Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing

Koustabh Dolui
Fondazione Bruno Kessler, Trento, Italy
Email: k.dolui@fbk.eu

Soumya Kanti Datta
EURECOM, Sophia Antipolis, France
Email: dattas@eurecom.fr

***Abstract*—When it comes to storage and computation of large scales of data, Cloud Computing has acted as the de-facto solution over the past decade. However, with the massive growth in intelligent and mobile devices coupled with technologies like Internet of Things (IoT), V2X Communications, Augmented Reality (AR), the focus has shifted towards gaining real-time responses along with support for context-awareness and mobility. Due to the delays induced on the Wide Area Network (WAN) and location agnostic provisioning of resources on the cloud, there is a need to bring the features of the cloud closer to the consumer devices. This led to the birth of the Edge Computing paradigm which aims to provide context aware storage and distributed Computing at the edge of the networks. In this paper, we discuss the three different implementations of Edge Computing namely Fog Computing, Cloudlet and Mobile Edge Computing in detail and compare their features. We define a set of parameters based on which one of these implementations can be chosen optimally given a particular use-case or application and present a decision tree for the selection of the optimal implementation.**

***Index Terms*—Cloud Computing; Cloudlet; Edge Computing; Fog Computing; IoT; Mobile Edge Computing.**

I. Introduction

The recent growth in services and applications leveraging the Internet has contributed to a steep rise in data storage and processing requirements. They are diverse in terms of the resources required by different applications and thus, often invoke tailor-made solutions. Cloud Computing provides as a suitable solution in in this context by leveraging the advancements in computing and network technologies. The backbone of the Cloud Computing paradigm is based on the data centers which are capable of handling storage and processing of large scales of data. These data centers are often connected with each other over optical networks to form data center networks (DCNs) appearing as a singular resource to the end user, with low-latency communication among the data centers. However, Internet of Things (IoT) systems have presented a new set of requirements the well established CC based solutions. IoT domains especially connected vehicles require near real-time processing of sensor data to take decisions and perform actuations. Even though the communication inside the DCNs are suitable for low-latency communication, the latency of communication between the end devices and the DCNs prove to be a bottleneck. This owes to the lack of location awareness while provisioning resources to an application or end device. Moreover, there is also a need for mobility support due to the agile nature of the end devices in various applications of these emerging technologies. In parallel, the number of connected devices is estimated to reach 30-50 Billion by 2020. The routing of massive scales of network traffic towards the DCs can prove to be a bottleneck degrading the latency and thus in turn the Quality of Service (QoS) and Quality of Experience (QoE). The vast number of requests to the DCNs in turn lead to the operation of the DCNs at a high duty cycle. This results in emissions of harmful greenhouse gases with a detrimental effect on the environment.

The Edge Computing (EC) attempts to overcome the described challenges. The EC leverages the storage and processing capacities of a large number of IoT devices connected to the Internet deployed for the purpose to provide an intermediate layer between the end devices and the cloud. With the presence of these "Edge devices", the computation load at the data centers are reduced by handling some of the requests directed to the cloud, locally, which do not require intervention from the cloud. This in turn, reduces the latency in resolving the requests and allows real-time handling of a subset of requests. Edge devices also support mobility due to the abundant availability and geo-distributed nature.

The Edge layer between the end devices and the cloud are implemented in different ways in terms of the devices which act as the intermediate edge nodes, the communication protocols and networks used by the Edge layer and also the services offered by the Edge layer. The implementation of the edge layer can be classified into three types, Mobile Edge Computing (MEC), Fog Computing (FC) and Cloudlet Computing (CC). Fog Computing presents a computing layer leveraging devices like M2M gateways and wireless routers. These are called Fog Computing Nodes (FCNs) and are used to compute and store data from end devices locally before forwarding to the Cloud. On the other hand, MEC proposes deployment of intermediate nodes with storage and processing capabilities in the base stations of cellular networks thus offering Cloud Computing capabilities inside the Radio Area Network (RAN). The Cloudlets are based on dedicated devices with capacities similar to a data center but on a lower scale present in logical vicinity to the consumers. This paradigm

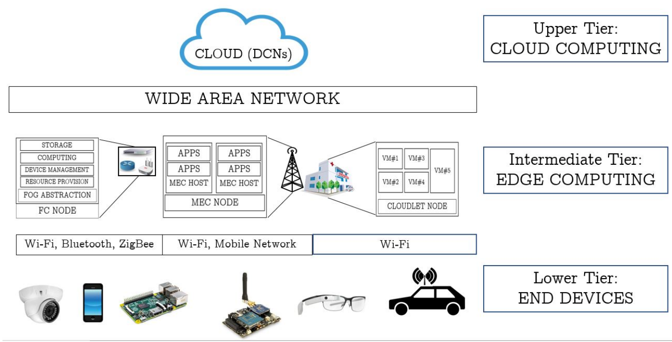

Fig. 1. N-tier Architecture for Edge Computing Paradigm.

allows end devices to offload Computing to the Cloudlet devices with resource provisioning similar to that of a data center.

Within the context of EC implementations, our contributions in this paper are manifold. We critically examine the different implementations and present a taxonomy to define and compare the features of these implementations. We propose a framework for selecting a specific implementation given a set of requirements from a use case. The framework, based on a decision tree, is devised to match the requirements of a particular application or use-case with the features offered by these implementations to make an optimal selection for the given use-case or application.

II. State-of-the-Art

The rapid growth of emerging applications like autonomous transport, smart cities, e-health monitoring among others, has pushed the case for Edge Computing as one of the major enablers for these applications. We study the related works in two parts. In the first part we discuss the related work on the architectural aspects while the second part explains several applications and use-case scenarios developed leveraging the EC paradigm.

A. Architecture and Definition

Bonomi et. al presented one of the first works on Fog Computing [1] assessing the suitability of Fog Computing for the IoT. This paper presents the requirements of emerging applications in terms of location awareness, real time interactions and need for geo-distributed end-points and how Fog Computing addresses these issues. The authors provide further insight into the suitability of FC for IoT applications with a few use-cases including a smart traffic light system and a smart wind farm in the following paper [2]. On the other hand, the authors of [3], [4] provide an overview of the Fog architecture and a comprehensive definition of the Fog nodes and their functions. The authors of [5] provide an application oriented representation of FC along with the security issues involved in these use-case scenarios.

Beck et. al presented the taxonomy and an architectural topology of Mobile Edge Computing [6] along with the purposes it would serve with respect to the requirements of emerging technologies and applications ranging from computational offloading to edge content delivery and aggregation. Surveys on Mobile Edge Computing [7], [8] illustrate the research efforts including MEC architectural and programming models. The discussions include FemtoClouds, REPLISOM and CloudAware while also highlighting open challenges in this area.

In comparison to MEC and FC, Cloudlet Computing is a newer paradigm. Satyanarayanan et. al provided the concept of virtual machine based Cloudlets as one of the first works [9] on Cloudlet Computing. Li et. al presents a study on the correlation of user mobility and how it affects computation in Cloudlets [10]. Authors of [11] propose a Cloudlet based architecture which attempts to leverage the computational abilities of devices in a LAN network by handling applications at a component level.

Existing work concerning multiple implementations of Edge Computing include that of Jararweh et. al [12] which presents a study on the integration of MEC and Cloudlets proposing a framework and performance evaluation of the combined architecture. However, in the existing literature there is a lack of studies which directly compare these three implementations under the umbrella of Edge Computing. We address this gap through our contribution in this paper.

B. Edge Computing based applications

The distributed computing philosophy has been applied to various emerging technologies and IoT applications to take advantage of different features offered by EC. Authors of [13] have developed an emergency alert service based on smart phones by computation offloading and preprocessing of data received from an emergency to Fog nodes. The work in [14] proposes a fall-detection algorithm and an implementation of the same using the computation capabilities of a network of Fog nodes. Authors of [15] have used the services of the Fog layer to implement a parking service which can be used to aggregate data collected from various Fog devices to find an optimal parking spot, thus leveraging the distributed nature of Fog nodes. Similarly authors of [16] leverage the distributed nature of Fog nodes to improve location awareness in VANETs and apply their model to a lane changing assistance use case. A Fog Computing architecture devised with a consumer-centric approach is presented in [17] with a use case scenario of connected vehicles.

On the other hand, for MEC based use cases, the developer is exposed to more information in terms of location awareness for the consumer due to the association of the MEC nodes with the Radio Access Network (RAN). Thus, proactive caching at the MEC nodes co-located with the base stations has been a major improvement for performance of websites. The authors of [18], [19] address this issue, pointing out the advantages of using proactive caching to avoid accessing the back-haul networks while also improving Quality of Experience (QoE) for the end-user. Resource virtualization is another key aspect of Mobile Edge Computing as addressed in [20], [21] enabling MEC nodes to run applications in containers offering Platform-as-a-Service (PaaS).

Satyanarayanan in his work [22] proposes use cases for Cloudlet Computing in Vehicle to Vehicle (V2V) communications stating different architectures for different use-cases. For example, in the case of an on-vehicle video game application, the Cloudlet is more suitably co-located with the device, also performing on-device processing from the sensor data while, for monitoring of road and traffic conditions an ad-hoc set of Cloudlets deployed in an area appears to be more suitable. The paper also discusses the suitability of Cloudlet Computing in the context of handling privacy and security measures as a middle-hop between end-devices and the cloud. The authors of [23] discuss the role of Cloudlets in improving crowd-sourcing applications in terms of scaling and pre-processing of massive scales of data generated from crowd-sourcing applications.

The above studies help us extract the important trends in the requirements of the emerging EC applications and we discuss these trends in the following sections.

III. COMPARISON AMONG EDGE COMPUTING IMPLEMENTATIONS

In this section, we address the gap in existing literature by presenting a detailed analysis of the implementations in the following subsections. In the first part of the section we look at the architecture of the implementations individually while in the second part we study the work flow through the different layers of the architecture while handling requests from the end users. in the final part, we compare the Edge Computing implementations and present our study.

A. EC implementation architectures

We compare the three EC implementations in terms of the architecture they follow, the function and location of their nodes serving as the intermediate layer between the end devices and the cloud, their offered services as well as their target applications in the following subsection. The N-tier architecture involving the cloud platforms, the end devices and the different implementations of Edge Computing is portrayed in Fig. 1.

1) Fog Computing: The Fog Computing implementation is a decentralized Computing infrastructure based on Fog Computing Nodes (FCNs) placed at any point of the architecture between the end devices and the cloud. The FCNs are heterogeneous in nature and thus can be based on different kinds of elements including but not limited to routers, switches, access points, IoT gateways as well as set-top boxes. The heterogeneity of FCNs paves the way for supporting devices at different protocol layers as well as support for non-IP based access technologies to communicate between the FCN and the end-device. The heterogeneity of the nodes is hidden from the end devices by exposing a uniform Fog abstraction layer which exposes a set of functions to perform resource allocation and monitoring, security and device management along with storage and compute services. These functions are utilized by the Service Orchestration Layer which receives requests from the end users and allocates resources in accordance to the requirements of the requests.

2) Mobile Edge Computing: MEC can be defined as an implementation of Edge Computing to bring computational and storage capacities to the edge of the network within the Radio Access Network to reduce latency and improve context awareness. The MEC nodes or servers are usually co-located with the Radio Network Controller or a macro base-station. The servers run multiple instances of MEC host which has the capabilities to perform computation and storage on a virtualized interface. The MEC hosts are overlooked by a Mobile Edge Orchestrator which handles information on the services offered by each host, the resources available and the network topology while also managing the Mobile Edge applications. The MEC servers offer real time information on the network itself including the load and capacity of the network while also offering information on the end devices connected to the servers including their location and networking information.

3) Cloudlet Computing: A Cloudlet can be defined as a trusted cluster of computers, well connected to the Internet, with resources available to use for nearby mobile devices [9]. A Cloudlet can be treated as ”data center in a box” running a virtual machine capable of provisioning resources to end-devices and users in real time over a WLAN network. The services are Cloudlets are provided over a one-hop access with high bandwidth, thus offering low latency for applications. The architecture proposed in [11] for Cloudlets is based on three layers, the component layer,the node layer and the Cloudlet layer. The component layer offers a set of services by providing interfaces to the higher layers overlooked by an Execution Environment. One or multiple Execution Environment(s) running on top of an OS form a Node, managed by a Node Agent. A group of co-located nodes form the Cloudlet layer managed by a Cloudlet Agent. Satyanaranan et. al proposes an architecture for cognitive assistance applications in [24], which involves a primary virtual machine which leverages cognitive functionalities offered by other virtual machines in the Cloudlet to serve a request. The data from the cognitive VMs is gathered on a user guidance VM which provides output to the end user.

B. EC implementation request handling

In this section, we study how requests performed to the different Edge Computing implementations are handled. We consider a use-case which requires offloading of processing tasks to the Edge, for example, an IoT based V2X application, where processing of data from car sensors are required. This processing of the data is offloaded to the different Edge Computing implementations.

1) Fog Computing: In the Fog Computing architecture, the Fog Orchestrator presides over the underlying Fog nodes communicating with the nodes through the functions exposed by the Fog abstraction layer. The requests from the end user arrive at the Fog Orchestrator with a set of requirements specified as policies. The required policy may include parameters like QoS, minimal Fog node configuration, load balancing among others.

TABLE I
COMPARISON OF EDGE COMPUTING IMPLEMENTATIONS

	Fog Computing	Mobile-Edge Computing	Cloudlet Computing
Node devices	Routers, Switches, Access Points, Gateways	Servers running in base stations	Data Center in a box
Node location	Varying between End Devices and Cloud	Radio Network Controller/Macro Base Station	Local/Outdoor installation
Software Architecture	Fog Abstraction Layer based	Mobile Orchestrator based	Cloudlet Agent based
Context awareness	Medium	High	Low
Proximity	One or Multiple Hops	One Hop	One Hop
Access Mechanisms	Bluetooth, Wi-Fi, Mobile Networks	Mobile Networks	Wi-Fi
Internode Communication	Supported	Partial	Partial

The Fog Orchestrator matches the policies with the services exposed by each of the nodes and returns an ordered list of nodes, in terms of suitability against the requested policy. The nodes which are most suitable are chosen based on availability.

2) Mobile Edge Computing: The MEC servers co-located with the base stations, receive the requests at the Mobile Edge Orchestrator from the end user. The orchestrator maintains a catalog of applications that are running on the underlying ME hosts and receives updates on the available resources from the ME Platform Manager. If an application is already running, the request is redirected to the application while if an application is not in a running state but is supported by the platform, the application is instantiated if resources are available and the request is accepted. Otherwise, the request is passed on to be handled in the cloud passing through the core of the network.

3) Cloudlet Computing: In case of Cloudlet Computing, the Cloudlet Agent overlooks the Cloudlets and the underlying nodes.The Cloudlet Agent communicates with the underlying components through the Node Agent and the Execution Environments. Policy violations in the components are passed on to the Cloudlet Agent from the components hierarchically. This allows the Cloudlet Agent to make an optimized choice for an underlying node when a request is received such that more complex queries are handled by nodes with higher processing capacities. The Cloudlet Agent can also provision and allocate more resources by instantiating new VMs if necessary to satisfy the received requests.

C. Comparison

Based on the features mentioned in the previous subsection, we present a comparative study of the implementations in Table I. Since the Fog Computing implementation proposes the presence of FCNs anywhere between the end devices and the cloud DCNs, Fog Computing offers more flexibility in the choice of devices for using them as FCNs. However, since FCNs leverage legacy devices by adding storage and processing to them, the computation and storage capacities are usually lesser than that of Cloudlets and Mobile Edge servers. On the other hand, due to the requirement of dedicated devices for Mobile Edge Computing and Cloudlet Computing the penetration of these implementations are slower than that of Fog Computing. However, these devices can be reused as both MEC servers as well as Cloudlets.

In terms of proximity to the edge, in case of Fog Computing, the FCN may not be the first hop access point for the end device due the leveraging of legacy devices as FCNs. For example, the first router connected to the end device may not be resourceful enough to run an FCN framework and thus, the closest FCN may be present multiple hops away. This leads to the support for inter-node communication support for Fog Computing as well. However, for Cloudlet and Mobile Edge Computing, the devices connect directly to the node over Wi-Fi and mobile network at the base station respectively. Fog Computing leverages gateways as devices for FCNs, thus offer support for non-IP based protocols like BLE and Zigbee. This allows Fog Computing to connect to a wider ranger of end devices and also offer protocol translation.

If we analyze the request handling mechanism and the services offered by each of these implementations, they follow a similar hierarchical approach, where there is a supervising entity overlooking the underlying nodes while communicating with them to gather information on the resource status and availability. However, the diversity and heterogeneity of the the Fog devices invoke the need for an abstraction layer while in the other implementations it is not necessary since dedicated devices are used as nodes. MEC, on the other hand presents an advantage of having fine grained information on the end user's location and network load for improved context awareness. For Cloudlets, the data stored and processed on the Cloudlet is in a soft state, i.e. the data is already backed up on the cloud and is updated once the processing is finished. Thus, another advantage offered by Cloudlets is that the end device can use a fresh Virtual Machine on the Cloudlet every time since the Cloudlet performs a pre-use customized resource provisioning and a post-use cleanup by backing up the processed data with the cloud

IV. USE CASE BASED DECISION MANAGEMENT

The Edge Computing paradigm offers support for a wide variety of end devices, applications and use-case scenarios. These use cases and applications have their own set of requirements and trade-offs which determine which one of the Edge Computing implementations is suitable given a use case. In this section, we define a taxonomy to evaluate a given use case and build a decision tree based on the taxonomy and the available implementations to make an optimal choice from the implementations of Edge Computing.

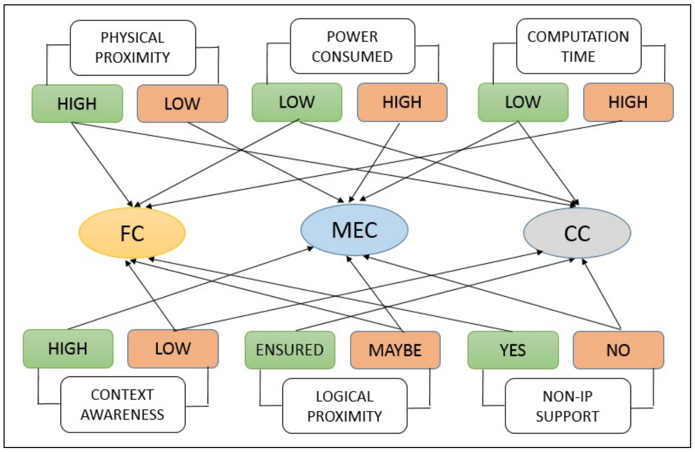

Fig. 2. Decision Tree for Edge Computing implementations.

A. Parameters for implementation selection

In this section we present a set of parameters based on which a user can compare the features and the performance of each of the implementations of the Edge Computing paradigm given a particular application or use case. We study each of these parameters comprehensively and illustrate how these parameters affect the Edge Computing paradigm.

1) Proximity: Proximity between the Edge Computing layer and the end devices can be defined in two ways. The first definition is that of logical proximity where the proximity is defined by the number of hops between the end device and the edge layer. The higher the number of hops, higher the chances of encountering queues along the path and thus chances of increased latency. Thus, the importance of logical proximity lies in the reduction of possibilities of encountering delays in the back haul network. On the other hand physical proximity is defined by the actual distance of the end device from its higher layer of computation. For example, in case of Cloud Computing, physical proximity plays a major role if the end device is located on one continent while the DC is located on another continent, delay becomes an important factor. Physical proximity might also affect the performance for Edge Computing when a single RNC handles devices over a large area, physical proximity also comes into play for MEC for real time and delay sensitive applications.

2) Access mediums: The connection to the Edge layer nodes can be established by end-devices using different mediums like Wi-Fi, Bluetooth, Zigbee, Mobile Radio Networks among others. Access mechanisms are important in more ways than one as it determines the bandwidth available to the end devices, the range of connectivity and also support for different types of devices. Fog Computing offers support for a wide array of mechanisms including Bluetooth and Zigbee which allows constrained devices with not enough memory to run the HTTP stack to connect to the FCNs and offload computation and storage. Moreover, these devices can also request for resources from the cloud through the FCNs in the edge layer. However, Cloudlets only support Wi-Fi as an access mechanism which offers high bandwidth to the end devices but lack of support for constrained devices.

3) Context awareness: Context awareness is a key parameter for applications and use cases where information about the network and surrounding devices is exposed to the Edge nodes. In this context, the MEC servers prove to be advantageous since they are placed in the RNCs, the servers receive information about the device location, load on the network and also the capacity of the network. However, since the nodes for Fog Computing are usually devices with a narrower view of the network, like routers or switches, the context awareness is lesser than that of MEC. However, the ability to communicate among the nodes themselves offers mitigates this issue to some extent. On the other hand, for Cloudlet Computing, the Cloudlets are designed to be standalone devices connected to the cloud, thus offering minimal context awareness.

4) Power consumption: The power consumption on the end devices is a major contributing factor if the end devices are resource constrained. The authors of [25] have pointed out the energy consumption with the use of LTE and radio networks is much higher than the energy consumption for WiFi. Thus, the power consumption while accessing Mobile Edge nodes is higher than that of Cloudlets. On the other hand, Fog Computing allows access to its nodes through access mediums which consume lower energy like 802.15.4 and BLE.

5) Computation time: Computation time can be defined as the time required by the Edge layer to perform the tasks assigned and responding to the end user with the desired results. In terms of computation time, MEC and Cloudlet Computing prove to be advantageous due to the virtualized nature of resources along with dynamic resource provisioning. On the other hand, since the Fog devices are usually legacy devices, the processing and storage capacities of these devices are lower, leading to a higher computation time.

B. Decision Management

In this subsection, we present a multi-entry decision tree based on the parameters defined in the previous section. The goal of this tree is to consider the desirable parameters for a particular use case or application and follow the tree to decide on a particular implementation of Edge Computing. The parameters and their possible characteristics and values are presented as entry points to the tree. The requirements of a use case needs to be presented in the form of these parameters and a choice is to be made as an entry point into the tree. Following the requirements of the use case in terms of the parameters specified, the path has to be followed to find the optimal choice. We illustrate the use of the decision tree with the following use case.

If we consider a use-case for a V2X communication application where the Vehicle offers video streaming applications streaming video from a list of videos cached at a nearby edge node. For this use case, we can classify the requirements to be logical proximity to avoid the need to access backbone networks, need for IP based access for high bandwidth, the need for physical proximity and high context awareness as well for information on the network. Since, this is a video based use case, the highest priority would be network information and load of the network. Choosing this as the starting point for the search, we have MEC as the possible outcome. On the other hand, if we set the priority higher to have proximity both physical and logical, the choice from the decision tree is Cloudlet Computing. Thus, having the requirements with the right set of priorities also plays a major role in the final decision obtained from the tree.

V. Conclusion and Future Work

Fog Computing, Cloudlet and Mobile Edge Computing share the vision of Edge Computing paradigm, however they have a different set of characteristics which sets them apart from each other, that we discussed in this paper. Even though a lot of research has been put into proposing and developing these features, they are interpreted differently by different consumers. Thus, there is a lack of standardization in terms of the actual implementation of FC, Cloudlet and MEC. This lack of standardization also affects the classification of the features of each of these implementations and thus makes the decision tree sparse. With gradual research into standardization, there would be more clarity on the features and implementation of the Edge nodes, which would facilitate in developing a denser decision tree with more choices to the end user.

Our contribution through the decision tree can be leveraged to build a recommender system for the choice of an EC implementation. The recommender system can take a particular use-case with a set of desired parameters or features as an input with corresponding priorities to recommend a particular implementation of EC to the consumer.

VI. Acknowledgments

This work has been partially funded through the French ANR Project DataTweet. EURECOM acknowledges the support of its industrial members - BMW Group, IABG, Monaco Telecom, Orange, SAP, ST Microelectronics and Symantec.

References

[1] F. Bonomi, R. Milito, J. Zhu, and 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*. ACM, 2012, pp. 13–16.

[2] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[3] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[4] E. Marín-Tordera, X. Masip-Bruin, J. G. Almiñana, A. Jukan, G. Ren, J. Zhu, and J. Farre, "What is a fog node A tutorial on current concepts towards a common definition," *CoRR*, vol. abs/1611.09193, 2016. [Online]. Available: http://arxiv.org/abs/1611.09193

[5] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1–8.

[6] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*. Citeseer, 2014.

[7] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Intelligent Systems and Control (ISCO), 2016 10th International Conference on*. IEEE, 2016, pp. 1–8.

[8] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2016 4th IEEE International Conference on*. IEEE, 2016, pp. 83–84.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, 2009.

[10] Y. Li and W. Wang, "The unheralded power of cloudlet computing in the vicinity of mobile devices," in *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE, 2013, pp. 4994–4999.

[11] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 2012, pp. 29–36.

[12] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: integrating cloudlets and mobile edge computing," in *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE, 2016, pp. 1–5.

[13] M. Aazam and E.-N. Huh, "E-hamc: Leveraging fog computing for emergency alert service," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, 2015, pp. 518–523.

[14] Y. Cao, S. Chen, P. Hou, and D. Brown, "Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2–11.

[15] O. T. T. Kim, N. D. Tri, N. H. Tran, C. S. Hong *et al.*, "A shared parking model in vehicular network using fog and cloud environment," in *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*. IEEE, 2015, pp. 321–326.

[16] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1202–1207.

[17] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric internet of things services," in *2015 International Symposium on Consumer Electronics (ISCE)*, June 2015, pp. 1–2.

[18] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[19] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data meets telcos: A proactive caching perspective," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 549–557, 2015.

[20] C. Pahl and B. Lee, "Containers and clusters for edge cloud architectures–a technology review," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, 2015, pp. 379–386.

[21] C. Pahl, S. Helmer, L. Miori, J. Sanin, and B. Lee, "A container-based edge cloud paas architecture based on raspberry pi clusters," in *Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on*. IEEE, 2016, pp. 117–124.

[22] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[23] Y. Xiao, P. Simoens, P. Pillai, K. Ha, and M. Satyanarayanan, "Lowering the barriers to large-scale mobile crowdsensing," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 9.

[24] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 2014, pp. 1–9.

[25] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 225–238.