mF2C: The Evolution of Cloud Computing Towards an Open and Coordinated Ecosystem of Fogs and Clouds

Xavi Masip-Bruin[1], Eva Marín-Tordera[1], Ana Juan Ferrer[2],
Antonio Salis[3](✉), John Kennedy[4], Jens Jensen[5], Admela Jukan[6],
Andrea Bartoli[7], Rosa M. Badia[8](✉), Matija Cankar[9],
and Marc Elian Bégin[10]

[1] Universitat Politècnica de Catalunya, CRAAX-UPC,
Vilanova i la Geltrù, Spain
{xmasip,eva}@ac.upc.edu

[2] ATOS Research and Innovation, Barcelona, Spain
ana.juanf@atos.net

[3] Engineering Sardegna, Cagliari, Italy
antonio.salis@eng.it

[4] Intel Research and Development, Leixlip, Ireland
john.m.kennedy@intel.com

[5] UK Research and Innovation/STFC RAL, Didcot, UK
jens.jensen@stfc.ac.uk

[6] Technische Universität Braunschweig, Brunswick, Germany
a.jukan@tu-bs.de

[7] Worldsensing Limited, Cambridge, UK
a.bartoli@worldsensing.com

[8] Barcelona Supercomputing Center, Barcelona, Spain
rosa.m.badia@bsc.es

[9] XLAB d.o.o., Lubjiana, Slovenia
matija.cankar@xlab.si

[10] SIXSQ, Geneve, Switzerland
meb@sixsq.com

Abstract. Fog computing brings cloud computing capabilities closer to the end-devices and users, while enabling location-dependent resource allocation, low latency services, and extending significantly the IoT services portfolio as well as market and business opportunities in the cloud and IoT sectors. With the number of devices growing exponentially globally, new cloud and fog models are expected to emerge, paving the way for shared, collaborative, extensible mobile, volatile and dynamic compute, storage and network infrastructure. When put together, cloud and fog computing create a new stack of resources, which we refer to as Fog-to-Cloud (F2C), creating the need for a new, open and coordinated management ecosystem. The EU Horizon 2020 program has recently funded a new research initiative (mF2C) bringing together relevant industry and academic players in the cloud arena, aimed at designing an open, secure, decentralized, multistakeholder management framework for F2C computing, including novel programming models, privacy and security, data storage techniques, service creation, brokerage solutions, SLA policies, and resource

U. Schwardmann et al. (Eds.): Euro-Par 2019 Workshops, LNCS 11997, pp. 136–147, 2020.
https://doi.org/10.1007/978-3-030-48340-1_11

orchestration methods. This paper introduces the main mF2C concepts, illustrates the need for a coordinated management ecosystem, proposes a preliminary design of its foundational building blocks and presents results that show the benefits mF2C may have on three key real-world scenarios.

Keywords: Cloud computing · Fog computing · Fog-to-cloud · Management · Distributed systems · Security · IoT

1 Introduction: The F2C Concept

The emergence of IoT –the networked connection of people, process, data and things – is expected to significantly increase the number of connected devices worldwide, from billions of units we have today, to tens of billions of units expected to be deployed in the coming years. Some predictions [1] suggest that 26 billion edge devices are to be connected by 2020, collecting more than 1.6 zettabytes (1.6 trillion GB) of data. According to Cisco reports, it is expected to have more than 50 billion devices connected by 2020, paving the way to fog computing [2]. At the same time, cloud service providers (Amazon AWS, Google Compute Engine, Microsoft Azure) today enable customers to quickly deploy a myriad of private and corporate services at comparably lower prices than buying and maintaining their own infrastructure. When combined, fog and cloud computing are without doubt setting standards in flexibility, cost, economy of scale, but also innovation in new services, devices and applications. Indeed, the computing and processing capacities offered by cloud computing can perfectly complement the comparably lower processing, storage and networking capacities of the edge devices building a novel, coordinated scenario between edge devices and the cloud.

In the combined scenario of cloud computing and a myriad of edge devices, one can observe that while data, users and decisions are at the edge side, processing capacities are primarily at the cloud side. As a result, today's systems need to address the challenges of overloading the network and inducing latency to transfer data from the edge to the cloud. Thus, the traditional approach of leveraging the centralized processing in the cloud premises may require a new thinking based on these two observations. First, the high latency values required to reach to the cloud in the centralized approach are not suitable for real time services. Second, forwarding data, stored and collected at the edge to the cloud to be processed, is non-optimal in terms of network resources allocation, and doubly so when results are to be returned to the device that sent them. This has set the stage for the evolution of fog computing, that can leverage a distributed approach based on bringing cloud capabilities closer to, or into, the edge devices, also referred to as mini-clouds, cloudlets, or small-scale clouds. Figure 1 illustrates the pyramid of today's fog and cloud ecosystem integrating the typically centralized cloud infrastructure, with various levels (or layers) of dispersed elements starting with smaller scale clouds, over to fog computing with various degrees of decision making and data processing capabilities [3].

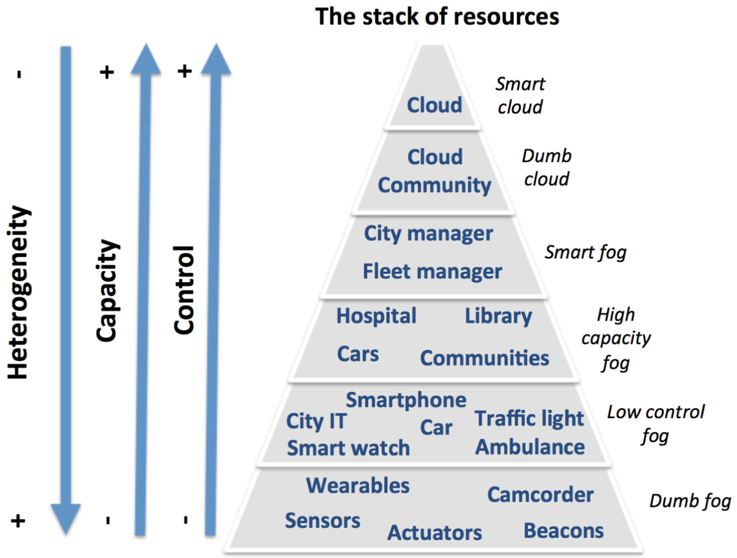

Fig. 1. Fog-to-cloud (F2C) layered structure: The stack of resources

In a combined Fog-to-Cloud (F2C) system, a critical question is how can a combined resource sharing and resource clustering approach efficiently extend the concept of a cloud provider to an unknown frontier, creating innovative resource-rich proximate infrastructures near to the user, while remaining profitable? To answer this question, we identified the need to provide a coordinated management of the combined F2C resources to ease and optimize the execution of existing and future services, through a myriad of new features including reduction of execution time, parallel execution, edge processing, fog security, locality, improved utilization of limited resources, improved energy efficiency ("green computing"), etc. To this end, a comprehensive control and management strategy is required, addressing efficient coordination and inter-operation of fog and clouds environments, as well as the innovative combined cloud/fog architecture.

This paper proposes a new research framework to achieve the same, which we refer to as mF2C focused at designing an open, secure, decentralized, multi-stakeholder management framework for F2C computing. An important feature of the system proposed is in its openness to integrating and supporting new functionalities and subsystems as they emerge, such as novel programming models, new privacy and security features, various data storage techniques, and brokerage solutions. This paper introduces the main idea behind the new mF2C concept, proposes a preliminary design of its foundational building blocks and presents results that show the benefits mF2C may have on three key real-world scenarios. This paper is structured as follows. Section 2 revisits the state of the art. Section 3 outlines main mF2C control and functionality, introducing the main architectural blocks as well as the main benefits expected from deploying mF2C in three real-world scenarios. Section 4 identifies main mF2C challenges and opportunities. Finally, Sect. 5 concludes the paper.

2 State of the Art

This section briefly revisits relevant contributions in four key F2C aspects (resource management, IoT management, programming models and security), emphasizing the need for designing innovative solutions to best match the computing demands of F2C.

2.1 Resource Management in Cloud and Fog Computing

Resource management in cloud computing has been subject to intense research with a myriad of important aspects, such as security, data privacy, data centers management, quality delivery, or energy consumption. Several cloud platforms are already available to manage cloud infrastructure, be it open source (CloudStack, Eucalyptus, OpenStack, and OpenNebula) or proprietary (Amazon EC2, Microsoft Azure, IBM Bluemix and Aneka). While there is no global consensus facilitating their seamless interaction in multi-cloud environment – no single, universal standard – standard APIs as well as libraries that abstract the cloud API have already been defined.

Recently significant efforts addressed "cloudification" of network functions under the umbrella of Network Function Virtualization Fig. 1. Fog-to-cloud (F2C) layered structure: The stack of resources (NFV), a software implementation of the network functions on "bare metal". However, their optimal placement and job scheduling especially in the cloud remains a hard problem, since network functions need to be managed in a dynamic fashion, and virtualized instances need to be able to migrate, grow, and shrink dynamically.

The combination of fog and cloud computing intensifies the resource management challenge. Several contributions exist aimed at managing how services are allocated into edge devices, or offloaded to execution, all based on meeting service level objectives, such as latency and VM setting capacity, see for instance [4]. However, fog computing as such is still in its infancy, lacking the standards and definitions of basic concepts. For instance, there is no a widely accepted definition for a fog node yet, mainly due to the diverse and heterogeneous set of edge devices. This diversity makes it very difficult to agree even on simple concepts, such as whether fog devices should be virtualized, and if so, whether the usage of the traditional VM concept, or containers is appropriate, etc. References can be found in the literature (see for example [5] and [6]) with divergent definitions of a fog node, defined to meet the needs of the specific application scenarios.

There are other contributions aimed at facilitating the management of IoT devices, ranging from pure data management to edge devices management. In the first area we can mention SENTILO [7] or IoT-LAB [8]. Both aim at easing the data collection from different IoT devices by putting all data together in a single repository for easy access. In device management, examples include the research projects FIWARE [9] and SOFIA [10], or in the commercial sector VORTEX [11]. Briefly, FIWARE consists of a catalogue of "enablers", i.e., enabling the development of applications and services in a cost-effective fashion. SOFIA's main goal is to ease systems interoperability aiming at promoting the development of new services and applications. The Vortex product contains different components to support different device data sharing configurations – Vortex Cloud for cloud data sharing, Vortex Fog for edge devices data sharing, etc.– aimed at data sharing and easing systems interoperability.

Overall, there is currently no coordination or integration strategy available which addresses the need for coordination among all cloud and fog resources.

2.2 Programming Models

Despite the plethora of programming models developed for the cloud (MapReduce, Aneka, Google app engine, etc.), applications to be executed in heterogeneous and distributed infrastructures – the ones considered in mF2C – cannot be supported directly. To the best of our knowledge, the only programming model that takes into account such an infrastructure is Mobile Fog [12]. However, the programming model proposed is very explicit with regard the infrastructure, and the availability of the system seems to be limited.

A particularly relevant programming framework for coordinated fog and cloud computing is COMPSs [13], a task based programming framework that enables the development of applications to be executed in distributed environments. COMPSs has two main aspects that may be used for mF2C deployment. First, it offers a simple programming interface and a set of tools that support the development of applications. Second, it comes with a powerful runtime environment able to build, at execution time, a workflow of the tasks comprising the application and execute them in a distributed environment. The runtime environment orchestrates the execution of the tasks in the workflow, and handles the required data transfers. The distributed computing platform can be composed of physical nodes or nodes in a cloud, and can include tasks deployed as web services.

2.3 Security Aspects

Security and privacy are well-known, widely addressed aspects, but remain greatly unsolved challenges in the cloud and fog areas, and are inherent to mF2C. Deploying fogs in fact exacerbates the traditional cloud security issues, since usually edge devices are located in non-controlled scenarios, and often misused by adversaries. This assessment is even extended when bringing together fog and cloud resources.

Information security in fog infrastructure currently builds on cloud, mobile, or network security. Many solutions are available when integrating with a single cloud provider, and several research initiatives have researched secure brokering of, and access to, multiple clouds. Mobile security is used by most of the apps, using either the user's telco account or their own app-specific security; in general, security is very application dependent and users have little control over it: the applications today either get all permissions they ask for, or nothing. Data confidentiality in-flight uses X.509 certificates or provider-specific symmetric keys; confidentiality at-rest is often via non-technical controls: contractual agreement – or trust. Authorization decisions are usually implicit – users who can access the service are authorized – or based on simple identity mappings or roles (RBAC). Intrusion detection is done via monitoring IaaS networks (e.g. Azure, Amazon, HPE) in addition to "traditional" methods of virus checking, etc.

We may conclude both: i) recent contributions in the security field for fog computing are not solid enough to be widely adopted by mF2C, and; ii) contributions in the cloud arena are too far from the specific mF2C needs, in terms for example of resources dynamicity or volatility.

3 mF2C Management Functionalities

Recognizing the need for a novel management ecosystem for F2C, this section outlines the mF2C management architecture, and shows preliminary benefits of an mF2C deployment in three illustrative real-world scenarios.

3.1 mF2C Management Architecture

The mF2C management architecture is structured into three architectural entities (mF2C Controller, mF2C Gearbox and Interfaces, shown in Fig. 2), all coordinated to work together in order to provide the different expected control and management functionalities.

mF2C Controller

This architectural entity consists of three connected and coordinated blocks. The three blocks share security and privacy aspects as a transversal requirement.

- *mF2C Resource Controller:* This function includes three main components Semantic Adaptation, Resource Management, and Security and Privacy. In turn, these components include methods and mechanisms implemented to guarantee both accurate knowledge about the available resources for each device and accurate information about the resource availability, including resource attributes, such as virtual/real, static/mobile, sharing capacities, clustering capacities, business policies, etc. (i.e., Resources Monitoring, Discovering, Virtualization). This is rather complex in F2C systems, due to the dynamicity inherent to its resources, the heterogeneity foreseen for the devices and systems comprising mF2C as well as the business relationship to be established among resources providers. To this end, mF2C can be envisioned as an opportunistic resources.
- *mF2C Service Controller:* Once the service request is validated, the service is categorized according to a dynamic taxonomy, which is yet to be designed, (i.e., Service Categorization and Decomposition). When required, the service can be decomposed into sub-services, ultimately turning it into a set of atomic services (sub-services) some of which can be executed in parallel. The set of sub-services may be preconfigured and stored in a repository. Challenging issues in this area include: to find the appropriate place to locate the service decomposition, to minimize the computing load and/or data transfer while keeping fast reaction time, to define to what extent these functions must be associated to the aggregation points, to define the dependency graph rules, and finally to develop strategies for sub-services search.
- *mF2C User Side:* mF2C must benefit from the user-specific context information to tailor service execution to specific user demands. To that end, a comprehensive set of functionalities must be defined, including but not limited to authentication, privacy, location, profiling, agreement policies, etc. (i.e., User and Context Functions). All these functionalities must meet the business policies in a real mF2C deployment. For example, a user may be willing to connect his/her smart car as a resource which requires appropriate economic incentives. The user could restrict the car compute system to send only anonymised data by default, as well as relay

emergency messages, or offer additional services via social networks (location, camera images, additional processing). A service request should be "validated" on the user side (or in an upper control layer if the device does not embed that functionality), checking for authentication and checking attributes release according to the user profile and the context (i.e. User Authentication and Profiling). To that end, novel strategies must be defined to describe the different new roles users may play, e.g., including fog providers or the capacity and incentives to share resources.

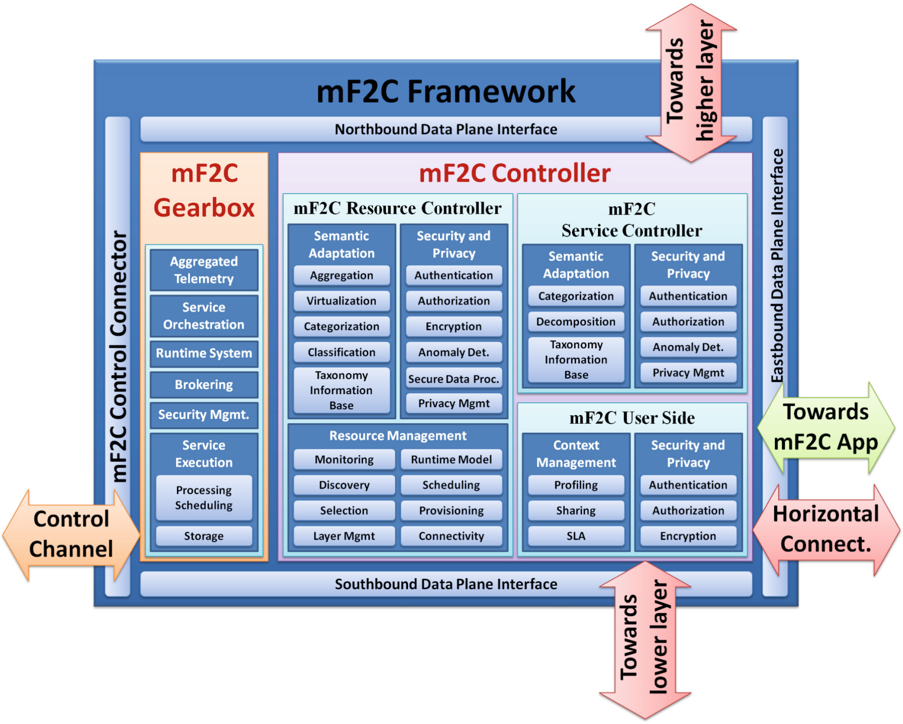

Fig. 2. Architectural blocks for the mF2C management framework

mF2C Gearbox

The set of preliminary components defined to build the Gearbox is:

- Aggregated telemetry: Rich, intelligent instrumentation and monitoring is required to inform decision making systems such as the service orchestration. For effective decision making and troubleshooting, this should cover the full-stack – from hardware up through operating system, middleware and hosted services be they deployed in containers or virtual machines. It should also be dynamically configurable, and support derived or aggregated metrics at the edge for maximum scalability of the overall solution.

- Service orchestration: This component is responsible for allocating services to the best available resources. The optimal allocation will depend on many factors. Considerations such as an analysis of historical invocations of the service, the precise nature and configuration of available resources in real time, and quality-of-service expectations and commitments could all have a bearing on where services, or elements of the service, are located. Effective abstractions and analytics will be required to ensure service orchestration systems are scalable at runtime.
- Runtime system: Different options may be considered for the runtime system in the F2C scenario, from traditional sequential execution to novel parallel execution. This component enables a transparent handling of the heterogeneous resources.
- Brokerage: Responsible for handling the dynamicity inherent to the edge devices while guaranteeing – or at least optimizing – that selected resources best match the services demands. Different resources registration policies may be considered depending on the context and the different devices.
- Service execution: Software execution and storage platform that unifies the model of all data (user, application and shared) into the potentially access controlled view seen by applications.

Interfaces

The *Interfaces* are key to the main feature of openness, modularity and extensibility of the mF2C framework and the platform. Since the mF2C is designed as an open layered framework for a customized usage by various devices and systems, the modules implemented by a specific F2C layer are connected with the overall system over these interfaces. Figure 3 illustrates the mF2C layered architecture (including agents and microagents to be deployed in edge devices with limited capacities) and the role of interfaces. The lowest layer represents the embedded devices, such as sensors with minimal processing capability, while the smart phone is in the middle layer (shown as a fog device), capable of processing an mF2C service on a small scale. Clouds are at the top layer, controlling the mF2C services at large scale. The control channels and data channels are separated. Data channels strictly follow the F2C layered hierarchy.

As it can be seen, multiple data channels connect to multiple child instances through the “Southbound” interface to lower layers. A single data channel connects to the (unique) parent instance through the “Northbound” interface. The “Eastbound” interface connects to the mF2C application as well as enables multi-cloud/fog communication within the same layer. All control channels (the “Westbound” interface) connect to the top layer instance that controls and manages the whole mF2C environment.

In addition to the three architectural entities mentioned above, security and privacy are cross-cutting concerns, transversal to the mF2C Controller and the mF2C Gearbox, meaning that all components in the overall mF2C management ecosystem must be designed, implemented, and operated to fulfill a common base set of security and privacy requirements and policies (these policies may of course depend on the device type or function). We expect that some security and privacy components will work in the same way, or at least in similar ways, for many mF2C components, including authorization decision that certain user data may be processed on a specific fog device. The basic functionalities for security and privacy for mF2C data are information classification, authentication, authorization, accounting, auditing, attack detection and finally secure data processing.

3.2 Applying mF2C to Real-World Scenarios

An mF2C coordinated resources management architecture is expected to help increasing revenues and product innovation to the businesses in various sectors. From a technology provider perspective this evolution (bottom-up) would boost the adoption of IoT devices and equipment in the various depicted scenarios (cities, buildings, etc.) and commercial development of value-added services.

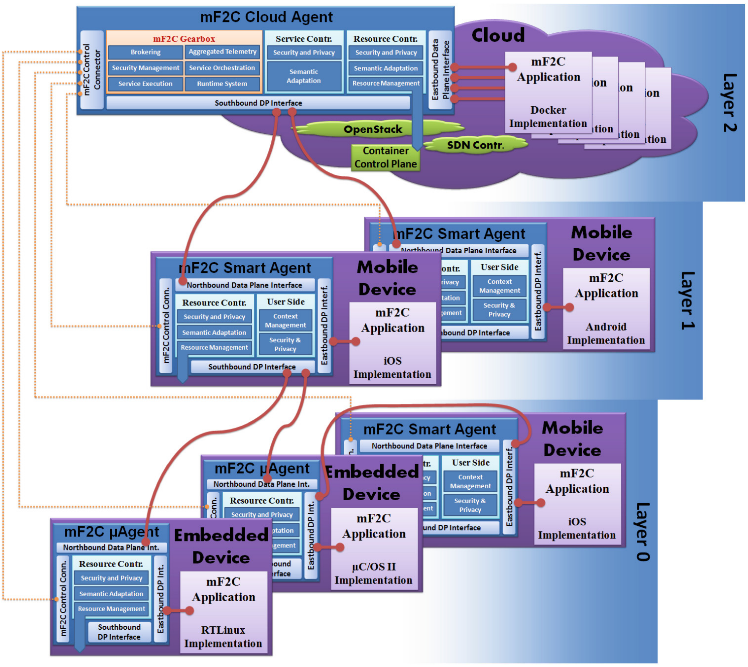

Fig. 3. Layered scenario with agents and interfaces

With the massive adoption of these devices, the revenues and requests for more sophisticated ones will increase as well. From a Service Provider perspective the availability of an extended platform (coming from the Cloud + Fog Providers) –with an elastic provisioning of resources that covers also the edge devices– offers them the opportunity to develop even more sophisticated services, like dependable e-health, or 3D real time navigation systems, thus widening the market scenario, extending their offering, and creating more value and revenues. Finally, from a Cloud provider perspective this evolution (top-down) creates ample opportunities for developing and extending the service chain offering, by adding one more ring (the Fog) in the provision of services, increasing the product/service portfolio and enabling new and challenging business models. In this way Cloud Providers could soon be renamed "Cloud + Fog Providers".

To illustrate the expected mF2C benefits and impact on these three different areas, we give examples of three real-world scenarios that can immediately deploy the systems akin to mF2C.

Scenario 1: Emergency Situation Management in Smart Cities (ESM): This application scenario is built upon real infrastructure developed in the city of Bogota, Colombia [14]. It consists of an implementation of distributed elements capturing signals and data, as well as a centralized traffic management system to integrate heterogeneous traffic-related information in a flexible and efficient cloud platform. A potential deployment of the mF2C management solution will enable cities to install fog computing infrastructure locally, for example in bus stops, and enable new real time services and push notifications without the need for tight connectivity infrastructure.

Scenario 2: Enriched Navigation Service (ENS): This scenario is based on the development and extension of the family of IoT devices and sensors that are oriented to operational support and monitoring in the marine sector, aiming at providing safer navigation even for less experienced sailors [15]. The example shows a relevant potential for making all the ship's sensors work together, processing and correlating the collected data in a combined fog and cloud computing system but also interacting with external data sources as well (e.g., other ships and marine vehicles, satellites). This achievement could lead to brand new added-value services of augmented reality in the marine sector. The mF2C management framework looks perfect for a technology like Sentinel [15], as the supporting technology for data processing orchestration and distribution, leveraging the access to open data databases and ontologies and the chance to develop new predictive models for forecast weather and travel related aspects, as part of new value-added services to support sailors' route planning. Currently the Sentinel devices work mainly individually but their crowd knowledge, which could be derived from combining and processing the data obtained from all distributed sensors, is not yet exploited.

Scenario 3: Smart Fog-Hub Service (SFHS): The third scenario is looking at the IoT evolution as a potential area where current cloud offering could be enriched and differentiated. Scenario 3 extends the concept of a "cloud hub" to a new concept of "fog hub", driven by real market needs. This scenario leverages the belief that value is generated at the business services level, particularly in spaces with recurring concentrations of people and objects that can communicate and interact. These scenarios are typical of airports, railway stations, seaports, shopping centers, hospitals, sports facilities, large parking areas, but also domestic scenarios with a communal clustering level. The scenario proposes to set up (Fog) Hubs in such scenarios to interact with all the objects within the scope of coverage, and to operate "in-proximity" marketing efforts, applying predictive algorithms to track (in an anonymized form) movements, choices and decisions of persons nearby, or even extend the hub with devices (e.g. beacons) capable of sending input (e.g. customized advertising) and determine the effectiveness of the specific campaign in terms of attention/visits rather than conversion (purchasing products/services). Potentially this model could be further extended by making different fogs, perhaps 5-15 km from each other, communicate, and by combining the results in terms of behavioral predictions in adjacent fogs.

4 Opportunities and Challenges in mF2C

There is no doubt that, to make the most out of the whole set of cloud and fog resources, that is for the overall F2C ecosystem to work, a new coordinated, open, secure, end-to-end management strategy must be developed to smartly orchestrate a large-scale, distributed, heterogeneous, open, dynamic and volatile set of resources in a decentralized, private, secure and trusted way, enabling open/multi–fog/cloud provider business models. But, this will not be enough. Users must endorse this computing strategy by sharing their resources (edge devices), thus enabling the collaborative model envisioned for F2C. User engagement should, in turn, incentivize the industrial sector to develop new business models and applications tailored to the F2C characteristics and the end users' engagement policies. We envision new cooperation modes to appear analogous to recent ideas in "sharing economy", such as Airbnb.

Cooperation can be fostered by shared interests and geographic proximity. For example, a group of cars in a parking lot may "decide" to "share" some of their resources to be offered to "other cars", hence becoming a local cloud or fog provider themselves (similar to the concept of micro data centers, small clouds or cloudlets) thus setting the stage for future business models. Already today models are emerging of negotiating and "selling" parking spaces, both on-demand use of vacant spaces outside homes as well as peer-to-peer selling between car owners in car parks.

5 Conclusions

This paper revisits the main cloud and fog computing concepts, envisioning their combination as next cloud evolution, making the best out of the set of distributed resources by combining cloud and fog computing. The paper introduces the need for a coordinated management of both systems and proposes a functional architecture of the management ecosystem able to intelligently manage the distributed set of resources, optimizing service execution according to resources availability and users' demands. The main functional blocks of the management architecture (referred to as mF2C) are proposed, along with an in-depth description of open challenges. We envision F2C as a key paradigm in the future as the next evolution in the cloud domain, and hence with a strong impact not only on the industrial sector but also on society and individuals. We believe in the prospect of collaborative computing model as foreseen for F2C, that can extend the well-known sharing economy model to edge devices owned by users.

Acknowledgments. This work was partially supported by the EU-funded mF2C project (Nr. 730929). Additionally, UPC authors are supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under contract RTI2018-094532-B-I00). BSC author is also supported by the Spanish Government (TIN2015-65316-P, SEV-2011-00067) and by the Generalitat de Catalunya (2014-SGR-1051).

References

1. Plummer, D.C., et al.: Top 10 Strategic Predictions for 2015 and Beyond: Digital Business is Driving “Big Change”, Gartner Inc, October 2014. http://www.gartner.com/technology/home.jsp
2. Bonomi, F., et al.: 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)
3. Masip, X., et al.: Foggy clouds or cloudy fogs; a real need for a coordinated management of F2C computing systems. IEEE Wireless Communications Magazine, in press (preprint version at http://www.craax.upc.edu/images/Publications/journals/Fog-to-cloud_preprint.pdf). HP report at http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676#.VqtAk8eC0hA De Filippi, P., McCarthy, S. “Cloud computing: Legal issues in centralized architectures. In: VII International Conference on Internet, Law and Politics (2011)
4. Barbosa, V., et al.: Handling service allocation in combined fog-cloud scenarios. IEEE ICC 2016, Malaysia, May 2016
5. Aazam, M., Huh, E.N.: Fog computing and smart gateway based communication for cloud of things. In: 2014 International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, August 2014
6. Cisco Fog Computing Solutions: Unleash the power of the Internet of Things. http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-solutions.pdf
7. Sentilo platform. http://www.sentilo.io
8. Future Internet-of-things Testbeds, IoT-LAB. https://www.iot-lab.info/
9. Fiware platform. http://www.fiware.org
10. Sofia platform. http://sofia2.com
11. VORTEX. www.prismtech.com/vortex
12. Hong, K., et al.: 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 2013, Hong Kong, July 2013
13. Rosa, M.B., et al.: COMP Superscalar, an interoperable programming framework, SoftwareX, vol. 3–4, December 2015, pp. 32–36. ISSN 2352-7110. http://dx.doi.org/10.1016/j.softx.2015.10.004
14. Smart city platform in Bogota. http://www.worldsensing.com/solutions/mobility/projects/worldsensing-iot-platform.html
15. Sentinel marine platform. http://www.sentinelmarine.net