



# Managing resources continuity from the edge to the cloud: Architecture and performance

Xavi Masip-Bruin<sup>a,\*</sup>, Eva Marin-Tordera<sup>a</sup>, Admela Jukan<sup>b</sup>, Guang-Jie Ren<sup>c</sup>

<sup>a</sup> Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

<sup>b</sup> Technische Universität Carolo-Wilhelmina zu Braunschweig, Germany

<sup>c</sup> IBM, Almaden Research Center, United States

## HIGHLIGHTS

- Layered and hierarchical approach to manage IoT devices.
- A solution for efficiently combining fog and cloud resources.
- A management architecture for resource continuity provisioning.
- Resources abstraction is crucial in the proposed management architecture.
- Lower databases stress in a smart city scenario as an initial performance analysis.

## ARTICLE INFO

### Article history:

Received 19 February 2017

Received in revised form 28 July 2017

Accepted 15 September 2017

Available online 3 October 2017

### Keywords:

Cloud computing

Fog computing

Edge computing

Resource management

Offloading

## ABSTRACT

The wide spread deployment of smart edge devices and applications that require real-time data processing, have with no doubt created the need to extend the reach of cloud computing to the edge, recently also referred to as Fog or Edge Computing. Fog computing implements the idea of *extending the cloud where the "things" are*, or in other words, improving application performance and resource efficiency by removing the need to processing all the information in the cloud, thus also reducing bandwidth consumption in the network. Fog computing is designed to complement cloud computing, paving the way for a novel, enriched architecture that can benefit from and include both edge(fog) and cloud resources. From a resources perspective, this combined scenario requires resource continuity when executing a service, whereby the assumption is that the selection of resources for service execution remains independent of their physical location. This new resources model, i.e., resource continuity, has gained recently significant attention, as it carries potential to seamlessly providing a computing infrastructure from the edge to the cloud, with an improved performance and resource efficiency. In this paper, we study the main architectural features of the managed resource continuity, proposing the foundation of a coordinated management plane responsible for resource continuity provisioning. We study an illustrative example on the performance benefits in relationship to the size of databases with regard to the proposed architectural model.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

There is hardly any technology report today, be it from academic or business sectors, without a reference to cloud computing. It is also widely accepted that cloud computing and its products, have been instrumental in facilitating a wider deployment of the so-called Internet of Things (IoT) services. First, as the technology is evolving to grant not just every user but also every object to

become *cloud-enabled*, cloud computing and cloud based presence of people and objects will continue to flourish. Second, the widely spread deployment of smart edge devices along with a huge number of new applications and 5G mobile network concepts, have with no doubt created the need to extend the reach of the traditional cloud computing towards the edge. These phenomena are unquestionably described by several recent studies. For example, predictions from [1] point out that 20.8 billions of edge devices are to be in use worldwide by 2020, collecting more than 1.6 zettabytes (1.6 trillion GB) of data, and IDC [2] estimates that “by 2019, 45% of IoT-created data will be stored, processed, analyzed, and acted upon close to, or at the edge of, the network”.

\* Corresponding author.

E-mail addresses: [xmasip@ac.upc.edu](mailto:xmasip@ac.upc.edu) (X. Masip-Bruin), [eva@ac.upc.edu](mailto:eva@ac.upc.edu) (E. Marin-Tordera), [a.jukan@tu-bs.de](mailto:a.jukan@tu-bs.de) (A. Jukan), [gren@us.ibm.com](mailto:gren@us.ibm.com) (G. Ren).

Aligned to this evolution, fog computing [3–5] has been recently proposed as a cloud computing concept implemented closer to the edge devices. Fog computing implements the idea of *extending the cloud where the “things” are*, or in other words, improving application performance and resource efficiency by removing the need to processing all the information in the cloud, thus also reducing bandwidth consumption in the network. Fog computing has been technologically conceptualized to deliver the following three main benefits: (i) minimizing latency – whereby the data is analyzed closer to where it is collected; (ii) balancing networking traffic – by offloading gigabytes of network traffic from the core network connecting to the cloud, thus reducing the bandwidth consumption, and; (iii) helping support security and privacy through proximity – by keeping sensitive data inside the proximate computer and network system. Substantial efforts are currently underway in the area of fog computing, devoted to analyzing the potential benefits brought by fog computing when applied to different sectors. Just for illustrative purposes we can mention recent contributions [6,7] and [8] in ehealth; [9–11] and [12] in smart cities; [13–15] and [16] in energy efficiency; [17,18] and [19] in IoT; [20] in industrial environments; [21] in radio access networks; as well as [22,23] and [24] in vehicular networks. Indeed, this set of efforts have turned into a notable record of scientific publications or even industrial-led forums (such as the OpenFog Consortium) putting together the industrial and academic sectors to discuss on new ideas and opportunities.

Designed not to compete but to complement cloud computing, fog computing when combined with cloud computing, paves the way for a novel, enriched scenario, where services execution may benefit from resources continuity from the edge to the cloud. From a resources perspective, this combined scenario requires resource continuity when executing a service, whereby the assumption is that the selection of resources for service execution remains independent of their physical location. Table 1, extending the data cited in [25] – considering different computational layers from the edge up to the cloud, – shows how the different layers can allocate different devices as well as the corresponding relevant features of each of them, including application examples. As it can be seen, an appropriate resources categorization and selection is needed to help optimize service execution, while simultaneously alleviating combined problems of security, resource efficiency, network overloading, etc.

Although many contributions in the literature are equally referring to edge computing or fog computing, the OpenFog Consortium in [26], distinguishes between the two and highlights three main differences between both concepts. The first refers to the fact that fog is designed to work with the cloud, whereas edge is per se defined by the exclusion of cloud. The second describes fog as hierarchical, while edge is limited to a reduced number of layers. Finally, while edge computing only focuses on computation, fog also includes networking, storage, control and acceleration. Considering this distinction and the definitions, our paper focuses on fog computing.

In this paper we study the issue of resource continuity and coordinated management of fog and cloud computing, and propose the foundational blocks for the system architecture. Our proposal considers a layered architecture, including all the systems in the resource chain from the edge to the cloud, all functions and components responsible for resource continuity provisioning, as well as novel service execution strategies. In the proposed approach, the resources are allocated to different computational layers depending on the actual resource capacities, in terms of computing, storage, and network. We also study an illustrative example on the performance benefits in relationship to the size of databases, with regard to the proposed architectural models presented and propose directions for further studies.

The functional description of the proposed architecture is designed to be open, agnostic to any specific scenario or context, thus extending its potential applicability. However, specific requirements demanded by real world deployments will be considered in the design of the algorithms, strategies and policies to be used for the different target scenarios, to adapt services needs, resources capacities and users demands. Although the proposed architecture is already split into three main components, each one covering these three sets of needs, the functional blocks to be designed must consider key aspects of fog but also cloud computing, such as devices heterogeneity, mobility, low control, efficiency, virtualization, security&privacy or new business models.

The rest of this paper is organized as follows. Section 2 presents the related work. The fog-to-cloud layered architecture is introduced in Section 3. In Section 4, we analyze an illustrative smart city scenario and finally we conclude the paper in Section 5.

## 2. Related work

The combined fog and cloud resources model is today attracting remarkable attention, driving to broad discussions highlighting open issues, research challenges, novel services opportunities and incipient architectural models. A layered view has been already identified in some contributions in different domains. Earlier work in [27] proposed a layered model for vehicular cloud clustering elements into three different layers, central (conventional) cloud, roadside cloud and vehicular cloud. The work in [28] proposed a hierarchical 4-layer fog computing architecture for big data analysis, specifically developed for smart cities. The OpenFog Consortium states in [26] that “Unfettered cloud-only architectural approaches cannot sustain the projected data velocity and volume requirements of the IoT. To sustain IoT momentum, the OpenFog Consortium is defining an architecture to address infrastructure and connectivity challenges by emphasizing information processing and intelligence at the logical edge. This approach is called fog computing. While the cloud itself may play a vital role in many deployments, fog computing represents a shift from traditional closed systems and a reliance on cloud-only focused models. Fog computing is complementary to, and an extension of, traditional cloud-based models”.

At the same time, organizing the whole set of resources from the edge to the cloud to guarantee resource continuity, will require an appropriate resources management strategy, that must be provided by a set of control and management functions, in form of an adequate control and management plane. To this end, the OpenFog Consortium explicitly states in [26], referring to its envisioned architecture, i.e., “The OpenFog architecture must implement elements of management, including network measurement, control and configuration, at or near the endpoint rather than being controlled primarily by gateways such as those in the LTE Core”. Also ETSI, working on delivering standards for Multi-Access Edge Computing (MEC) [29], describes the technical requirements and a reference architecture for a mobile edge platform, endowed with the “essential functionality required to run mobile edge applications on a particular virtualization infrastructure”.

Relevant to the need to manage the resource continuity within a fog-to-cloud management framework, Greco et al. in [30] proposed a two-layer architecture consisting in two interworking planes, namely the control and management planes of the cloud, in a multi-cloud scenario. The Open Cloud Computing Interface (OCCI), proposed a protocol and API for management tasks, focusing on cloud management systems [31]. In the specific area of fog computing, most of the current research efforts focus on application scenarios, as already introduced in the illustrative references included in Section 1. However, none of these proposals focus on a coordinated resource management, from the edge up to the cloud.

**Table 1**

Resource continuity possibilities in the layered architecture according to [25].

|          |                             | Resource continuity from edge to cloud |                                |                                    |                               |
|----------|-----------------------------|--|--------------------------------|------------------------------------|-------------------------------|
|          |                             | Fog                                    |                                |                                    | Cloud                         |
|          |                             | Edge devices                           | Basic/aggregation nodes        | Intermediate nodes                 | Cloud                         |
| Device   |                             | Sensor, actuator, wearables            | Car, phone, computer           | Smart building, cluster of devices | Datacenter                    |
| Features | Response time               | Milliseconds                           | Subseconds, seconds            | Seconds, minutes                   | Minutes, days, weeks          |
|          | Application examples        | M2M communication haptics              | Dependable services (e-health) | Visualizations simple analytics    | Big data analytics statistics |
|          | How long IoT data is stored | Transient                              | Minutes, hours                 | Days, weeks                        | Months, years                 |
|          | Geographic coverage         | Device                                 | Connected devices              | Area, cluster                      | Global                        |

Aligned to the need to provide a coordinated resources continuity management, the contribution in [32], proposes a new Fog-to-Cloud (F2C) layered model, demonstrating the benefits brought by both considering the proposed layered resources architecture – in terms of service execution time, – and enabling novel parallel service execution strategies—leveraging resource sharing and collaborative models. Aimed at a similar objective, the OpenFog Consortium has recently released their proposal for an OpenFog Reference Architecture (OpenFog RA) [26]. Although both approaches are conceptually quite similar and are intended to a similar goal, there are two main differences that, at this preliminary stage in the architectural definition, are worth mentioning. First, F2C is explicitly envisioned as a collaborative model, where users are expected to play as resources consumers but also as resources providers through deploying collaborative strategies, with a strong impact on new business models and opportunities. Second, while F2C is envisioned to manage the whole resources continuum from the edge up to the cloud, OpenFog RA is mainly proposed to manage the fog scenario, by defining different layers (Tiers) interoperating with the edge (i.e., endpoints/things), as read in [26] “It (the OpenFog RA) is a medium to high-level view of system architectures for fog nodes and networks”. Indeed, although interoperability with cloud is considered in the OpenFog RA, the current version of the architecture does not clearly describe how this will be deployed. In fact, the still high level description of both approaches makes it difficult to making a comprehensive and detailed comparison. Finally, it is also worth mentioning that the deployment of the F2C management plane concept is supported by a recently funded EU project (mF2C [33]).

Last but not least, we need to highlight that security and privacy provisioning is envisioned as a big challenge in the proposed continuum resources scenario, mainly due to both the inherent edge devices characteristics, such as heterogeneity, mobility and low control, and the – yet to be defined – business relationships in a multi-owned scenario, all together exacerbating the traditional security issues usually considered in cloud scenarios. Devices heterogeneity makes difficult to converge on a single security solution. Mobility fuels devices volatility and system dynamics, what also drives hard constraints on security. Finally, it is worth mentioning that the level of control on devices at the edge is usually lower than the one supposed for static systems or cloud premises, thus increasing systems vulnerabilities. A comprehensive list of threats and potential attacks to a fog infrastructure is provided by [26], along with a high-level view of a tentative OpenFog Node Security Architecture. Authors in [34], and lately extended in [35], propose for the first time a security based architecture for the F2C layered model, through the deployment of different cluster heads responsible for providing secure communications by handling security as a global system function. The paper also defines the protocol strategy proposed to handle communications between the different components in the architecture, and ends up showing preliminary results highlighting the benefits of the proposed architecture.

In this paper we extend the work in [32] to propose and analyze in more detail a distributed management framework tailored to efficiently manage resource continuity within a generic layered architecture combining the cloud and edge resources.

### 3. Management architecture for resource continuity

Recognizing the benefits of resource continuity when jointly managing cloud and fog systems and devices, this section outlines the main architectural concepts for resource management. It focuses on two main contributions: (i) the hierarchical and distributed management architecture with a layered approach and; (ii) the main functional blocks within the architecture. For illustration and easier understanding, we exemplify the concepts proposed in a smart city scenario.

#### 3.1. Layered, hierarchical and distributed management architecture

We present the envisioned coordinated fog and cloud scenario in the layered architecture shown in Fig. 1, where resources are allocated to different layers depending on their capacities and features, as earlier presented in Table 1. Fig. 1 illustrates a particular case considering 4 layers, with traditional cloud as the top layer and edge devices at the bottom. In this approach, all fog devices need to be mapped into three layers, which we refer to as edge, basic/aggregation and smart, similar to Table 1. As shown in Fig. 1, the higher the layer the higher the capacities, the control, with implications on the comparably lower number of devices and, perhaps, higher security, and a lower privacy (e.g., in cloud).

In regard to the resource continuity concept, the main idea is to abstract the layered resources when allocating resources. We note that the physical resources can be first categorized, similar to Table 1. The categorization has to goal to advertise the resources from the edge to the cloud, as an abstracted entity distributed within one unique layer. As we can see in Fig. 1, the physical view consisting of 4 layers (FLO, FL1, FL2 and CL) is mapped into a single logical resources view. Individual resources are abstracted to be coherently managed by a Resource Management function, part of the envisioned control and management plane, responsible for handling devices mobility, monitoring and resources categorization.

The resources abstraction can be implemented through virtualization. What is important here is to provide a coherent view of the whole set of available resources, from the edge to the cloud. For instance, if virtual machines are used to manage datacenters at cloud, virtual machines should be also used to abstract edge devices, otherwise the management strategy must support different abstraction strategies and some sort of adaptation, such as between containers and virtual machines. For any architecture to deliver on its promise on resource continuity, the following requirements must be met: (i) coherent view of the whole set of resources to optimize service requirements, (ii) unique control and

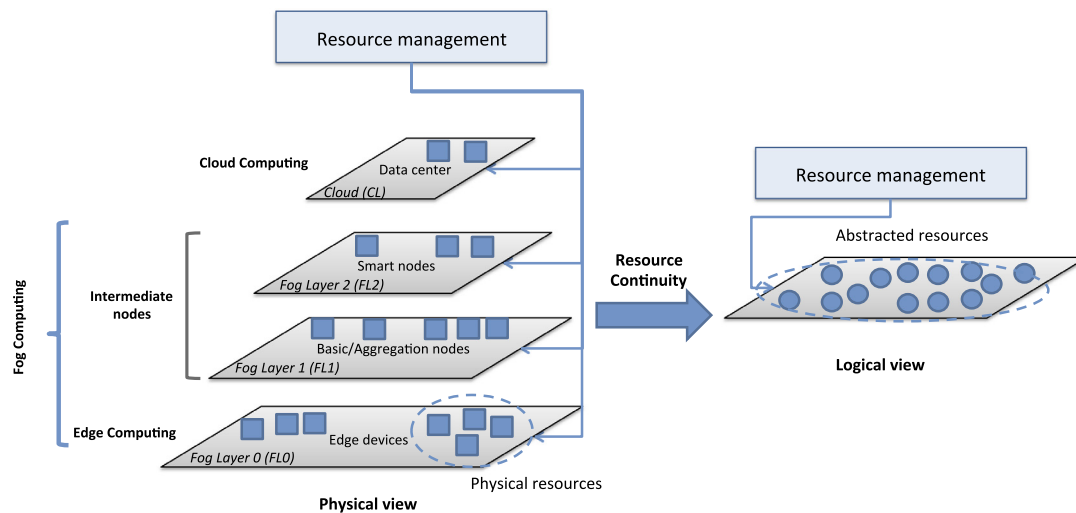


Fig. 1. The layered architecture and the abstractions relevant to resource continuity.

management plane managing all resources, independently of their location or ownership, (iii) resources categorization and classification to efficiently tailor resource selection to service demands, (iv) resource selection depending on resources capacities, and (v) efficient usage of resources.

We now present an example of a specific implementation of the proposed architecture, on an example of the so-called smart city. Fig. 2 illustrates an example of a total of three layers – two layers are defined as Fog Layers (Layers 1 and 2) and one layer as a Cloud Layer (Layer 3), – aggregating the existent devices and other computing, networking and storage resources. In this example, simple actuators and sensors can be in the lowest layers, while “smarter” devices, such as user phones can build layers above. We assume devices are grouped into fog domains (hereinafter referred to as “Fogs”, see Fig. 2), according to a certain policy, depending for example on real connectivity, proximity, capacity or some business goals. There are 3 different clouds in Layer 3, with no particular restrictions on whether they are public, private or hybrid clouds, – interoperability between clouds is considered an innate feature. There are 2 different Fogs (fog domains) in Layer 2 (Fog 1.2 and Fog 2.2) and 4 in Layer 1 (Fog 1.1, Fog 2.1, Fog 3.1 and Fog 4.1).

We here propose a distributed management architecture that includes two main management artifacts, namely *Areas* and *Control Elements*. An *Area* consists of a group of Fogs located at different layers, facilitating vertical resources coordination among different layers. It is open to research, and out of the scope of this paper, to go into the details of *Areas* configuration, their definition or the interaction mechanisms between them. *Control Elements*, on the other hand are responsible for providing the different management tasks required in each corresponding domain. Two different *Control Elements* are proposed, Dew Point (DP) and Control Area Unit (CAU). A Dew Point is proposed as the control element responsible for handling all control and management functionalities for a specific Fog. We can see in Fig. 2 that all fogs include an individual Dew Point, implemented through different devices (e.g., a traffic light in Fog 1.1, a bus stop in Fog 4.1 or a base station in Fog 1.2). Each Dew Point embeds control functionalities as required to handle the specific functions included in the said Dew Point—these functions are all currently in the domain of open research and architecture challenges. A Control Area Unit (CAU), on the other hand, is responsible for the management of a specific Area and its location presents an interesting future research challenge. Fig. 2 shows two CAUs, responsible for managing the two identified Areas (A and B). CAUs communicate towards different fronts. First,

a CAU is connected to all Dew Points in its area, thus enabling vertical coordination among resources with different capacities and hence located at different layers. Second, different CAUs are connected (shown as inter-area control communication) enlarging the set of available resources through distinct CAUs and thus enabling horizontal coordination. Finally, CAUs are also connected to the cloud (shown as Control Communication), to include the cloud resources in the whole management, hence enabling the deployment of the whole resource continuity concept.

Having set the main components for the proposed management architecture, we now propose the basic set of functional blocks that would be required to develop the proper management functionalities.

### 3.2. Functional block design

Fig. 3 illustrates the proposed set of control and management blocks in the architecture designed previously, providing a coordinated and distributed management solution aiming at handling resources continuity from the edge to the cloud. The figure shows three main blocks. The first and also the largest block allocates the different functionalities we expect to co-exist, i.e., in reference to the user, service and resource-specific functions. The second block allocates the set of functions used for instantiations of the different modules, referring to the different control components and edge devices, i.e., the Dew Points (DP) and the Control Area Units (CAU). The last one, Brokering, is designed to be applied in a typical scenario of multi-ownership infrastructure, where different fog/cloud layers may belong to different parties, and a brokering concept is necessary for their joint deployment.

The users domain in the first block in Fig. 3 includes different functions that contextualize how users interact (i.e., interface) with the resources and the services to be executed. Every user needs to follow the security and privacy rules (Authentication, Authorization and Encryption) when requesting a service. The resource continuity in the F2C layered architecture strongly depends on the user and context information (Profiling), to tailor service execution to the user demands. Novel strategies are expected to include the different new roles users may play in a layered architecture, for example, including fog providers capacity to share resources (Sharing). Similarly, users data must meet privacy concerns (Privacy). All corresponding functions need to be in accordance with the business policies (SLA and other to be defined) in place to guarantee a real-world deployment. For example, a user willing



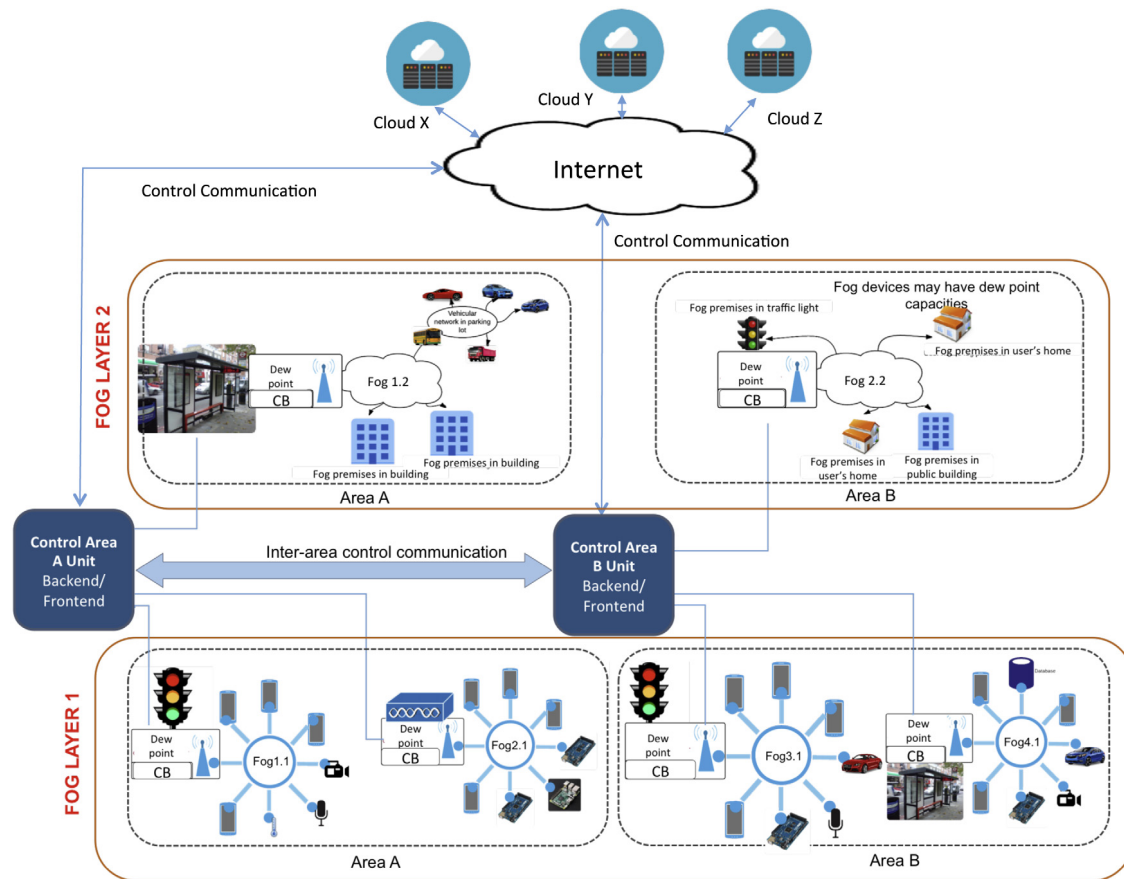


Fig. 2. Distributed management architecture for the layered model in a F2C ecosystem.

to share (“to grid”) his/her car computing resources must know what the obtained benefits would be, whereby sharing would be implemented with authenticated and valid users only.

On the services domain, once the user’s request is granted, the service is categorized (Categorization) according to a dynamic service taxonomy, whereby a service can be either pre-configured, or created on-the-fly. When requested, the service can be decomposed (Decomposition) into sub-services (also known as atomic services), commonly requiring fewer resources and facilitating new approaches, such as parallel execution, to achieve better performance. The set of sub-services may be either already stored in a database (Repository), or newly defined by the requested service. Challenging issues include finding out the appropriate place to locate the service decomposition, while taking into consideration the performance, such as computing load and service execution time. An interesting open issue is to define to what extent the service functions must be associated with the Dew Point; defining the graph rules that include not only the sub-services but also their relationships and dependences; defining strategies for searching for available resources giving preference to Layer-Area or Area-Layer; defining strategies for searching missing sub-services in case of unavailability in an Area/Layer (depending on the strategy above), or; defining what may be done when the sub-services found do not guarantee the expected service constraints, etc. Finally security and privacy guarantees also deserve scientific attention to work on strategies for anomaly detection or privacy management.

On the resources domain, an accurate knowledge about the available resources for each device and an accurate resource representation – including important aspects of resources, such as being virtual/physical, static/mobile, shared/exclusive, etc., – must be guaranteed (Monitoring and Discovery). This information, once

categorized (Categorization) and classified (Classification) is stored into a database (Distributed Repository), whereby the methods to update policies, representation rules, etc., is an open issue here. This Repository also includes per-layer resource capacities (be it computing, storage, etc.) and network Connectivity features (such as topology information). Resources are assigned to the service to be executed, depending on the service demands, runtime policies required (e.g., parallel or sequential execution, what unquestionably impacts the way computer, network and storage resources are selected) and resource availability (all managed through the functions Provisioning, Selection, Routing, Allocation, Runtime model as well as Scheduling and Sharing policies). It is worth noting that resource representation is a fundamentally complex endeavor, mainly due to the dynamics inherent to the envisioned layered architecture, the heterogeneity of the edge devices, as well as the business relationship in a multi-ownership scenario. Indeed, we envision resource continuity in the layered model as an opportunity resources framework, where services and users benefit from open resources to optimize overall performance.

Nevertheless, as already pointed out in Section 2 when reviewing the state of the art, the specific envisioned characteristics about heterogeneity, dynamics and business relationships, will undoubtedly have a enormous influence and impact on the security and privacy guarantees (e.g., secure data processing, anomaly detection or privacy management) in the F2C architecture. Hence, further research efforts will be required in the coming future to handle the challenges imposed by the foreseen F2C scenario, aimed at proposing a comprehensive solution for security and privacy provisioning.

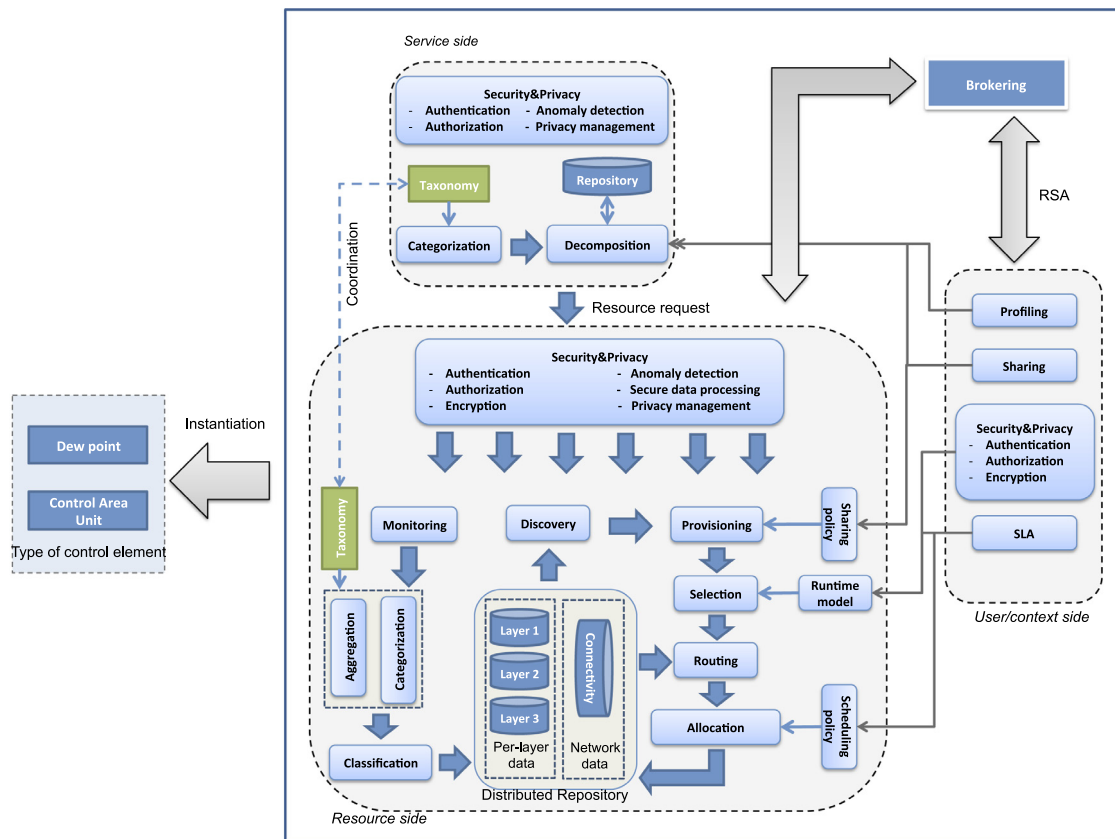


Fig. 3. Distributed management architecture for the layered model in a F2C ecosystem.

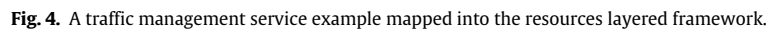
#### 4. Performance study

This section is intended to present illustrative benefits obtained when deploying the proposed management architecture on a specific applicability area (smart cities). We illustrate the features discussed previously on a traffic management service deployed on a smart city. We note that the results presented are not significant on any performance benchmarking or quantifiable benefits, but serve the purpose of revealing the potential performance benefits that can be achieved by the resource continuity concept in the proposed layered architecture and are in fact meant to encourage further research in this area. With that objective and based on the concepts presented so far, we now illustrate the operation of a potential traffic management service deployed in a hypothetical smart city and smart transportation context, incorporating the proposed coordinated and distributed management framework. We start describing the resource layered architecture, illustrating the different components considered in the chosen service. Then, we tailor the topology settings to three different real-world cities (Los Angeles, Terrassa and Barcelona) and finally we show preliminary results for database size and database lookup time.

Fig. 4 illustrates the example. It considers three fog layers, whereby each layer may include different fogs. Fog layers are defined according to the corresponding resource capacity, whereby the higher layer in the hierarchy implies that more resources are available in that layer, and that all layers may serve the purpose of monitoring traffic conditions. For instance, fogs in Layer 1 can be formed of mobile edge devices; fogs in Layer 2 have more capacity (compute, storage, networking) than those in Layer 1, etc. An example of Layer 2 can be a computer network system installed in a bus stop or in a building. Similarly, fog resources in Layer 3 can be formed by the IT capacity of public buildings, such as libraries, universities or a city council. We assume the three

fog layers fall into a single Area, controlled by the Control Area Unit (CAU) deployed at Fog Layer 2. Fog Layer 1 includes three different fogs, one located in a car and two located in traffic lights. Fog Layer 2 consists in only the capacities installed in a bus stop. Finally, Fog Layer 3 illustrates the sharing (“gridding”) of resources among multiple cars in a parking lot. We also assume that layers publish their resources into a brokering module that implements global policies for multi owner/provider resource visibility. Each layer and fog system embeds their corresponding control functions that can be used to manage the infrastructure and configure the required coordination functions in order to orchestrate the services and allocate resources.

To analyze the traffic monitoring service, we find the approach from [36] most useful, based on creating a data matrix collecting information about all vehicles following a certain route between an origin and a destination in a city, obtained from distinct types of sensors. In [36], it is considered that collected data processing is performed at centralized city data-centers. The issue comes up when dealing with traffic management services only impacting on a single specific city area, thus with no need to shift the decision process to geographically distant data-centers. Unlike the work in [36], and in order to show the impact on the amount of data to be forwarded, we consider that all the information coming from fogs within a specific city area may be processed in the same area, thus with no need for forwarding data to centralized data-centers. As a consequence, a traffic data sub-matrix will be generated for each one of the city areas, containing enough information to facilitate the deployment of services requiring local decisions. Beyond the potential beneficial effects on latency, service execution time and bandwidth reduction, we next analyze the impact on database size, lookup time and traffic load motivated by the need to update the proposed databases.



- Fogs of Layer 1 are traffic sensor nodes (TSN) detecting number of cars and MAC addresses.
- Traffic sensor nodes (TSNs) in Layer 1 are 100 m (linear) distance each, in the city streets.
- In the layered approach, the baseline city area will have 10,000 m of linear meters of streets.
- We assume a car occupying 4 m, 1 m between car and car, and 2 lanes city streets (average).
- Services running in a city area only require data providing from the same city area.

we consider different city area sizes. First approach is to consider a city area as a district. For Barcelona city, for instance, we need to consider 10 areas, each with 1,300 TSNs (fogs of Layer 1). In a different approach, a bigger number of areas would be considered, for example, set to 130 areas, thus including approximately 100 TSNs each (fogs of Layer 1), see [Table 2](#). Therefore, the information stored and required to process services in this specific city would be  $48 \times 40 \times 1300$  bits (approx. 300 Kbytes) with 10 city areas, and  $48 \times 40 \times 100$  bits (approx. 23 Kbytes) with 130 city areas, to be updated every few seconds. Moreover, assuming a time access to the tables databases proportional to  $O(\log(N))$ ,  $N$  being the size of the database, [Table 2](#) shows the table sizes for both approaches as well as the lookup time normalized to the time to access to a table of 23 Kbytes for Barcelona, Terrassa (a medium size city, with 431,540 m of streets and 6 districts), and for Los Angeles (a large city, with 5,521,563 m of streets and 15 districts).

On the other hand, Table 2 also shows the impact on the traffic load motivated by the need to update the proposed databases. Only as an example, updating a cloud centralized database in the city of Los Angeles every 5 s, will represent 217 Gbytes of data every day being sent throughout the network, hence with a strong impact on energy consumption, network dimensioning and infrastructure optimization used, only to update the data base.

**Table 2**

City database information and databases updating traffic.

| Barcelona city                    |                                 | Layered management approach |                       |                |
|-----------------------------------|---------------------------------|-----------------------------|-----------------------|----------------|
| Average distance between TSNs (m) | Number of TSNs in an area       | Number of city areas        | Table size in an area | Lookup time    |
| 100                               | 1,300                           | 10 (districts)              | 300 kbytes            | 1.21           |
| 100                               | 100                             | 130                         | 23 Kbytes             | 1              |
|                                   |                                 | Cloud management approach   |                       |                |
| Average distance between TSNs (m) | Number of TSNs in the city      | Table size in all the city  |                       | Lookup time    |
| 100                               | 13,000                          | 3 Mbytes                    |                       | 1.40           |
| Terrassa city                     |                                 | Layered management approach |                       |                |
| Average distance between TSNs (m) | Number of TSNs in an area       | Number of city areas        | Table size in an area | Lookup time    |
| 100                               | 708                             | 6 (districts)               | 165 kbytes            | 1.17           |
| 100                               | 100                             | 43                          | 23 Kbytes             | 1              |
|                                   |                                 | Cloud management approach   |                       |                |
| Average distance between TSNs (m) | Number of fog nodes in the city | Table Size in all the city  |                       | Lookup time    |
| 100                               | 4,250                           | 1 Mbytes                    |                       | 1.31           |
| Los angeles city                  |                                 | Layered management approach |                       |                |
| Average distance between TSNs (m) | Number of TSNs in an area       | Number of city areas        | Table size in an area | Lookup time    |
| 100                               | 3,680                           | 15 (districts)              | 860 kbytes            | 1.29           |
| 100                               | 100                             | 550                         | 23 Kbytes             | 1              |
|                                   |                                 | Cloud management approach   |                       |                |
| Average distance between TSNs (m) | Number of TSNs in the city      | Table size in all the city  |                       | Lookup time    |
| 100                               | 55,000                          | 12 Mbytes                   |                       | 1.52           |
| Databases updating traffic        |                                 |                             |                       |                |
| Management approach               | City                            | 1 second                    | 5 seconds             | 10 seconds     |
| Cloud                             | Barcelona                       | 3 Mbytes/sec                | 0.6 Mbytes/sec        | 0.3 Mbytes/sec |
|                                   | Terrassa                        | 1 Mbytes/sec                | 0.2 Mbytes/sec        | 0.1 Mbytes/sec |
|                                   | Los angeles                     | 12 Mbytes/sec               | 2.4 Mbytes/sec        | 1.2 Mbytes/sec |
| Layered                           | Any of the cities               | 23 Kbytes/sec               | 4.6 Kbytes/sec        | 2.3 Kbytes/sec |

## 5. Conclusions

Fog and cloud computing, when put together, lay the foundation for creating a new, and highly heterogeneous computing and networking architecture, recently enticing high attention due to its potential in easing the development of innovative Internet-of-Things (IoT) services and novel resource sharing strategies, turning into incipient architectural models, such as the one from the Open-Fog Consortium or the recently proposed F2C model. The capability to coordinate and manage the entire set of resources from the edge up to the cloud, drives the resource continuity concept, responsible for mapping the whole set of physical resources into a simplified abstracted view, easing the overall coordination and efficiently improving the services execution. This paper proposes the main foundations for a management framework responsible for resource continuity provisioning, efficient resources selection, novel service execution strategies, and also points out several challenges fueling promising research lines. The proposed framework creates a layered model for the fog-to-cloud management architecture, and we illustrated a few key control components in a smart city scenario. A simple traffic control monitoring service was analyzed in terms of database related characteristics, aimed at showing incipient benefits that we believe motivate further research in this new area.

## Acknowledgments

This work was partially supported for the 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] D.C. Plummer, Top 10 strategic predictions for 2015 and beyond: Digital business is driving big change, Gartner Inc., <http://www.gartner.com/newsroom/id/3165317> (Online; accessed January-2017).
- [2] IDC futurescape: Worldwide Internet of Things 2016 predictions at <https://www.idc.com/research/viewtoc.jsp?containerId=259856> (Online; accessed January-2017).
- [3] F. Bonomi, et al., Fog Computing: A platform for Internet of Things and analytics, in: N. Bessis, C. Dobre (Eds.), in: Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence, vol. 546, Springer, 2014. <http://dx.doi.org/10.1007/978-3-319-05029-4.7>.
- [4] 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, MCC'12, ACM, 2012.
- [5] I. Stojmenovic, S. and Wen, The Fog computing paradigm: Scenarios and security issues, in: 2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014, Warsaw, Poland, 2014.
- [6] O. Fratu, C. Pena, R. Craciunescu, S. Halunga, Fog Computing System for Monitoring Mild Dementia and COPD Patients, TELKSIS 2015, Serbia, 2015.
- [7] A.M. Rahmani, et al., Exploiting smart e-Health gateways at the edge of health-care Internet-of-Things, a fog computing approach, Future Gener. Comput. Syst. (2017). <http://dx.doi.org/10.1016/j.future.2017.02.014>.
- [8] O. Ferrer, et al., Fog computing in health 40 the role of small data, Int. J. Embedded Syst. (2015).
- [9] M. Sapienza, E. Guardo, M. Cavallo, G. La Torre, G. Leombruno, O. Tomarchio, Solving critical events through mobile edge computing: An approach for smart cities, in: IEEE International Conference on Smart Computing, SMARTCOMP, MO, St. Louis, USA, 2016.
- [10] C. Perera, Y. Qin, J.C. Estrella, S. Reiff-Marganiec, A.V. Vasilakos, Fog computing for sustainable smart cities, arXiv:1703.07079v1 [cs.NI].
- [11] A. Sinaeipourfard, J. Garcia, X. Masip-Bruin, E. Marin-Tordera, A novel architecture for efficient fog to cloud data management in smart cities, in: 37th IEEE International Conference on Distributed Computing Systems, IEEE ICDCS, Atlanta, USA, June 2017.
- [12] I. Stojmenovic, Fog Computing: A cloud to the ground support for smart things and machine-to-machine networks, in: Australasian Telecommunication Networks and Applications Conference, Melbourne, Australia, November 2014.
- [13] F. Jalali, K. Hinton, R. Ayre, Fog computing may help to save energy in cloud computing, IEEE J. Sel. Areas Commun. 34 (5) (2016).
- [14] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, J. Stefa, Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: Review, challenges, and a case study, IEEE Netw. 30 (2) (2016).
- [15] M.A. Al Faruque, K. Vatanparvar, Energy management-as-a-service over fog computing platform, IEEE Internet Things J. 3 (2) (2016).



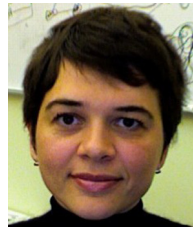
- [16] E. Baccarelli, P.G. Vinueza, M. Scarpiniti, Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study, *IEEE Access* 5 (2017).
- [17] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of Things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13–16, Helsinki, Finland, August 2012.
- [18] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: A platform for Internet of Things and analytics, in: *Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence Book Series, SCI*, vol. 546, 2014, pp. 169–186.
- [19] A.V. Dastjerdi, R. Buyya, Fog computing: Helping the Internet of Things realize its potential, *Computer* 49 (8) (2016).
- [20] G. Beauregard, Fog computing virtualizing industry, in: LocalGrid Technologies Inc., March 2015.
- [21] M. Peng, S. Yan, K. Zhang, C. Wang, Fog-computing-based radio access networks: Issues and challenges, *IEEE Netw.* 30 (4) (2016).
- [22] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: A viewpoint of vehicles as the infrastructures, *IEEE Trans. Veh. Technol.* 65 (6) (2016).
- [23] Y. Xiao, C. Zhu, Vehicular fog computing: Vision and challenges, in: *IEEE International Conference on Pervasive Computing and Communications Workshops PerCom Workshops*, Kona, HI, USA, 2017.
- [24] M. Sookhak, F. Richard Yu, Y. He, N. Kumar, Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing, *IEEE Veh. Technol. Mag.* (2017) 99.
- [25] Fog computing and Internet of Things: Extend the cloud to where the things are, in: OpenFog Consortium, 2015.
- [26] Open fog consortium working group, in: OpenFog Reference Architecture for Fog Computing White paper, Feb 2017.
- [27] R. Yu, Y. Zhang, S. Gjessing, W. Xia, K. Yang, Toward cloud-based vehicular networks with efficient resource management, *IEEE Netw.* 13 (2013) 48–55.
- [28] Tang, et al., A hierarchical distributed fog computing architecture for big data analysis in smart cities, in: *li Proceedings of the ASE BigData and Social Informatics, ACM*, 2015, p. 28.
- [29] ETSI work in MEC at <http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing> (Online; accessed May-2017).
- [30] S. Greco, A. Costanzo, Cloud control and management planes for service provisioning, in: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, December 2011.
- [31] Open cloud computing interface at <http://occi-wg.org> (Online; accessed January-2017).
- [32] X. Masip-Bruin, E. Marin-Tordera, G. Tashakor, A. Jukan, G.J. Ren, 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).
- [33] The mF2C project at <http://www.mf2c-project.eu>.
- [34] S. Kahvazadeh, V. Barbosa, X. Masip-Bruin, E. Marin-Tordera, J. Garcia, R. Diaz, Securing combined fog-to-cloud system through SDN approach, in: 4th Workshop on CrossCloud Infrastructures and Platforms, ACM Digital Library, Serbia, Belgrade, 2017.
- [35] S. Kahvazadeh, V. Barbosa, X. Masip-Bruin, E. Marin-Tordera, J. Garcia, R. Diaz, An SDN-based architecture for security provisioning in fog-to-cloud (F2C) computing systems, in: *Future Technologies Conference, FTC*, Vancouver, Canada, November 2017.
- [36] J.J. Fernández Lozano, Miguel Martín-Guzmán, Juan Martín-Vila, A. García-Cerezo, A wireless sensor network for urban traffic characterization and trend monitoring, *Sensors* 15 (10) (2015).



**Xavi Masip-Bruin**, M.Sc. and Ph.D. in Telecommunications Engineering from the Universitat Politècnica de Catalunya, (UPC) is an associate professor at the Computer Science Dpt. at UPC and serves as Director of the CRAAX lab. In 2011 and 2012 he was visitor professor at UPEC, Paris. His current research interests focus on cloud and fog computing, network management, IoT and particularly on analyzing the benefits brought by combining fog and cloud paradigms. He has been involved in many different research initiatives at national and international level. Xavi has been recognized with a 2016 IBM Faculty Award.



**Eva Marín-Tordera**, received her M.Sc. degree in Physics from Universitat Barcelona (UB) and her Ph.D. degree in Computer Science from UPC. She is with the CRAAX lab from 2007. Her research focuses on applying predictive strategies for resources management in combined fog-cloud computing solutions, particularly applied to smart transportation and ehealth sectors. Eva has participated in several national and EU projects, and has published around 100 papers in international conferences and highly impacting journals.



**Admela Jukan** is Chair Professor of Communication Networks at the Technische Universität Carolo-Wilhelmina zu Braunschweig (Brunswick) in Germany. Prior to that post, she was research and visiting faculty at the Institut National de la Recherche Scientifique (INRS), University of Illinois at Urbana Champaign (UIUC) and Georgia Tech (GaTech). In 1999 and 2000, she was a visiting scientist at Bell Labs, Holmdel, NJ, and in 2015 at MIT. From 2002–2004, she served as Program Director in Computer and Networks System Research at the National Science Foundation (NSF) in Arlington, VA. Admela Jukan received her Dr.tech. degree (cum laude) in Electrical and Computer Engineering from the Technische Universität Wien, the M.Sc. degree in Information Technologies from the Politecnico di Milano, Italy, and her Dipl. -Ing. degree from the Fakultät Elektrotechnik i Racunarstva (FER), in Zagreb, Croatia.



**Guang-Jie Ren** joined IBM Almaden Lab in August 2009 and is currently leading the Mobile Research Group. His work is currently focused on creating enterprise apps that address industry pain points with intuitive design, mobile technologies, advanced analytics, and user empowerment. His prior work includes business analytics, enterprise architecture, and business modeling. Trained as an industrial engineer with Bachelor and Master degrees from Tsinghua University in Beijing, he received his Ph.D. in engineering and management from Cambridge University.