



Do we all really know what a fog node is? Current trends towards an open definition

Eva Marín-Tordera^{a,*}, Xavi Masip-Bruin^a, Jordi García-Almiñana^a, Admela Jukan^b,
Guang-Jie Ren^c, Jiafeng Zhu^d

^a Universitat Politècnica de Catalunya-Advanced Network Architectures Lab, UPC-CRAAX, Vilanova i la Geltrú, Spain

^b Technische Universität Braunschweig, Germany

^c IBM, Almaden Research Center, San Jose, USA

^d Huawei, Santa Clara, USA

ARTICLE INFO

Article history:

Received 22 December 2016

Revised 1 March 2017

Accepted 24 May 2017

Available online 27 May 2017

Keywords:

Fog computing

Fog node

IoT

Edge devices

Edge computing

ABSTRACT

Fog computing has emerged as a promising technology that can bring cloud applications closer to the physical IoT devices at the network edge. While it is widely known what cloud computing is, how data centers can build the cloud infrastructure and how applications can make use of this infrastructure, there is no common picture on what fog computing and particularly a fog node, as its main building block, really is. One of the first attempts to define a fog node was made by Cisco, qualifying a fog computing system as a “mini-cloud” located at the edge of the network and implemented through a variety of edge devices, interconnected by a variety, mostly wireless, communication technologies. Thus, a fog node would be the infrastructure implementing the said mini-cloud. Other proposals have their own definition of what a fog node is, usually in relation to a specific edge device, a specific use case or an application. In this paper, we first survey the state of the art in technologies for fog computing nodes, paying special attention to the contributions that analyze the role edge devices play in the fog node definition. We summarize and compare the concepts, lessons learned from their implementation, and end up showing how a conceptual framework is emerging towards a unifying fog node definition. We focus on core functionalities of a fog node as well as in the accompanying opportunities and challenges towards their practical realization in the near future.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Cloud computing has become an essential information technology power horse, commonly used by a myriad of applications, and valued by users to seamlessly run business, entertainment and social network applications at remote data center premises. The IT outsourcing feature of the cloud is not only bringing value added services, but also lowering expectations on the ability of edge devices to process the applications locally. The recent proliferation of Internet of Things (IoT)-related services, including eHealth [1], smart cities [2], smart transportation systems [3] and industrial scenarios [4], to name a few, are however challenging the performance of cloud computing, mostly for the reasons of unpredictable and often high communication latency, privacy gaps and related traffic loads of networks connecting cloud computing to end-users.

To address some of these limitations of cloud computing, the research community has recently proposed the concept of *Fog Computing* [5], aiming at bringing cloud service features closer to what is referred to as “Things,” including sensors, embedded systems, mobile phones, cars, etc.

Fog computing was initially proposed in the area of IoT to help execute applications and services. The work by Al-Fuqaha et al. [6] surveyed IoT concepts with fog computing to deploy IoT applications, such as, location, distribution, scalability, density of devices, and mobility support. The first more formal definition of fog computing, by Bonomi et al. in [5], stated that ‘*Fog computing is a highly virtualized platform that provides compute, storage and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of the network*’. A similar definition can be found in [7] stating that ‘*Fog computing is proposed to enable computing directly at the edge of the network, which can deliver new applications and services especially for the future of Internet*’.

In fact, a number of surveys focused on fog computing exists, see [8–10], aiming at revisiting fog computing concepts, thus defin-

* Corresponding author.

E-mail addresses: eva@ac.upc.edu (E. Marín-Tordera), xmasip@ac.upc.edu (X. Masip-Bruin), jordig@ac.upc.edu (J. García-Almiñana), ajukan@tu-bs.de (A. Jukan), gren@us.ibm.com (G.-J. Ren), jiafeng.zhu@huawei.com (J. Zhu).

ing what fog computing is, its challenges, possible applications as well as scenarios where fog computing may undoubtedly contribute to. Unlike the set of existing contributions surveying what fog computing is, this paper is not intended to revisit fog computing as a novel cloud paradigm. Instead, our aim is to emphasize on fog computing infrastructure deployment, also discussing about the need for conceptualizing what a fog node is, also considering how and where it might be deployed. Aligned to that objective, this paper extends the traditional scope of fog computing surveys by proposing novel resource organization concepts for a fog node leveraging abstractions and virtualizations of heterogeneous physical edge devices as the key pillar to accommodate physical edge devices heterogeneity.

This paper is structured into two main parts. The first part (Sections 2–4) surveys the state of the art in technologies for fog computing nodes, including functional and conceptual approaches to define a fog node and its relationship with edge devices, paying special attention to the contributions that analyze the role edge devices play in the fog node definition. The survey is conducted first responding to the need to know “what” a fog node may be (Section 2), “how” and “where” a fog node may be deployed (Section 3), and finally briefing new ideas on collaborative models highly relying on edge devices capacities (Section 4). In this survey part, we end up showing that, in theory, different research trends, such as fog computing, cloudlets, mobile edge computing, etc., propose similar things, namely, the use of proximate computational resources instead of remote resources at datacenters. We clearly put the focus on highlighting edge devices heterogeneity, from simple sensors and actuators to some other endowed with computational capacities, such as mobile phones or microcomputers, and show how different research trends handle such heterogeneity, from fog computing and cloudlets (edge devices are only mere data producer/consumers) to Mobile Device Cloud (edge device are used for application offloading relying on their computational capacity). The survey part of this paper ends with Section 4, aimed to open up the discussion by considering sharing or collaborative policies, particularly relying on edge devices endowed with enriched capacities.

After revisiting main contributions in the current literature, the second part of the paper (Sections 5 and 6) opens up the discussion on what a fog node could be, showing a conceptual framework towards a unifying fog node definition. After a deep review of the fog computing and more in general, the edge computing literature, we come back to the fog node concept, considering the lessons learnt from the different edge computing research areas, and make an attempt to define what a fog node might be on a scenario considering all different views. We then propose in Section 5, an open fog node definition as a potential strategy to properly manage all different fog node definitions and implementations as well as the edge devices heterogeneity, what is in fact, the main rationale supporting the need for conceptualizing a fog node. The accompanying opportunities and challenges towards the practical realization of such an open fog node definition are carefully emphasized in Section 6, paving the way to further research areas.

Finally, Section 7 summarizes and concludes the paper.

2. Moving cloud to the edge, introducing the “what”

This section aims at positioning current references in the area of fog computing discussing on how fog computing can be framed in the wider area of edge computing, and how the concept of fog node may support service execution at the edge. It is worth highlighting that the formal concept of fog computing is not disruptive. Since its inception, the main fog computing model has been perceived as what is known as edge computing, including cloudlets [11], Mobile Edge Computing [12], Intelligent Transport Systems

Clouds (ITS-Clouds) [13] and VANET (Vehicular Ad Hoc NETWORKS)-Clouds [14]. The overarching idea in all these concepts is to make it possible to run applications based on location and closer to the user, on virtualized hardware devices, as we have seen in mobile clouds, cloudlets, ITS-clouds, etc. Related to it are also efforts in the so-called Mobile Grid Computing (see [15–17]). The Mobile Cloud Computing [18] paradigm (MCC) is also close to the edge computing concept, as it aims at providing solutions to guarantee an efficient offloading of applications and services from mobile devices to remote resource providers –cloud, fog or cloudlets [19]. Intimately related to MCC, in Mobile Device Clouds (MDC) mobile devices offload their tasks to local clouds built by grouping neighboring edges (see for example [20–22]). Similar ideas can be found in Content Delivery Networks (CDN) [23]. In CDNs, cache servers are deployed at the edge of the network to reduce the latency when downloading content from remote sites. Table 1 illustrates some of the relevant edge computing proposals as they appear in the various categories and flavors. Although significant differences may be highlighted between these concepts, they all essentially propose the use of proximate computational resources rather than remote resources in datacenters. Some differences, we believe, stem from the research communities addressing them. For example, cloudlets come from the cloud computing research community while fog computing comes from the area of networking. The similarity/differences between these concepts are summarized in [8], stating that “some other concepts, not declared as ‘fog computing’, might fall under the same ‘umbrella’ e.g., cloudlets.”.

Recognized such conceptual proximity, the open question remains whether there can be a clear and well-accepted definition of the basic functional and conceptual entity in fog computing, i.e., the fog node. We envision the fog node concept as a key component in fog (and any fog-based) computing, to guarantee (i.e., ease control and management) the set of resources and functionalities requested by services to run on such a highly volatile, dynamic, and heterogeneous scenario. So far, a fog node was considered as a physical device that implements fog computing. – i.e., “what” –, as read in [7] ‘In fog computing, facilities or infrastructures that can provide resources for services at the edge of the network are called fog nodes.’, or in [24], ‘a fog node is the physical device where fog computing is deployed’. Indeed, in [24] authors do not conclude with a single fog node implementation strategy, but they propose a variety of devices as candidates, including routers, switches, wireless access points, video surveillance cameras, and Cisco Unified Computing System (UCS) servers. A common characteristic for all these devices to become potential fog nodes, is that they all embed computing, storage and networking capabilities, all essential to ease the execution of IoT applications [25].

However, unfortunately there is no consensus on “how” and “where” a fog node is implemented either. From our perspective, the most interesting aspect of a fog node definition is that we effectively need a system that can efficiently select the set of devices building the fog node as well as its main functionalities.

Moreover, we do not envision an “static” fog node definition, bounded to current fog computing demands and constraints, but an open definition where the fog node concept may be applied to any novel computing paradigm stemmed from fog computing. Following such an open definition for a fog node, this paper pays particular attention to conceptualizing a fog node in the recently coined Fog to Cloud (F2C) concept [26], as an advanced extension of the current fog computing. In short, F2C proposes a coordinated and layered management of all existing resources, from the cloud down to the edge, in order to both optimize services execution and enable new collaborative models based on resources clustering, sharing, etc. However, extending the fog node concept far beyond the current fog computing may have an immediate impact on “what” a fog node should be. The question now is: if a fog node is

Table 1
Edge computing categories.

Technology	References	Including mobile edge devices
Fog computing	[1,2,3,4,5,9,20,31,24,27,28,29,30,26,32,35,34,47,48,49]	Only when fog nodes include underlying edge devices
Cloudlets	[11,19,59]	No
Mobile edge computing	[12]	Yes, by definition
ITS and VANET clouds	[13,14,36,62]	Yes, vehicles
Mobile device cloud	[20,21,22,60,61,33]	Yes, the mobile user offloads to cloud and to other edge mobile devices
Content delivery networks	[23]	No
Grid proposals including edge devices (such as mobile grid computing)	[39,15,16,17]	Yes

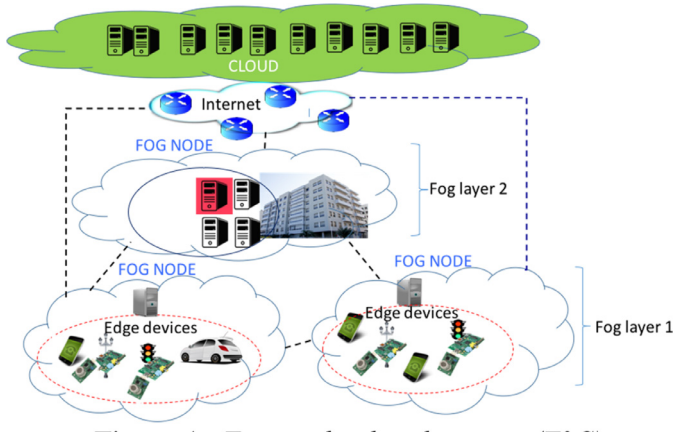


Fig. 1. Fog-to-cloud architecture (F2C).

the physical device where fog computing is deployed, what is a fog node when deploying a fog computing-based paradigm (e.g., F2C)? For the sake of illustration let us assume a F2C layered architecture connecting a set of devices and capabilities with a pool of resources, drawing a cloud layer and a few different hierarchical fog layers vertically and horizontally distributed. Fig. 1 depicts a possible F2C scenario including one cloud layer and two fog layers: fog layer 1 directly connected to the edge devices – mobile phones, sensors, small processing boards, etc. –, and fog layer 2 standing for an intermediate computing capacity layer, between fog layer 1 and cloud.

When applying the definition of “what” a fog node should be on the F2C scenario in Fig. 1, we may highlight two factors. First, at upper layers – see layer 2 in Fig. 1 –, where edge devices are not considered, it is very clear what a fog node should be, according to its definition. However, when considering the lowest fog layer – the one including edge devices –, two approaches come up. The first considers the fog node as the server (or mini cloud) connecting edge devices and responsible for running IoT services using data coming from the edge devices. The second considers the fog node as the whole set of components (including the server and the edge devices), thus requiring mechanisms for aggregating and controlling the edge devices capacities – storage, sensing, computing and network. Towards this vision, there is a myriad of concepts, approaches and ideas that can be evaluated on their suitability to help define fog nodes in future systems. The main goal of this paper is to systematically address issues relevant to providing a common and open definition for a fog node.

Finally, it is worth mentioning the absence of the concept of a cloud node, mainly because there is no need for defining such a concept. This assessment is supported by the static and much more centralized performance of cloud computing – even when considering distributed data centers, from our point of view they

are still centralized when compared with the distribution of “edge” devices. However, when moving to fog computing, its distributed nature, volatility, high mobility and heterogeneity make the reason for conceptualizing a fog node.

3. Revisiting fog node concepts: how and where

We start by categorizing contributions related to fog nodes found in the literature, into two categories, basically differing in the role edge devices play – i.e., characteristics and functionalities, namely the “how” –, as follows:

- fog nodes as mini-clouds with (“dumb”) edge devices acting as data producers/consumers, and;
- fog nodes as mini-clouds with (“smart”) edge devices enriched with significant IT capacities.

The two proposed categories stem from the edge devices description details provided by the papers referenced. When these details are not available, we infer the said characteristics and functionalities from other information, such as apps and services to be developed at that device, their location, etc.

There are several ideas in the recent literature falling into the first category, mini-clouds with (“dumb”) edge devices, including the industry-led proposals [24] and [25], the Smart Gateway proposed in [27], the eHealth services in [28], the micro data centers proposed in [29], or the proposal of fog nodes serving as caches in Information Centric Networking found in [30].

The second category, mini-clouds with (“smart”) edge devices, includes the early contribution on fog computing in [31], proposing a three-layered architecture, consisting of cloud data centers, fog nodes at the edge of the network and devices as end points. Closely related contributions in the second category may be also found in [2] and [32], leveraging the fact that applications are distributed across different layers and all assuming that (“smart”) edge devices collect and process the raw data gathered from sensors.

Let us illustrate the two said categories through the instructive example shown in Fig. 2. Suppose a smart city platform includes several bus stops equipped with fog nodes designed as mini-clouds that comprise at least one server with processing capabilities. Assume first that bus stops are fog nodes falling into the first category, i.e., the edge devices connected (sensors and actuators) are only data producers/consumers (see Fig. 2a) For example, sensors would measure the CO and CO₂ levels in the city and forward this data for processing to the fog node installed in the bus stop. The fog node in the bus stop will process the data collected from the different sensors in its coverage area, possibly merged with either other data gathered from other sensors (e.g., temperature, number of detected vehicles, etc.) or even with information downloaded from existing and data-related repositories at cloud. For instance, the goal of this processing may be to issue a warning to the city management, aimed at limiting the number of cars in that area of the city.

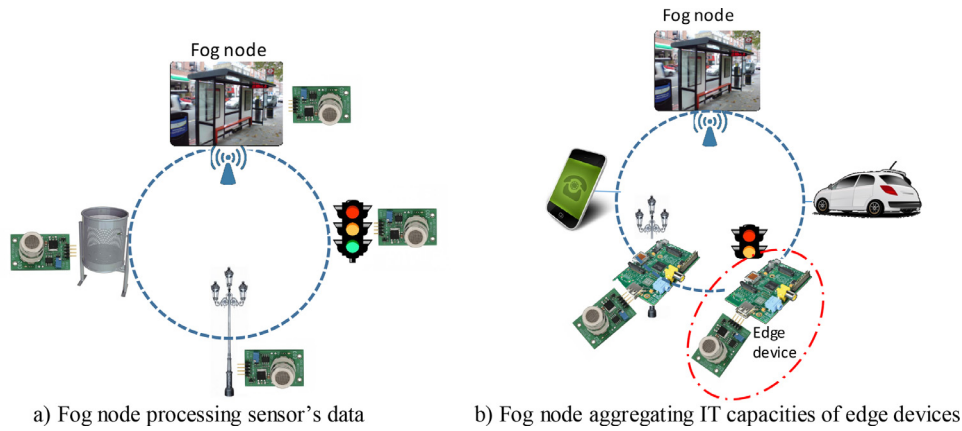


Fig. 2. Illustrative example for the two categories of fog nodes.

Table 2
Fog node location.

Gateways	[1,27,28,29]
Intermediate compute nodes	[2,3,5,31]
Network elements such as routers	[3,4,5,31,24,30,32]

Let now us analyze the second category in the same scenario, as illustrated in Fig. 2b. Here, the edge devices are now endowed with sensing but also computing capacities, including a sensor connected to a compute board, a mobile phone and a car (we assume that the car and the mobile phone also include a temperature sensor). The data processing is now possible by the edge device itself, hence producing information – pre-processed data –, to be forwarded to the mini-cloud within the fog node.

The illustrative example shown in Fig. 2 deals with the “how”, thus categorizing fog nodes depending on the edge devices characteristics and functionalities. The yet missing aspect is the “where”, that is, where fog nodes are located. Table 2 summarizes and categorizes references according to most common fog node locations.

Some of the revisited contributions propose to locate fog nodes in highly capable devices, such as routers or smart gateways. In such a group we may include contributions in [1,27–29], proposing the use of gateways for deploying fog computing in different scenarios (for example [1] and [28] both focus on eHealth). More in detail, in [1] the gateway is proposed to act as an intermediate point between sensors connected to the patient and the local switch/Internet, receiving data from the sensors, running some protocol conversion, and feeding the upper layer with services, such as data aggregation, filtering and dimensionality reduction. Authors in [28] propose to enrich the traditional gateway functionalities, with the capacity to pre-process data coming from an ECG device (Electrocardiogram). In a different scenario, authors in [27] propose a smart gateway to connect IoT producing audio or video data to the cloud. The proposed smart gateway is augmented with the capacity to process the data to be forwarded to the cloud via Internet. In [29], a smart gateway is proposed to implement the so-called fog micro-data center supporting functions of resource estimation and management.

On a different approach, works [3,5,31], proposed by Bonomi et al., established the foundations of fog computing as an intermediary computing layer between cloud resources and edge devices, designed in an open fashion such there is no dependency from specific devices. Authors in [2] propose the use of three fog computing layers for big data analysis in smart cities. The first fog layer, called intermediate computing nodes, consists of computers

with intermediate computer power. The second fog layer, called edge computing nodes, is built by small computing nodes (e.g., mobile phones). Finally, the third layer, called the lowest fog layer, consists of sensors with sensing capacities only.

Other proposals directly point out to network elements as the proper location for a fog node. In [4], centered on deploying fog computing in the industrial environment, fog nodes are implemented in Cisco edge routers, as it was first proposed by Cisco in [24]. Authors in [30] propose to deploy fog computing in routers at the edge of the network to implement ICN (Information Centric Networking). In [32] authors also propose a three-layered architecture, known as Cloud, Fog and Dew. In this structure, the Dew layer refers to the edge devices (e.g., sensors or cameras) and the Fog layer is implemented at devices at the edge of the network (e.g., network routers), and is responsible for providing compute, storage and application services closer to edge devices producing the data.

So far in this section, we have categorized existing references according to the “how” and the “where” aspects for a fog node. Summarizing the revisited contributions, we may with no doubt conclude that the scenario is very large, heterogeneous and diverse, and that there is no a globally accepted consensus on the “best” strategy for the “how” and the “where” to conceptualize a fog node. In such an uncertain scenario, we propose to move far beyond existing contributions, thus opening the fog node definition towards innovative ideas. Aligned to this approach, we highlight that the two fog node categories mentioned when addressing the “how” – i.e., the fog nodes with “dumb” edge devices that can only produce data, and fog nodes with “smart” edge devices pre-processing data –, do not paint the full picture for the envisioned scenarios to come in the near future. Innovative highly demanding services (for example a medical emergency service [26]) may require additional processing and storage capacity from a richer set of resources not included in the two categories above. Different solutions are possible depending on the services envisioned and the resources management strategy in place. For example, in emergency scenarios (natural disasters, critical events, etc.), computing capabilities can be acquired on demand, from volunteer sources, such as cars parked nearby, or individuals offering their smart phone resources to the emergency personnel if they happen to be close by. In [15], an interesting example was given of sharing smart phone computation resources only when phones are connected to the grid. Thus, adding processing capacities to edge devices (the second category) seems to be not enough to handle such highly demanding scenarios, therefore further concepts must be developed. Certainly, this new highly demanding scenario is un-

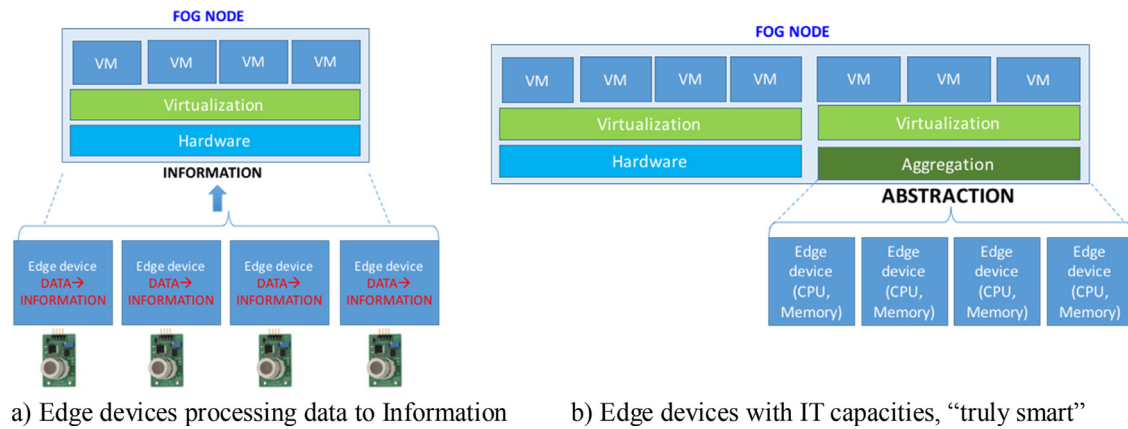


Fig. 3. Edge devices function.

questionably impacting on the fog node concept. More specifically, in this scenario, where edge devices can run external services, we pose the need for a global view of the fog node, meaning that the fog node will not only be the server acting as mini-cloud, but will also include the edge devices capacities. Nonetheless, although the deployment of such advanced concepts is innovative and rather interesting, it will likely increase the complexity of the overall system, requiring not only common but also standardized abstractions of the heterogeneous edge devices. To that end, an additional fog node category is discussed in Section 4 and the need for abstraction policies is developed in Section 5.

4. Meeting innovation at the edge: collaborative scenarios

As previously introduced, the concept of a fog node is essentially based on the characteristics and functionalities of edge devices, turning into two categories. In the first category, "dumb" edge devices are producing data (such as sensors), or acting as actuators. In the second category, "smart" edge devices include various compute, storage and networking capabilities. Section 3, however, ends up highlighting an innovative scenario where edge devices are "open" to new sharing policies, that is not covered by the previous categories and that will surely demand for new ideas on the way edge devices are managed. The main aim of this section is to dig into such scenario, emphasizing the challenges and surveying the related work.

4.1. Additional smart edge devices categorization

The need for an additional edge devices category is motivated by what we believe is the vision of a near future, where different IoT devices with IT/sensing capacities (i.e., those at the second category) can be used to create a large scale *grid* (or more than one *grid*) by means of novel sharing or collaborative policies. In a smart city scenario, for example, what is today a cloud service may be an entity able to request IT/sensing resources from the *grid*. A salient feature in this collaborative scenario is the ability of edge devices to broadly offering their capacities, be it sensing and/or processing.

To make it simple, we envision a third category of edge devices, stemmed from the second one, differentiating between specialized "smart" devices which only process the data from the sensors they are connected to (the second category), and general-purpose devices offering their resources for sharing, with various degrees of IT capacities, from smart phones to multi-platform management, such as clusters, grids, and ITS clouds. We refer to these edge devices in the third category as "truly smart". Fig. 3a and b illustrates the two categories of smart edge devices ("smart" and "truly

smart"), including the need for IT capacity abstraction for the latter (see Fig. 3b).

Therefore we summarize the role of edge devices in fog computing into three basic categories:

- "dumb" devices as mere data producers/consumers,
- "smart" edge devices with the capacity to process (only) their own data,
- "truly smart" edge devices offering their IT capacity to run distributed applications.

The novel sharing policies suggested for edge devices pave the way to deploy new business segments leveraging innovative collaborative models. But, as said before, adding new functionalities to the edge devices, now considered not only as single devices but also as a cluster of them, will undoubtedly increase the whole management complexity. Certainly, complexity will also depend on the set of functionalities to be provided by the edge devices. Thus, a comprehensive knowledge of the possible edge devices functionalities is required before devoting efforts to design a proper solution to manage them.

4.2. Related work on edge devices functionalities

We summarize in Table 3 the prior art in fog computing according to the set of functionalities embedded into the edge devices. The first three rows correspond to fog proposals with edge devices in the role of: i) mere data producers ("dumb"); ii) edge devices processing their own local raw data from connected sensors ("smart"), and; iii) edge devices offering their IT capacity to execute external services ("truly smart"). Recall that the classification about the role of edge devices is obtained from the details provided by the papers reviewed, and when these details are not explicit enough, we inferred this information from the overall context in the paper.

In the fourth row of the table we list proposals dealing with distributed computing. Though not directly linked to fog computing, proposals into the fourth row are listed intended to both, comparison purposes and because some of them include edge devices that could be categorized as "truly smart". In this fourth row we summarize some of the existing work categorized as distributed computing that includes edge devices, such as VANET clouds, Jungle computing, mobile grid computing or volunteer computing, where edge devices may be cars, mobile phones, etc. A common characteristic for all proposals listed in these four rows is that they require a triggering feature to run the application. In other words, all these options include some sort of management system and/or resource coordinator that allocates tasks to resources.

Table 3
Edge devices functionality.

Fog with edge devices as mere data producer/consumer “dumb”	[1,3,5,27,28,29,30]
Fog with edge devices processing local data “smart”	[2,31,32]
Fog with edge devices offering their IT capacity “truly smart”	[20,26,35,9,34]
Edge devices offering their IT capacity (distributed computing)	[14,39,15,16,17,36,40]
Offloading to edge devices	[21,22,33]

Contributions in the fifth row, also considers the distributed execution using edge devices (mobile phones). However, unlike the papers listed in previous rows, in this case the application is initiated at an edge device and part of the execution is offloaded to other edge devices. From the point of view of task offloading (cf. [33]), two main trends may be highlighted in the research literature, mini-cloud based architectures – where edge devices offload tasks to the mini-cloud (First and second rows) –, and collaborative architectures – where edge devices offload tasks to other edge devices (Fifth row). Thus, contributions in the third, fourth and fifth rows assume edge devices execute external applications, hence edge devices may be considered as “truly smart devices”. However, the question here is, should these edge devices be considered fog nodes?, or rather just as a part of the fog node capacity assuming the fog node as the mini-cloud? We make an attempt to address these questions in Section 5, when proposing an open definition for a fog node.

Coming back to the second row in Table 3, (“smart” edge devices), and for the sake of graphical illustration, Fig. 3a illustrates what is closest to the proposals of this category, whereby the output information from the edge device is not just the raw data, but rather a piece of elaborated information obtained through pre-processing in the edge device. In other words, edge devices process data collected from sensors/actuators they are connected to. The pre-processing is a highly beneficial feature as it reduces the amount of data sent to the fog node throughout the network, while offloading the pre-processing to the edge devices. This was in fact studied in [32], where the idea was to endow edge devices with collecting/generating and pre-processing capacities, turning raw data into information, which is propagated to higher levels, be it fog and/or cloud. Paper [2], on the other hand, proposes a hierarchical, layered-based architecture for big data analysis in Smart Cities, whereby Layer 1 is Cloud and Layer 2, Layer 3 and Layer 4 are considered three fog layers, as already described in Section 2. More in detail, smart edge devices in layer 3 can collect, aggregate, identify potential threat patterns – with applications of machine learning algorithms –, and finally convert the sensors collected raw data into information.

A key difference, highly impacting on the whole management complexity, refers to the strategy for handling IT capacities at the “smart” and “truly smart” edge devices. When the computing capacity is embedded in edge devices only to process local sensor’s data, as previously described, there is no need to either abstract or aggregate their capacity such it can be offered to another process of resource discovery, see Fig. 3a. This is not the case with the third category of edge devices, “truly smart”, where the role of edge devices further extends towards richer IT capacities, as shown in Fig. 3b, hence driving the need for resource aggregation and abstraction. Indeed, Fig. 3b illustrates the features of aggregation and abstraction of edge devices. A practical example may be inferred from Fig. 2b, assuming that the board attached to the sensor can also run applications not necessarily related to the data collected from the sensor itself. In a similar fashion, the car or the mobile phone, can also share their computational power to process service requests coming from an external service management system. As the computational power of edge devices is offered to run services in a distributed fashion, the challenge is to integrate such

a distributed set of edge devices with a cloud computing system, another fog computing system or with a new edge device.

In such a context of global resources integration, authors in [9] analyze the definition and role of fog computing, and specifically discuss the edge-cloud – referred in this paper as mini-cloud – and virtualized sensor networks. In their approach, applications are divided into *droplets*, tiny pieces of code running at edge devices, thus removing the unnecessary upload of data to central servers. Another proposal, called Mobile Fog [34], suggests a programming model to run hierarchically distributed applications according to their workload at cloud, fog and edge devices. Here, the fog nodes were defined as physical devices located inside the network infrastructure, and connected to mobile edge devices to include smart phones, vehicles, etc. The paper proposes an application, to be executed by invoking a specific function – called *connect_fog()* –, enabling the edge device to set a fog process connecting to the global Mobile Fog process running on a fog node.

Another related example with “truly smart” edge devices” can be found in [35], where it was proposed to use mobile phones to perform data analytics in IoT applications, whereby users offered any available resources based on some access policies and resource sharing principles. The main innovation here was to match the partitioning of the application data according to the capacity of the existing resources at the participating edge devices. In [20], a service is divided into different tasks and subtasks, hence enabling potential offloading towards either neighboring edge devices or to the cloud. Similar proposals to distribute application execution have been made also elsewhere. VANET (Vehicular Ad Hoc NETworks) Cloud proposals in [14] and [36] have also considered sharing of resources through edge devices (in this case vehicles could be considered as “truly smart” edge devices) as compute entities. In [36] it was proposed that in a fleet of cars, either one vehicle is appointed as the cloud controller, or all vehicles can act as the interface to the cloud networks. Similar proposals in the area of Mobile Device Clouds (MDC) can be found in [21,22,33]. Work in [21] proposes a task offloading mechanism from a mobile device to a MDC consisting of a set of mobile devices. The goal of the offloading mechanism is to maximize the lifetime of the MCD, that is, maximize the time that no device has exhausted its battery therefore balancing energy consumption on the different devices – based on similar ideas coming from opportunistic wireless networks. Authors in [22] propose an offloading mechanism where a mobile device initiates an application that is, totally or partially offloaded to other mobile devices. The goal of the proposal is twofold. On one hand, minimize the local power consumption – the consumption of the mobile device initiating the application –, and on the other hand, decrease the overall computation time; all constrained by the fact that connectivity with the other mobile devices is intermittent. In [33], mobile devices can offload their tasks to other mobile devices, with the control assistance of the base station that brings a better view of the global mobile devices’ state. In this scenario, mobile devices must report their available resources, be it either network or computational, to the mentioned base station. The proposed offloading mechanism, formulated as a Lyapunov optimization problem, has three main objectives: 1) to achieve an optimal energy conservation for all the users, 2) to provide a reasonable incentive scheme for the

users participating in the collaborative scheme, and finally 3) to get adapted to the continuous changes in the resources state due the unpredictable behavior of mobile devices.

Finally, the application execution in different clouds has been referred to as multi-platforms (clusters, grids and cloud) – not included in Table 3 –, stemming from the areas of high performance computing and parallel programming. Among a myriad of previous work proposing usage of using multiple platforms, of particular interest are those that combine cloud with other type of resources (grid and clusters) and the corresponding multi-platform resource allocation methods. For instance, applications are executed in a distributed fashion in different clouds [37], in a combination of cloud and grid resources [38], or in a combination of heterogeneous, hierarchical distributed and high performance resources, such as in *Jungle Computing* [39]. *Jungle Computing* represents the extreme case of distributed computing systems including stand-alone machines, clusters, grids, clouds and edge mobile devices, all meeting a common requisite, namely to share CPU, memory and communication capacities over a wide-area network. In [38] it was proposed to unify the view of all available computing resources (community grids, collaborative grids and cloud) by means of a grid overlay constructor. Cloud@Home is another known approach of combining cloud computing with shared resources at home [40]. In Leveraging Volunteer Computing [41], users share their own resources to be presented at cloud as virtual instances. A similar proposal for sensors and actuators can be found in [42], where a hypervisor was proposed for the abstraction and virtualization of sensors and actuators. The layer of abstraction provided by the hypervisor presents these sensors as virtual instances in cloud. The idea is interesting, and could be extended to include a broader set of edge devices. Regarding the limited resources of a sensor to be virtualized, the virtualization may be done not by slicing the sensor resources but by sharing one physical sensor among various services. In this way, each service can see the sensor as an isolated and exclusive resource, as this service is the only at a time obtaining data from that sensor.

5. Towards an open definition for a fog node

Previous sections focused on existing contributions particularly dealing with aspects related to what a common definition of a fog node might be. The focus was on fog infrastructure – that is, the set of server resources which related work denotes as fog nodes –, and the related edge devices connected to these nodes. After a careful analysis of the existing contributions and driven by the large heterogeneity of the proposed ideas, we may conclude that the work reviewed so far is more diverging than converging towards a common fog node understanding. In order to sort this out, in this section we put the focus on converging aspects towards a common definition of a fog node, while we remain cognizant of the ongoing evolution that is blurring the differences between clouds, fogs, and edge devices as the services become more oblivious of the infrastructure used.

To start off the discussion, let us consider the widest fog scenario, that is the one including all possible resources from cloud to the edge, to be represented by the distributed Fog-to-Cloud (F2C) system proposed in [26], where several layers are considered, some at cloud and some at fog (also including edge devices, see Figs. 1 and 4). Assuming this scenario to be the widest one in terms of resource capacities and heterogeneity, we envision fog nodes as a set of elements, including servers (e.g., mini-clouds) and edge devices (be it “dumb”, “smart” and/or “truly smart”), thus integrating the edge devices’ capacity (data processing, sensing) into the fog node definition. In this way, we foresee the whole resources orchestration and offloading strategies as an integral part of the functions of a fog node. For this idea to work, the compute and

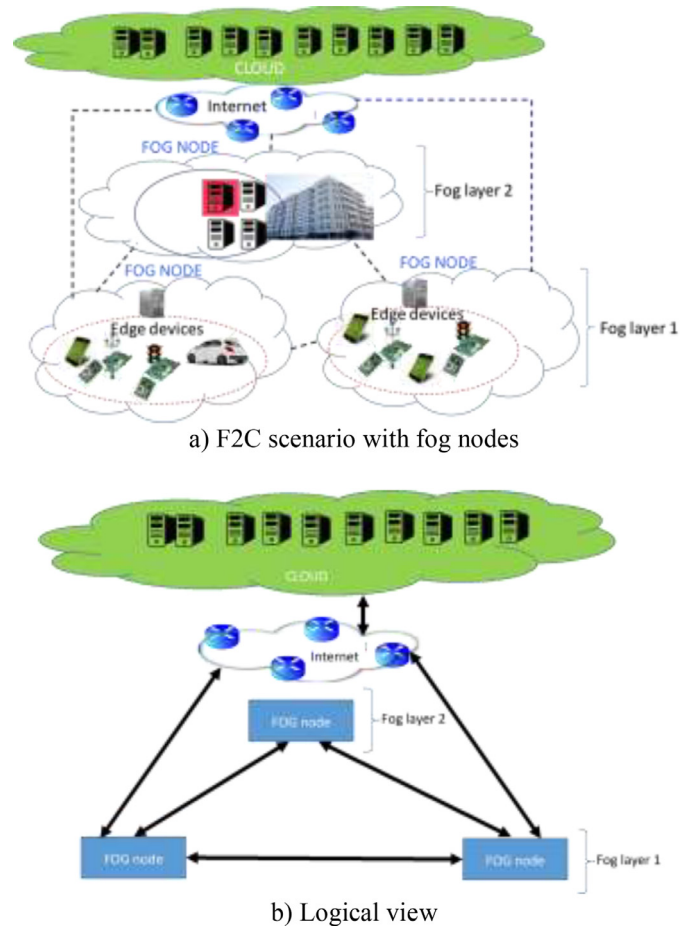


Fig. 4. Fog node model.

storage capacity at edge devices should be presented in terms of abstracted computing units, hence all resources, from cloud (for example, in the form of virtual machines or containers) to the edge must be properly managed. Given that, each fog node would include two type of resources: i) one or more computing servers and ii) the aggregated capacity of the edge devices, as illustrated in Fig. 3b. In line with this concept, a fog node would not be a specific device, or a set of specific devices, but rather a logical concept, with heterogeneous type of devices as its physical infrastructure. In Fig. 4 we present both the physical view and the logical view of different fog nodes in a typical F2C scenario to illustrate the logical concept for a fog node. In the example illustrated in Fig. 4, the fog node (layer 1) at the very edge of the network, will put together edge devices along with a mini-cloud. However, at higher levels in the F2C hierarchy, a fog node does not need to include the abstractions of edge devices, but only the server/s setting the mini-cloud. Such a layered abstraction is in fact the essence of the future joint Fog-to-Cloud architecture.

How various features of edge devices can be presented as logical instances in a fog node is yet an open question. Also, what the computing entity in the whole system is (the “how”), and its locality (the “where”) where these abstractions are created and managed are both open to discussion. For instance, one of the physical devices building the fog node (preferably the one with higher computing capacity) can be made responsible for deploying the abstraction strategy, similar to the concept of cluster head, while also granting communication between all fog layers and the cloud. In a more appropriate parlance of today’s systems, this could be referred to as the *fog node controller*. The resource discovery is an

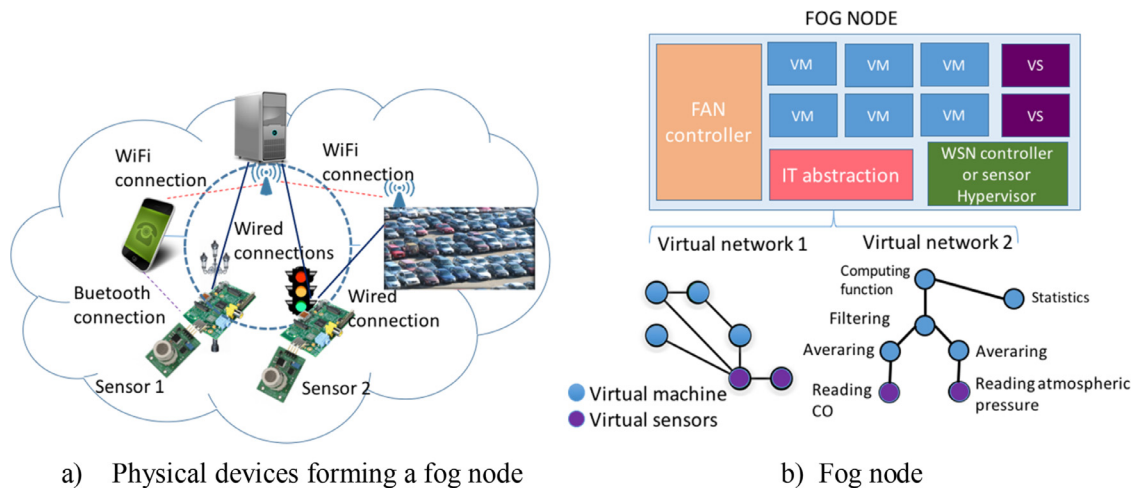


Fig. 5. Fog node proposal.

other open challenge, whereby various IT capacities (CPU, memory, storage) of a fog node can be presented in form of few virtualized computing unities. The same analogy applies with the sensors forming part of a fog node, and the network connecting the different fog node devices. We argue that all resources managed by a fog node should be abstracted, not only the IT resources but also the sensing and network resources.

In sum, we believe that a fog node can be defined along the following lines:

Fog nodes are distributed fog computing entities enabling the deployment of fog services, and formed by at least one or more physical devices with processing and sensing capabilities (e.g., computer, mobile phone, smart edge device, car, temperature sensors, etc.) All physical devices of a fog node are connected by different network technologies (wired and wireless) and aggregated and abstracted to be viewed as one single logical entity, that is the fog node, able to seamlessly execute distributed services, as it were on a single device.

Whether this is a lasting definition, our goal is to post the question of what a fog node is in the context of a holistic, combined fog and cloud computing ecosystem, where the notion of a fog node is used to serve and present to higher layers an abstracted and virtualized view of the underlying fog resources and the networks connecting them. This is illustrated in Fig. 5. Fig. 5a depicts the physical devices and the physical network forming the fog node. Fig. 5b shows a potential abstraction of the physical resources, in form of Virtual Machines (VMs), Virtual Sensors (VSs) and possible Virtual Networks, as seen by higher layers in the fog-to-cloud architecture, setting all together a preliminary approach to a candidate Fog node architecture, including a FAN (Fog Area Network) controller, as well as two modules, the IT abstraction and the Wireless Sensor Network (WSN) controller. Fig. 5a also illustrates the fact that fog node devices can be physically connected between them, using different network technologies such as 3 G/4 G, LTE, Ethernet, WiFi, Bluetooth, etc., while the FAN controller would take over the network virtualization (virtual networks VN1 and VN2). We believe that this abstraction and the consequent integration with the cloud will not only ease the fog computing deployment, but also fundamentally change the cloud systems as we know them today, towards a more distributed and more decentralized operation, with all the qualities of the today's data center-based service provisioning.

5.1. Running smart city services

After introducing the logical concept for an open fog node definition, we will show how the organization of resources in form of an abstracted fog node can help the development of different services in a smart city. Thus, let us consider a smart city deploying a specific portfolio of services, including an environmental service, a traffic control service and an ehealth service, leveraging some IoT infrastructure deployed in the city. For the sake of illustration, let us map the city infrastructure into a single fog node topology as shown in Fig. 5a, including a set of heterogeneous devices, a PC, one mobile phone, a cluster of cars and two sensors connected to two Raspberry boards. More concretely, static sensor 1 and sensor 2 are respectively a CO (or other contaminant gas) sensor and an atmospheric pressure sensor, and the mobile phone and the cluster of cars represent the resources volatility characteristic inherent to devices mobility. This physical infrastructure is responsible for executing the different services deployed in the city. According to our logical concept, we envision different virtual entities to be created, each putting together the set of resources required to properly run a specific city service. Fig. 5b, shows two virtual entities, virtual network 1 and virtual network 2, that for example, may be used to deploy the ehealth and environmental services respectively. The third virtual entity – not included in Fig. 5b to facilitate overall reading –, may be a different combination of physical resources, as required by the traffic control service.

Focusing, for example, on the environmental service, we may see that virtual network 2 includes two virtual sensors – corresponding to the two physical sensors –, and different VMs devoted to run such a service (computing function, statistics, filtering and two of them for averaging). The two VMs used for computing average values could be (some policy must be defined) implemented in the two boards connected to the sensors. We consider the two boards as mini-computers, such as Raspberries. In this context, the fog node should be able to manage all type of resources, either non-virtualized or virtualized with any technology. In this specific case, probably the most suitable virtualization strategy in mini-computers such as Raspberries would be containers, and thus the VMs shown in Fig. 5b running the averaging process would be containers. Afterwards, the average values produced by both Raspberries would be filtered in a VM located at, for example, the mobile phone. The results after the filtering process would be sent to a VM in the PC, where a specific environmental function is computed based on the received values of CO and atmospheric pressure. The PC will trigger a warning to the smart city manager

responsible for taking the suitable actions. This is only an example of possible configuration, but other variants are also possible.

It is worth highlighting that the configuration of the virtual resources will be tuned depending on the specific services requirements. For example, in the previous example the filtering function has been allocated in a VM at the mobile phone, which is arguable due to its mobility. Another possibility would be to allocate both VMs, running the filtering and computing function processes, in the PC, setting a kind of trade-off between the service requirements and the mobility.

Simultaneously to the environmental service execution in virtual network 2, virtual network 1 may be used to deploy the ehealth service. We may now consider that the two virtual sensors, also linked to the two physical sensors, collect data that is processed at a virtual instance in virtual network 1 (e.g., cluster of cars) and forwarded to the PC, that will trigger a warning to either the city manager – responsible for taking the appropriate actions –, or directly to citizens with respiratory problems. Also setting a new different virtual entity (network, VMs and VSs), the traffic control service can be executed, running a well-defined set of different functions intended to smartly controlling the traffic in a specific city area, depending for example on the level of pollution measured by the sensors. The output of this service would be a warning forwarded to cars drivers advising about the permission or prohibition of circulation. We may also consider that other more complex services can run in parallel – as long as there are resources enough according to some policy in place –, including for example computationally complex services, such as those requiring a significant amount of resources – e.g., augmented reality/virtual reality –, utilizing different fog nodes as a grid.

Thus, the main benefit of the proposed logical concept for a fog node is the abstraction of resources. This means that, a particular set of physical resources, may be configured to simultaneously run, in parallel distinct services, using the same resources, but virtualized, thus enabling resources sharing but isolated execution. We may observe that a coordinated management of the abstraction strategy deployed in the open fog node definition, may undoubtedly contribute to:

- Guarantee services are suitably executed utilizing the required amount of resources.
- Optimize resources in order to maximize the amount of services to execute.
- Isolated services execution on the same resources.
- Easy scalability.
- Guarantee the required abstraction to manage a huge set of highly heterogeneous devices.
- Develop novel collaborative models based on resources sharing.
- Better management of resources volatility. A reduction on the available capacity on a physical resource does not necessarily mean to reduce the resources allocated to all services in execution.

Certainly, the proposed abstraction model is also bringing some challenges, we think may unquestionably pave the way to new research avenues and opportunities, as emphasized in Section 6.

6. Open issues and challenges on conceptualizing a fog node

In this paper, we have made an attempt to conceptualize a fog node, emphasizing the need for and rationale behind this quest. This section analyses open issues and challenges driven by that fog node concept, including existing references to learn from. Since the context of previous sections was to distinguish between capabilities of edge devices –“dumb”, “smart” and “truly smart” –, we structure this section also aligned to this classification. To this end, we put special focus to “smart” and “truly smart” edge devices,

while also outlining the challenges related to “dumb” edge devices (sensors), paying particular attention on how edge devices can be abstracted (i.e., virtualized), aggregated, and how to handle their mobility and information uncertainty. We finish the section with a discussion on Quality of Service, as well as on security and privacy aspects.

Before going deeper into the discussion, it is important to take a moment to recognize that the scenarios envisioned are putting together a large set of highly heterogeneous resources, which creates a fundamentally complex system to be managed in a coordinated fashion. For instance, it is well known that basic computing units in the clouds are usually virtual machines. Hence, cloud management can be reduced to the management of a set of virtual machines. Moving to the edge, fog computing includes both mini-clouds as well as edge devices. Hence, in order to coordinate management between the clouds and the fogs, and assuming virtualization is the strategy of choice, the compute capability of edge devices also needs to be virtualized (i.e., abstracted) and aggregated. Finally, while we have emphasized the management of heterogeneous edge devices in a fog node as a great challenge, we have not yet tackled the network management issues in the context of fog computing, which further increases the complexity of the overall system analysis.

6.1. Edge device virtualization

Today, there is a plethora of heterogeneous edge devices and systems with rather different characteristics and capabilities, such as sensors, actuators, wearables, embedded systems, mobile phones or cars. Let us focus here on individual edge devices that have computing capabilities – CPU and memory –, hence devices with enough capacity to run some lines of code setting a service, an application, part of an application, or a function. In its basic capability, the edge device includes the hardware of the CPU and memory units, its corresponding operating system (OS), and the network interface, as illustrated in Fig. 6. Fig. 6a shows the basic configuration for an edge device, including network interface, hardware and the dedicated operating system. Building on this basic system, edge device resources can be further virtualized, to optimize and extend its performance and applicability, further illustrated in Fig. 6b, c and d. Indeed, Fig. 6b shows an edge device whose resources are virtualized over its own OS, such as it is the case of VirtualBox [43]. A different approach is depicted in Fig. 6c where virtualization is handled using a hypervisor such as VMWare ESX-Server [44]. Fig. 6d illustrates virtualization on the operating system level, creating Linux Containers (LXC) [45] by means of a known container based system Docker [46].

Each one of the options in Fig. 6 has its advantages and drawbacks. That said, we argue that, to fully integrate fog nodes with clouds, all these scenarios need to be supported, or else, dynamic features like resource sharing cannot be implemented in an open fashion. At the same time, opens the question of the best configuration for each specific application scenario. For instance, what would be the proper configuration for a low-cost commercial board device (e.g., Arduino or Raspberry) connected to a sensor? Is the selected configuration to stay for long and what is the power consumption? If the configuration is only for short time, what is the policy triggering a configuration change? Should we guarantee all applications to use the same configuration for an edge device? etc. By reviewing the existing literature, Bonomi et al. in [31] suggested edge devices to be configured as either virtualized (in VMs) or offered as bare metal. However, other contributions, see [47–49], use the containers to run applications in fog nodes – considered in these works as mini-clouds at the edge of the network –, due to their reduced memory capacity, computing footprint, and small size. Aligned to the latter concepts, if the mini-cloud component

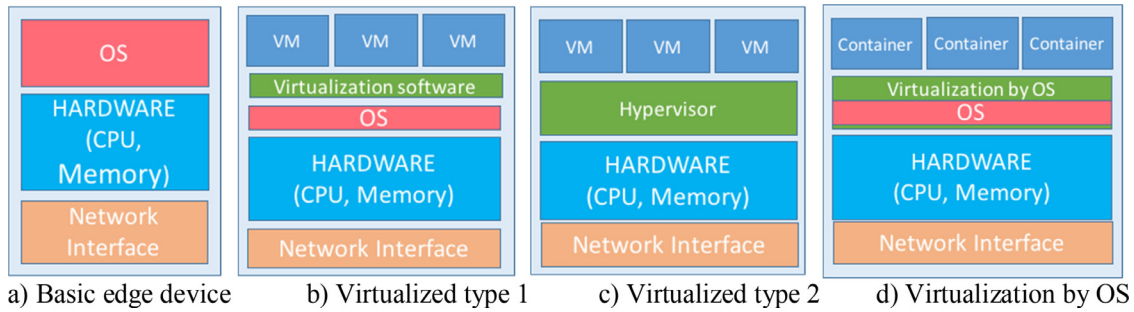


Fig. 6. Possible edge devices configuration.

Table 4
Mini-clouds and edge devices virtualization.

Fog node virtualization not specified or bare metal	[1,2,4,27,29] ^a [28] ^b [32,35,29]
Fog node providing virtualized resources	[31,27,29,9,34]
Fog node offering virtualized resources as Containers	[47,48,49]
Edge device as bare metal	[31,15,17] ^c
Edge device virtualized as VMs	[13,14,31,39,36,40] ^c

^a Works in [27,29] argue the possibility of dealing with both type of resources, virtualized and no virtualized.

^b In [28] sensors are virtualized however there is not information about how the computer capacity of Smart Gateway acting as fog it is represented.

^c These proposals are not specifically in the fog area, but include vehicular clouds (VC), grid computing, mobile grid computing, etc.

in the fog node is virtualized by means of containers, shall we assume that edge devices should be virtualized in the same fashion? Shall we use the same virtualization strategy for all edge devices? Indeed, edge devices capacities are much modest than a mini-cloud, hence new algorithms, methods and policies are needed to set the proper virtualization strategies for edge devices.

In sum, further research is needed to define and set policies for the best virtualization configuration when dealing with various edge devices. This is, in our understanding, one of the main challenges to be addressed when conceptualizing a fog node. With the aim of bringing some light to the challenge of edge-device virtualization, Table 4 lists some current contributions classified depending on: i) whether the fog nodes' IT capabilities (i.e., mini-clouds) are considered, and; ii) whether the IT edge devices are virtualized. References of first, second and third rows in Table 4 deal with fog nodes without including edge devices. Notice that contributions listed in the first row – *Fog-nodes virtualization not specified or bare metal* –, also include references that do not specify the way fog nodes offer their resources. Furthermore, the overall set of works reviewed here also includes contributions from other research areas – vehicular clouds, grids, mobile grids, etc. We can conclude based on all work reviewed, that a fog node will be shown as a logical entity, with its computing, processing and networking capabilities virtualized, open to include any type of edge device, either virtualized or not.

6.2. Abstraction and aggregation of edge devices

Recognized the need for presenting the available IT resources of a fog node as a set of available virtualized resources, such as virtual machines (VMs) or containers, we state that the whole set of virtualized resources in a fog node must encompass: i) the virtualization of the mini-cloud hardware resources, and; ii) the virtualization of the edge devices. The approach to represent a fog node as a virtual concept was earlier illustrated in Fig. 3b, where we showed that a fog node can include different virtual machines that are jointly managed. Some of these resources can be hardware resources of the mini-cloud and some other brought by an abstraction layer – represented in Fig. 3b by “Abstraction” of the

edge devices –, all creating a joint topology that is hardware, software and technology agnostic. In other words, we showed that the abstraction and aggregation are the key features.

The challenges in implementing the abstraction and aggregation layer, include some of the salient features that a fog node would need to include, such as:

- **Uniform representation of edge devices:** From the fog node point of view, the heterogeneous edge devices need the same representation – in terms of characteristics, features, parameters –, which is critical to facilitating the overall management of fog computing. To that end, every edge device would run a client management software – yet to be defined –, including the required functionalities to keep such a uniform representation (as earlier illustrated in Fig. 4).
- **Aggregation of resources:** The abstraction layer may virtualize multiple edge devices together as “aggregated”. This means that abstraction is used after an aggregation process, carried out for example by clustering the resources. For instance, a gridding software (see [50] or [51]) deployed at the abstraction layer can aggregate and present the available resources at the edge devices as if they were from a single device. This is illustrated in Fig. 7a that depicts edge devices aggregation by a grid, also including an emulation software layer (optional). Emulation may be necessary in some cases when running an application for a type of hardware different from that provided by the edge device. For example, an x86 application is run over edge devices with ARM hardware. A different option may be to install a distributed operating system (DOS) [52,53] running in the edge devices (see Fig. 7b). Fig. 7a and b depicts a zoom-out view for the abstraction, both showing the advantages of aggregating the edge device computer resources with a grid or a DOS. Another example of aggregation is shown in Fig. 7c, similar to vehicular clouds (VC), where different vehicles and their IT capacities are aggregated forming a cloud, handled by specific cloud software like OpenStack [54], OpenNebula [55], etc., taking into account which cloud software would be the more suitable for architectures usually utilized in edge devices, such as ARM.

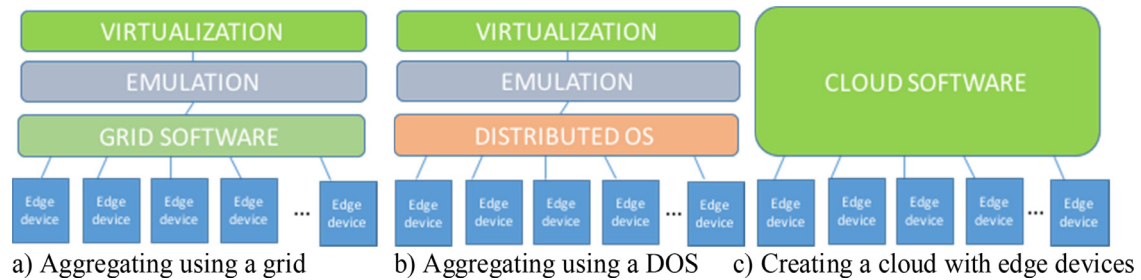


Fig. 7. Aggregation of edge device resources.

- *Resource selection, or flexible resource aggregation:* The entire set of edge devices can be aggregated to appear as a single resource to the F2C management system. More dynamic aggregation assumes however that only a subset and a more flexible configuration of aggregated resources is possible. In other words, the grid/cloud can be set by a different number of real physical resources tailored to the specific request. This flexibility can not only be more resource efficient but also contribute to a better energy management. As an example, a Raspberry Pi 2 with 4 cores Cortex A7 has power consumption between 3.5–4 W [56], whereas an i7 consumes at least 45 W.
- *Edge devices mobility:* Some edge devices, such as a mobile phone or a car, can be on the move. In this scenario, a strategy based on volunteering was proposed in [41], where edge devices join the fog node voluntarily, leaving the system when they are not available. The reasoning behind this idea is twofold: edge devices are on the move and hence may leave the area of influence of the fog node; or edge devices are offline, out of battery, switched off, etc. (next subsection is devoted to address the specific aspects related to mobility).

In sum, the fog node concept necessitates a uniform or standardized view of the available resources, in order to coordinate with the cloud computing management systems. The optimal aggregation strategy will depend on the type of resources and scenarios addressed, which is subject of future research.

6.3. Mobility and inaccuracy

Mobility is inherent to the edge devices. In fog nodes, this makes the above discussed abstraction and aggregation even more challenging. Moreover, a mobile edge device can be connected to a fog node through a high variety of networks, be it Bluetooth, WiFi, 3G/4G/5G. Regardless of the network technology, the fog node connectivity is generally limited by its geographical coverage area. This also means that the amount of resources brought in by edge devices linked to a fog node is not static. By looking back at Fig. 3b, we may undoubtedly assess that the amount of VMs physically corresponding to edge devices will be dynamically changing.

Different proposals have addressed the mobility problem in the context of collaboration and gridding. When adopting volunteer computing (see [40,41]), nodes voluntarily join the grid. To that end, an application installed on an edge device manages the process of joining/leaving “the grid.” Often, there is a context to this decision. For example, one needs to consider the device’s CPU idle cycle, or the battery life time. In [16], various solutions for mobile grid computing were considered, such as the Quality of service, scheduling and resource management, security, fault tolerance, etc. One of the interesting ideas to deal with scheduling and resource management was presented in [17], where the mobility pattern of the resources was analyzed to estimate the resource availability, classifying resources into full available, partial available, and unavailable.

The mobility problem and the intermittent connectivity have also been addressed in the context of Mobile Device Clouds (MDC), where task offloading mechanisms strongly depend on the type of connection between the different mobile devices and their mobility pattern. Some of the offloading mechanisms already proposed are supported by the Device-to-Device communication (D2D) paradigm. D2D was initially proposed to skip the usual constraint for two mobile devices to communicate through the Base Station (BS), by allowing them to directly communicate. Asadi and Wang [57] surveys D2D communications in cellular networks. Regarding the mobility pattern, [58] proposes a proactive mechanism predicting the mobility of mobile devices. Based on the location of the mobile devices the proactive mechanism predicts the time two mobile devices could be connected. This mobility prediction is an additional input for the offloading algorithm to decide where to offload a task.

The cloudlet concept proposed in [11] is, to many, a synonym to the fog computing idea. In [59], a study on the impact of user mobility on cloudlet computing performance was presented, and the relationship between the user mobility patterns, the probability a device accesses a particular cloudlet and the probability of successful tasks execution was investigated. The work concludes that the user mobility affects not only cloudlet access probability but also the cloudlet computing performance. The work in [60] proposes an offloading architecture, including different heterogeneous devices, e.g., Mobile Device Clouds (MDC), cloudlets, mobile cloudlets and clouds. The estimation of the resource availability due to mobility is computed based on the history of its performance. The main outcome includes an estimation algorithm responsible for predicting the disruption factor between the device and the cloudlet as well as an estimator for the duration of the connectivity of each mobile device. Furthermore, mobile devices have an application to activate and deactivate the collaboration mode, indicating whether they openly can offer its computational power.

Specifically close to the fog computing area, the work in [61] analyses the edge device mobility from a different perspective. Fog nodes are considered static mini-clouds located at the edge of the network, whereas the edge devices are continuously moving. The most interesting contribution of this work is that based on the user movement, an event traffic application starts being processed at some fog nodes before the mobile user reaches the location predicted. Only live event processing begins at the moment the user reaches one of the fog nodes. In order to address the problem of a potential inaccurate prediction of the future location of a mobile user, authors propose to start the processing in parallel, at several locations. The location to be finally selected will be the closest to the real position when the mobile user arrives.

In the area of VANET clouds the mobility has also been addressed to a large extent and for space reasons, we mention here several references only, closest to the area we address. In [36], a specific VM migration strategy for vehicular networks was proposed, whereby different vehicles form a cloud and the mobility

of one of them causes the disruption of the connection with the other vehicles. Therefore, guest VMs in this vehicle need to be migrated to either one or more of the rest of the vehicles forming the vehicular cloud, or to the roadside unit, (RSU, which are fixed stations located on the road side), or to the central cloud, depending on the resource availability. Moreover, methods are proposed to reserve part of the resources of the mobile devices to allocate migrated VMs in order to reduce the dropping rate of cloud services. A similar work can be found in [62], where authors propose to analytically model the arrival and departure of vehicles in the *Vehicular Cloud* following a Poisson distribution.

In sum, consideration of edge devices mobility is essential to conceptualizing a fog node. Although solutions have already been proposed, the area is wide open for research. Mobility also introduces challenges in the resources abstraction process. Indeed, in a resource discovery process, a fog node would advertise its available capacities, including the available capacity of the underlying edge devices. In a highly dynamic scenario, however, this information may change rather frequently. It is worth noticing that the inaccuracy of the fog node management information is definitely linked to the time dimension, and holds for a specific time period. Hence, a policy to define when a fog node must perform the aggregation of resources is also required. Similar to [17], smarter policies should be investigated and applied in mobile fog computing scenarios to analyze the same.

6.4. Network abstraction

The fog node, including the mini-cloud located at the edge of the network as well as the edge devices in the area of the fog node, must be correctly managed to optimize resources utilization and services performance. The management architecture of a fog node should include challenging strategies for resource discovery, resource allocation and edge devices management. We argue that all these strategies and policies must be handled together by what we refer to as *the management plane*, from cloud to the edge, (setting the foundations for a management plane for the envisioned F2C architecture).

In the F2C scenario envisioned (recall that we pose this scenario as the widest one, including the whole set of resources from the edge up to the cloud, so theoretically positioning the most endeavoring context), past sections first analyzed and later emphasized, the need for abstraction and aggregation of the IT capabilities of the edge devices as part of the fog node resource discovery. So far, however, we still have not considered the network. Recognizing the existence of the network, as the “glue” for edge devices, how can the network be abstracted and aggregated to be jointly managed with compute and storage resources? Just as there is a wide heterogeneity of edge devices, there is a wide heterogeneity in network technologies as well, including WiFi, LTE, 3 G/4 G, Bluetooth, or more recently, LoRaWan [63], and SigFox [64].

Past works, like [9,33,65,66] proposed the fog area network to be managed by means of Software Defined Networks (SDNs) or/and Network Functions Virtualization (NFV). These proposals are aligned to the current trend of softwarization of Telecommunications [67]. In these scenarios, the fog node includes an SDN-like controller handling the programmability of the network of edge devices under the fog node control. The communication between the different fog nodes and between the fog nodes and cloud can be handled through traditional routing mechanisms – e.g., OSPF – following either a fully distributed or a centralized management using an SDN approach as proposed in [66], where the whole network from cloud to fog is managed by SDN.

In a different set of scenarios, several contributions propose to manage the vehicular network, VANET, by means of SDN. Authors in [68] propose to centralize the VANET network intelligence in

the Road Side Unit (RSU). Then, vehicles only have to forward data packets either to other vehicles or to the RSU, based on the decisions made by the SDN controller in the RSU. The RSU is also taking the control of the overall data dissemination. The work in [69] proposes the Fog-SDN (FSDN) VANET architecture, where the fog network management is shared between the SDN controller, the RSU and the base station, all physically located at different devices. The SDN-controller sends abstract policy rules, but the final decision is taken by the RSU or the base station based on the local knowledge of its networks and resources.

Finally, the use of SDN in Wireless Sensor Networks (WSN) is worth mentioning. In [70] SD-WSN (Software Defined-WSN) is proposed, enabling the separation between the data plane, formed by sensors forwarding data, and the control plane, formed by one or more controllers centralizing the network functions, such as routing or QoS. The underlying idea is to make the sensor network customizable by programming, which is well-aligned with the previously discussed F2C management plane objectives.

In line with a widely recognized trend in networking referred to as network softwarization, fog area networks need also to be softwarized, whereby a physical network can also be configured in terms of different virtual networks, of which the functions can run in either the fog node or the cloud. This is a highly endeavoring characteristic in an IoT scenario, mainly built by putting together a lot of small and heterogeneous devices forming the network. For example, while an IoT application may require a network formed by all existing sound sensors, another application may only require temperature sensors, hence claiming for a different sensor network. Towards this vision, a number of challenges need to be addressed, including the strategies for a right placement of virtual network functions.

6.5. Quality of service

Sections 6.1–6.3 highlight different challenges related to the abstraction of resources setting a fog node, thus including aspects related to IT, sensor and network resources. This subsection deals with QoS while the next one (see Section 6.6) faces security and privacy issues.

There is with no doubt that QoS is yet a challenge in cloud computing, and even though fog computing is able to address some critical aspects of QoS, such as network latency, QoS also remains a challenge in fog computing. In fact, many characteristics inherent to fog computing even exacerbates the QoS issue in fog computing (driven by the effects of handling a distributed, dynamic, volatile and heterogeneous set of fog devices). For example, due to mobility or limited battery lifetime, it is difficult to guarantee that a resource, once discovered, can guarantee its presence for the duration of the service lifetime. Another issue relevant to the QoS provisioning is the multi-provider environment – which in fog computing is as possible as we know it in cloud computing – where the restrictions in the amount of data to be disclosed among providers, undoubtedly affects the overall QoS. Also the clustering capacity required to support the envisioned collaborative model opens many foreseen and yet unforeseen issues. The economic factors also play a role, such as whether the cellular network operator would participate in a multi-provider fog service, or the user, as owner of the end device. These and similar question require further studies.

6.6. Security and privacy

Security and privacy is also a key challenge yet requiring substantial research efforts in fog computing [8]. According to Vaquero and Roderio-Merino [9], fog computing will inherit the security concerns coming from current virtualized environments, such as

cloud computing, but augmented with the fact that fog computing is executed at the edge of the network, in a highly heterogeneous set of devices. This unquestionably makes some of the security solutions proposed for cloud computing not suitable for fog computing. One of the main security unsolved issues in fog computing is authentication at the different levels. For example, a gateway serving as fog node may be compromised or replaced by a fake one (e.g., man-in-the-middle attack). On the other hand, the fact that fog computing shifts some computational capabilities to the edge devices, drives the edge of the network to handle private, sensitive or confidential information, so highlighting issues related to privacy and trust. Thus, secure communications must be granted in order to guarantee data privacy at the edge of the network, as well as some kind of isolation mechanism when running applications (or service, or part of a service) in fog nodes.

7. Conclusions

Although early work in fog computing would have been sufficient to define a fog node as a highly virtualized platform, details were missing about the role of edge devices, as well as whether the fog nodes are to be general purpose, or defined in the context of specific applications, such as eHealth, industrial environment, Smart Cities, etc. The state of the art research, most of the time, identifies a fog node as a mini-cloud, located at the edge of the network, and close to the IoT devices connected to it. Instead, we argue that the coming IoT services will undoubtedly demand strict constraints in service performance (latency, security, QoS, etc.), thus requiring an efficient management of the whole set of available resources, be it from cloud, fog or a combination of both (as set on the F2C scenario). Leveraging such a complex scenario, we conceptualize a fog node as a logical entity, putting together different heterogeneous resources (the mini-cloud as well as the edge devices), to be managed by abstraction and aggregation policies.

In this paper we focus on core functionalities of a fog node as well as in the accompanying opportunities and challenges towards their practical realization in the near future. We first survey the state of the art in technologies for fog computing, paying special attention to the contributions that analyze the role edge devices play in conceptualizing a fog node. We then summarize and compare the concepts, lessons learned from their implementation, and show how a conceptual framework is emerging towards a unifying fog node definition. After that, we present for the first time a logical view (Fig. 4) and an architectural approach (Fig. 5) about what a fog node may be. In short, we propose to develop an abstraction strategy so as the fog node may show its devices (mini-clouds, “dumb”, “smart” and “truly” smart edge devices) as a homogeneous set of logical resources. Finally, we discuss about open issues and challenges arising when the fog node must present an abstracted and virtualized view of its physical resources (i.e., computing, sensing and networking) to higher layers in a fog-based hierarchical scenario.

Acknowledgments

This work was partially supported for UPC authors by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund under Contract TEC2015-66220-R (MINECO/FEDER).

References

- [1] A. Rahmani, N. Thanigaivelan, T. Gia, J. Granados, B. Negash, P. Liljeberg, H. Tenhunen, Smart e-health gateway: bringing intelligence to internet-of-things based ubiquitous healthcare systems, in: Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas USA, 2015.
- [2] B. Tang, Z. Chen, G. Heffernan, T. Wei, H. He, Q. Yang, A hierarchical distributed fog computing architecture for big data analysis in smart cities, in: Proceedings of the ASE BigData & Social Informatics, Kaohsiung, Taiwan, 2015.
- [3] F. Bonomi, The Smart and Connected Vehicle and the Internet of Things, 2013, http://tf.nist.gov/seminars/WSTS/PDFs/1-0_Cisco_FBonomi_ConnectedVehicles.pdf
- [4] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kirikas, R. Sudhaakar, Components of fog computing in an industrial internet of things context, in: Proceedings of 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON Workshops), Seattle, USA, 2015.
- [5] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 2012.
- [6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.
- [7] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: Proceedings of the 2015 Workshop on Mobile Big Data, ACM, 2015, pp. 37–42.
- [8] I. Stojmenovic, Fog computing: a cloud to the ground support for smart things and machine-to-machine networks, in: Proceedings of Telecommunication Networks and Applications Conference (ATNAC), Australasian, IEEE, 2014, pp. 117–122.
- [9] L.M. Vaquero, L. Roderio-Merino, Finding your way in the fog: towards a comprehensive definition of fog computing, ACM SIGCOMM Comput. Commun. Rev. Arch. 44 (5) (2014) 27–32 Newsletter ACM 2014.
- [10] S. Yi, Z. Hao, Z. Quin, Q. Li, Fog computing: platform and applications, in: Proceedings of the 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), IEEE, 2015, pp. 73–78.
- [11] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Comput. 8 (4) (2009) 1536–1268.
- [12] European Telecommunications Standards Institute, ETSI at <http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing>
- [13] S. Bitam, A. Mellouk, ITS-cloud: cloud computing for intelligent transportation system, in: Proceedings of IEEE Global Communications Conference (GLOBECOM), Anaheim, USA, 2012.
- [14] S. Bitam, A. Mellouk, S. Zeadally, VANET-CLOUD: a generic cloud computing model for vehicular ad hoc networks, IEEE Wirel. Commun. 22 (1) (2015) 96–102.
- [15] F. Büsching, S. Schildt, L. Wolf, DroidCluster: towards smartphone cluster computing – the streets are paved with potential computer clusters, in: Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Macau, China, 2012.
- [16] A. Bichhawar, R.C. Joshi, A survey on issues in mobile grid computing, Int. J. Recent Trends Eng. Technol. 4 (2) (2010).
- [17] J. Lee, S. Song, J. Gil, K. Chung, T. Suh, H. Yu, Balanced scheduling algorithm considering availability in mobile grid, Advances in Grid and Pervasive Computing Volume 5529 of the series Lecture Notes in Computer Science, Springer, 2009, pp. 211–222.
- [18] N. Fernando, S.W. Loke, W. Rahayu, Mobile Cloud Computing: A Survey, Future Generation Computer Systems, 29, Elsevier, 2013.
- [19] Y. Jararweha, L. Tawalbeh, F. Ababneha, A. Khreishah, F. Dosarib, Scalable cloudlet-based mobile computing model, Procedia Comput. Sci. 34 (2014) 434–441.
- [20] T. Nishio, R. Shinkuma, T. Takahashi, N.B. Mandayam, Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud, in: Proceedings of the First International Workshop on Mobile Cloud Computing & Networking, MobileCloud '13, July–August, 2013.
- [21] A. Mtiaba, A. Fahim, K.A. Harras, M.H. Ammar, Towards resource sharing in mobile device clouds: power balancing across mobile devices, in: Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, MCC'13, Hong Kong, 2013.
- [22] C. Shi, V. Lakafosis, M.H. Ammar, E.W. Zegura, Serendipity: enabling remote computing among intermittently connected mobile devices, in: Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '12, South California, USA, 2012.
- [23] G. Peng, CDN: Content Distribution Network, Technical Report TR-125 of Experimental Computer Systems Lab, Stony Brook University.
- [24] CISCO, “Cisco fog computing solutions: Unleash the power of the internet of things (whitepaper),” 2015. [Online] http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-solutions.pdf
- [25] Roberto Mora. Cisco iox: Making fog real for iot, blogs@cisco-cisco blogs, June 2015. Online <http://blogs.cisco.com/iox/cisco-iox>
- [26] X. Masip-Bruin, E. Marín-Tordera, A. Jukan, G. Ren, G. Tashakor, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud (F2C) computing systems, IEEE Wirel. Commun. Mag. 23 (5) (2016) 120–128.
- [27] M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: Proceedings of the 2014 International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 2014.
- [28] T.N. Gia, M. Jiang, A.M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Fog computing in healthcare internet of things: a case study on ECG feature extraction, computer and information technology, in: Proceedings of the 2015 IEEE International Conference on Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK, 2015.

- [29] M. Aazam, E.N. Huh, Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT, in: Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications (AINA), Gwangju, Korea, 2015.
- [30] I. Abdullahi, S. Arif, S. Hassan, Ubiquitous shift with information centric network caching using fog computing, in: Computational Intelligence in Information Systems, 331, Springer, 2014, pp. 327–335.
- [31] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, in: Fog Computing: A Platform for Internet of Things and Analytics, Big Data and Internet of Things: A Roadmap for Smart Environments, 546, Springer, 2014, pp. 169–186.
- [32] K. Skala, D. Davidovic, E. Afghani, Z. Sojat, Scalable distributed computing hierarchy: cloud, fog and dew computing, *Open J. Cloud Comput.* 2 (1) (2015) 16–24.
- [33] Lingjun Pu, Xu Chen, Jingdong Xu, Xiaoming Fu, D2D fogging: an energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration, *IEEE Journal on Selected Areas in Communications* 34 (12) (2016) 3887–3901.
- [34] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, B. Koldehofe, Mobile fog: a programming model for large-scale applications on the internet of things, in: Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, MCC'13, Hong Kong, 2013.
- [35] A. Mukherjee, H.S. Paul, S. Dey, A. Banerjee, ANGELS for distributed analytics in IoT, in: Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 2014.
- [36] R. Yu, Y. Zhang, S. Gjessing, W. Xia, K. Yang, Toward cloud-based vehicular networks with efficient resource management, *IEEE Netw.* 27 (5) (2013) 48–55.
- [37] K. Senthil, M. Tech, Performance Analysis of multi-cloud Deployment in Many task Applications, In International Journal of Engineering Research and Technology 1 (5) (2012) ESRSA Publications.
- [38] M. Silverstein, Building an online domain-specific computing service over non-dedicated grid and cloud resources: the superlink-online experience, in: Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Newport Beach, USA, May 2011.
- [39] F.J. Seinstra, J. Maassen, R.V. van Nieuwpoort, N. Drost, T. van Kessel, B. van Werkhoven, J. Urbani, C. Jacobs, T. Kielmann, H.E. Bal, Jungle computing: Distributed, supercomputing beyond clusters, grids, and clouds, in: M. Cafaro, G. Aloisio (Eds.), Grids, Clouds and Virtualization, Computer Communications and Networks, Springer London, 2011, pp. 167–197.
- [40] V.D. Cunsolo, S. Distefano, A. Puliafito, M. Scarpa, Volunteer computing and desktop cloud: the Cloud@Home paradigm, in: Proceedings of the Eighth IEEE International Symposium on Network Computing and Applications (NCA 2009), Cambridge, USA, 2009.
- [41] D.P. Anderson, BOINC: A System for Public-Resource Computing and Storage, in: 5th IEEE/ACM International Workshop on Grid Computing, Pittsburgh, PA, 2004, pp. 365–372.
- [42] S. Distefano, G. Merlino, A. Puliafito, A. Vecchio, A hypervisor for infrastructure-enabled sensing clouds, in: Proceedings of the 2013 IEEE International Conference on Communications Workshops (ICC), Budapest, Hungary, 2013.
- [43] VirtualBox Commercial product Online at <https://www.virtualbox.org/>
- [44] VMware ESX-Server Commercial product Online at <https://www.vmware.com/products/vsphere-hypervisor>
- [45] Linux Containers Commercial product Online at <https://linuxcontainers.org/>
- [46] Docker Commercial product Online at <https://www.docker.com/>
- [47] D. Willis, A. Dasgupta, S. Banerjee, ParaDrop: a multi-tenant platform to dynamically install third party services on wireless gateways, in: Proceedings of the 9th ACM Workshop on Mobility in the Evolving Internet Architecture, MobiArch '14, Maui, Hawaii, 2014.
- [48] B.I. Ismail, E. Mostajeran Goortani, M.B. Ab Karim, W. Ming Tat, S. Setapa, J.Y. Luke, O. Hong Hoe, Evaluation of docker as edge computing platform, in: Proceedings of the 2015 IEEE Conference on Open Systems (ICOS), Melaka, Malaysia, 2015.
- [49] A Blog of the ZHAW Zurich University of Applied Sciences, October 2015, Online at <https://blog.zhaw.ch/icclab/making-fogcomputing-real-research-challenges-in-integrating-localized-computing-nodes-into-the-cloud/>
- [50] HTCondor research product Online at <https://research.cs.wisc.edu/htcondor/>
- [51] Globus grid software Commercial product Online at http://toolkit.globus.org/grid_software/
- [52] Plan 9 from Bell Labs Commercial product Online at http://doc.cat-v.org/plan_9/
- [53] The Inferno Operating System/Virtual Machine Commercial product Online at <http://doc.cat-v.org/inferno/>
- [54] OpenStack Commercial product Online at <https://www.openstack.org/>
- [55] OpenNebula Commercial product Online at <http://opennebula.org/>
- [56] Raspberry Pi 2 Commercial product Online at <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- [57] Arash Asadi, Qing Wang, Vincenzo Mancuso, A survey on device-to-device communication in cellular networks, *IEEE Commun. Surv. Tutor.* 16 (4) (2014) 1801–1819.
- [58] Bo Li, Zhi Liu, Yijian Pei, Hao Wu, Mobility Prediction Based Opportunistic Computational Offloading for Mobile Device Cloud, in: Proceedings of 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE), Chengdu, China, December 2014.
- [59] Y. Li, W. Wang, The unheralded power of cloudlet computing in the vicinity of mobile devices, in: Proceedings of Wireless Networking Symposium, Globecom, Atlanta, USA, 2013.
- [60] A. Mtibaa, K.A. Harras, K. Habak, M. Ammar, E.W. Zegura, Towards mobile opportunistic computing, in: Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing (CLOUD), New York, USA, 2015.
- [61] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, B. Koldehofe, Opportunistic spatio-temporal event processing for mobile situation awareness, in: Proceedings of the 7th ACM International Conference on Distributed Event-based Systems, DEBS '13, Arlington, Texas, USA, 2013.
- [62] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, X. Shen, An SMDP-based resource allocation in vehicular cloud computing systems, *IEEE Trans. Ind. Electron.* 62 (12) (2015) 7920–7928.
- [63] Commercial product LORA Online at <https://www.lora-alliance.org/For-Developers/LoRaWANDevelopers>
- [64] Commercial product SigFox Online at <http://www.sigfox.com/>
- [65] W.S. Chin, H. Kim, Y.J. Heo, J.W. Jang, A Context-based Future Network Infrastructure for IoT Services, *Procedia Computer Science* 56 (1) (December 2015) 266–270.
- [66] A view of fog computing from networking perspective, *Procedia Comput. Sci.* 56 (2015) 266–270 July 2015.
- [67] A. Manzalini, Softwareization: RD & Innovation status of activities and next steps, ITU-T Focus Group on IMT-2020 Pre-meeting workshop on network softwareization Turin, Italy, September, 2015, Online at <http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/Workshop-Turin/manzalini-slides.pdf>
- [68] K. Liu, J.K.Y. Ng, V.C.S. Lee, S.H. Son, I. Stojmenovic, Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a Software defined network, *IEEE/ACM Transactions on Networking* 24 (3) (2016) 1759–1773.
- [69] N.B. Truong, G.M. Lee, Y. Ghamri-Doudane, Software defined networking-based vehicular adhoc network with fog computing, in: Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, Canada, 2015.
- [70] T. Luo, H.P. Tan, T.Q.S. Quek, Sensor openflow: enabling software-defined wireless sensor networks, *IEEE Commun. Lett.* 16 (11) (2012) 1896–1899.