

UNIVERSIDADE ESTADUAL DE CAMPINAS
SISTEMA DE BIBLIOTECAS DA UNICAMP
REPOSITÓRIO DA PRODUÇÃO CIENTÍFICA E INTELECTUAL DA UNICAMP

Versão do arquivo anexado / Version of attached file:

Versão do Editor / Published Version

Mais informações no site da editora / Further information on publisher's website:

<https://jisajournal.springeropen.com/articles/10.1186/s13174-018-0086-3>

DOI: 10.1186/s13174-018-0086-3

Direitos autorais / Publisher's copyright statement:

©2018 by Springer Nature. All rights reserved.

DIRETORIA DE TRATAMENTO DA INFORMAÇÃO

Cidade Universitária Zeferino Vaz Barão Geraldo

CEP 13083-970 – Campinas SP

Fone: (19) 3521-6493


<http://www.repositorio.unicamp.br>

RESEARCH

Open Access



Fog orchestration for the Internet of Everything: state-of-the-art and research challenges

Karima Velasquez¹, David Perez Abreu^{1*} , Marcio R. M. Assis², Carlos Senna², Diego F. Aranha², Luiz F. Bittencourt², Nuno Laranjeiro¹, Marilia Curado¹, Marco Vieira¹, Edmundo Monteiro¹ and Edmundo Madeira²

Abstract

Recent developments in telecommunications have allowed drawing new paradigms, including the Internet of Everything, to provide services by the interconnection of different physical devices enabling the exchange of data to enrich and automate people's daily activities; and Fog computing, which is an extension of the well-known Cloud computing, bringing tasks to the edge of the network exploiting characteristics such as lower latency, mobility support, and location awareness. Combining these paradigms opens a new set of possibilities for innovative services and applications; however, it also brings a new complex scenario that must be efficiently managed to properly fulfill the needs of the users. In this scenario, the Fog Orchestrator component is the key to coordinate the services in the middle of Cloud computing and Internet of Everything. In this paper, key challenges in the development of the Fog Orchestrator to support the Internet of Everything are identified, including how they affect the tasks that a Fog service Orchestrator should perform. Furthermore, different service Orchestrator architectures for the Fog are explored and analyzed in order to identify how the previously listed challenges are being tackled. Finally, a discussion about the open challenges, technological directions, and future of the research on this subject is presented.

Keywords: Fog, Cloud, Internet of Everything, Orchestration, Research challenges

1 Introduction

A new industrial revolution driven by digital data, computation, and automation is arriving. Human activities, industrial processes, and research lead to data collection and processing on an unprecedented scale, spurring new products, services, and applications, as well as new business processes and scientific methodologies [1].

The applications and services of the Internet of Everything (IoE) [2] can be the link between extremely complex Information and Communication Technology (ICT) network infrastructures and general activities of the whole society in general. These applications and services usually rely on the use of Cloud computing to achieve elasticity, on-demand self-service, resource pooling, among other

characteristics. However, new generation applications and services (e.g. IoE-based applications and services) have requirements that are only partially met by existing Cloud computing solutions [3].

In recent years there has been a paradigm shift to bring Cloud services towards the edge of the network. In this peripheral area, there is an abundance of heterogeneous IoE resource-constrained devices both generating and consuming data [4]. This represents an increment on the amount of data, that would lead to increased traffic and response time to transport to the Cloud and back. It is possible thus to place storage and processing devices at the rim of the network to help preprocess this data and alleviate the load sent towards the core network, while also reducing response times which benefits real-time applications particularly. This solution is known as Fog computing.

*Correspondence: dabreu@dei.uc.pt

¹ Department of Informatics Engineering, University of Coimbra, Polo II - Pinhal de Marrocos, 3030-290, Coimbra, Portugal

Full list of author information is available at the end of the article

Fog computing is an important paradigm to help address the requirements of the IoE that are not completely covered by the Cloud; nonetheless, the use of this technology creates new challenges. The Fog needs to support the orchestration of applications and services on demand, with adaptability, while providing flexible performance. In practice, traditional service orchestration approaches that were applied to Cloud services are not suitable for the large scale and dynamism of Fog services, since they can not effectively treat the prominent characteristics of the Fog's distributed infrastructure. It is crucial to clearly identify the challenges that differentiate the Fog from the Cloud, in order to create innovative orchestration solutions able to meet its required characteristics such as high mobility, high scalability, and performance in real-time. Some Fog orchestration architectures have already been proposed, but it is still not clear how well they meet the Fog's requirements.

This paper presents a review of the main Fog challenges that impair the migration of the orchestration mechanisms from the Cloud to the Fog. Furthermore, it shows a revision of different Fog service orchestration architectures (SORTS, SOAFI, ETSI IGS MEC, and CONCERT), in order to evaluate how these major challenges are being addressed.

The remainder of the paper is structured as follows. Section 2 presents the scope of the paper, the Internet of Everything, including the involved paradigms namely Internet of Things (IoT), Fog, and Cloud computing. Section 3 defines more thoroughly the Fog computing paradigm, what are the main differences between Cloud and Fog computing environments, to finish with some scenarios of applicability of Fog computing. Section 4 discusses some research challenges that a Fog Orchestrator must handle. Section 5 describes a set of Fog orchestration architectures, including how they deal with some of the previously identified Fog challenges. A comparative analysis of the reviewed architectures is presented in Section 6. Conclusions are drawn in Section 7.

2 The Internet of Everything

According to Byers and Wetterwald [5], about 50 billion of devices will be connected to the Internet by 2020. One consequence of this trend is the production of an unprecedented volume of data in the most diverse segments. Such data can be used to provide new services for the improvement of various areas of the society (e.g. transport, health, economy). In this context, *IoT*, *Fog*, and *Cloud* computing paradigms of service provision stand out.

The term IoT [6] is widely used, although still a blurry one, to refer to a vision of a future Internet where any object can communicate with other devices using Internet communication protocols. The IoT paradigm has been defined as a technology to connect objects that surround

us providing a reliable communication and making available the services provided by them. Additionally, the IoT brings ubiquity providing a new dimension to the ICT, known as "Any THING" communication involving interaction between computers, humans, and things to complement the previous "Any TIME" and "Any PLACE" communication paradigm presented in the ICT [7]. The IoT will ultimately comprise virtualized sensors, actuators, and platforms, which will result in a set of software "things" [8].

The Fog paradigm envisions a set of micro-data centers, placed at the edge of the network, with the following characteristics [9]: location awareness, mobility support, real-time interactions, low latency, geographical distribution, heterogeneity, interoperability, security, and privacy.

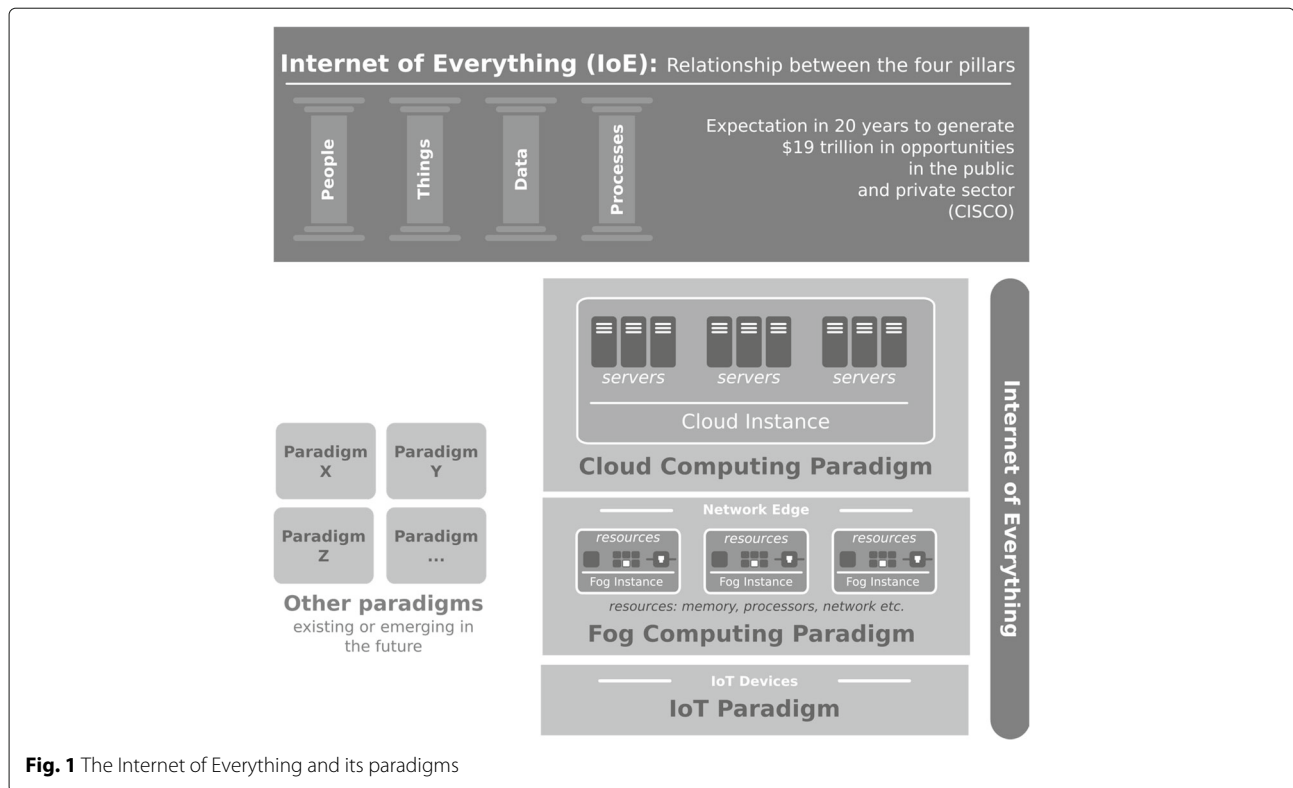
Cloud computing [10, 11] is a well-known paradigm to providing basic computing assets as a service. Cloud Services Providers (CSPs) offer specialized servers in data centers with large storage area, high computing capacity, and a powerful network infrastructure. According to NIST [12], CSPs must offer their customers the *on-demand self-services*, *broad network services access*, *resource pooling*, *rapid elasticity* and *measured services*.

Cisco Systems Inc. [13] describes IoE as a set of relationships derived from the connectivity between *people*, *processes*, *data* and *things*. According to Cisco, the IoE could generate \$4.6 trillion in value for the global public sector in the next 15 years. In addition, there is an expectation of generating \$14.4 trillion in the private sector over the same period. The IoE enables the emergence of new services based on the IoT, Fog, and Cloud computing paradigms to enhance the quality of life of citizens, creating a strong dependency on them [14]. The relation between these three main paradigms is depicted in Fig. 1. At the bottom level is the IoT, where reside different resource constrained devices (e.g. sensors and actuators) gathering data. Next, to the IoT layer comes the Fog computing level, where the data is aggregated and preprocessed. Finally, at the upper layer is the Cloud computing where the data can be stored and analyzed.

Out of the three paradigms involved in the IoE (IoT, Fog, and Cloud), Fog computing is the most recent and least explored in the research field. To clarify the concept and the new demands it imposes, the following section offers a description of the Fog, where it can be applied, and its new challenges that differentiate it from the Cloud.

3 The Fog computing paradigm

The frontier between the Cloud and the end devices is known as the Fog. The Fog is an environment with a plethora of heterogeneous devices that work in a ubiquitous and decentralized manner, communicating and cooperating among themselves [15]. Thus, the Fog emerges as an extension of the Cloud paradigm escalating from



the core of the network towards its edge; it is comprised of a heavily virtualized platform able to perform storage, processing, and networking activities between the Cloud servers and the end devices [16, 17]. The Fog comes to support novel applications and services that are not completely fit for the Cloud, granting them ubiquity, high resilience, low latency, decentralized management, and cooperation [18, 19].

The OpenFog consortium [20] defines Fog as:

“A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum.”

A Cloudlet is an autonomous Fog instance domain that can maintain relationships with other domains. Cloudlets are typically used to offload applications when devices are not capable of executing them, but without incurring in the costs (both of time and monetary) of using a more specialized data center in the Cloud [21, 22].

The use of Cloudlets in the context of Fog environments has proven to provide several advantages in terms of response times and energy consumption, among other benefits [23]. Thus, most multi-layer architectures (that include Cloud and Fog) incorporate the concept of Cloudlets for the deployment of services and applications in order to improve response times, Quality of Service (QoS), and other factors.

The scenario at the edge of the Cloud is vastly different from the one at its core. Thus to fully understand their peculiarities, next subsection reviews the main aspects where Cloud and Fog diverge.

3.1 From Cloud to Fog computing

The Fog introduced a paradigm shift from the traditional concept where the core of the network is in charge of providing information that will be consumed at the edge of it. To address the challenges arising from Cloud to Fog computing, the latter has to fulfill the following key features [3, 15, 20, 24]:

- Heterogeneity and Interoperability, to deal with a broad diversity of physical and virtualized devices deployed in wide-ranging environments;
- Edge location, location awareness, and low latency, to guarantee that the most time-sensitive data is processed closest to who is requesting it;
- Wireless communication, to reach a variety of devices at the edge avoiding the installation of a fix communication network and contributing to reduce the amount of traffic in the core network;
- Real-time support, to satisfy services and applications with time-sensitive requirements;
- Mobility support, to allow continuity in the services provided to devices and final users.

In the Fog, final users (i.e. mobile devices and IoT sensors) generate ample quantities of data at the edge of

the network making them producers and consumers at the same time. The treatment of this unprecedented quantity of data represents a challenge for traditional paradigms like Grid and Cloud computing; thus the Fog arises to overcome these limitations [3, 25].

Although Cloud and Fog computing share overlapping features, the Fog becomes a non-trivial extension of the Cloud that must deal with characteristics inherent to its placement in the overall network infrastructure, such as location awareness, geographical distribution, low latency, real-time, and mobility support [17, 26–28]. Another significant difference regarding the Cloud is that the Fog encompasses a huge set of heterogeneous devices, mostly wirelessly connected, that are more constrained in terms of resources in comparison with the servers that reside in the Cloud [29].

Fog computing provides to the Cloud an alternative to manipulate exabytes of data generated daily from the IoT [25]. With the ability to process the data near where it is generated and required, it is possible to tackle the challenges regarding data volume, interoperability, and time-sensitivity. By eliminating the round-trip time related to traveling to the Cloud and back, it is feasible to accelerate the awareness and response time of services and applications. Furthermore, by preventing the need to send the data to the Cloud, or at least by aggregating it first, capacity of the communication channel is saved by the reducing the amount of traffic in the core network.

The Fog and IoT paradigms are used together in different application scenarios, some of which are described in the next subsection.

3.2 Application scenarios

The combination of the Fog and IoT paradigms could be used in many scenarios to achieve and improve applications and services requirements. In this subsection some examples of the collaborative use of Fog and IoT are described within four specific areas: *Automation*, *Healthcare*, *Smart Cities*, and *Infotainment*.

3.2.1 Automation

One important scenario where the Fog could play a major role is related to Automation. These systems refer to the integration of cyber technologies that make devices Internet-enabled to implement services for different industrial tasks such as Internet-based diagnostics, maintenance, and efficient and cost-effective operation [30].

So far the common user interaction involves a person behind the vast majority of the endpoints connected to the data network. With the advent of the IoT the landscape becomes different, with many sensors and actuators communicating with them as end users. Thus, it is necessary

to adopt a *system view* instead of an *individual view* of the endpoints [26].

Other constraints for this environment include data and service security since many decentralized endpoints could lead to security disruptions; scalability, dealing with naming and addressing a huge amount of devices, and also with the heterogeneity of a massive amount of services and execution options [31]. Furthermore, minimizing latency and timing jitter are other crucial factors for this type of scenario. Many critical automation systems such as in-flight control systems, medical applications, and internal vehicle networking require stable real-time behavior [5].

Moreover, operators infrastructures can contain devices from multiple manufacturers communicating with different technologies, sometimes proprietary, creating additional problems regarding interoperability and service deployment that must rely on manual intervention by network managers [32]. This suggests the need of a management solution, possibly based on Software Defined Networking (SDN) and Network Function Virtualization (NFV) providing low-level abstractions that ease the configuration and administration tasks of the devices.

This clearly represents one scenario that benefits from the Fog characteristics; however, the Fog must have a management process able to communicate with such a heterogeneous environment with stringent requirements and provide it with the QoS needed to maintain these industrial systems running smoothly.

3.2.2 Healthcare

Another scenario of applicability of Fog computing is healthcare. Several wearable devices (e.g. Fitbit [33]) and platforms (e.g. Google Fit SDK [34] and iOS health API [35]) that let users collect and monitor their fitness and health data have been emerging in recent years. This growth has been promoted mainly by the advances in manufacturing technology and the emergence of interoperability standards (e.g. Bluetooth Low Energy -BLE- and Radio-Frequency Identification -RFID-). However, it is necessary to integrate these devices and the data they collect into a new health model [36, 37]. Such a model should allow the integration of physicians, health professionals, clinics, and the patients. Through the integration and analysis of this data, it will be possible to obtain more accurate diagnoses and improve the effectiveness of the treatment of diseases.

In this context, the processing of data generated by wearable devices requires low latency among other requirements related to the speed of events, such as interoperability, scalability, and security. This is due to the fact that in an emergency situation, such as an epileptic seizure [38], the processing and generation of alerts for the most diverse stakeholders (i.e. doctor, emergency

unit, ambulance) must be the fastest in order to avoid undesirable situations (e.g. death or permanent sequelae). Fog computing brings processing to the edge of the network, significantly improving technical factors that allow shorter responses in emergency scenarios. Fog can also interface with other paradigms such as Cloud computing, allowing persistent data to be driven by the Clouds for permanent storage and performing tasks such as analytics.

Hosseine et al. [38] describe a layer-based framework for collecting, aggregating, and analyzing data for automatic and actual detection of epileptic seizures. In this solution, Fog computing is used as a middleware to perform real-time data processing, feature extraction and classification to feed machine learning and data caching mechanisms. A Fog-driven IoT interface, named as FIT is proposed in by Monteiro et al. [39]. The purpose of their solution is to enable the communication between IoT devices (e.g. smart watches) and the Cloud for the specific purpose of analyzing acoustic data to detect speech-language pathologists. To do this, Fog nodes are placed on the LAN-level in the network hierarchy. The Fog nodes are used to collect, store and process raw-data, before sending it to the Cloud for permanent storage.

3.2.3 Smart Cities

The Smart City paradigm emerged to describe the use of new technologies in everyday urban life, providing the management of its services (e.g. energy, transportation, lighting, public safety) using ICT. These technologies implement a logical/virtual infrastructure to control and orchestrate physical objects to accommodate the city services to the citizen needs [40].

In the context of Smart Cities, mobility is a key requirement that should be explored allowing devices and services to capture information about the environment and act in real-time. For example, a mobility scenario should take into consideration vehicles and pedestrians into the city providing them relevant information and services according to their location.

In this urban mobility scenario, inside the Smart City paradigm, a Fog Orchestrator faces major challenges, for example, consider the complexity of the resource management related to a Cloud-based traffic sensing and travel planning service/application.

The Orchestrator must be able to maintain low latencies, high resilience, and trustworthiness according to the applications and users expectations, even in times when the infrastructure is stressed under heavy loads of traffic. The key issue here is that such loads are intrinsically specific to a particular application and cannot be reutilized from other scenarios or domains; thus further research in this direction to achieve the requirements mentioned above is required.

3.2.4 Infotainment

For applications with stringent latency requirements, the Fog offers an attractive alternative to the Cloud. Many applications, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Roadside (V2R) communication, virtual reality applications, gaming applications, video streaming, financial trading applications among others, require very low latency levels (i.e. around tens of milliseconds) [18], thus rendering the Cloud services insufficient. The term *Infotainment* refers to a combination of Information and Entertainment served together [41], and is used in this subsection to group the applications mentioned above.

Infotainment applications are becoming more global every day with about 82% of the Internet traffic estimated for 2021 corresponding to video alone (both business and consumer), virtual and augmented reality traffic to increase 20-fold between 2016 and 2021, consumer Video-on-Demand (VoD) to be equivalent to 7.2 billion DVDs per month, and Internet gaming traffic to grow tenfold by 2021 (i.e. 4% of consumer Internet traffic) [25]. Furthermore, by 2021 the number of devices connected to IP networks will be more than three times the global population [25]. Many of these new connected devices will correspond to devices located at the IoT level.

With such an increment in the demand of traffic that additionally has rigorous demands regarding latency, the inclusion of Fog helps bringing the services closer to the users. To efficiently handle these services, new management functions at IoT and Fog level must be conceived to deal with the scenario requirements efficiently; for flexible connectivity in heterogeneous and highly mobile environments, with strong latency guarantees, network operators require innovative orchestration mechanisms that support dynamic multi-technology resource management [42].

To properly achieve the application scenarios described above, it is important to thoroughly recognize the characteristics of the environment, and the challenges they impose. Orchestration and resource management in Fog environment have to deal with different requirements and objectives. A selection of the challenges that a Fog Orchestrator must overcome is presented in the following section.

4 Research challenges in Fog orchestration

Fog computing brings challenges at many different levels. Looking from a broader perspective, one of the first challenging issues is the modeling of the orchestration element that needs to be able to perform the deployment of the Cloudlets [43, 44] and handle tasks inside the environment.

The combination of IoT, Fog, and Cloud embraces a complex scenario where in some cases it is not suitable to migrate or apply well-known solutions or mechanisms

from other domains or paradigms. This statement is already considered by important Cloud providers, such as Amazon and Microsoft who have released new services (i.e. AWS Greengrass [45] and Azure IoT Edge [46] respectively) focused on addressing the new requirements at the edge of the Cloud. Some aspects to take into consideration are:

- Resource Management, which requires the design and development of mechanisms that handle tasks such as *Scheduling, Path Computation, Discovery and Allocation*, and *Interoperability*;
- Performance that deals with *Latency, Resilience*, and *Prediction and Optimization* from the gauges, mechanisms, and algorithms point of view;
- Security Management that must include mechanisms and policies to cope with *Security and Privacy* and *Authentication, Access, and Account*.

Each task represents a challenge to be addressed by the Fog Orchestrator. These tasks are described in the following subsections.

4.1 Scheduling

Scheduling is one of the main tasks of an Orchestrator. In Fog environments, it is necessary to consider how to exploit the collaboration between nodes to offload applications efficiently. In general, the processing nodes should be managed by a resource broker in the Orchestrator to perform a smart scheduling of the resource taking into consideration the applications' workflows [47].

Capacity planning and Cloudlets positioning models can be derived from the cellular phone antenna placement problem. The additional challenge is to extend those models to look beyond the maximum number of users expected but also understanding applications behavior and load patterns. An ideal capacity planning would be able to avoid applications that require low delays to be offloaded to the Cloud, which can hinder applications functioning. On the other hand, it would also minimize Cloudlets size to avoid underutilization and reduce operating expenses.

As soon as Cloudlets are deployed, they bring many new interesting challenges to scheduling. Among those, we consider *application classification* and *user mobility* as two key aspects to be associated with scheduling in providing efficient resource management for the Fogs and their users.

Application classification must provide the scheduler with information about application requirements, which will allow the scheduler to prioritize the Cloudlet use and optimize other (potentially conflicting) objectives (e.g. reduce network usage, reduce Cloud costs). By making use of such information, a Fog scheduler can decide which application(s) should run in the Cloudlet and which

should run in the Cloud. Moreover, application classes could also allow a system-level scheduler to prioritize applications within a Cloudlet, allowing smaller granularity control over the delays observed by applications at each class.

User mobility is another challenging issue, as it can determine the amount of load in Cloudlets over time. Users data and processing are widely used to support mobile applications in smart devices. Thus, understanding user behavior and mobility patterns can improve resource management by better planning the scheduling of the application beforehand. This planning is of paramount importance to avoid application delays perceived by the users during mobility. For example, when two applications with different requirements are to be scheduled to the same Cloudlet, if a predictive mechanism can accurately determine when the less QoS-strict application can be migrated to the Cloud, the other application can experience lower delays since its arrival to the Cloudlet. Note that this planning can also involve data movement, depending on the application being migrated. In this case, planning should also consider the time taken to move data between parts (i.e. Cloudlets or Cloudlets-Cloud).

Although mobility can be reasonably predicted in general [48], prediction misses will eventually occur from lack of information or user unpredictable behavior. Prediction misses can incur in additional computing and networking costs: the same data/application movement will be needed to contour the incorrect prediction results. Scheduling strategies to deal with mobility prediction failure are also an interesting problem to be studied in the Fog computing context.

Scheduling in Fog computing environments should prioritize several factors. Three of them stand out: i) the diversity of workloads submitted, ii) the high degree of heterogeneity of the resources present in Cloudlets, and iii) a new class of mobility applications [49, 50]. Even though some solutions at Cloud level, such as Kubernetes [51], provide mechanisms for deployment, maintenance, and scaling application across multiples hosts; these should be rethought to achieve Fog requirements efficiently. In Cloud environments, application scheduling used to follow a centralized approach taking advantage of the global knowledge of the relatively small quantity of homogeneous data centers. Nonetheless, in the heavily distributed scenario of Fog where there is a large number of heterogeneous micro-data centers potentially located over large geographical areas, the legacy scheduling approaches are not suitable considering that it is necessary to achieve latency requirements.

Thus, new approaches to prediction and scheduling execution based on hybrid mechanisms are necessary to guarantee low latency and service-continuity during

users' mobility. Local scheduling at Fog level following a choreography approach in combination with optimization mechanisms at the upper levels (i.e. Cloud level) to take advantage of a global view of all infrastructure, can be the next step to efficiently exploit the collaboration between all the nodes in a Cloud/Fog scenario and fulfill an end-to-end application awareness scheduling.

4.2 Path computation

Path computation is playing a key role on the Internet and has evolved to support new types of applications and services, as well as network structures. These demands tend to be heterogeneous since there is an increasing number of users accessing the Internet. In addition, path computation must consider the characteristics of entities connected to the Internet, new services and applications provided over the Internet, and communication platforms such as wireless technologies and Cloud/Fog systems.

The main objectives of path computation are: (1) *maintaining end-to-end connectivity*, how to find the best way of reaching a destination which is not directly connected to the source?; (2) *adapting to dynamic topologies*, how can the best new path be found?; (3) *Maximizing network and application traffic performance*, how can users be provided with a high level of QoS at the lowest possible cost, while providers obtain the highest profits possible with the lowest investment?; and (4) *providing network resilience*, how does the routing protocol behave when failures occur and what is its impact on traffic performance?

Path computation in Cloud, Fog and IoE environments plays a crucial role, since it goes beyond packet and flow based decisions, and involves supporting dynamic services. In particular, routing can provide information to support functions such as service placement by Orchestrator. This is especially important within the Fog computing paradigm where the aim is to ensure these services can be accessed with the lowest latency possible (see SubSection 4.5) as well as to reduce energy consumption [15, 52].

One key aspect of path computation in multi-hop wireless networks concerns the best way to characterize the links in the network. Although this was already a problem in wired networks, it has become of critical importance in wireless environments owing to the rapidly changing characteristics of the medium and topologies in IoT/IoE environments, the existence of multiple channels, and inter-flow interference. In view of this, it is critical to select metrics that, in addition to traffic load levels, depict the characteristics of the links and paths in the network [53–55].

Connectivity and routing in Fog is a challenge given the heterogeneous nature of its mostly wireless links, in comparison with other distributed systems such as the Cloud. Just take into consideration that it is necessary

to maintain the connectivity between the services and devices deployed in an IoT large-scale scenario. Nevertheless, these challenges also provide new opportunities for cost-reduction and enlarging the network connectivity scope. For example, a multi-hop wireless network could be partitioned into different clusters due to the coverage of available resources in Fog nodes (i.e. Cloudlets, sink nodes, smartphones) to enhance the support of the services in the IoE. In this kind of scenarios, the SDN and virtualization approaches could be used to instantiate particular devices in real-time, and adapting routes to changing conditions.

4.3 Discovery and allocation

One important issue that must be addressed in a scenario as the one described so far is related to the discovery of the physical and virtual devices in the Fog, as well as the resources associated with them. The resource discovery is a process where different computational capacities (e.g. CPU memory, storage) can be reported to the management unit so it can account for the resources available in the overall system [56]. This process also refers to the associations of devices grouped, known as Cloudlets, that embody the Fog.

For the discovery process, a device could advertise its available capacities, or it could be sensed by the management entity. However, it is worth noticing that in a highly dynamic scenario (unlike the Cloud), this information might vary rapidly [57]. Thus, the time frame on which the information must be updated becomes a critical factor to guarantee the accuracy of the information reported.

The management entity in charge of the Fog is therefore responsible for the accounting of the resources available in the Fog, and then selecting the one that best fit for the service's requirements [58]. As for Cloud computing, at the Fog level the goal usually is to maximize the utilization of resources while the idle periods are minimized [59]. Nevertheless, according to the needs and the nature of the environment (e.g. high mobility), different policies can be used for the allocation process. Some examples are [49]: (1) *Concurrent*, requests are allocated to the receiving Cloudlet, regardless of usage or capacity; (2) *First come first served*, requests are served according to their arrival time; and (3) *Delay-priority*, applications requiring lower latency are prioritized.

Another thing to consider is the tradeoff between different (sometimes competing) optimization parameters. For instance, it could be required to minimize the energy consumption while also minimizing the latency [60]. These represent a multi-objective optimization problem that renders the allocation process into a non-trivial problem.

The management entity, using the data from the discovery process, should apply the preferred policy in order to

achieve the applications' requirements. Both the discovery and allocation processes represent two challenges that must be addressed by an Orchestrator.

4.4 Interoperability

In a macro view, interoperability [61] is the ability that distributed system elements have to interact with each other [62]. Several factors influence the interoperability of a system, such as the heterogeneity of the elements present in it. Thus, in Fog computing environments where the set of Cloudlets have a high degree of heterogeneity, there are several challenges to maintain the interoperability between its elements [20].

Considering the similarities between Cloud and Fog paradigms, it is possible to derive the problem and the solutions to maintain the interoperability found in the Inter-Clouds for Fog environments. It is feasible to group the solutions to keep interoperability into: *translators*, *standard interfaces* and *ontologies*. Translators, or brokers [63], concentrate the communication protocols supported by the Orchestrator to perform the communication between the involved parties. It is a solution widely used by Inter-Clouds to achieve interoperability between Cloud providers. However, it may not be applicable by the Orchestrator when considering the low latency requirements of applications and services. The insertion of a new translation layer may increase the overhead to the process. Standard interfaces provide a straightforward and standardized way of communicating between elements of a distributed system. In addition, standardized interfaces [20] make the process of insertion and diffusion of new functionalities more controlled and homogeneous, since in most cases there are working groups or consortiums involved in the development of the interface.

On the other hand, heterogeneity causes certain interfaces to be accepted only by portions of the Cloudlets, also the problem of the *egg-and-chicken* may appear when considering new interfaces. The variability of standard interfaces can lead to the need for brokers, which may refer to problems arising from their use (e.g. overhead). Ontologies [64–67] are a representation of knowledge (e.g. W3C Semantic Web standard web ontology language [68]). They “hide” the technologies used by delegating the implementation to local contexts of Cloudlets. Implementing and maintaining an ontology increases the complexity of the Orchestrator, creating problems similar to those generated by brokers.

Providing interoperability is a problem inherent to distributed environments. Associations of multiple Clouds, as well as other distributed systems, have already faced this challenge and it is no different in the Fog. However, since the Fog intends to address the applications that have latency constraints and mobility support as main properties, the approach considered for interoperability is

an open challenge that must be addressed [15, 20]. The choice of a broker can serve interoperability but can insert an overhead that can lead to denial of applications with a certain latency requirement [69]. Added to this, the diversity of Cloudlets can lead to the maturity of several protocols by the broker. Already adopting standard interfaces can decrease the amount of Cloudlets available in the environment which can compromise the amount of applications that can be executed.

4.5 Latency

One of the characteristics of Fog environments is that they provide low levels of latency [15, 70]. This allows the deployment of a different kind of services with real-time and low latency restrictions that are not necessarily fit for the Cloud; but also requires a new set of mechanisms that guarantee that these low latency levels are met.

Smart routing and forwarding mechanisms should be designed, aiming at faster response time. A multipath approach [71] could be employed to achieve this goal, especially when dealing with huge bulks of data; however, for small but critical tasks, the use of redundant packets has proven to be efficient [72, 73].

Another possibility is designing intelligent service placement mechanisms [74, 75] for the Orchestrator. It is also important to take into consideration the mobile nature of the devices (e.g. sensors in cars), for which location awareness [76] and dynamism support [77, 78] must also be included.

Given that the tendency is shifting time-constrained services and applications towards the edge of the Cloud into the Fog, it is imperative to guarantee that the time restrictions are met [79]. The service orchestrator must incorporate novel mechanisms, different from those already available for other distributed systems such as Cloud, that are sensitive to time constraints, and that support other features such as mobility, dynamism, and geo-distribution [17].

Another issue to take in consideration is the more limited resources regarding the bandwidth of the links in comparison with Cloud systems, given their wireless nature and narrower capacity [80].

4.6 Resilience

In the complex and diverse environment where the IoE acts, a seamless interaction between all the actors that build the IoE paradigm, from the physical (e.g. sensors, actuator, smart objects) to the logical perspective (e.g. service, applications, protocols), is a critical aspect. Even more, in this kind of scenarios, the availability of the physical and logical devices and their services represent a key requirement, given that some critical applications such as assisted driving, augmented maps, and health monitoring

require continuous availability while providing real-time feedback to users.

An improved connectivity between Cloud [81] services and devices in the IoT [82] is necessary to support the emerging applications enabled by the IoT Cloudification. To deal with disruptions in the IoE, it is required to have mechanisms that enhance its resilience both at infrastructure and service levels. Sterbenz et al. [83] defined resilience as the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation. Considering that the IoE includes components from the IoT to the Fog and the Cloud, the main resilience challenges emerge from the objects and communication point of view.

To increase the resilience of smart objects that enable the interaction with the physical world, replication and backup schemes must be implemented; however, how is it possible to adopt aforementioned schemes efficiently? One traditional approach is using a primary and backup model, where devices and services are duplicated for robustness purposes. This would not be adequate considering that this strategy could waste valuable resources in the already constrained Fog nodes in case no failure occurs. In this context, virtualization mechanisms have proven to be useful from the cost and operational perspectives. Emerging virtualization paradigms like Containers [84] and NFV [85] allow the improvement of the performance and availability of service and device components. Once mapped as a logical item, physical objects can be handled like any other piece of software, granting the possibility to apply migration, instantiation, and other well-known techniques over them. Thus, a failure concerning a service or device can be recovered by migrating or instantiating a logical object over a different physical device.

From the communication point of view, the traditional approach of distributed systems relies in trying to hide the distributed nature of the system to offer a perspective of a single machine. In Cloud environments, this “hiding” approach remains for the network layout considering that Cloud services just expose high-level information about their setup and distribution. On the other hand, in Fog scenarios it is essential to know about the network topology to take advantage of the geographical distribution which requires a more fine-grained topology abstraction. Thus, it is necessary to have an efficient and flexible way to control the route of the data and the topology of the communication infrastructure in the IoE.

Regarding the resilience at the communication infrastructure level, an approach in two phases could be applied using a detailed fine-grained topology abstraction at Cloud and Fog levels. In a first step, an offline mechanism to find disjoint paths between the components of the IoE could be executed to obtain backup paths that can be

switched in case of failures. In a second stage, the detection of a failure and the migration of the data flows will be performed inline. To achieve this last task, the use of path-splitting and multipath routing strategies appears to be a feasible solution [86].

To guarantee a smooth work of the proposals mentioned above from the resilience perspective, an Orchestrator should be in charge of intelligent migration and instantiation of resources and services providing a global view of the status of the IoE. Furthermore, the interaction between federative Clouds and services represents an additional challenge, since the Orchestrator has to unify politics from different administrative entities smoothly.

4.7 Prediction and optimization

A proper management of resources and services in an IoE environment, where these are geographically distributed generating multi-dimensional data in enormous quantities, is only possible if the orchestration process takes into consideration prediction and optimization mechanisms of all overlapping and interconnected layers in the IoE [63]. Thus it is necessary that the Orchestrator has a global view of all the resources and services, from the edge of the infrastructure to the computation and storage place on the Cloud.

Although a global view of the infrastructure helps in the management process, an efficient orchestration at the Fog level remains a challenge. The service oriented computing model applied in Cloud environments is based on hiding implementations and runtime details to services and applications to ease its deployment. In the case of the Fog paradigm, geo-distribution, physical location, and the type of node/device actually matters. Thus, it is necessary to find a middle ground that exposes enough details about the distribution and physical location of the edge devices to take advantage of prediction and optimization.

To achieve successful results in prediction mechanisms, it is necessary to collect an enormous amount of resources, services, and users' data to feed the proper algorithms. Here, data analytics plays a vital role to extract useful information from the data gathered and perform prediction and optimization tasks. Taleb et al. [87] introduced the term Follow Me Cloud (FMC) to denote a framework that enables mobile Cloud services to follow their respective mobile users during their journeys by migrating all or portion of the services to optimal computational centers ensuring the best Quality of Experience (QoE). This new paradigm brings a huge challenge for the orchestration and management of resource in Cloud and Fog environments considering that in many cases prediction mechanisms should be used in order to guarantee the proper QoE.

Some efforts have been performed in prediction. Nadembega et al. [88] present a mobility-based service migration prediction for Cloud and Fog environments to ensure QoE to users while avoiding signaling messages and reducing the amount of data transferred between data centers and Cloudlets. The prediction mechanisms used in this research used three schemes: (1) a data transfer throughput estimation scheme that aims to estimate in advance the time required for data to travel from the computational nodes to users; (2) a service area hand-off time estimation scheme to estimate that a user will remain inside a specific coverage area; (3) a service migration management scheme to split the requested service into offloading portion.

Patel et al. [89] measured the performance impact of prediction during live virtual machine migration using machine learning algorithms. Specifically, a time series prediction using historical analysis of past data relating dirty memory pages was performed using two prediction models; the first one using an autoregressive integrated moving average model, and the second one based on a learning model using the support vector regression technique.

For the researchers above, the prediction impact in the overall performance was remarkable; however, to achieve these results the datasets used for the machine learning algorithms played a key role pointing out the importance of their significance for prediction mechanisms. Thus prediction is still an open issue in the context of orchestration and resource management into Cloud and Fog environments.

The planning and continuous optimization of resource management such as the placement of virtual services and other components on physical resources have an enormous impact on the effectiveness and performance of Cloud and Fog solutions. To address these issues, an analytical framework able to provide a mathematical foundation for optimizing or finding a trade-off between the (possibly conflicting) objectives involved in the optimization problems, that the Cloud and Fog Orchestrator have solved, will be useful [90].

A valuable analytical framework for orchestration optimization could be based on queuing theory [91]. In this context, computational tasks can be modeled as clients, and resources can be modeled as queueing nodes. Clients move across the network of queues following a routing policy and are serviced by each node according to some scheduling policy decided by the Orchestrator. Thus, different performance metrics can be evaluated. For example, the service latency can be measured as the total time the corresponding to the movement of the client from the source node to the destination node, and resource utilization could be mapped to the usage of corresponding nodes.

The use of offline techniques and frameworks like the ones explained before should be combined with monitoring and inline optimization mechanisms that enable flexibility and self-configuration in order to guarantee users' QoS. These challenges from the orchestration point of view on Fog environments remain open for deeper research and contributions.

4.8 Security and privacy

Many benefits may arise from the decentralization of Cloud-based solutions through the distribution of computation, storage and communication responsibilities towards nodes closer to data sources.

From the point of view of security and privacy, there are mixed effects that must be taken into account. In the positive side, decentralization usually improves resilience, by simply removing central points of failure or compromise. Intermediate nodes have a better opportunity to detect and mitigate threats, filter malicious traffic and perform other security checkpoint activities. Distributed storage may also reduce the volume of a potential data breach or impact of government surveillance efforts, with clear improvements in privacy.

It is important to point out that the aforementioned benefits do not come without more complex deployment, policy, control and coordination requirements. The added complexity can reduce the overall security and privacy guarantees, as more components come under the influence of an *attacker*. Immediate and major challenges arising from decentralization are distributed infrastructure protection, identity lifecycle and cryptographic key management (i.e. secure generation, distribution, exchange, storage, use, and replacement of credentials and keys). These are rather classical problems, but they receive new undertones which may allow interesting trade-offs and novel solutions in the context of Fog computing.

Infrastructure protection includes many different types of threats, ranging from network security solutions to malware detection and elimination. When interest is restricted to applications of cryptography, secure routing protocols become of particular interest, since route manipulation is a relatively simple attack vector for performing denial of service attacks or simply deviating sensitive traffic towards nodes under control of an attacker [92]. There are several proposals available in the research literature for securing routing protocols [93], but surprisingly they have not been adopted or even considered for standardization.

The Fog computing model presents another opportunity for improving this situation, due to higher computing and storage resources typically available in routers and other pieces of modern network infrastructure. Additional challenges in the infrastructure protection space are to coordinate distributed detection of malicious code and

traffic, potentially involving different service providers and manufacturers of infrastructure equipment [94, 95]. The fierce competition in these markets introduces additional privacy requirements, where threats must be detected in a privacy-preserving way without disclosing critical information.

Security mechanisms are not restricted to the basic networking level and can also be important at much higher abstractions, for example at the service provisioning level. As services become more distributed, information such as service type and interface, device hostname and ownership may be considered sensitive and require protection. Significant attention has been dedicated to the design of protocols for private (as in *privacy-preserving*) service discovery over the network [96]. Unfortunately, many of these protocols were proposed very recently and have not been thoroughly analyzed regarding security, performance or ease of deployment, what amounts to an interesting research challenge. These are important aspects that must be considered by a Fog Service Orchestrator.

In the scope of privacy, there are other relevant research goals beyond the service provisioning level. The Fog computing model transfers many Cloud-based functionalities to the infrastructure, such as data aggregation. In such a task, near-user edge devices combine partial observations about monitored metrics or characteristics to provide a complete view to the upper layers in the hierarchy, preferably by preserving the privacy of the lower-level users. It turns out that privacy-by-design mechanisms offer interesting solutions to this problem that can be efficiently implemented at these points using differential privacy techniques [97].

Despite the fact that solutions for security and privacy issues are well-studied in Cloud environments, not all of them are suitable for the Fog due to their different characteristics as well as the vast scale of devices at the edge of the network. From the privacy perspective, the main challenge lays into how to preserve the end user privacy since the Fog nodes are deployed near them collecting sensitive data concerning to identity and usage patterns. Regarding security, a significant challenge is how to deal with the massively distributed approach of the Fog to guarantee the proper authentication mechanisms and avoid massive distributed attacks. Thus, it is necessary to outline future research directions to cope with the challenges discussed in this subsection.

4.9 Authentication, access, and account

To perform activities related to application life cycle management (i.e. deployment, migration, application of policies) the Orchestrator interacts with the Cloudlets in the environment. However, due to the degree of heterogeneity of the Cloudlets in relation to the security aspects, new challenges arise when carrying out this

interaction [98]. The interaction between the Orchestrator and each Cloudlet follows the steps described in the well-known Authentication, Access, and Account (AAA) framework. Authentication makes it possible to identify the Orchestrator in the Cloudlet by querying its credentials. Allocation allows the Orchestrator to perform a set of actions (e.g. access to selected resources and execution commands) after it is authenticated. Finally, the account records the amount and time of use of the resources by the Orchestrator.

Regarding Authentication, the Orchestrator must provide the means to handle the most diverse Cloudlet authentication approaches and protocols. Two challenges are inherent in this action, manage the authentications and the passage of credentials by the Orchestrator to Cloudlets over a network. Stojmenovic et al. [99] also emphasize the need for the Orchestrator to prevent malicious attacks that act on the authentication process, such as the Man-in-the-Middle attack [100]. The use of a cryptographic key distribution infrastructure [101, 102] can improve the authentication process in terms of security and control.

The Fog environment is oriented to low latency, which may lead to the use of symmetric keys because of its low complexity in relation to the asymmetric keys in the implementation of the key infrastructure for authentication. Nevertheless, this choice brings with it one of the main problems in using symmetric keys: the possibility of compromising the whole environment if a key is compromised (e.g. stealing). As mentioned by Dastjerdi et al. [101], another approach that can be used to “hide” credentials is Trusted Execution Environment (TEE) [103]. However, factors such as complexity and speed decrease also arise when using this approach. In addition to these solutions, it is possible to delegate the authentication and authorization process to Internal or External Identity Providers (IdP) [104]. IdPs are prepared to use standard protocols for these purposes (e.g. X.509 [105] and Security Assertion Markup Language [106]). In contrast, while addressing the heterogeneity problem, the insertion of a new layer in the process may increase the cost of the Orchestrator to perform authentication and authorization.

To use the resources and perform actions in the Cloudlets, it is necessary that the Orchestrator is authorized to do so. The set of actions and resources available to the Orchestrator is determined and disseminated by the Cloudlets. Consequently, it is required to design and develop mechanisms to ensure that this information matches the current state of the Cloudlets and that the information collected reflects the set of actions and resources to which the Orchestrator has access. This disclosure feeds a catalog that the Orchestrator uses to manage the applications. The discrepancy in the

information can lead to the commitment of the orchestration plan consuming more time in an application deployment or migration. The role of the accounting process in this context is to help feed the catalog, accounting for the resources used and the resources that will be released after the migration or termination of an application.

Note that due to the dynamicity of the Fog environment, considering the Cloudlet and applications lifecycle, AAA-related processes can be executed several times in a short space of time which can cause overhead. Additionally, the diversity of Cloudlets can expose the environment to malicious attacks, compromising the orchestration. Thus, it is necessary to consider this factor and the consequent impacts on the orchestration of each approach employed for AAA.

The Fog requires a well-constructed Orchestrator able to deal with the full management function for this complex environment, and properly handle all the challenges previously described. Several efforts have been carried out towards this direction, and some of them are outlined below.

5 Fog orchestrator architectures

The OpenFog Reference Architecture (OpenFog RA) was designed by the OpenFog Consortium as a guide to help in the design and maintenance of hardware, software, and system elements that are needed for Fog computing environments [20]. The architecture is structured by a set of *pillars*, which represent key attributes needed to provide distribution of computing, storage, control, and networking functions in the vicinity of the data source (i.e. users, things). These pillars are:

- Security, which ensures that the deployment will offer a secure end-to-end environment;
- Scalability, which allows adaptation to workloads, system costs, performance, and other needs;
- Openness, which permits Fog nodes to exist anywhere, be pooled by discovery, and be dynamically created;
- Autonomy, which enables Fog nodes to continue functioning and delivering services in case of an external failure;
- Programmability, which provides highly adaptive deployments, like retasking a Fog node for accommodating operational dynamics automatically;
- Reliability, Availability, and Serviceability, which guarantees the delivery of expected functionality under normal and adverse operating conditions;
- Agility, which focuses on the transformation of massive amounts of data into manageable formats, and also deals with the highly dynamic nature of the Fog handling sudden changes;

- Hierarchy, which helps standardizing the organization of multiple Fog islands in a single or federated system.

The OpenFog RA provides guidelines for the features a proper Fog system should offer; however, it does not include instructions about the management or orchestration of the scenario and the actors playing key roles in it (e.g. devices, services, and applications).

The main focus of this research is on orchestration; thus all the architectures evaluated in this section specifically describe how to deal with orchestration functions in Fog environments. Given that the Fog paradigm is relatively recent, not much research is available on the topic of Fog service orchestration. In certain stages of new technologies, there is a moment where the concepts are diffuse and can be applied or restricted to more situations. In the case of Fog computing, there are divergences on the definition when considering other technologies with similar purposes.

For example, some authors describe Fog and Edge computing as distinct technologies [107, 108], while other authors interpret both as synonymous when considering Fog as a paradigm of computation [109, 110]. It is also possible to assume Mobile Edge Computing (MEC) [111] as an interpretation of Fog environment but with a specific niche [16]: mobility. In MEC, Cloudlets are mobile devices that interact directly with an IoT or cellular layer. Such Cloudlets are orchestrated centrally in an upper layer (Cloud). Following this idea, this section presents some Fog-based architectures and some MEC-based architectures, to later on analyze them.

For the selection of the architectures to analyze, we reviewed existing Fog computing orchestration literature published between January 2008 and September 2017. The publications were located using keyword search on Google Scholar and other academic databases, such as ScienceDirect, Springer, IEEE Xplore, and ACM digital Library. The keywords we used included “Fog computing orchestration”, “Cloud computing orchestration”, “Edge computing orchestration”, and “Mobile Edge computing orchestration”.

The search performed left us around seventy-five papers published in the context of Cloud/Fog orchestration. From this subset, approximately twenty researches proposed Fog-enabled architectures regarding orchestration but just half of them (i.e. ten works) described in detail the modules inside the architecture and their roles. These works were published after 2014, confirming the novelty of the topic.

Finally, among the few works we were able to find regarding Fog orchestration or other environments that could be extrapolated (i.e. ten works), this section describes a subset including the ones with higher citation

number (which reflects the impact that the publications generated in the scientific community), and more recent publication date (which covers the most contemporary research in Fog Orchestrators). Additionally, the works that did not address in enough detail at least half (five out of nine) of the research challenges identified in Section 4 were not included in this study because we considered they do not fulfill the minimum requirements of an end-to-end orchestration approach, or did not provide enough details on how to handle them, thus it could lead to an unbalanced analysis.

The four Fog service orchestration architectures selected to be discussed on this section are: SORTS, SOAFI, ETSI IGS MEC, and CONCERT; which cover at least half of the identified research challenges with enough detail so that they can be objectively compared.

5.1 SORTS

Velasquez et al. [112] proposed a hybrid approach for service orchestration in the Fog, framed in the SORTS

(Supporting the Orchestration of Resilient and Trustworthy Fog Services) project [113]. The infrastructure is divided by levels that are managed using both choreography and orchestration, according to the needs of the different levels. The architecture allows the use of different Orchestrator instances, corresponding to the optimization goals of various scenarios.

5.1.1 Logical network infrastructure

The SORTS infrastructure is divided into three levels, shown in Fig. 2 from bottom to top: (1) the IoT, (2) the Fog, and (3) the Cloud. The IoT level is composed of Virtual Clusters that represent a group of terminal communication devices (e.g. smartphones, vehicles); these devices can communicate with each other inside their Virtual Cluster or with neighboring Virtual Clusters, allowing mobility of the devices. At this level, a choreography approach is used, meaning that the devices cooperate among each other for managing purposes. This allows quicker response times in case of changes in the topology (e.g. shift from one Virtual

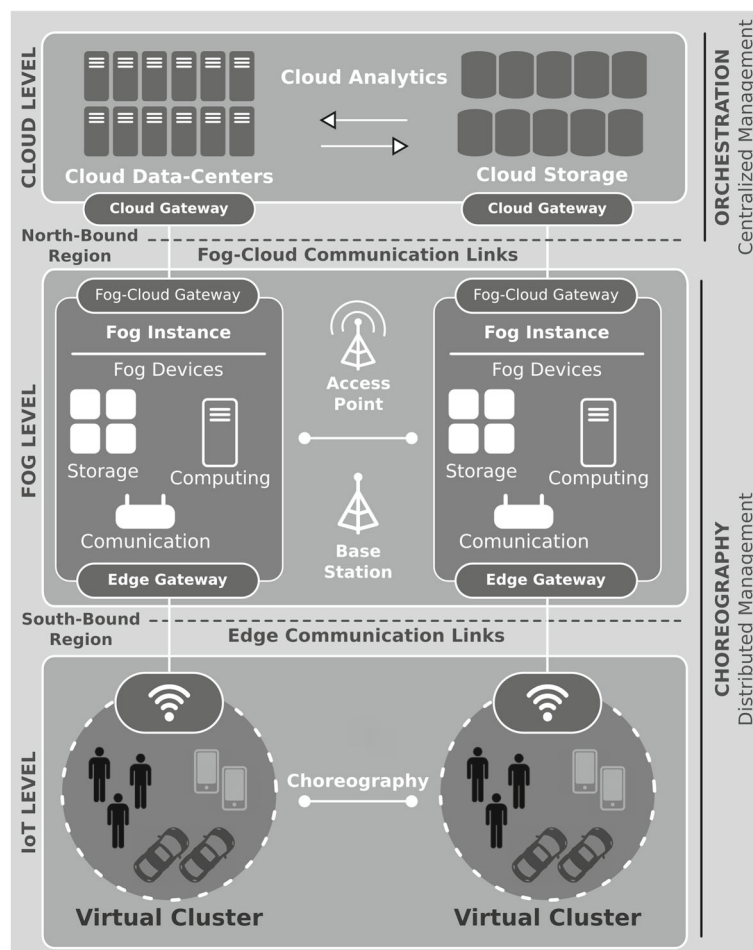


Fig. 2 SORTS Logical Infrastructure

Cluster to the next) thus increasing the resilience of the system.

The Fog level is divided into the South-Bound and the North-Bound regions. The South-Bound region, closer to the IoT, is composed of Fog computing devices, able to perform migration of services (code offloading); Fog communication devices, able to establish connections between the different levels of the infrastructure; and Fog storage devices, able to perform caching functions for the IoT users. These activities are controlled using choreography mechanisms.

The orchestration is used at the North-Bound region of the Fog level and the Cloud level (top level of the infrastructure). Fog/Cloud gateways and links are used to connect the Fog (North-Bound region) and Cloud levels. The Cloud level enables the use of a massive amount of resources for demanding storage and processing tasks. The hybrid approach (i.e. choreography plus orchestration) facilitates the achievement of a higher independence for the lower levels, granting them more dynamism and quicker response time in case of failures; at the same time, at the upper levels permits maintaining a global view that allows the implementation of optimization tasks involving the overall system.

5.1.2 SORTS orchestrator architecture

The architecture presented in Fig. 3 was designed to manage the resources and communication in the scenario previously described. Overlapped instances of the architecture are to be replicated at different Fog Instances and Virtual Clusters allowing the use of the distributed choreography mechanisms; and also at the Cloud level, where a single instance is deployed for global orchestration.

The Orchestrator is composed of different modules. The *Communication Manager* handles the communication among the different Orchestrator instances; the

Resource Manager monitors the resource usage of the infrastructure; the *Service Discovery* enables the lookup of services available in the nearest location; the *Security Manager* provides different authentication and privacy mechanisms.

The *Status Monitor* oversees the activities in the system; the *Planner Mechanisms* schedule the processes in the system and the location where they will be placed; and the *Optimization Mechanisms* which are meant to be applied at the upper levels, are used to improve the performance of the system.

5.2 SOAFI

The Service Orchestrator Architecture for Fog-enable Infrastructure (SOAFI) is a reference architecture proposed by Brito et al. [98]. The development of this architecture was performed to demonstrate the importance of the Orchestrator in a Fog environment. In addition, the work is focused in the portability of concepts from other environments into building an Orchestrator in the context of Fog, such as the use of precepts of the TOSCA [114] and ETSI NFV MANO [115].

This architecture (see Fig. 4) consists of two main elements organized on a client-server model: *Fog Orchestrator* and *Fog Orchestration Agent*.

5.2.1 Fog orchestrator

The Fog Orchestrator (FO) is a centralized entity that organizes the Fog nodes into logical groups called Logical Infrastructure. Through this grouping, it is possible to create the hierarchy of capacity and objectives within the framework. The responsibilities of the FO are divided into *Infrastructure Management*, *Orchestration* itself, *Security*, and *Monitoring*.

- *Infrastructure Manager*, handles all resources present in the Fog, maintaining the tasks of discovery, allocation, and catalog of resources. Such information generated by infrastructure management components is used by the Orchestrator to perform its activities;
- *Orchestrator*, carries out the composition of resources to offer new services. After receiving a template with information about the characteristics of the service, the Orchestrator requests information from the Infrastructure Manager to create and execute an orchestration plan. Another component that the Orchestrator relates to is the Monitoring, from which obtains information to support its operations. The process can be performed manually, through an external request. It can also be performed automatically, responding to an internal request;
- *Security*, provides data security mechanisms. In addition, it performs authentication management activities that are required in a heterogeneous and dynamic environment such as the Fog;

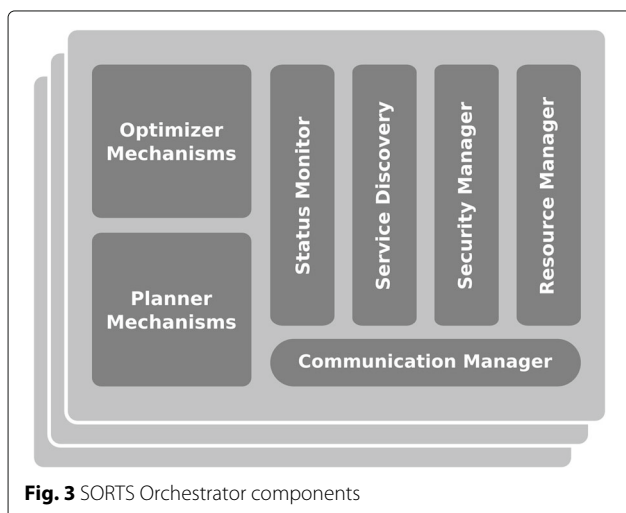
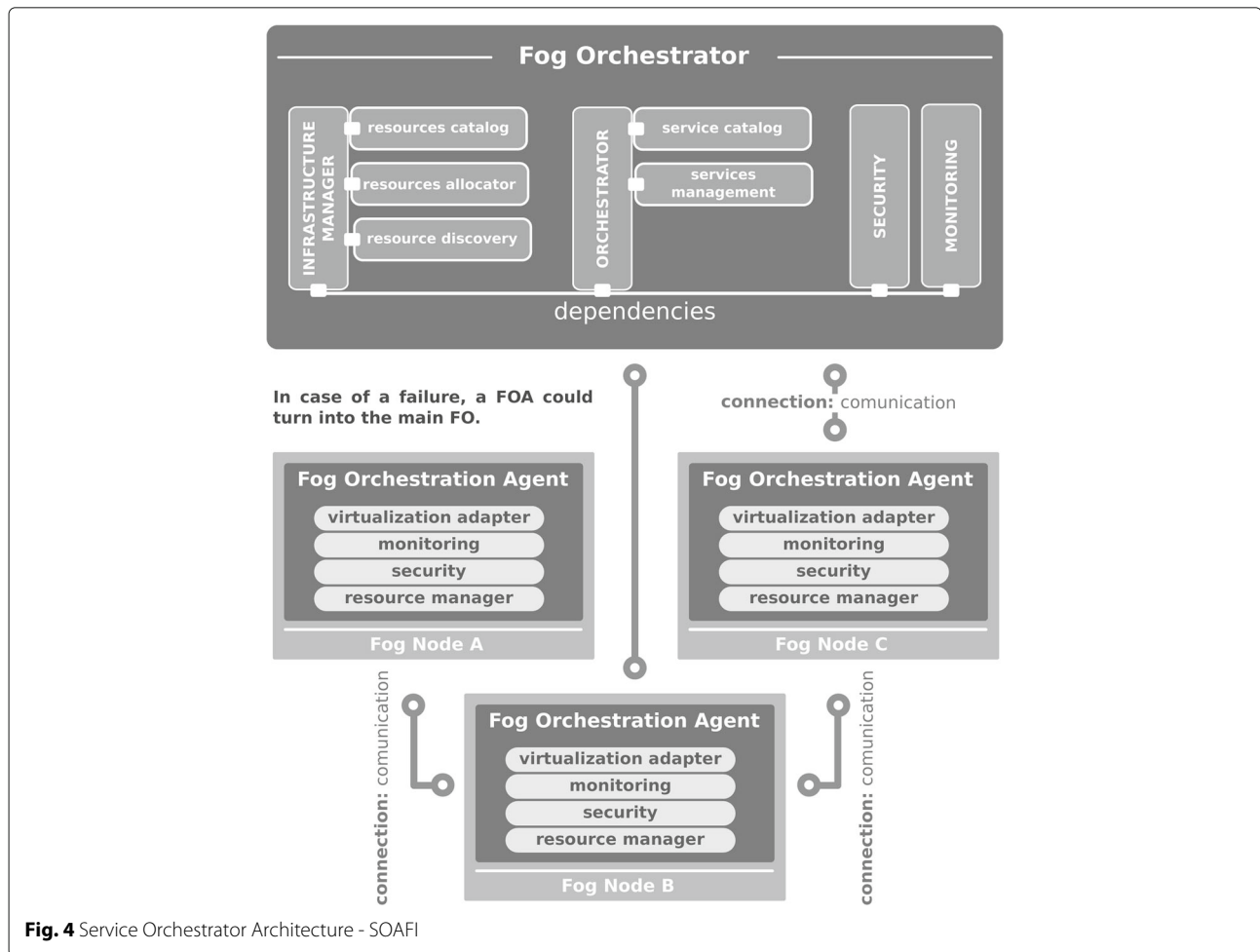


Fig. 3 SORTS Orchestrator components



- *Monitoring*, interacts with the Cloudlets to obtain data about the system and create a global view of all Cloudlets.

5.2.2 Fog orchestration agent

Within SOAFI, each Cloudlet has a Fog Orchestration Agent (FOA) installed which is an interface with the Fog Orchestrator. The authors of the architecture describe that in addition to interfacing with the FO, the FOA has a set of responsibilities, such as:

- *Allocation*, manages resources present in the Cloudlet. The management activities are conditioned to the degree of authorization that FOA has in relation to the resources of interest;
- *Discovery*, detects the connection and disconnection of resources in a Cloudlet. Another related activity is the announcement that each Cloudlet makes regarding its presence to the rest of the Cloudlets present in the environment. This publication allows the FO to discover new Cloudlets;
- *Optimization*, locally manages the running services and enables the creation of a set of policies for the execution of virtualized environments;
- *Interoperability*, maintains the Machine to Machine (M2M) interoperability through a standardized communication way that considers different communication protocols. It also provides support for various virtualization approaches (e.g. Containers and VMs).

According to Section 3, Cloudlets are autonomous entities. Because of this behavior, Cloudlets may never communicate with the FO. This absence of communication can be derived from the absence of the FO in the environment or due to other reasons such as the impossibility of communication as results of problems in the network. In this situation, the architecture allows the FOA to temporarily become a FO, as long as it has the conditions to support all Fog Orchestrator responsibilities. With this, the FOA can temporarily perform the orchestration of its services in the Cloudlets.

5.3 ETSI IGS MEC

This subsection presents the reference architecture for Mobile Edge Computing ETSI GS MEC [116] described by the ETSI Industry Specification Group. In the reference architecture, the Cloudlets are the Mobile Edge Hosts (ME Hosts). The ME Hosts contain the resources (i.e. compute, storage and network) and available components within the architecture. In addition to the ME Hosts, the architecture contemplates more modules (see Fig. 5). Among them is the Orchestrator that is the component with the main function within the architecture.

5.3.1 Mobile edge orchestrator

The Mobile Edge Orchestrator (MEO) performs the planning, deploying and managing of the application's lifecycle. To achieve this, it communicates with other components of the architecture to obtain the state of the resources (available and used), the executing applications and the current capacity. These communications are described in the reference architecture by points of interaction that are channels of communication between components present in the architecture (see Fig. 5):

- *Mm1*, receives requests to start and/or terminate applications in the environment;
- *Mm3*, allows obtaining information about the state of the applications. It also enables the Orchestrator to maintain up-to-date service information and manage application-related policies;
- *Mm4*, maintains resource management and application deployment images by the respective ME Hosts. The information obtained in these interactions will help the Orchestrator to build a catalog of the resources;
- *Mm9*, handles the requests to migrate an application. This migration may be internal to the MEC domain, or external to another domain.

The architecture describes the need for the presence of several points discussed in Section 4 (i.e. *Scheduling, Discovery, Allocation, Optimization, Authentication, Access, and Account*). However, as it is a reference architecture, how these points will be implemented depends on the technology used and the needs of the niche to which the architecture will be applied. It is important to say that the

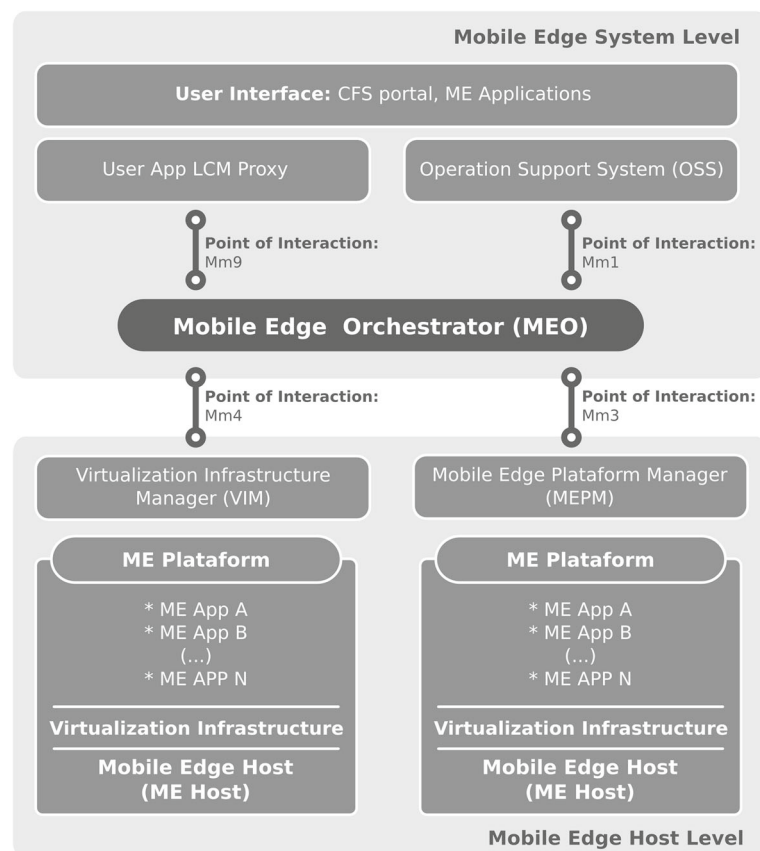


Fig. 5 ETSI GS MEC reference architecture

official documentation does not indicate clearly whether these tasks are performed by the Orchestrator.

5.4 CONCERT

CONCERT is a Cloud-based architecture for next-generation cellular systems proposed using two prominent approaches, decoupling the control and data plane and deploying services closer to users (at the edge of the network). In CONCERT, the data plane deals with different physical resources which are coordinated/orchestrated at the control plane to expose them as virtual resources to services and applications placed according to final users requirements. Thus, Liu et al. [91] proposed a converged edge infrastructure for cellular network communication and mobile edge computing services.

The CONCERT architecture is depicted in Fig. 6. From the bottom to the top the data plane encompasses different physical resources which are managed in an upper and decoupled control plane by the *Conductor entity*. The Conductor orchestrates and virtualizes all the data plane resources. On the top of the architecture, software-defined services are deployed using virtual resources.

The CONCERT data plane includes Radio Interfacing Equipments (RIEs), software-defined switches and computational resources. The RIEs deal with the signaling process between radio and digital domains besides taking care of radio resource slicing functions. These system's components provide the last-hop communication to final users. Conjointly with the base stations, it is possible to provide local servers at the edge of the network (Fog) to minimize the latency of applications and provide ubiquity to final users.

The software-defined switches interconnect the RIEs and the computational resources under the supervision of the Conductor, which is responsible for constructing and updating the forwarding tables of the switches enabling a smooth communication between all the data plane components.

The computational resources are in charge of all the data plane computation. These resources are distributed in different location taking into considerations their computational capabilities, for example, placing them next to the RIEs to achieve better response times as it was mentioned before. Another possibility is to aggregate the data by small regions in regional servers to decide which data could be processed locally and which one must be for-

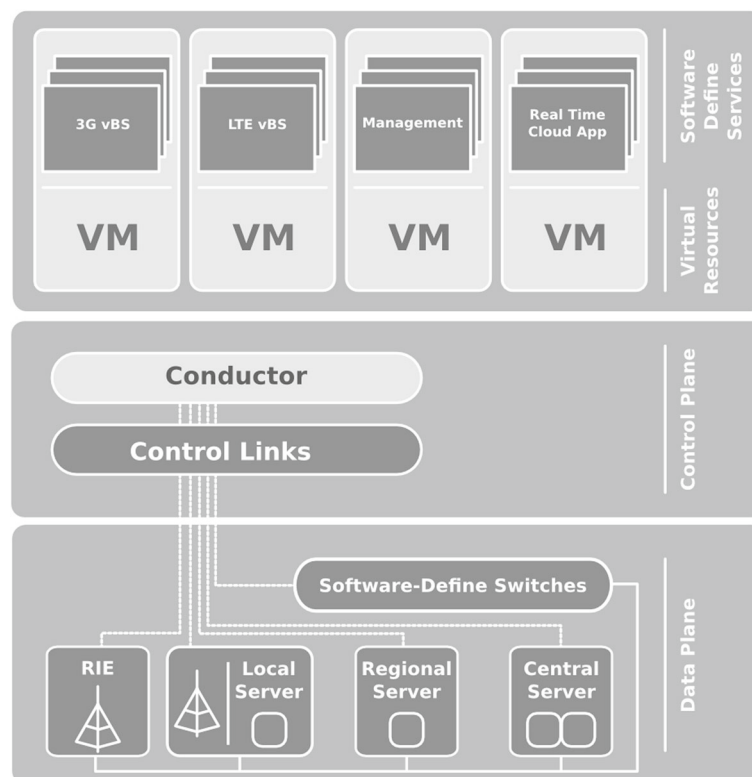


Fig. 6 CONCERT Architecture

warded to central servers in charge of hard demanding tasks.

The Conductor is the main component of the control plane, and its role is to orchestrate the physical resources available in the data plane and export them as virtual resources to the upper layers. The Conductor's mechanisms take care of the resource management and the communication infrastructure focusing on achieving applications' and users' requirements.

In Fig. 6, the mechanisms at the south-bound include radio interfacing management, networking management, and location-aware computing management; thus the data plane resources are orchestrated from the control plane. During the orchestration process, information about the state of the network infrastructure and the computational resource is gathered for optimization purposes.

The Conductor performs a variety of tasks via different mechanisms; for example, the Location-Aware Computing Management (LCM) mechanism is one of the most important given that it schedules computational tasks to computational resources. At this level, applications and services have different requirements, on the other hand, the computational resources may have different computational capabilities and are deployed in several locations. Thus the LCM has to decide the schedule strategies to fulfill applications' and users' requirements trying to find an optimal tradeoff among various objectives, such as low latency, resource utilization ratio, and resilience. In CONCERT, the Conductor can perform the tasks discussed above after collecting information about the deadlines, resource demands, location, and result destination of the applications and services in order to feed the proper mechanisms and algorithms embedded on it.

At the north-bound of the control plane, all the physical resources orchestrated are virtualized, creating a virtual infrastructure for the software-defined services.

At the top level of CONCERT, services are deployed following a software-defined approach taking advantage of the virtualization and orchestration performed by the Conductor in the control plane. Moreover, mobile edge Cloud services could be set up according to the user's requirements (e.g. low latency, high resilience) taking advantage of the LCM and other mechanisms provided by the Conductor.

In order to evaluate the reviewed architectures, a comparative analysis based on the challenges discussed in Section 4 is provided in the next section.

6 Discussion

A Fog Orchestrator requires a re-design of several mechanisms from well-known distributed systems (such as Cloud), in order to deal with the particular characteristics of the Fog. For instance, in the case of *Resource Management* (i.e. *Scheduling*, *Path Computation*, *Discovery* and

Allocation, and *Interoperability*) it is important to consider the heterogeneity of the Fog, as well as the resource constraints of the devices at this level, which is very different from the reality in the Cloud.

Also, it is important to integrate solutions to handle the production of massive amounts of data at this level, which represents another shift in the previously known paradigm, and the heavy distributed nature of this scenario that also implies the need of keeping a precise knowledge of the devices and their locations.

In the case of *Performance* (i.e. *Latency*, *Resilience*, and *Prediction and Optimization*), new solutions are arising to deal with the high dynamism of the Fog and their mostly wireless links, using some knowledge of the edge environment. Also, to increase the performance, data is usually aggregated at the Fog level before being sent to the Cloud, thus the new solutions should consider data aggregation and preprocessing, diverging from mechanisms used on classic distributed systems.

Finally, for *Security Management* (i.e. *Security and Privacy* and *Authentication, Access, and Account*), it is important to handle the vastly distributed environment which enables more points of failure and the possibility of massive distributed attacks, and also consider the new threat regarding confidentiality of the sensitive data located in the proximity of the users.

The architectures reviewed in Section 5 are evaluated in this section considering how efficiently they handle these challenges. Table 1 summarizes the reviewed architectures, and how they deal with the challenges present in Fog environments, discussed in Section 4. For each challenge, Table 1 provides a short description on how the challenge is handled. In the case that an architecture does not provide information about the support for a particular challenge, N/A (Not Available) is listed in the table. Furthermore, since the ETSI IGS MEC architecture is just a reference, and no details are provided regarding the implementation, no more specifics are included in the table.

In general, the most addressed challenges are *Scheduling*, *Discovery* and *Allocation*, *Prediction and Optimization*. These challenges are the classic ones on distributed and Cloud environments; thus various mechanisms to deal with these challenges have been migrated to the Fog from its predecessors. On the other hand, the less natively included properties are *Path Computation* and *Interoperability*. This could be due to the majority of solutions use well-known external paradigms to deal with them. For *Path Computation*, SDN and NFV are commonly imported by Fog Orchestrators to deal with dynamic routing; meanwhile, standard description languages, such as ontologies, are utilized to achieve *Interoperability*.

For *Latency* and *Resilience*, the majority of the solutions are working towards embedded mechanisms. This

Table 1 Comparative analysis fog orchestrator architectures

Challenges	SORTS	SOAFI	ETSI IGS MEC	CONCERT
Scheduling	Planner mechanisms schedule processes and their locations	Service management and catalog done by the Orchestrator	Planning, deployment and management carried out by the MEO	LCM implements scheduling strategies, services deployed using a SDN approach
Path Computation	N/A	N/A	Indicated as needed but no further details provided	N/A
Discovery and Allocation	Service discovery mechanisms at the Orchestrator to enable lookup	Handled by infrastructure manager	Indicated as needed but no further details provided	Orchestrate physical resources into virtual ones, carried out by the Conductor
Interoperability	N/A	M2M interoperability through standard communication	N/A	N/A
Latency	Service placement mechanisms at the Orchestrator to minimize latency	N/A	N/A	Provide local servers at the Fog to minimize latency
Resilience	Survivability mechanisms at the Resource Manager	N/A	N/A	Use of resilience metrics for scheduling purposes
Prediction and Optimization	Global mechanisms to improve performance of the system	Set of policies for virtual environments	Indicated as needed but no further details provided	State of the network used for optimization mechanisms
Security and Privacy	Security manager provides different privacy mechanisms	Data security mechanisms as dependencies of the Orchestrator	N/A	N/A
Authentication, Access, and Account	Authentication mechanisms supported by its security manager	N/A	Indicated as needed but no further details provided	N/A

represents a currently active and challenging field of research, taking in consideration that they support important metrics for final users that heavily impact QoS and QoE. Thus, more research is needed in these areas to better support applications and services in the Fog, looking towards improving services performance and user satisfaction.

In the case of the ETSI IGS MEC architecture [116], since it refers to a reference architecture, it only mentions that the challenges should be addressed, but does not specify how it must be done. From this review, SORTS [112] is the architecture that supports the most challenges. An hybrid approach of choreography and orchestration was proposed in this research to enable Cloudlets and IoT islands to take decisions in a distributed way while at the Cloud level the Orchestrator can aggregate information from the lower layers to manage all the resources using a global view of the infrastructure allowing optimal decision-making over the entire system.

Overall, the reviewed architectures are still under development. The majority of them propose theoretical

approaches to deal with the challenges present in Fog environments. Nevertheless, practical and experimental solutions are coming forward; once the research field becomes more mature, stronger solutions are to be expected.

7 Conclusions

In an Internet of Everything environment, smart devices communicate with each other and with the users through the Internet to gather, process, and analyze data, without much (or any) human intervention. This inevitably will enable the rise of new generation services and applications where unique and customized information will be processed for users on demand. This brings along different challenges that have to be addressed in order to guarantee their proper function while providing acceptable QoE for final users.

Fog Orchestration refers to the process of automating application workflows in the sense of providing dynamic policy-based lifecycle management of Fog infrastructure and services. The Orchestration includes the provisioning,

management, and monitoring on a large number of Fog nodes (i.e. Cloudlets) with a broad range of capabilities that include computing (computer resources), routing (network) and distributed databases (storage). The Fog Orchestration system must manage heterogeneous, and distributed systems spread across a wide geographical area. This requires a hierarchical organization with effective policies integrated with the Cloud orchestration system via intelligent interfaces.

In this paper, a revision on the Fog paradigm and its challenges is provided, to later on introduce a set of Fog service Orchestrator architectures, and how these deal with the challenges of the Fog. A comparative analysis is provided on the different architectures.

Further research has to be carried out to come up with stronger and more efficient architectures that include mechanisms and processes to handle the identified challenges and other issues not covered in this research. Future works also include determining additional research challenges and proposing how to manage them.

Acknowledgements

Karima Velasquez and David Perez Abreu wish to acknowledge the Portuguese funding institution FCT - Foundation for Science and Technology for supporting their research under the Ph.D. grants <<SFRH/BD/119392/2016>> and <<SFRH/BD/117538/2016>> respectively. The work presented in this paper was partially carried out in the scope of the projects: <<MobiWise: From mobile sensing to mobility advising>> (P2020 SAICTPAC/0011/2015), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), European Union's ERDF (European Regional Development Fund), and the Portuguese Foundation for Science and Technology (FCT); and SORTS, financed by the the CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior <<CAPES-FCT/8572/14-3>> and by the FCT - Foundation for Science and Technology <<FCT/13263/4/8/2015/S>>. This work is part of the INCT of the Future Internet for Smart Cities <<CNPq 465446/2014-0, CAPES 88887.136422/2017-00 and FAPESP 2014/50937-1>>.

Authors' contributions

KV, DPA, and MRM carried out the analysis of the State of the Art related with Internet of Everything, Fog and the Orchestration of resources in these environments; even more, they participated in the sequence alignment and drafted the manuscript. CS, DFA, LFB, and NL participated in the identification and description of the orchestration research challenges presented in Fog environments. MC, MV, EM¹, and EM² conceived and designed the study, besides reviewed the manuscript critically for important intellectual content. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹ Department of Informatics Engineering, University of Coimbra, Polo II - Pinhal de Marrocos, 3030-290, Coimbra, Portugal. ² Institute of Computing - University of Campinas, Av. Albert Einstein, 1251, Campinas - São Paulo, Brazil.

Received: 19 September 2017 Accepted: 4 May 2018

Published online: 18 July 2018

References

1. EU Commision. Communication from the Commision to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. Towards a Thriving Data-Driven Economy. 2014. <http://bit.ly/2wIVbVj>. Accessed 21 Feb 2017.
2. Senna C, Batista DM, Soares Junior MA, Madeira ERM, Fonseca NLS. Experiments with a self-management system for virtual networks. In: Proceedings of the XXIX Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2011) - II Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF 2011). Campo Grande: Brazilian Computer Society (SBC); 2011. p. 7–10.
3. Dastjerdi AV, Buyya R. Fog computing: Helping the internet of things realize its potential. *Computer*. 2016;49(8):112–6. <https://doi.org/10.1109/MC.2016.245>.
4. Kazmi A, Jan Z, Zappa A, Serrano M. Overcoming the heterogeneity in the internet of things for smart cities. In: Podnar Žarko I, Broering A, Soursoos S, Serrano M, editors. Interoperability and Open-Source Solutions for the Internet of Things. Cham: Springer; 2017. p. 20–35.
5. Byers CC, Wetterwald P. Fog computing distributing data and intelligence for resiliency and scale necessary for iot: The internet of things (ubiquity symposium). *Ubiquity*. 2015;2015(November):4–1412. <https://doi.org/10.1145/2822875>.
6. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener Comput Syst*. 2013;29(7):1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>.
7. ITU-T. Overview of the Internet of Things: ITU; 2012. <http://www.itu.int/rec/T-REC-Y.2060-201206-I>. Accessed 29 May 2017.
8. Minerva R, Biru A, Rotondi D. Towards a definition of the internet of things (iot): IEEE; 2015. <http://bit.ly/2rMn5uS>. Accessed 26 Jan 2017.
9. Kai K, Cong W, Tao L. Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues. *J China Univ Posts Telecommun*. 2016;23(2):56–96. [https://doi.org/10.1016/S1005-8885\(16\)60021-3](https://doi.org/10.1016/S1005-8885(16)60021-3).
10. Vaquero LM, Roderio-Merino L, Caceres J, Lindner M. A break in the clouds: towards a cloud definition. *SIGCOMM Comput Commun Rev*. 2008;39(1):50–5.
11. Geelan J, Klems M, Cohen R, Kaplan J, Gourlay D, Gaw P, Edwards D, de Haaff B, Kepes B, Sheynkman K, Sultan O, Hartig K, Pritzker J, Doerksen T, von Eicken T, Wallis P, Sheehan M, Dodge D, Ricadela A, Martin B, Kepes B, Berger IW. Twenty-one experts define cloud computing. *Electron Mag*. 2009. <http://cloudcomputing.sys-con.com/node/612375>. Accessed 05 Sept 2017.
12. Mell P, Grance T. The NIST definition of cloud computing. Technical report, National Institute of Standards and Technology, Information Technology Laboratory. 2009.
13. Bradley J, Loucks J, Macaulay J, Noronha A. Internet of everything (ioe) value index. Technical report, Cisco Systems Inc. 2013.
14. Atzori L, Iera A, Morabito G. The internet of things: A survey. *Comput Netw*. 2010;54(15):2787–805. <https://doi.org/10.1016/j.comnet.2010.05.010>.
15. Vaquero LM, Roderio-Merino L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *SIGCOMM Comput Commun Rev*. 2014;44(5):27–32. <https://doi.org/10.1145/2677046.2677052>.
16. Yi S, Li C, Li Q. A survey of fog computing: Concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data. Mobidata '15. New York: ACM; 2015. p. 37–42. <https://doi.org/10.1145/2757384.2757397>.
17. Yi S, Hao Z, Qin Z, Li Q. Fog computing: Platform and applications. In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb); 2015. p. 73–8. <https://doi.org/10.1109/HotWeb.2015.22>.
18. Chiang M, Zhang T. Fog and iot: An overview of research opportunities. *IEEE Internet Things J*. 2016;3(6):854–64. <https://doi.org/10.1109/JIOT.2016.2584538>.
19. Hung SC, Hsu H, Lien SY, Chen KC. Architecture harmonization between cloud radio access networks and fog networks. *IEEE Access*. 2015;3:3019–34. <https://doi.org/10.1109/ACCESS.2015.2509638>.
20. Group OCAW. OpenFog Reference Architecture for Fog Computing. Technical report, OpenFog Consortium. 2017.

21. Verbelen T, Simoens P, De Turck F, Dhoedt B. Cloudlets: Bringing the cloud to the mobile user. In: Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services. MCS '12; 2012. p. 29–36. <https://doi.org/10.1145/2307849.2307858>.
22. Jararweh Y, Tawalbeh L, Ababneh F, Dosari F. Resource efficient mobile computing using cloudlet infrastructure. In: 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks; 2013. p. 373–7. <https://doi.org/10.1109/MSN.2013.75>.
23. Gao Y, Hu W, Ha K, Amos B, Pillai P, Satyanarayanan M. Are cloudlets necessary? Technical report, Carnegie Mellon University, School of Computer Science. 2015.
24. Cisco. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. Technical report, Cisco. 2015.
25. Cisco. The Zettabyte Era: Trends and Analysis. Technical report, Cisco. 2016.
26. Bonomi F, Milito R, Natarajan P, Zhu J. In: Bessis N, Dobre C, editors. Fog Computing: A Platform for Internet of Things and Analytics. Cham: Springer; 2014. pp. 169–86.
27. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. MCC '12; 2012. p. 13–6.
28. Botta A, de Donato W, Persico V, Pescapé A. Integration of cloud computing and internet of things: A survey. *Futur Gener Comput Syst*. 2016;56:684–700. <https://doi.org/10.1016/j.future.2015.09.021>.
29. Varshney P, Simmhan Y. Demystifying fog computing: Characterizing architectures, applications and abstractions. In: 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC). Madrid: IEEE; 2017. p. 115–24. <https://doi.org/10.1109/ICFEC.2017.20>.
30. Jazdi N. Cyber physical systems in the context of industry 4.0. In: 2014 IEEE International Conference on Automation, Quality and Testing, Robotics; 2014. p. 1–4. <https://doi.org/10.1109/AQTR.2014.6857843>.
31. Breivold HP, Sandström K. Internet of things for industrial automation – challenges and technical solutions. In: 2015 IEEE International Conference on Data Science and Data Intensive Systems; 2015. p. 532–9. <https://doi.org/10.1109/DSDIS.2015.11>.
32. Rotsos C, Farshad A, Hart N, Aguado A, Bidkar S, Sideris K, King D, Fawcett L, Bird J, Mauthe A, Race N, Hutchison D. Baguette: Towards end-to-end service orchestration in heterogeneous networks. In: 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS); 2016. p. 196–203. <https://doi.org/10.1109/IUCC-CSS.2016.035>.
33. Fitbit, Inc. Fitbit. <https://www.fitbit.com>. Accessed 01 Aug 2017.
34. Alphabet Inc. Google Fit SDK. <https://developers.google.com/fit>. Accessed 30 Aug 2017.
35. Apple Technology Company. HealthKit. <https://developer.apple.com/healthkit/>. Accessed 29 Aug 2017.
36. Farahani B, Firouzi F, Chang V, Badaroglu M, Constant N, Mankodiya K. Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare. *Futur Gener Comput Syst*. 2017.
37. Kraemer FA, Braten AE, Tamkittikhun N, Palma D. Fog computing in healthcare: A review and discussion. *IEEE Access*. 2017;5: 9206–22.
38. Hosseini MP, Hajisami A, Pompili D. Real-time epileptic seizure detection from eeg signals via random subspace ensemble learning. In: 2016 IEEE International Conference on Autonomic Computing (ICAC). Würzburg: IEEE; 2016. p. 209–18. <http://dx.doi.org/10.1109/ICAC.2016.57>.
39. Monteiro A, Dubey H, Mahler L, Yang Q, Mankodiya K. Fit: A fog computing device for speech tele-treatments. In: 2016 IEEE International Conference on Smart Computing (SMARTCOMP). St. Louis: IEEE; 2016. p. 1–3. <https://doi.org/10.1109/SMARTCOMP.2016.7501692>.
40. Borgia E. The internet of things vision. *Comput Commun*. 2014;54(C): 1–31. <https://doi.org/10.1016/j.comcom.2014.09.008>.
41. Saini M, Alam KM, Guo H, Alelaiwi A, El Saddik A. Incloud: a cloud-based middleware for vehicular infotainment systems. *Multimedia Tools Appl*. 2017;76(9):11621–49. <https://doi.org/10.1007/s11042-015-3158-4>.
42. Rotsos C, King D, Farshad A, Bird J, Fawcett L, Georgalas N, Gunkel M, Shiimoto K, Wang A, Mauthe A, Race N, Hutchison D. Network service orchestration standardization: A technology survey. *Comput Stand Interfaces*. 2017;54:203–15. <https://doi.org/10.1016/j.csi.2016.12.006>. SI: Standardization SDN&NFV.
43. Chen M, Hao Y, Li Y, Lai C, Wu D. On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Commun Mag*. 2015;53(6):18–24. <https://doi.org/10.1109/MCOM.2015.7120041>.
44. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Comput*. 2009;8(4):14–23. <https://doi.org/10.1109/MPRV.2009.82>.
45. Amazon, Inc. AWS Greengrass. <https://aws.amazon.com/greengrass>. Accessed 07 Feb 2018.
46. Microsoft. Microsoft Azure - IoT Edge. <https://azure.microsoft.com/en-us/services/iot-edge>. Accessed 07 Feb 2018.
47. Pham X-Q, Huh E-N. Towards task scheduling in a cloud-fog computing system. In: 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS). Kanazawa: IEEE; 2016. p. 1–4. <https://doi.org/10.1109/APNOMS.2016.7737240>.
48. Song C, Qu Z, Blumm N, Barabási A-L. Limits of predictability in human mobility. *Science*. 2010;327(5968):1018–21. <https://doi.org/10.1126/science.1177170>.
49. Bittencourt L, Diaz-Montes J, Buyya R, Rana OF, Parashar M. Mobility-aware application scheduling in fog computing. *IEEE Cloud Comput*. 2017;4(2):26–35. <https://doi.org/10.1109/MCC.2017.27>.
50. Sun X, Ansari N. Edgeiot: Mobile edge computing for the internet of things. *IEEE Commun Mag*. 2016;54(12):22–9. <https://doi.org/10.1109/MCOM.2016.1600492CM>.
51. The Linux Foundation. Kubernetes. <https://kubernetes.io>. Accessed 18 Apr 2018.
52. Ooi BY, Chan HY, Cheah Y-N. Dynamic service placement and redundancy to ensure service availability during resource failures. In: 2010 International Symposium on Information Technology, vol. 2; 2010. p. 715–20. <https://doi.org/10.1109/ITSIM.2010.5561605>.
53. Campista MEM, Esposito PM, Moraes IM, Costa LHM, b. Duarte OCM, Passos DG, Albuquerque CVND, Saade DCM, Rubinstein MG. Routing metrics and protocols for wireless mesh networks. *IEEE Netw*. 2008;22(1): 6–12. <https://doi.org/10.1109/MNET.2008.4435897>.
54. Alotaibi E, Mukherjee B. Survey paper: A survey on routing algorithms for wireless ad-hoc and mesh networks. *Comput Netw*. 2012;56(2): 940–65. <https://doi.org/10.1016/j.comnet.2011.10.011>.
55. Paris S, Nita-Rotaru C, Martignon F, Capone A. Cross-layer metrics for reliable routing in wireless mesh networks. *IEEE/ACM Trans Netw*. 2013;21(3):1003–16. <https://doi.org/10.1109/TNET.2012.2230337>.
56. Marín-Tordera E, Masip-Bruin X, García-Almiñana J, Jukan A, Ren G-J, Zhu J. Do we all really know what a fog node is? current trends towards an open definition. *Comput Commun*. 2017;109:117–30. <https://doi.org/10.1016/j.comcom.2017.05.013>.
57. Marín-Tordera E, Masip-Bruin X, Almiñana JG, Jukan A, Ren G, Zhu J, Farre J. What is a fog node A tutorial on current concepts towards a common definition. *CoRR*. 2016;abs/1611.09193. <http://arxiv.org/abs/1611.09193>.
58. Masip-Bruin X, Marín-Tordera E, Tashakor G, Jukan A, Ren G-J. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications*. 2016;23(5):120–28. <https://doi.org/10.1109/MWC.2016.7721750>.
59. Mahmud R, Buyya R. Fog computing: A taxonomy, survey and future directions. *CoRR*. 2016;abs/1611.05539. <http://arxiv.org/abs/1611.05539>.
60. Deng R, Lu R, Lai C, Luan TH, Liang H. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet Things J*. 2016;3(6):1171–81. <https://doi.org/10.1109/JIOT.2016.2565516>.
61. Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M. Blueprint for the intercloud - protocols and formats for cloud computing interoperability. In: Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services. ICIW '09. Washington: IEEE Computer Society; 2009. p. 328–36.
62. Assis MRM, Bittencourt L, Tolosana-Calasanz R. Cloud federation: characterisation and conceptual model. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. London: IEEE; 2014. p. 585–90. <https://doi.org/10.1109/UCC.2014.90>.
63. Wen Z, Yang R, Garrahan P, Lin T, Xu J, Rovatsos M. Fog orchestration for internet of things services. *IEEE Internet Comput*. 2017;21(2):16–24. <https://doi.org/10.1109/MIC.2017.36>.

64. Barnaghi P, Wang W, Henson C, Taylor K. Semantics for the internet of things: Early progress and back to the future. *Int J Semant Web Inf Syst*. 2012;8(1):1–21.
65. Shin S, Seo S, Eom S, Jung J, Lee KH. A pub/sub-based fog computing architecture for internet-of-vehicles. In: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). Luxembourg City: IEEE; 2016. p. 90–3. <https://doi.org/10.1109/CloudCom.2016.0029>.
66. Singh D, Tripathi G, Alberti AM, Jara A. Semantic edge computing and iot architecture for military health services in battlefield. In: 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC). Las Vegas: IEEE; 2017. p. 185–90. <https://doi.org/10.1109/CCNC.2017.7983103>.
67. Abreu DP, Velasquez K, Pinto AM, Curado M, Monteiro E. Describing the internet of things with an ontology: The suscity project case study. In: 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). Paris: IEEE; 2017. p. 294–9. <https://doi.org/10.1109/ICIN.2017.7899427>.
68. W3C. OWL 2. <https://www.w3.org/TR/owl2-primer>. Accessed 25 July 2017.
69. Pham X-Q, Man ND, Tri NDT, Thai NQ, Huh E-N. A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing. *Int J Distrib Sens Netw*. 2017;13(11):1–16. <https://doi.org/10.1177/1550147717742073>.
70. Velasquez K, Perez Abreu D, Curado M, Monteiro E. Service placement for latency reduction in the internet of things. *Ann Telecommun*. 2017;72:105–15. <https://doi.org/10.1007/s12243-016-0524-9>.
71. Chen Y-C, Lim Y-S, Gibbens RJ, Nahum EM, Khalili R, Towsley D. A measurement-based study of multipath tcp performance over wireless networks. In: Proceedings of the 2013 Conference on Internet Measurement Conference. IMC '13. New York: ACM; 2013. p. 455–68. <https://doi.org/10.1145/2504730.2504751>.
72. Vulimiri A, Michel O, Godfrey PB, Shenker S. More is less: Reducing latency via redundancy. In: Proceedings of the 11th ACM Workshop on Hot Topics in Networks. HotNets-XI. New York: ACM; 2012. p. 13–8. <https://doi.org/10.1145/2390231.2390234>.
73. Vulimiri A, Godfrey PB, Mittal R, Sherry J, Ratnasamy S, Shenker S. Low latency via redundancy. In: Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies. CoNEXT '13. New York: ACM; 2013. p. 283–94. <https://doi.org/10.1145/2535372.2535392>.
74. Moens H, Hanssens B, Dhoedt B, Turck FD. Hierarchical network-aware placement of service oriented applications in clouds. In: 2014 IEEE Network Operations and Management Symposium (NOMS); 2014. p. 1–8. <https://doi.org/10.1109/NOMS.2014.6838230>.
75. Xiong G, Hu Y-X, Tian L, Lan J-L, Li J-F, Zhou Q. A virtual service placement approach based on improved quantum genetic algorithm. *Front Inf Technol Electron Eng*. 2016;17(7):661–71. <https://doi.org/10.1631/FITEE.1500494>.
76. Steiner M, Gaglianella BG, Gurbani V, Hilt V, Roome WD, Scharf M, Voith T. Network-aware service placement in a distributed cloud environment. In: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. SIGCOMM '12. New York: ACM; 2012. p. 73–74. <https://doi.org/10.1145/2342356.2342366>.
77. Zhang Q, Zhu Q, Zhani MF, Boutaba R, Hellerstein JL. Dynamic service placement in geographically distributed clouds. *IEEE J Sel Areas Commun*. 2013;31(12):762–72.
78. Wang S, Urgaonkar R, Chan K, He T, Zafer M, Leung KK. Dynamic service placement for mobile micro-clouds with predicted future costs. In: 2015 IEEE International Conference on Communications (ICC); 2015. p. 5504–10. <https://doi.org/10.1109/ICC.2015.7249199>.
79. Dolui K, Datta SK. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In: 2017 Global Internet of Things Summit (GIOTS); 2017. p. 1–6. <https://doi.org/10.1109/GIOTS.2017.8016213>.
80. Osanaiye O, Chen S, Yan Z, Lu R, Choo KKR, Dlodlo M. From cloud to fog computing: A review and a conceptual live vm migration framework. *IEEE Access*. 2017;5:8284–300. <https://doi.org/10.1109/ACCESS.2017.2692960>.
81. Fajjari I, Aitsaadi N, Pujolle G. Cloud networking: An overview of virtual network embedding strategies. In: Global Information Infrastructure Symposium - GIIS 2013; 2013. p. 1–7. <https://doi.org/10.1109/GIIS.2013.6684379>.
82. Xiao J, Boutaba R. Reconciling the overlay and underlay tussle. *IEEE/ACM Trans Netw*. 2014;22(5):1489–502. <https://doi.org/10.1109/TNET.2013.2281276>.
83. Sterbenz JPG, Çetinkaya EK, Hameed MA, Jabbar A, Qian S, Rohrer JP. Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation. *Telecommun Syst*. 2013;52(2):705–36. <https://doi.org/10.1007/s11235-011-9573-6>.
84. Pahl C, Lee B. Containers and clusters for edge cloud architectures – a technology review. In: 2015 3rd International Conference on Future Internet of Things and Cloud; 2015. p. 379–86. <https://doi.org/10.1109/FICloud.2015.35>.
85. Matias J, Garay J, Toledo N, Unzila J, Jacob E. Toward an sdn-enabled nvf architecture. *IEEE Commun Mag*. 2015;53(4):187–193. <https://doi.org/10.1109/MCOM.2015.7081093>.
86. Perez Abreu D, Velasquez K, Curado M, Monteiro E. A resilient internet of things architecture for smart cities. *Ann Telecommun*. 2017;72:19–30. <https://doi.org/10.1007/s12243-016-0530-y>.
87. Taleb T, Ksentini A. Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Netw*. 2013;27(5):12–9. <https://doi.org/10.1109/MNET.2013.6616110>.
88. Nadembega A, Hafid AS, Brisebois R. Mobility prediction model-based service migration procedure for follow me cloud to support qos and qoe. In: 2016 IEEE International Conference on Communications (ICC); 2016. p. 1–6. <https://doi.org/10.1109/ICC.2016.7511148>.
89. Patel M, Chaudhary S, Garg S. Machine learning based statistical prediction model for improving performance of live virtual machine migration. *J Eng*. 2016;2016:9. <https://doi.org/10.1155/2016/3061674>.
90. Amato F, Moscato F. Exploiting cloud and workflow patterns for the analysis of composite cloud services. *Futur Gener Comput Syst*. 2017;67: 255–65. <https://doi.org/10.1016/j.future.2016.06.035>.
91. Liu J, Zhao T, Zhou S, Cheng Y, Niu Z. Concert: a cloud-based architecture for next-generation cellular systems. *IEEE Wirel Commun*. 2014;21(6):14–22.
92. Goldberg S. Why is it taking so long to secure internet routing? *Commun ACM*. 2014;57(10):56–63. <https://doi.org/10.1145/2659899>.
93. Butler KRB, Farley TR, McDaniel P, Rexford J. A survey of BGP security issues and solutions. *Proc IEEE*. 2010;98(1):100–22. <https://doi.org/10.1109/JPROC.2009.2034031>.
94. Oberheide J, Cooke E, Jahanian F. Cloudav: N-version antivirus in the network cloud. In: van Oorschot PC, editor. Proceedings of the 17th USENIX Security Symposium. Berkeley: USENIX Association; 2008. p. 91–106.
95. Krishnan S, Taylor T, Monrose F, McHugh J. Crossing the threshold: Detecting network malfeasance via sequential hypothesis testing. In: 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Budapest: IEEE; 2013. p. 1–12. <https://doi.org/10.1109/DSN.2013.6575364>.
96. Wu DJ, Taly A, Shankar A, Boneh D. Privacy, discovery, and authentication for the internet of things. In: Askoxylakis IG, Ioannidis S, Katsikas SK, Meadows CA, editors. Proceedings of the 21st European Symposium on Research in Computer Security. Lecture Notes in Computer Science. vol 9879. Cham: Springer International Publishing; 2016. p. 301–19.
97. Dwork C. Differential privacy: A survey of results. In: Agrawal M, Du D, Duan Z, Li A, editors. Proceedings of the 5th International Conference on Theory and Applications of Models of Computation TAMC. Lecture Notes in Computer Science, vol. 4978. Berlin: Springer Berlin Heidelberg; 2008. p. 1–19.
98. de Brito MS, Hoque S, Magedanz T, Steinke R, Willner A, Nehls D, Keils O, Schreiner F. A service orchestration architecture for fog-enabled infrastructures. In: 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). Valencia: IEEE; 2017. p. 127–32. <https://doi.org/10.1109/FMEC.2017.7946419>.
99. Stojmenovic I, Wen S, Huang X, Luan H. An overview of fog computing and its security issues. *Concurrency Comput Pract Experience*. 2016;28(10):2991–3005.

100. Conti M, Dragoni N, Lesyk V. A survey of man in the middle attacks. *IEEE Commun Surv Tutor*. 2016;18(3):207–51.
101. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R. Fog computing: Principles, architectures, and applications. *CoRR*. 2016;abs/1601.02752. <http://arxiv.org/abs/1601.02752>.
102. Misra P, Simmhan YL, Warrior J. Towards a practical architecture for the next generation internet of things. *CoRR*. 2015;abs/1502.00797. <http://arxiv.org/abs/1502.00797>.
103. Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: What it is, and what it is not. In: 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1. Helsinki: IEEE; 2015. p. 57–64. <https://doi.org/10.1109/Trustcom.2015.357>.
104. Celesti A, Tusa F, Villari M, Puliafito A. Evaluating a distributed identity provider trusted network with delegated authentications for cloud federation. In: CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization; 2011. p. 79–85.
105. ITU. X.509: Information technology - Open Systems Interconnection. <http://www.itu.int/rec/T-REC-X.509>. Accessed 28 July 2017.
106. OASIS. SAML Version 2.0. <http://saml.xml.org/saml-specifications>. Accessed 29 July 2017.
107. Tran TX, Hajisami A, Pandey P, Pompili D. Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. *IEEE Commun Mag*. 2017;55(4):54–61. <https://doi.org/10.1109/MCOM.2017.1600863>.
108. Ahmed A, Ahmed E. A survey on mobile edge computing. In: 2016 10th International Conference on Intelligent Systems and Control (ISCO); 2016. p. 1–8. <https://doi.org/10.1109/ISCO.2016.7727082>.
109. Satyanarayanan M, Chen Z, Ha K, Hu W, Richter W, Pillai P. Cloudlets: at the leading edge of mobile-cloud convergence. In: 6th International Conference on Mobile Computing, Applications and Services; 2014. p. 1–9. <https://doi.org/10.4108/icst.mobcase.2014.257757>.
110. Ruay-Shiung-Chang, Gao J, Gruhn V, He J, Roussos G, Tsai W-T. Mobile cloud computing research - issues, challenges and needs. In: 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering; 2013. p. 442–53. <https://doi.org/10.1109/SOSE.2013.96>.
111. ETSI, Vodafone, IBM, Huawei, Intel, Nokia Networks, NTT DOCOMO. Mobile-edge computing: Introductory technical white paper. Technical report, ETSI Industry Specification Group. 2014.
112. Velasquez K, Abreu DP, Gonçalves D., Bittencourt L, Curado M, Monteiro E, Madeira E. Service orchestration in fog environments. In: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud); 2017. p. 329–36. <https://doi.org/10.1109/FiCloud.2017.49>.
113. Centre for Informatics and Systems - UC. FCT/ CAPES - SORTS: Supporting the Orchestration of Resilient and Trustworthy Fog Services. <https://www.cisuc.uc.pt/projects/show/228>. Accessed 29 Aug 2017.
114. TOSCA. Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0. 2013. <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf>. Accessed 17 Feb 2015.
115. Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG). ETSI NFV Work Group. Network Functions Virtualisation (NFV); Management and Orchestration. <http://bit.ly/2UqtqAR>. Accessed 28 Apr 2017.
116. ETSI. Technical report, ETSI Industry Specification Group. 2016.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com