Fig. 4. Edge computing solution using an IoT and edge devices. [19]

One example of how edge computing changes the IoT network (see Figure 4) was presented in [19], where the authors place an edge device, i.e., a smartphone, between the IoT sensors and actuators level, and the rest of the network where more powerful devices reside. It is important to mention that, any other device, e.g., desktop computers, laptops, and tablets, capable of processing locally gathered data, can be considered an edge node. Every such device has different characteristics which define what functionality can be performed, being capable of controlling IoT actuators based on input data and processing data for high levels like fog nodes and cloud.

In conclusion, edge computing can be considered a key enabler for scenarios where centralized cloud-based platforms are impractical. Processing data near to the logical extremes of a network reduces significantly latency and bandwidth cost while decreasing the distance data has to travel. As a consequence, this paradigm can address concerns in energy consumption, security, and privacy [20]. However, it is not a trivial task to adopt the edge computing paradigm, since there are multiple challenges that must be overcome.

A. Edge Computing Architecture

Every edge computing architecture can have different components and specifications depending on the targeted use case. For example, Jiafu Wan et al. [21] proposes an architecture

of edge computing for IoT-based manufacturing and analyzes its role from four different points of views, including edge equipment, network communication, information fusion, and cooperative mechanism with Cloud Computing. On the other hand, Zhang et al. [22] propose the edge-based architecture and implementation for the smart home scenario composed of three functional layers, the sensor layer, the edge layer, and the cloud layer.

A more general overview of an edge computing architecture is shown in Figure 5. Such an architecture can be structured into three different layers, the front-end, near-end, and far-end as described by Wei Yu et al. in [23]. A detailed description of every layer is given below:

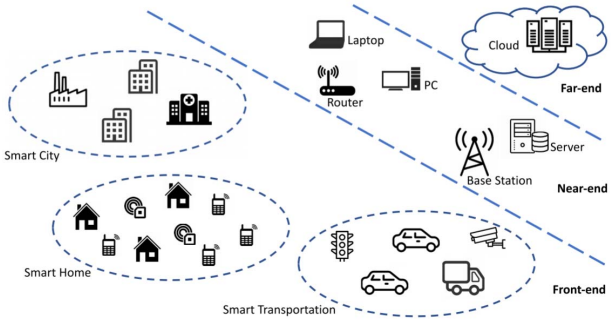

Fig. 5. A typical architecture of edge computing networks [23]

- **The front-end** represents the closest layer to the end user capable of processing locally collected data from IoT sensors. Due to being in close proximity, this layer provides real-time response times for critical applications and better privacy since private data is processed in close proximity of the user. However, the devices at this layer are resource constrained devices which cannot provide sufficient resources to met all requirements of an application. As a result, these devices pre-process the data and forward the result to a higher level layer.
- **The near-end** layer is composed of more powerful edge devices, e.g., servers and laptops, capable of handling most of the data processing and storage required by an IoT application. However, these devices are further away from the source of data, meaning that the system can provide only near real-time responses. Most of the time, this extra layer is enough to ensure that an IoT application can perform at the edge. Nonetheless, if there is a case when more computational resources are needed, then the data is sent to the last layer.
- **The far-end** layer represent the cloud servers which provide an almost unlimited amount of high processing capabilities and storage. Moreover, it can provide different levels of security and control for users and developers as well, offering them access to the thousands of servers to perform a task [24], [25]. However, the latency of an IoT application faces increased delays, since these servers are farther away from the end devices.

B. Illustrative use case

Recently, the fast adoption of IoT devices and the emergence of new IoT applications for use cases like smart cities, healthcare, automotive, and manufacturing has made edge computing paradigm as one of the major topics in academia and industry alike. Each system requires that these new IoT applications demand rigid requirements that the cloud cannot satisfy. As a consequence, the requirements of these applications can benefit from the architectural placement of edge devices closer to the end user. This shift in deploying an application introduces many benefits such as fast response time and increased availability. Both critical in use cases like healthcare or smart city where reacting to an event must be performed in real-time. One example of an application that can benefit from this paradigm is a wearable ECG sensor. In this case, if the data is processed in the cloud instead of the edge, a high communication latency and low-reliability characteristics are obtained. In real-time applications, such characteristics are not accepted, since the wait time could prove to be fatal to the user. Hence, for this type of critical applications, a local decision at the edge must be taken, rather than sending data to the cloud.

Many IoT applications can benefit from the advantages that edge computing offers, i.e., storing and processing data locally. One such use case is a smart home, where multiple applications can be deployed such as energy monitoring application that ensures an efficient consumption of energy by scheduling the operational time of each appliance in the house [26]. Besides fast response time and decreased bandwidth consumption, the security and privacy of personal data are ensured by processing it locally. Moreover, one particular advantage of deploying applications at the edge is increased availability since it can work without a stable connection to the cloud. Other IoT applications can benefit from the Edge Computing paradigm and to understand better the advantages, we describe in this section a healthcare application [19].

1) A wearable ECG sensor: The case study consists of a wearable ECG sensor attached to the human body through a smartwatch and a smartphone that acts as an edge device as it is presented in Figure 6. The communication between the wearable sensor and the edge device is via Bluetooth and via WiFi for the Internet.

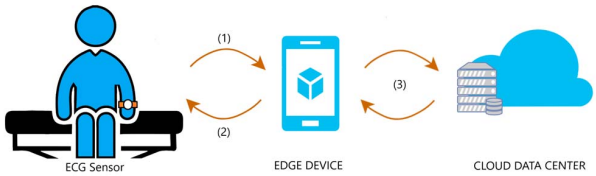

Fig. 6. A wearable ECG sensor

Usually, for this type of applications where sensors must be in close contact with the user to be able to monitor any

vital signs, a system architecture composed of a smartwatch and a smartphone represents the perfect choice. In this case, the user does not have to wear uncomfortable or unnecessary equipment, since anyway these two edge devices are used every day for normal activities. However, these devices are resource constrained, the gathered data cannot be stored locally. Consequently, the smartphone must either transfer the processed data to the cloud for further analysis or it can discard once it was processed.

In Figure 6, we can observe the system architecture required to deploy such an IoT application. Here, the smartphone represents an edge device that monitors the incoming data generate by the wearable device and react in real-time if a critical event occurs. The device pre-processes the data and sends it to the cloud if a more computational demanding analysis is needed like determining the health of the user based on data gathered over a period of time. The application executes as follow: the process starts with (1) when the smartwatch sends real-time data to the edge device, at this point the device process the data and sends the data for storage to the cloud (3). Depending on the situation, if there is a critical event, the edge device starts (2) and sends an alert notification to the emergency unit and the user.

Since edge devices have limited resource capabilities one must consider designing a system that takes into consideration the energy consumption, communication latency, storage, and computational resources. Hence, developers have to code software with highly efficient streaming algorithms, storing essential monitoring information and avoiding continuously data transfers to the central cloud.

IV. Future challenges

In recent years, the number of Internet of Things (IoT) devices connected to the overlay network is increased continuously. Stringent requirements of IoT systems have recently suggested the architectural placement of a computing entity closer to the network edge. Fog and Edge Computing visions 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 large number of interconnected distributed heterogeneous devices having the purpose to deploy and manage demanding applications closer to the user. Yet, it is a non-trivial task to design such platforms where all their 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.

A. Resource management

Fog and Edge computing paradigm have emerged as an approach to bring computational resources from cloud environments closer to the end nodes, respectively closer to the physical IoT devices at the edge of the network. Therefore, for the successful adoption of these systems, new resource management is imperative to make full use of the available resources and process applications in close proximity to the user. Edge devices often are considered as resource-constrained devices, therefore, resource management techniques are required. Such techniques allow edge devices to optimize their resource utilization (e.g., energy-aware smart devices can reduce energy consumption by offloading computation to other nearby edge nodes), improve data privacy, and enable devices to collaborate and share resources to process IoT applications.

The efficient usage of the resources provided in the proposed paradigm plays a key role to achieve desired performance and scalability. Hence, a taxonomy of resource management at the edge, based on the current state-of-the-art research in this area, is presented in [27]. In view of the objective of the technique, the authors present a total of five different categories. In addition, they summarize the benefit of an edge paradigm and through the surveyed papers they define which resource types can be managed in a better way compared to a centralized architecture.

Resource estimation is one of the fundamental requirements in resource management, i.e., the ability to estimate the amount of resources required by a particular task. This is important for handling the uncertainties found in an IoT network and providing at the same time a satisfactory Quality of Service (QoS) for deployed IoT applications. *Resource discovery* is the second category identified and is one of the critical challenges in IoT application performance in edge computing. This category discovers available resources already deployed at the edge node or resources in a large scale and geographically distributed nodes connected in Peer-to-Peer (P2P) manner. Such resources may refer not only to hardware capabilities of the edge node (i.e., CPU, storage, and memory) but also it may refer to context data, sensing or other types of domain-specific resources. Resource discovery complements resource estimation by keeping the pool of available computational resource updated.

Resource allocation is the third category of resource management classification which aims to utilize the knowledge of discovered resources and then map the different parts of the application at different edge devices. Such mapping is based on the criteria that edge device must fulfill the applications prior requirements. In the other words, the main objective of this category is to allocate IoT applications in close proximity to the users. There are two different perspectives of the allocation process: (i) represents the initial deployment to the edge of the network deciding *where* to map the application and (ii) serves as a migration technique by self-adapting *when* a node has failed. A challenging task arises when the distributed edge devices share their resources in order to fulfill applications goal i.e., in such cases in the network of the edge nodes a close collaboration between nodes enforced by security and privacy is required. Solving such challenge introduces the fourth category known as *resource sharing*.

Resource optimization is the last category of resource management classification produced as a result of combining the

above-mentioned resource management approaches. The main aim is to optimize the utilization of available resources at the edge according to the constraints of the IoT application. Usually, the developer creates the QoS requirement of his application before deploying to the edge.

B. Security and privacy

Fog and Edge computing paradigm are considered as the promising extension of the cloud computing paradigm. As a result, many cloud applications adopt the vision of fog and edge computing by migrating some computational resources toward the edge of the network. By embracing and applying these changes to our environment, we can transform our cities and homes by enabling them to automatically react to different events. However, to benefit from these advantages researchers must propose new technologies to ensure security and user privacy. Additionally to the new security and privacy issues risen in a dynamic IoT network, the paradigms inherits the security and privacy issues from the cloud as well. As an example, let us consider the scenario where an intruder can track and learn the family location and activities simply by accessing the edge network deployed in the house. Consequently, these two properties represent one of the most important challenges that a developer must overcome when such an IoT system is deployed.

To ensure the privacy and security of a system composed of edge and fog devices, an engineer must assess if it satisfies the following three most important characteristics, i.e., confidentiality, and availability known as the CIA triad model [28]. Two of these characteristics, such as confidentiality and integrity, are providing data privacy guarantees, while the availability assure that an edge node is available to share its resources when required. Yi et al. identify the most important security issues of fog computing as authentication, access control, intrusion attack, and privacy [9].

One of the main security issue in fog and edge computing is represented by authentication [13]. In a dynamic IoT network where edge devices can join or leave the network without any restrictions, a connectivity mechanism to ensure the security is preserved by authenticating an edge or fog node must be implemented. A node is successfully authenticated only if its credentials are verified and validated properly. One solution to securely authenticate edge devices is presented in [29]. Existing security solutions must be updated to fog/edge computing scenario to account for threats nonexistent in a controlled cloud environment [14].

A fog/edge ecosystem is composed of multiple devices distributed in a multi-layer architecture, each of them having its own security problems. Furthermore, new security challenges appear from combining these devices to form a new IoT ecosystem. For this reason, the authors in [30] propose a comprehensive study in which a threat model for possible security issues of the entire system and each individual component is discussed. This study is achieved by examining the scope and nature of potential security attacks (see Table 1).

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

Table 1: Threat model for fog and edge computing [14]

An analysis of all attacks that can occur against an edge ecosystem and every component individual is shown in Table 1. Here, we can observe that multiple different targets are identified like network infrastructure, service infrastructure composed of edge data center and core infrastructure, virtualization infrastructure and user devices [30]. By network infrastructure, the authors group the various communications networks used to exchange data between edge devices and can suffer an attack from an adversary. Some of the following attacks are possible on such a communication network, i.e., Denial 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 [31]. Another point of attack is represented by the architecture infrastructure which is divided into service infrastructure which resides at the edge of the network and core infrastructure which represents the cloud. An adversary could attack the service infrastructure, by using physical damage, rogue component privacy leakage, privilege escalation, and service or VM manipulation. In contrast, the core infrastructure is more secure being prone to attacks like rouge component, privacy leakage, and VM manipulation [30]. Finally, the virtualization infrastructure can suffer 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 refers to the ability of an edge device to protect the personal data of a user. Personal data is considered protected if the user can have the power of deciding where data should be processed. As a rule, if the data is processed locally, then chances of intercepting the data by a malicious attacker are close to zero [32]. Currently, when data is processed in the cloud, the personal information of the user is more vulnerable since it has to be transferred to a remote location. Consequently, edge and fog paradigms enforce privacy by moving the computation closer to the user. However, some privacy challenges remain unsolved like, (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 [11].

C. Network management

Network management plays an important role in both edge and fog paradigms, as it is the way to connect all smart devices at the edge and ultimately provide 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. New emerging technologies like software-defined networks (SDN) and network function virtualization (NFV) are seen as a possible solution that may have a significant impact on the network's implementation and maintenance by increasing the scalability and reducing cost [12].

Since both mobile and stationary devices coexist in the network, it is essential to provide a seamless connectivity mechanism in view of the volatile nature of the network. Therefore, connectivity is another aspect of network management. This mechanism must be able to easily connect/disconnect from the network in order to accommodate the uncertainty created by mobile devices. In addition, this promotes the increased deployment of intelligent devices by both users and manufacturers without additional costs or expert knowledge.

The intelligent IoT integrator (I3) developed by USC [33], aims to create a marketplace where users can share their private data with different stakeholders and receive incentives. The main advantage of this market is that it encourages users to deploy more advanced devices, which in turn increase the IoT network. Furthermore, a pool of data is provided to the developers in order to improve their IoT applications.

V. Towards a IoT application platforms

As we are now acquainted with fog and edge computing paradigms, as well as the challenges that each of these architectures must overcome, in this section, we propose a new use case that combines the two of them. It is a use case deployed in a smart city scenario aiming to provide a platform that offers users the possibility of customizing their environments by downloading different IoT applications. An overview of our proposed IoT platform is presented in Figure 7.

The platform is composed of a three-tier layer architecture, i.e., edge, fog, and cloud respectively. The cloud layer contains all the IoT applications models that a user can download. Additionally, it offers the possibility for each user to test their current IoT network and verify if any extra resources are needed for a certain application they desire to download. Next, the fog layer is in charge of managing the networks created by each user, ensuring that the user has installed the edge devices correctly. On top of this, the fog layer is responsible to assure that the newly formed edge device network is secured and trustworthy. This verification is performed by validating the user and the edge network. Finally, the edge layer consists of multiple edge devices where the application is installed after the user downloads it. In this way, it ensures that any stringent requirements are met and the quality of service is satisfied.

To achieve this behavior, the proposed system combines two different resource management objectives, i.e., resource allocation and resource discovery. Resource allocation is used to install IoT applications at the edge of the network. For this reason, we can deploy our decentralized resource management system described in [34]. This is a decentralized edge resource allocation techniques that deploy IoT applications at the edge

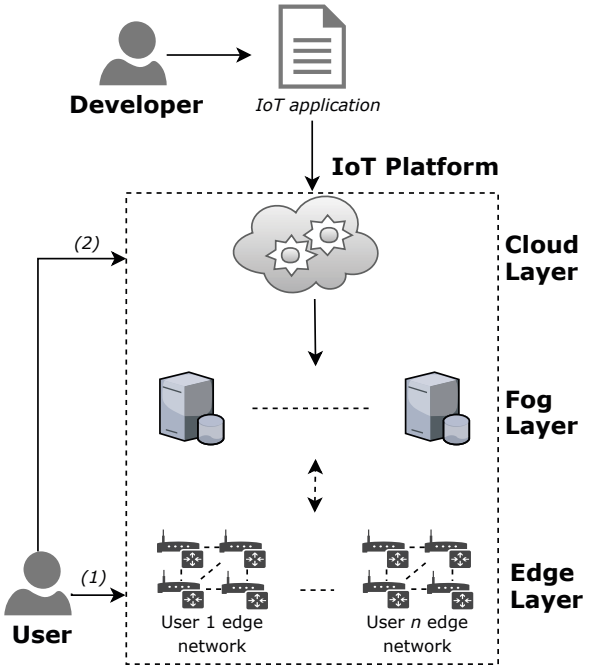

Fig. 7. IoT platform overview

of the network is there are available resources. On the other hand, to validate that indeed the user has the required resources for an application in his edge network, a resource discovery technique is implemented for that particular group of edge devices. Combining these two techniques we can enable seamless deployment of IoT applications at the edge of the network.

With the help of such a platform, we envision a world where general purpose edge devices are sold to the user and enable them to build their own personal edge networks (1). By general purpose devices, we refer to resource-constrained edge nodes containing a set of sensors and actuators. It is important to specify, that these devices have all their computational resources available in the beginning when they were acquired by the user. With such devices installed in their homes or buildings, the owner can select IoT applications from the platform and download them at the edge to satisfy a specific situation (2), giving the possibility to customize as the owner desire. For example, if the user wants to have a safer home, he can download applications for smoke, leak and gas detection. Furthermore, if there are pets living in the house, an IoT application that controls the food dispenser or let the dogs out can be installed. As a final example, a user can download applications for healthcare monitoring if the owner requires special care. In conclusion, we envision a world where edge devices are bought and used as smartphones are used today.

VI. Conclusion

Two of the most prominent paradigms that researchers have proposed to overcome the challenges that the actual cloud-

centric state of the art faces are fog and edge computing. Both share the same vision of migrating some computational resources closer to the edge of the network. With such an approach, the stringent requirements of new IoT applications like fast response time, better security and privacy, and increased availability can be satisfied.

When first introduced, both paradigms were differentiated by the IoT nodes placement in a network. On one hand, fog computing aims to extend the cloud capabilities by creating a cloud to things continuum by placing fog nodes closer to the end users devices. On the other hand, edge computing proposes a solution where an IoT application can be deployed on the available resources found on edge devices like smartphones and laptops. However, considering the big improvements of IoT devices in the last couple of years, the premises of fog and edge computing overlap. As a result, we conclude that fog computing architecture consist of the same design as an edge computing architecture.

Finally, to conclude our paper and show a combination of fog and edge devices, we have introduced a smart city scenario where users can select and download different IoT applications from a platform to customize their homes.

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] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[2] F. B. et al., "Fog computing and its role in the internet of things," *1st ACM Mobile Cloud Computing Workshop*, pp. 13–15, 2012.

[3] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, oct 2009. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2009.82http://ieeexplore.ieee.org/document/5280678/

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

[6] A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, "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)*, April 2017, pp. 247–258.

[7] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy." Citeseer.

[8] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013, 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] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 workshop on mobile big data*. ACM, 2015, pp. 37–42.

[10] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, *Fog Computing: A Platform for Internet of Things and Analytics*. Cham: Springer International Publishing, 2014, pp. 169–186. [Online]. Available: https://doi.org/10.1007/978-3-319-05029-4_7

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

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

[13] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *2014 Federated Conference on Computer Science and Information Systems*, Sept 2014, pp. 1–8.

[14] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live vm migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[15] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug 2016.

[16] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 46–59, Jan 2018.

[17] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, May 2015, pp. 1–5.

[18] "(2018). openfog architecture overview. openfog consortium architecture working group. accessed on dec. 7, 2016. [online]. available: http://www.openfogconsortium.org/wp-content/ uploads/openfog-architecture-overview-wp-2-2016.pdf."

[19] M. Gusev and S. Dustdar, "Going back to the roots—the evolution of edge computing, an iot perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, 2018.

[20] J. Pate and T. Adegbija, "Amelia: An application of the internet of things for aviation safety," in *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. IEEE, 2018, pp. 1–6.

[21] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, "Edge computing in iot-based manufacturing," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, 2018.

[22] S. Zhang, W. Li, Y. Wu, P. Watson, and A. Zomaya, "Enabling edge intelligence for activity recognition in smart homes," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 228–236.

[23] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2018.

[24] Z. Chen, G. Xu, V. Mahalingam, L. Ge, J. Nguyen, W. Yu, and C. Lu, "A cloud computing based network monitoring and threat detection system for critical infrastructures," *Big Data Research*, vol. 3, pp. 10–23, 2016.

[25] X. Xu, Q. Z. Sheng, L.-J. Zhang, Y. Fan, and S. Dustdar, "From big data to big service," *Computer*, vol. 48, no. 7, pp. 80–83, 2015.

[26] C. Xia, W. Li, X. Chang, F. Delicato, T. Yang, and A. Zomaya, "Edge-based energy management for smart homes," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 2018, pp. 849–856.

[27] K. Toczé and S. Nadjm-Tehrani, "A taxonomy for management and optimization of multiple resources in edge computing," *CoRR*, vol. abs/1801.05610, 2018. [Online]. Available: http://arxiv.org/abs/1801.05610

[28] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A critical analysis on the security concerns of internet of things (iot)," *International Journal of Computer Applications*, vol. 111, no. 7, 2015.

[29] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 60–65, May 2018.

[30] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680 – 698,

2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16305635

[31] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *2015 IEEE 39th Annual Computer Software and Applications Conference*, vol. 3, July 2015, pp. 53–59.

[32] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in *2010 Sixth International Conference on Semantics, Knowledge and Grids*, Nov 2010, pp. 105–112.

[33] U. of Southern California, "I3: The intelligent iot integrator (i3)," https://i3.usc.edu/.

[34] C. Avasalcai and S. Dustdar, "Latency-aware distributed resource provisioning for deploying iot applications at the edge of the network," in *Advances in Information and Communication*, K. Arai and R. Bhatia, Eds. Cham: Springer International Publishing, 2020, pp. 377–391.