Kuo, P.H., Mourad, A., Chenguang, L., Berg, M., Duquennoy, S., Chen, Y.Y., Hsu, Y.H., Zabala, A., Ferrari, R., González, S., Li, C.Y., Chien, H.T. (2018). An Integrated Edge and Fog System for Future Communication Networks, *in 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Barcelona, Spain, 2018, pp. 338-343.

DOI: 10.1109/WCNCW.2018.8369023

An Integrated Edge and Fog System for Future Communication Networks

Ping-Heng Kuo[1], Alain Mourad[1], Chenguang Lu[2], Miguel Berg[2], Simon Duquennoy[3], Ying-Yu Chen[4], Yi-Huai Hsu[4],
Aitor Zabala[5], Riccardo Ferrari[6], Sergio Gonzalez[7], Chi-Yu Li[8], Hsu-Tung Chien[8]

Correspondence: ping-heng.kuo@interdigital.com

[1]InterDigital Europe LTD, [2]Ericsson Research, [3]SICS, [4]ITRI, [5]Telcaria SRL,
[6]AZCOM, [7]University Carlos III of Madrid, [8]National Chiao Tung University

***Abstract*—Put together, the edge and fog form a large diverse pool of computing and networking resources from different owners that can be leveraged towards low latency applications as well as for alleviating high traffic volume in future networks including 5G and beyond. This paper sets out a framework for the integration of edge and fog computing and networking leveraging on ongoing specifications by ETSI MEC ISG and the OpenFog Consortium. It also presents the technological gaps that need to be addressed before such an integrated solution can be developed. These noticeably include challenges relating to the volatility of resources, heterogeneity of underlying technologies, virtualization of devices, and security issues. The framework presented is a Launchpad for a complete solution under development by the 5G-CORAL consortium.**

Keywords—Edge; Fog; MEC; 5G; RAN; IoT

I. INTRODUCTION

The need to provide networking, computing, and storage capabilities close to the end users has arisen recently to meet challenging KPIs of low latency and high bandwidth for the emerging applications. Examples of these applications include connected cars, Internet of Things (IoT), and virtual reality (VR). The European Telecommunications Standards Institute (ETSI) has been the first to acknowledge such a need and has started the development of Multi-Access Edge Computing (MEC) [1]. ETSI MEC sets focus on supporting applications to better serve the end user by running directly at the edge closer to the user and by exploiting context information available through the edge. ETSI MEC adopts a distributed computing infrastructure that embraces virtualization technologies to enable flexible and scalable deployment of applications. These applications in one form may also be virtual network functions, hence the overlap with ETSI NFV (Network Functions Virtualization) [2]. ETSI MEC is currently addressing challenges relating to the portability of MEC applications from one MEC host to another, the federation of heterogeneous MEC resources from different domains and different owners, and the (de-)composition of applications from elementary tasks distributed anywhere in the MEC infrastructure.

The OpenFog consortium [3] was also established recently with the aim to leverage virtualization for a cloud-to-thing continuum where applications can be distributed anywhere between cloud and things, and this may well be outside the control of mobile operators. From a terminology point of view, some might prefer to differentiate a non-data center view of the edge as the fog, but functionally speaking they are basically the same, simply more or less capable distributed computing and networking nodes strategically positioned to ensure some target level of service.

The fog adds new degrees of freedom where computing and networking resources may be volatile, handled on demand, and not necessarily fully owned and controlled by a mobile operator. The two organizations ETSI MEC and OpenFog have recently reached an understanding with intent to share work related to global standards development for fog-enabled mobile edge applications and technologies [4]. This paves the way towards a holistic solution that integrates both edge and fog, going all the way down from edge nodes at aggregation points, to fog nodes at access points, terminal devices, and even particular chips inside the terminal device.

Recently, a 5G-PPP Phase 2 project entitled **"*5G-CORAL: A 5G Convergent Virtualised Radio Access Network Living at the Edge*"** [5] has been launched with objectives of providing solutions for edge and fog integration. This paper aims to lay down the foundations of the integrated edge and fog solution that has been envisioned by 5G-CORAL. It proposes a framework dubbed as EFS (Edge and Fog System) where the edge and fog resources in a given local area are pooled together, abstracted and jointly orchestrated in one single platform. Applications, functions and services are virtualized and hosted over the EFS virtualization infrastructure. They provide and consume data accessible through EFS in order to optimize performance or behaviour and provide value-added services. For instance, the information relating to a plurality of different radio access technologies (RATs) can be exposed to the EFS platform, which may permit new approaches of multi-RATs convergence. In addition, the interaction with other EFSs in other areas or distant cloud systems is also considered.

The rest of this paper is organised as follows: Section II presents the state of the art from ETSI MEC and OpenFog. Section III then introduces the EFS concept, its architecture and components. Section IV analyses key technical gaps and challenges that need to be addressed. Finally, Section V draws the conclusion and prospects for future work.

II. STATE OF ART – ETSI MEC AND OPENFOG

Prior to delving into the details of the proposed EFS framework, it is important to review the efforts that have been made for edge and fog computing by industrial standards. In particular, the main frameworks in the state of art around edge and fog computing, namely ETSI MEC and OpenFog Consortium, are elaborated in this section as the background.

A. *ETSI MEC*

ETSI MEC is a cloud-based computing environment, the idea of which is to enable computing at the edge of the network. Apparently, by running applications in proximity to the end user terminal, services require very low latency and high bandwidth become much more feasible. It is worth noting that the term MEC was originally referred to as Mobile Edge Computing, but ETSI has recently re-branded MEC to Multi-Access Edge Computing [7], in order to stress the goals of supporting LTE as well as several other different access technologies, including Wi-Fi or 5G. Hence MEC can be regarded as a standards-based interface for multi-access hosts. MEC framework consists of the following entities, as described in [1], (i) mobile edge host, (ii) mobile edge host level management and (iii) mobile edge system level management. This is depicted in **Figure 1** below.

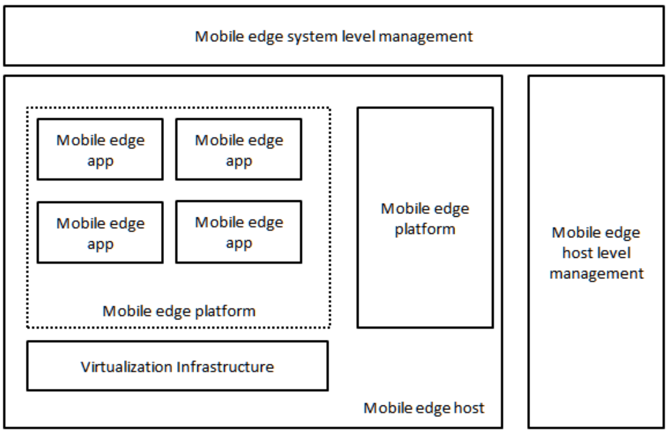

Figure 1. MEC architecture.

The mobile edge host is an entity that contains the mobile edge platform and the virtualization infrastructure. The mobile edge platform offers an environment where the mobile edge applications can discover, advertise, consume and offer mobile edge services. The virtualization infrastructure provides compute, storage, and network resources for the mobile edge applications [6]. The mobile edge applications are run as virtual machines (VM) providing mobile edge services. The mobile edge host level management comprises the mobile edge platform manager and the virtualization infrastructure manager, and takes care of the mobile edge applications. The mobile edge platform manager handles the life cycle of the applications. The mobile edge system level management controls the whole system which is composed of multiple mobile edge hosts. In addition to **Figure 1**, there is a virtualization infrastructure manager (VIM) supervising the compute storage and networking functionalities of the virtualization infrastructure on which the applications run. The mobile edge orchestrator maintains an overall view of the mobile edge system. It is also responsible for the on-boarding of application packages and the selection of appropriate mobile edge host(s) for application instantiation.

B. *OpenFog Consortium*

The OpenFog Consortium [3] was formed on the principle that an open architecture is required for the success of fog computing system for IoT platforms and applications. OpenFog Consortium defines fog computing as a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking functions close to the users along the continuum from Cloud to Things. The OpenFog reference architecture is designed based on the definition of fog computing defined by OpenFog Consortium. The OpenFog reference architecture is constructed by a set of principles called pillars which are security, scalability, open, autonomy, Reliability, Availability, and Serviceability (RAS), agility, hierarchy, and programmability. Based on these pillars, OpenFog proposed an architecture description with five perspectives, namely performance, security, manageability, data analytics and control, and IT business and cross fog applications, as shown in **Figure 2**.

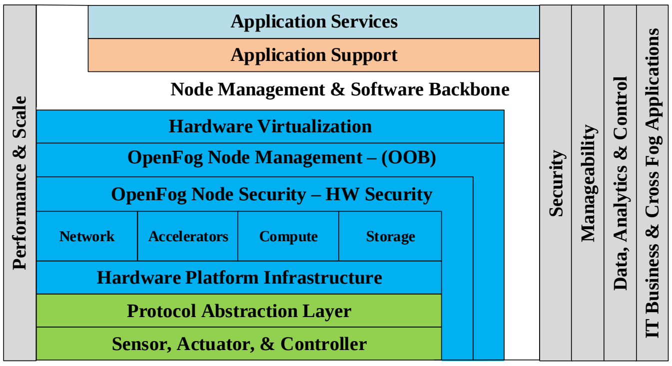

Figure 2. Architecture description with perspectives.

The most critical metric for performance of a fog computing environment is the latency. Obviously, it is crucial to guarantee end-to-end security from the security perspective. For manageability, on the other hand, the abilities of managing various aspects of fog computing deployment are of paramount importance. In view of data analytics and control, fog nodes need local data analytics control to form an autonomous fog computing system. Finally, the IT business and cross fog applications aim to enable the migration and properly operation among fog computing systems of multi-vendors.

For the protocol stack shown in **Figure 2**, application services are dependent on infrastructure and fulfill specific end use cases requirements. Application support is infrastructure software that does not fulfill any use case, but support and facilitate the application services. Node management and software backplane is general operation and management of the node and its communications with other nodes and systems. Hardware virtualization means the fog node should support virtualization technique such as VMs or containers. OpenFog node management is a discrete manageability system that will include a hardware platform management (HPM) which is responsible for controlling and monitoring the other components inside the fog node (e.g. storage, accelerators, et al). OpenFog node security should guarantee the end-to-end security for fog nodes. The hardware platform infrastructure provides a common set of hardware infrastructure components and software modules are plugged in the hardware infrastructure to customize the function of specific fog nodes. Protocol abstraction layer allows the sensors or actuators which have their own protocol to communicate with the fog network directly. Sensors, actuators, and control are the lowest level IoT devices which may or may not have fog capability.

III. EDGE AND FOG INTEGRATION

As discussed previously, MEC is based on a centralized computing architecture, aiming to build up a dedicated edge cloud infrastructure closer to the end users than the distant cloud. For example, the Telecom Central Offices (CO) are typical locations to host such smaller scale data center infrastructure, which basically follows the cloud design principles but with smaller number of servers than a cloud data center. However, Fog Computing features a distributed computing architecture, utilizing the computing resources available in diverse types of computing devices around the end users. To give some examples, such computing devices can be base stations, access points, network switches and routers, connected cars, IoT gateways, smart phones, and even wearable devices.

The two paradigms of MEC and Fog clearly overlap in some cases and extend each other in some other cases. Put together they form a larger and more diverse pool of computing and networking resources of different owners that can be leveraged towards low latency applications as well as for alleviating high traffic volume in future networks including 5G and beyond. This clearly calls for an integrated solution across Edge and Fog to be envisioned, which we refer to in this paper as EFS (Edge and Fog System), illustrated in **Figure 3**.

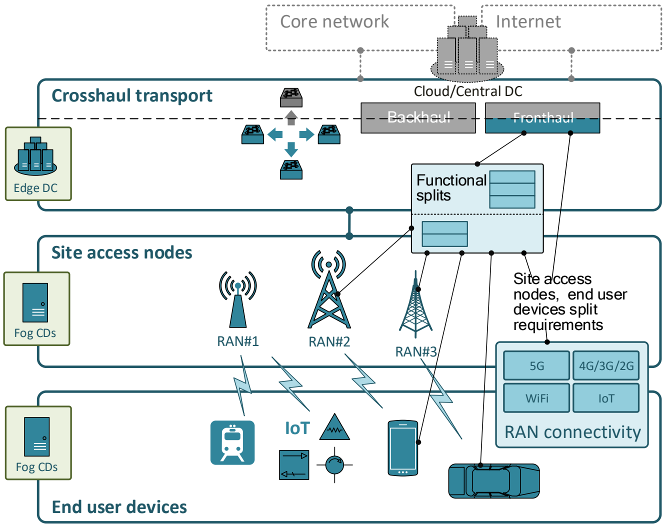

Figure 3. Physical view of EFS.

As shown in **Figure 3**, EFS is composed of three layers, a first (MEC) layer denoted as Edge Data Centers (DCs) and two Fog computing layers underneath denoted as Fog Computing Devices (CDs). Basically, the edge DCs layer is located at the network segment capable of serving many end users, handling heavy computational tasks and provide a large amount of storage. For example, virtualized Radio Access Network (RAN) functions can be hosted in Edge DCs. The first Fog layer is on the base station level where the base stations, access points or small computing gears nearby (e.g. mini PC, Raspberry PI etc.) for different Radio Access Technologies (RATs) can also serve as Fog CDs. For example, it can host Content Delivery Network (CDN) functions to reduce the backhaul load. The second Fog layer is around the end users, such as the on-board small cells on trains, the computing gears in connected cars, smart phones and IoT devices. Some light computational tasks like analytics can be offloaded to these Fog CDs.

In EFS, the computing, storage and networking resources in the Edge DCs and Fog CDs layers are abstracted, virtualized and managed in one common virtualization platform, as shown in **Figure 4**. The functional blocks of EFS functions, applications and services will operate in this EFS virtualized infrastructure. Note that the **EFS service platform** (ESP) manages the data services between EFS and non-EFS functions and applications, where non-EFS functions and applications are those hosted outside of the EFS, e.g. in a central cloud. The functions and applications exchange data through the EFS service platform, e.g. via a subscribe/publish mechanism.

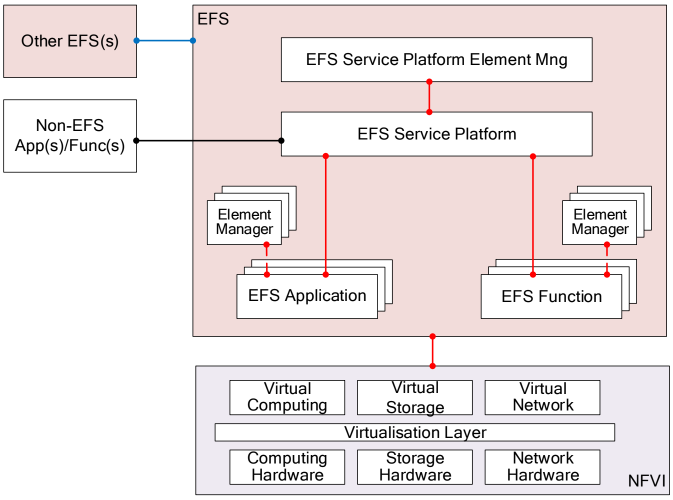

Figure 4. Functional view of EFS.

More details regarding these three key functional blocks with an EFS, namely applications, functions, and services are expounded below.

1) *EFS applications* perform certain computational tasks and are virtualized on the EFS. They can provide services by publishing data, e.g. context information, through the ESP via Application Programming Interfaces (APIs). They can consume the data from the services available on the ESP by subscribing these services provided by other EFS and non-EFS applications and functions. The EFS can also host the applications that neither publish data nor consume services. The applications can be either user or third-party applications. Specifically, the user applications are referred to as the

applications directly consumed by the user, e.g., Augmented Reality (AR) and User-Targeted Advertisements (UTA). For developers, such a user application is usually developed into three parts running on user equipment, at the edge (i.e. EFS) and in cloud, respectively. The part running at EFS is the EFS application defined here. For example, in AR, UEs can offload heavy computational tasks like object recognition to EFS applications running on Fog CDs or/and Edge DCs. A third-party application, on the other hand, is employed by certain vertical industries or products for various types of purposes. For example, a car safety application may be launched to give pre-crash warnings and collision avoidance signaling in a timely manner, by fetching precise information relating to the location of nearby vehicles with sufficiently low latency. Another example for a third-party application is an IoT gateway application to facilitate the coordination and management of IoT devices and enhance the overall security.

2) EFS functions are classified into **Virtualized Network Functions (VNFs)** and **Performance Enhancement Functions (PEFs)**. The VNFs pertaining to RAN, transport networks, and core networks could find it beneficial in terms of latency to execute in the EFS instead of a distant cloud. However, the trade-offs between latency and bandwidth must be considered. A finer granularity of the RAN functional split could be also envisaged, where instead of splitting between stack layers or functions in the same layer as today, the splitting could be done inside the same function into elementary virtualized functions executing in tailored EFS resources. For example, this could be the case of authentication functions whose decomposition and virtualization could be advantageous to facilitate seamless session continuity between different RATs. The virtualized core network functions like Mobility Management Entity (MME) and PDN Gateway (P-GW) in 4G networks can be hosted in EFS to facilitate data offloading and mitigate heavy-signaling caused by frequent handovers in the local access. The PEFs, on other hand, may obtain context information from different RATs and/or nodes from the ESP for the sake of making certain decisions to optimize network performance or achieve various connectivity KPIs. For instance, the coordination of spectrum, interference, and energy among multiple RATs could be carried out by PEFs within the EFS by exploiting the context information.

3) EFS service platform is in charge of the data services, to store and distribute the subscribed data of a service to the data subscribers, while the service data are published by the data publishers and organized as EFS services by the EFS service platform. It specifies the protocols and mechanisms for data communication, storage and management and serves both EFS and non-EFS functions and applications though APIs. The non-EFS functions and applications are hosted outside of EFS, such as on the TN) and CN networks, as well as distant clouds. For example, the RAN functions can publish the RAN context information and the platform can abstract and organize the information as a RAN context service. The subscribing applications of the RAN context service get the context information and use them for their own purposes. For example, a load balancing application can avoid using overloaded RATs based on the RAN context information.

IV. TECHNOLOGICAL GAPS AND CHALLENGES

The EFS brings many opportunities of improving future networks. Nevertheless, several foreseeable technological gaps and challenges must be first investigated and addressed to realize such a concept. Some of the issues are discussed below.

A. *Volatility of Resources*

The fog computing, storage and networking resources are borrowed from devices close to the end user, such as a smartphone, a smart TV, or a connected vehicle. This implies that the availability of fog resources may not be continuous and persistent due to their volatile nature. To be specific, the devices that contribute these fog resources may move away or switched off anytime, and hence causing interruptions to the operations of functions and/or applications that are hosted or facilitated by the computing system amalgamating both edge and fog resources. How the tasks of these functions and applications can be carried out in a seamless manner is indeed a challenging issue that needs to be addressed.

B. *Heterogeneity of RATs*

Various technologies are anticipated to provide the connectivity service between the different edge and fog devices in the EFS infrastructure. These may include any RAT, cellular (e.g. 4G, 5G) and non-cellular (e.g. WiFi, Bluetooth, Zigbee, etc.). The context information that may be extracted from all these different RATs is certainly beneficial to expose into the EFS so that performance optimization can be sought for applications and network functions alike. The challenges here are clearly in: (i) determining what context information may be useful to extract from the different RATs, and how to extract and expose these as services into the EFS; (ii) designing mechanisms that consume these context information services in order to optimize the performance of applications and the underlying multi-RAT network.

C. *Applicability to Internet of Things*

IoT scenarios can benefit greatly from edge and fog computing [8], with geo-distribution, mobility, location awareness, low latency, heterogeneity of technologies, and support for real-time interactions. For enhanced flexibility and integration of new technologies, it is possible to virtualize IoT gateways in the EFS. In such a setup, the gateway is simply an interface to the wireless environment, but technologies and protocols are run in the EFS, in such a way that they can benefit from the EFS ecosystem and services. How to efficiently enable such an architecture while meeting the stringent performance requirements of the wireless protocols is an interesting challenge. Another one is how to best combine different sources of information obtained from EFS services, to improve localization and/or communication.

D. *End User Terminal Virtualization*

Thanks to technological advancement in virtualization, it is possible that some of the computing or networking tasks of an end user terminal may be moved for execution in the EFS. For instance, to avoid fast battery depletion of a user terminal, some of the more battery-consuming tasks (e.g. high-complexity computations) can be offloaded to a virtualized user terminal shadow in the EFS [9]. Moreover, some of the radio access mechanisms, such as Automatic Repeat request (HARQ) for reliability enhancement, may be better handled in the EFS through the virtualized terminal shadow on behalf of the end user terminal (as illustrated by Figure 5). This may help for example to save the end user terminal's battery, as it may enter a sleep mode earlier without waiting for the acknowledgement from its communicating peer (e.g. a base station). Before realizing such ambitious end user terminal virtualization, there is a number of challenges that need to be addressed first such as: (i) when to create such virtual (shadow) terminal, where to host it, which scope (of tasks) is advantageous to assign to it, and how dynamically changing are all these; (ii) what interface(s) are needed to connect the end user terminal with its virtualized shadow in the EFS, but also with other peers (both physical and virtual); (iii) what constraints (e.g. security, privacy) may apply for deciding where in the EFS to host the shadow of a given user terminal, and how are these constraints complied with and controlled.

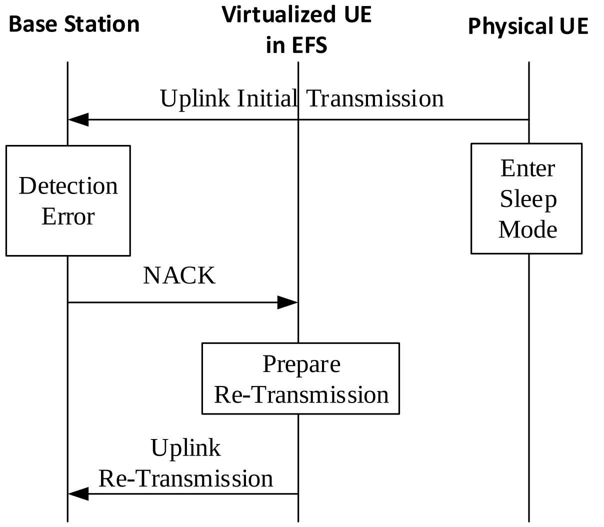

Figure 5. Illustration of HARQ with User Virtualization

E. Security Challenges

The challenges to secure the EFS platform can be threefold. First, EFS applications which can be developed by various parties may be buggy/malicious, thereby exposing the platform or benign applications to security threats. Except for isolating them based on virtual machines or containers, the interaction between them and the EFS core should be secured. Second, the EFS platform can support outside non-EFS functions and applications, which may be malicious. It should not only interact with them based on a set of secure APIs, but also deploy a firewall-like feature to defend against external attacks. Third, two EFS platforms, e.g., two fog nodes, should have mutual authentication, security control over resource sharing, and end-to-end transport security. For one EFS platform, others may have different levels of trustworthiness. It thus needs to provide multiple levels of access control on the sharing of various resources.

V. CONCLUSIONS

This paper presented the concept of an integrated edge and fog system. Such a system is characterized by the volatility of its resources, the heterogeneity of its nodes and technologies, and the multiplicity of its resource providers. It offers however a higher scalability, higher resource usage efficiency (pooling gain), and higher flexibility in executing various applications and functions in the continuum between fog nodes, edge data centers, and central clouds. Several challenges lie ahead before such a system can be fully developed. These include: (i) ways of coping with the volatility of the resources to ensure seamless execution of the applications and functions; (ii) mechanisms to abstract, integrate, federate and dynamically allocate the large pool of heterogeneous resources; (iii) mechanisms to gather, provide and consume context information in a timely manner from across the large base of resources and technologies; (iv) algorithms that can act timely on the context information when it is still relevant to optimize the system performance and quality of service; (v) schemes that enable virtualization when and where it is needed, down to the level of end user terminals; (vi) secure and trustworthy procedures for all exchanges between applications, functions, and services, no matter where they are executing. All these challenges are to be addressed by the research and development community, such as ETSI MEC, OpenFog, and the recently launched 5G-CORAL project [5].

ACKNOWLEDGMENT

This work has been partially funded by the H2020 collaborative Europe/Taiwan research project 5G-CORAL (grant num. 761586).

REFERENCES

[1] European Telecommunications Standards Institute, "Mobile Edge Computing (MEC); Framework and Reference Architecture," ETSI GS MEC 003, March 2016.

[2] European Telecommunications Standards Institute, "Network Functions Virtualisation (NFV); Acceleration Technologies; VNF Interfaces Specification," ETSI GS NFV-IFA 002, April 2016.

[3] OpenFog Consortium, https://www.openfogconsortium.org/

[4] European Telecommunications Standards Institute, "ETSI and OpenFog Consortium collaborate on fog and edge applications," 2017. [Online]. Available: http://www.etsi.org/news-events/news/1216-2017-09-news-etsi-and-openfog-consortium-collaborate-on-fog-and-edge-applications. [Accessed: October 20, 2017]

[5] 5G-CORAL, www.5g-coral.eu

[6] R. Mijumbi, et al. "Network function virtualization: State-of-the-art and research challenges", *IEEE Commun. Surveys & Tutorials*. vol. 18, no. 1, pp. 236-262 1st Quart. 2016.

[7] T. Taleb, et al, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration", *IEEE Commun. Surveys & Tutorials*, Vol. 19, No. 3, 3rd Quarter, 2017, pp.1657-1681.

[8] Z. Wen, et al, "Fog orchestration for internet of things services" *IEEE Internet Computing*, vol. 21, no. 2, pp. 16-24, 2017.

[9] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities", *IEEE Internet of Things Journal*, Vol. 3, No. 6, Dec. 2016, pp.854-864.