5G.NRW Competence Center

Edge Computing

An Overview

Author

Offen im Denken

Lehrstuhl für Software Engineering

Contact

Institute for Software Engineering

Prof. Dr. Volker Gruhn

Schützenbahn 70

45127 Essen

David Schuster, M. Sc.

david.schuster@uni-due.de

Gefördert durch:

Ministerium für Wirtschaft, Industrie, Klimaschutz und Energie des Landes Nordrhein-Westfalen

1 Introduction

In 2011, the National Institute of Standards and Technology (NIST) published its definition of Cloud Computing, describing it as a model that provides ubiquitous and convenient network access to a distributed pool of configurable resources. These are, for example, networks, servers, storage, applications or services that can be dynamically scaled on demand with minimal administrative effort or interactions with the service provider [1]. A variety of cloud providers offer a wide range of services in the form of IT infrastructure, platforms and software. This is made possible by globally distributed data centers with an enormously high number of IT resources such as CPU, memory and storage. However, depending on the geographical distance between user devices and data centers, network connectivity characteristics such as latency or data rates can become critical bottlenecks. To address this challenge, Edge Computing was proposed.

Edge Computing is defined as a paradigm that brings computation, storage, and network capabilities and resources to the edge of the network close to the user's location. The structure of an edge cloud environment including user devices is shown in the Figure 1.1. By providing a layer between the user devices and the cloud infrastructure composed of so-called edge nodes, server or hosts the requests directed to the distant data centers can be significantly decreased. Since the edge nodes are close to the user devices, the latency will be reduced compared to the conventional communication between user devices and a cloud infrastructure. In addition, offloading computation-intensive tasks to the edge layer leads to save transmission energy and in turn to a decreasing energy consumption. This is an important benefit for mobile devices because of their limited battery lives. Another supported feature of Edge Computing is

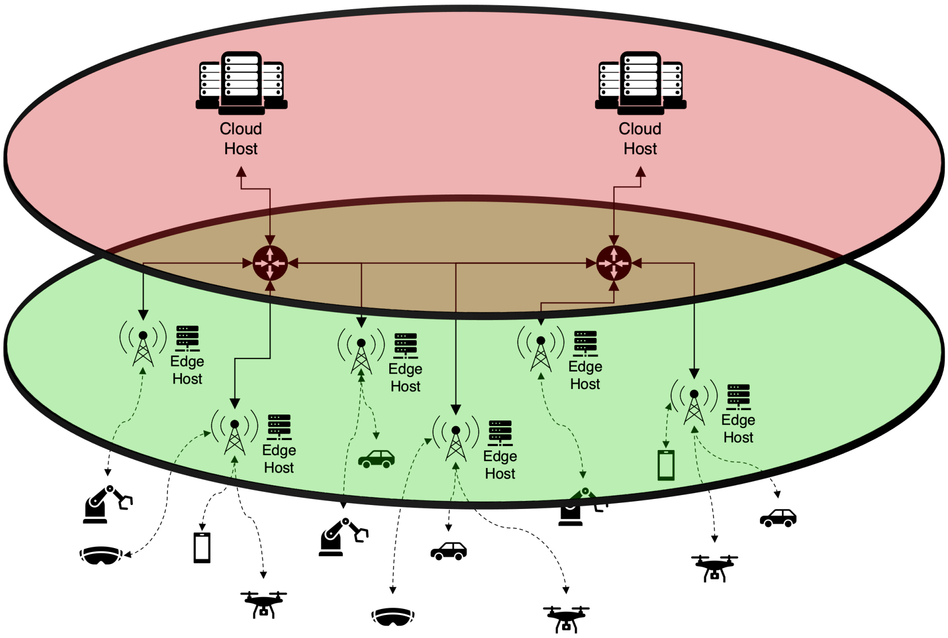

Figure 1.1: Structure of an Edge Cloud Environment

the ability of an edge node to receive information about location of user devices and the environment, which enables the provisioning of context-aware services and applications. In recent years, various Edge Computing implementations have emerged, which differ in system architecture of the edge nodes, the communication between user, edge and cloud layer, and the services offered by the edge environment [2].

2 Cloudlet

In 2009 Satyanarayanan et al. [3] introduced the concept of Cloudlets. A Cloudlet is defined as a trusted and resource-rich edge node, offering computation and storage resources using a connection with the Internet to nearby mobile user devices. Located at business premises or populated areas (e.g. coffee shop, shopping mall, airport) Cloudlets form the intermediate layer of a 3-layered hierarchic architecture additional consisting of a device layer and a cloud layer. The objective of this approach is to handle resource-intensive tasks offloaded by mobile applications. For this purpose a Cloudlet appear as a datacenter in a box providing high-performance computing resources and accessible to user devices over a single-hop wireless local area network. Cloudlets are based on Virtual Machines (VMs), which allow the instantiation and use of customized services by the user devices [3, 4].

To enable service requests by mobile user devices to a Cloudlet-based infrastructure the state of a VM has to be delivered over the network. Therefore, Satyanarayanan et al. [3] present two distinct approaches The *VM migration* approach considers the suspending of an already executing VM. Afterwards its state including information about processor, memory and storage are transferred to the destination, whereby execution is resumed from the point in which the VM was suspended. Using the *dynamic VM synthesis* approach, a mobile device transfer a VM overlay to a Cloudlet. The overlay is a derivation of a base VM, which is available on the Cloudlet. Applying the overlay on top of the base VM by the Cloudlet results in a so-called launch VM, that starts providing the requested service. At this point the user device can begin to interact with the launch VM and offload its tasks. After all tasks are completed, the state of instance is discarded and the launch VM is destroyed. In some cases, the Cloudlet creates a VM residue out of a part of the VM state, which can be transferred to the mobile device [5].

Based on the dynamic VM synthesis approach Simanta et al. [6] introduce a reference architecture for code offload by using a Cloudlet-based infrastructure, which is presented in Figure 2.1.

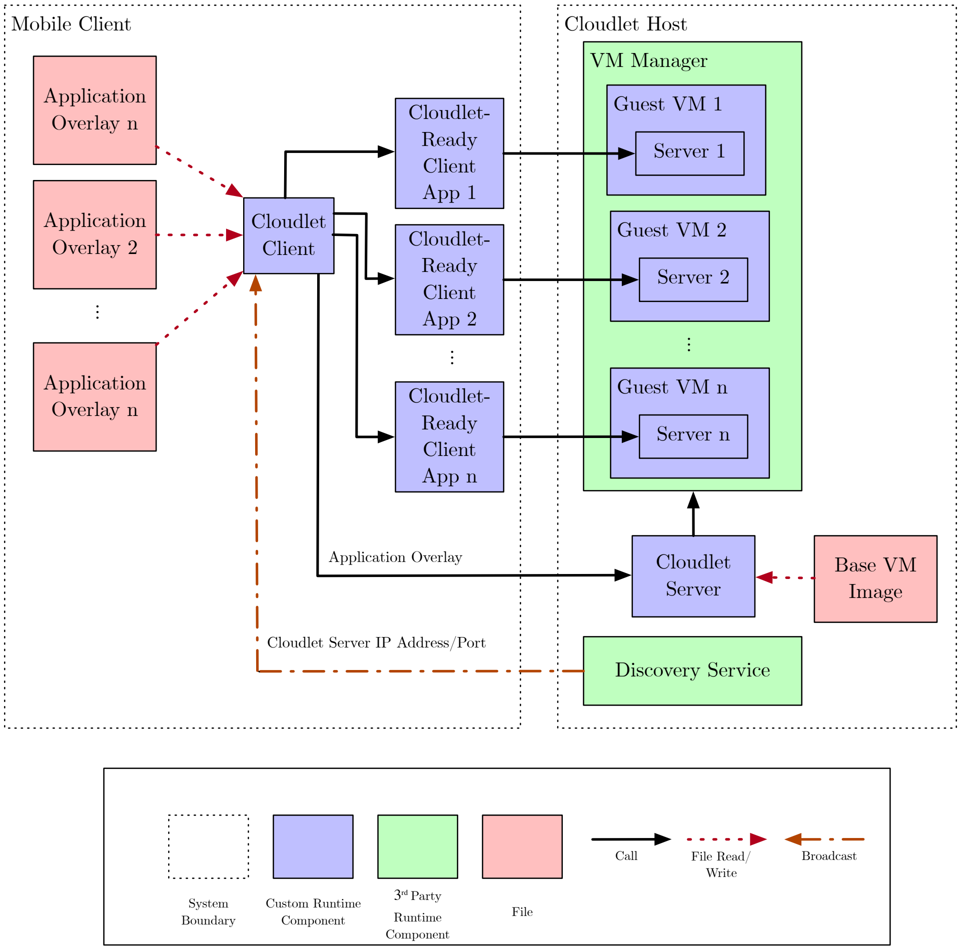

Figure 2.1: Own representation of reference architecture for Cloudlet-based system based on [6].

The architecture consists of the two main systems. The Cloudlet Host is a server located in proximity to user devices, which contains the following components:

- Discovery Service broadcasts the IP address and port of the Cloudlet to the mobile devices. Hence, they are able to find the Cloudlet, which enable communication between user devices and Cloudlet.
- Cloudlet Server receives the overlay from Cloudlet Client to perform VM synthesis. The result is a VM image, that will be used to instantiate the guest VM. On startup, the guest VM transfer its IP address and port to the Cloudlet Server and the server communicates the information to the Cloudlet Client.
- VM Manager contains all instantiated Guest VMs, which execute the computation-intensive tasks commissioned by the mobile client.
- Base VM Image includes an operating system and additionally functionalities to enable communication between the Cloudlet Server and Guest virtual machines executed on the VM Manager.

The mobile client is a device, for example a smartphone, tablet, or smart wearable, that includes the following components:

- Application Overlay is created on the basis of the Base VM Image, that is located on the Cloudlet Host. For each overlay the Mobile Client hold a corresponding Cloudlet-Ready Client App.
- Cloudlet Client discovers Cloudlet servers. In addition, it transmits the overlays to them.
- Cloudlet-Ready Client App are used by a user device. The App communicates with the server provided by the Guest VM to offload computation-intensive tasks.

In June 2015 the Open Edge Computing (OEC) Initiative has been launched [4]. The initiative is consisting of the active members Carnegie Mellon University, Crown Castle, Intel, InterDigital, Microsoft, Nokia, NTT, Seagate, Deutsche Telekom, VMWare and Vodafone. The main objective of the group is driving Cloudlet-based technologies to provide edge applications for live demonstration. In this context Wang et al. [7] present a system architecture, that enables client side workload reduction and an efficient Cloudlet resource allocation for edge-native applications. An edge-native application is a specially designed and written software, that requires typical Edge Computing properties (e.g. low latency offload or bandwidth scalability) in order to be used without restrictions [8]. The architecture consists of a client device and a Cloudlet. Both have a component, which monitors their device-specific system resources, such as battery level on client side or CPU and memory on the Cloudlet. Network information are monitored at both devices to get better estimates about the network status. The collected information is processed together with forecasting knowledge and delivered to a policy component in the Cloudlet. Depending on an external policy specification the policy component decides the allocation of Cloudlet resources for competing user devices. Possible policies definitions are for example latency requirements or priorities. Based on monitored resource consumption and expected processing demand a planner component on the client device determine a specific method to reduce the workload. This leads to a better performance of the executed native-edge application [7].

3 Fog Computing

First initiated by Cisco Systems in 2012 the concept of Fog Computing is a paradigm, that offers computation, storage and network capabilities between user devices and the datacenters of a Cloud Computing infrastructure. The decentralized location of fog architecture at the edge of the network implies several characteristics. It provides context awareness by collecting real-time information of the user devices, such as locations and environments, which can be realized through their geographical distributed computation and storage resources. Since, the Fog Computing platform is close to user devices, it supports applications with low latency requirements. The Fog Computing architecture consists of a large number of computational, networking, storage and acceleration units, so-called fog nodes, which communicate directly with the user devices to enable the provision of service and application mobility. The nodes can be heterogeneous devices with a network connection, such as routers, switches, access points, or server. This allows a deployment and use of fog nodes in a variety of locations. A main characteristic of Fog Computing is the supported interactions with the cloud. Since physical resources of fog nodes are limited, the Fog Computing paradigm provides that the nodes have interplay capabilities to the cloud. This allows offloading of computation intensive tasks, such as big data analytics to the centralized cloud, which has much more resources than a fog node [9].

To accelerate the development of Fog Computing, the Open Fog Consortium has been founded by ARM, Cisco, Dell, Intel, Microsoft and Princeton University in November 2015. The main objective of the consortium is the standardization of Fog Computing to support the development and maintenance of hardware, software and system components, which are required to create a fog-based ecosystem. In addition, they conceive test-beds to support the integration of Fog Computing in real world scenarios. Open Fog Consortium defined Fog Computing as a horizontal, system-level architecture to provide computation, control, storage and network resources and services located in proximity to the users along the continuum from cloud to user devices. Based on this specification the consortium defined a reference architecture, that is abstract description of a fog node instance. It represents a framework with the objective to support stakeholders specify their requirements to design and develop a Fog Computing environment at the edge of network. Possible stakeholders are business leaders, silicon engineers and manufacturers as well as system, software and application architects and developers [10]. The structural components including and perspectives of the Open Fog reference architecture are illustrated in Figure 3.1.

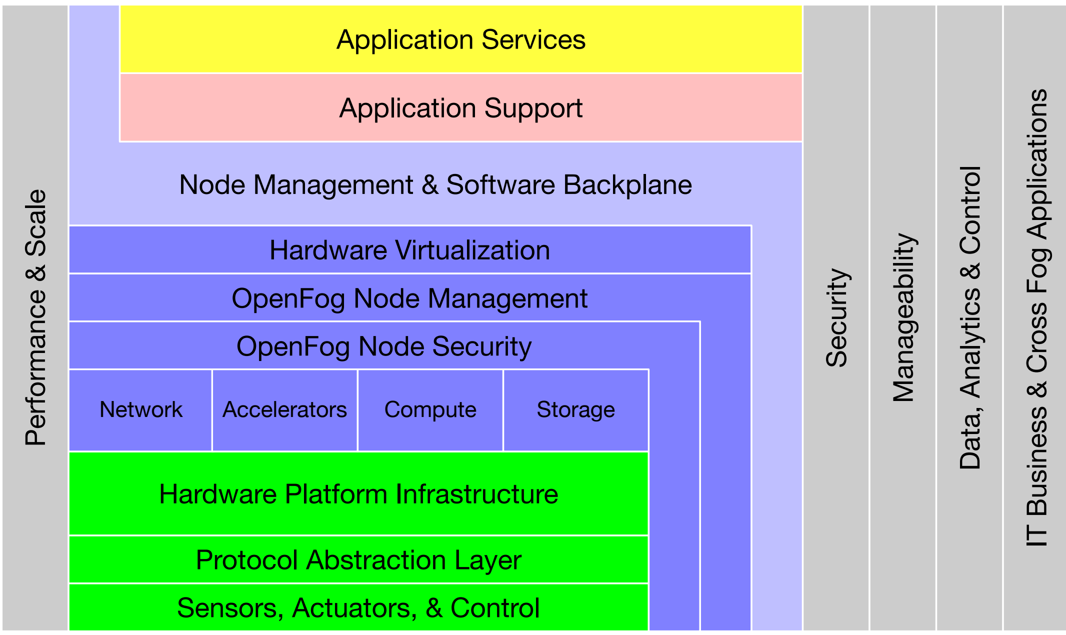

Figure 3.1: Own schematic representation of OpenFog reference architecture description with perspectives based on [10].

The reference architecture is a composite of several structural aspects, so-called views, to represent the various stakeholders in the Fog Computing continuum and consequently realize a fog-based deployment. The architecture consists of three views:

- Node view represents the lowest level view of the architecture. It includes Sensors, Actuators, and Control, which have a wired or wireless connection to a fog node to capture information about the environment, move a mechanism, or control a system. These hardware or software-based devices have no or only a few processing resources. Above these devices layer the Protocol Abstraction Layer enables the data transfer from the devices up to the higher system layers.

- System view is composed of a number of node views combined with other components to create a platform. The view contains the Hardware Platform Infrastructure including power, cooling and mechanical components, and the physical resources of the fog node, such as computation, networking or storage units. The OpenFog Node Management layer has capabilities to control, manage and configure the intern components of the fog node. In addition, the view contains an OpenFog Security Layer, which describes the security tasks for the fog node. This includes security measures for the physical resources, an efficient method for trust establishment and maintenance and a policy implementation, that realizes a secure providing of the fog node in various environments. The Hardware Virtualization allows an abstract view of the physical resources. Hence, components can be individual isolated from each other, which leads to a higher security and a dynamic provision of resources.

- Software view is deployed on top of fog node hardware layer and can be divided into three layers. The Node Management and Software Backplane accomplish the general operation and management of the fog node. Furthermore, it facilitates fog nodes communication and data transfer with other fog nodes. The Software Backplane is responsible for executing any software on the node, such as operating system, software drivers and firmware as well as communication services. The Node Management has the capabilities to keep the hardware and software of the fog node in a desired state and to ensure specific properties such as availability, resilience and performance. Above Node Management and Software Backplane layer the Application Support provides a number of software, which can be used by the applications. Possible application support software are runtime engines (e.g. Java Virtual Machines (JVMs), Node.js, .NET Framework), app or web servers, message and event bus systems, app storage (e.g. MySQL, MongoDB, Redis), or analytic tools. The Application Services Layer provides the services, which meet domain specific demands. This layer uses the infrastructure and software of the two underlying layers to satisfy specific use case requirements.

The OpenFog Reference Architecture contains five perspectives, which represent aspects and functionality that cut across the several architectural layers. The architecture includes the following perspectives:

- Performance and Scale: The provision of applications and services at the edge of the network through Fog Computing will improve the performance of the system to fulfill various Quality of Service (QoS) requirements. The performance perspective assumes that several applications or processes that guarantee a specific QoS do not restrict each other. In the reference architecture the scalability is realized by virtualization technologies. This enables a flexible and on-demand resource allocation to the applications and services.
- Security: The architecture implements measures that enable end-to-end security for all components between the cloud and the user devices. That involve the security every component of the fog node including the node hardware, fog network, and software as well all interfaces. Considered security aspects are trust, integrity, availability, privacy, authentication, authorization, and identity protec-

tion.

- Manageability: The reference architecture considers two different management interfaces. The choice for using In Band (IB) or Out of Band (OOB) interfaces or both depends on the scenario in which the fog node is deployed. IB manageability interface can communicate with the software of the system. For example IB manageability component can use periodic request to check the health of the system. In case the IB management component does not send the request, a higher level management component informs the service systems to solve the problem. Manageability subsystem with a OOB manageability interface is not executed on the host operating system. This allows management of the system independent of the systems state. A OOB manageability interface can still communicate with the host, although the fog platform is shut down and software is not running on the platform. This allows to perform for example power the fog platform on or a system health check. Additionally, the manageability perspective refers to a management lifecycle. Every system has a number of so-called management agents, which ensure that each element of the fog node pass every phase of the management lifecycle without human interaction. The individual phases of lifecycle management are commission, provision, operate, recovery and decommission.

- Data, Analytics and Control: In consequence of the huge amount of data that must be transferred and stored in cloud datacenters to process and analyze the data by business applications the traditional cloud model approach is not efficient anymore. The proximity of resources and application to the data source enables a procession of this data for local purpose-specific analytics and transfer an action back to the data source. At the same time the data or other datasets can be sent to the cloud to perform further specific processing. The hierarchical structure of Fog Computing enables partition of analytics to various layers of the network and in turn an efficient way to capture, process and analyze data.

- IT Business and Cross Fog Applications: To realize a multi-vendor fog-based environment the applications and services have to interoperate with the several layers of the fog hierarchy. This includes sharing of captured or generated data between several fog nodes and over various hierarchies.

In January 2019 the OpenFog Consortium merged with Industrial Internet Consortium (IIC). The goal of this union is to drive the development of fog-based procedures and to contribute the advancement of fog concept in the industrial internet.

4 Multi-access Edge Computing

In September 2014 the European Telecommunications Standards Institute (ETSI) Mobile Edge Computing Industry Specification Group (MEC ISG) introduced the concept of Mobile Edge Computing (MEC). They defined its objective as the provision of IT and Cloud Computing resources and services located close to the mobile users within the Radio Access Network (RAN). MEC enables mobile devices resource optimization by the deployment of applications and services, which handle compute-intensive tasks in the edge of a network. Since the communication is proceeded over the radio access network, MEC can provide context-aware services. Furthermore, MEC makes measures for preprocessing of big data sets available before transferring them to the cloud. These capabilities of MEC in combination with network characteristics, such as low latency and high bandwidth, enables a large number of new mobility and real-time application scenarios within a mobile networking environment [11].

In 2017 ETSI MEC ISG has renamed Mobile Edge Computing as Multi-access Edge Computing to reflect the requirements of non-cellular operators and to take into account heterogeneous communication technologies, e.g., 4G, 5G, and Wi-Fi. The expansion of scope allows the deployment of MEC systems located at various locations within RAN as well as colocated with other components of the network edge, for example Basestations (BSs), optical network units, Wi-Fi access points, routers, or switches. In cellular network basestations have traditionally only been used for communication purposes. The collocation of MEC systems with basestations enables the provision of resources to offer computation-intensive and latency-sensitive applications and services for the resource-limited mobile user devices. In addition, this concept reduces the amount of data, that is transferred to the core network [11]. ETSI MEC ISG defined in [12] the MEC reference architecture, as presented in Figure 4.1, which shows the high level functionality entities of a MEC system and the interfaces between the components.

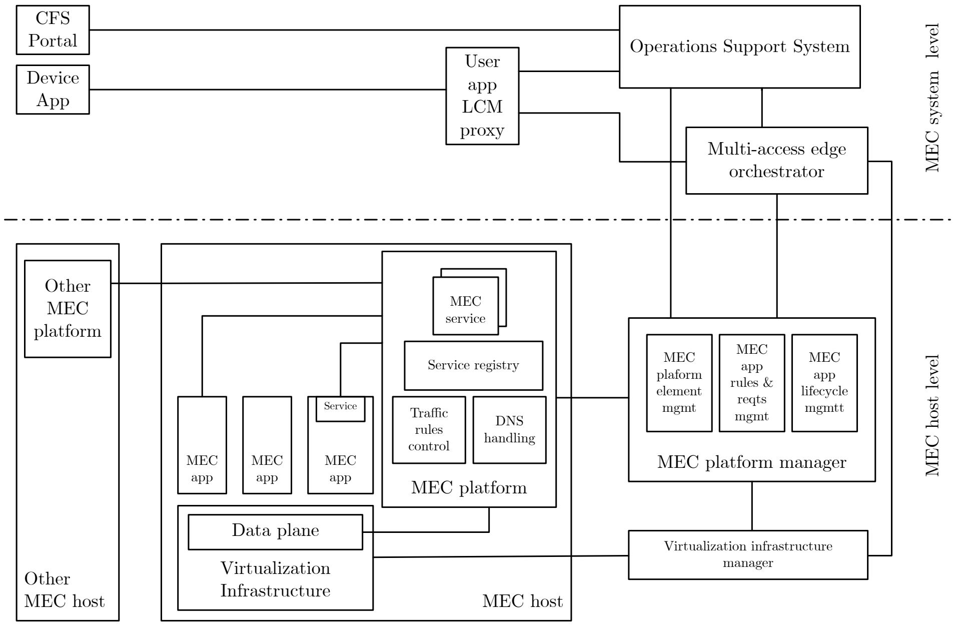

Figure 4.1: Own schematic representation of ETSI Multi-access Edge Computing system reference architecture based on [12].

The entities are grouped into two layers. The MEC host level contains the MEC host including the MEC platform and virtualization infrastructure. This infrastructure provides computation, networking and storage resources for executing MEC applications. The MEC platform receives traffic rules from the platform manager, MEC applications, or services and send them to the data plane of the virtualization infrastructure, which facilitate the traffic routing between applications, services, and other networks, such as a 3GPP, local, or external networks. Additionally, the MEC platform enables discovering, advertising and consuming of services by MEC applications. Examples of MEC services are the Radio Network Information service, which communicates data about the radio network to authorized MEC applications or the location service which transfer information about location of the user devices the applications. MEC applications are software units executed as VMs on top of a virtualization infrastructure. They can define a set of rules and requirements in terms of required resources, latency or services, which are managed by the platform manager and used to decide the MEC host selection to instantiate the application. In addition, the application's life cycle is managed by the platform manager by receiving information about performance and fault of the virtualized resources from the Virtualization infrastructure manager. Further functions of the Virtualization infrastructure manager are to allocate, release, configure, and manage the resources of the virtualization infrastructure.

The MEC orchestrator is the core component of the MEC system level. It has the overall visibility over the MEC system including the deployed MEC Hosts, physical resources and MEC services. The orchestrator maintains a catalog of available MEC applications and has capabilities to select a MEC host and to initiate the instantiation and termination of MEC applications based on a number of constraints and requirements (e.g. available resources, or services). A request to instantiate or terminate a MEC application are created by Customer facing service portal or device application. The customer facing service portal enables access for third party customers of MEC system operators to select a number of MEC applications. Device applications are software entities of user devices, that can communicate and interact with the MEC system through the user application lifecycle management proxy. The proxy accepts incoming requests from device applications and communicate with the MEC orchestrator and the Operations support system to authorize the request and initiate instantiation or termination of a MEC application.

5 Conclusion

Edge Computing provides computation, storage, and network capacity and resources to the edge of the network into the geographic proximity of the user's location. The goal is to place applications closer to the source of the request to ensure better network connectivity characteristics, such as low latency or high data rates, between the user device and the application. The implementation of Edge Computing can be divided into three different approaches, which differ in various aspects such as the basic structure of the system architecture, resource management or services offered by the edge environment [2].

The concept of Cloudlets was introduced by Satyanarayanan et al. [3] in 2009. A Cloudlet is a small-scale trusted datacenter in a box providing high-performance computational and storage resources to mobile devices in the proximity. The implementation of Cloudlets is based on VMs to handle resource-intensive tasks offloaded by mobile applications. As part of the development of Cloudlets, several concepts were introduced to enable the provisioning of VMs and the transfer of their states, leading to on-site processing of requests from mobile user devices.

Fog computing is a holistic conceptual model that also focuses on providing computational, networking and storage units between the end devices and the cloud. However, Fog Computing also considers other aspects such as security, monitoring, manageability or device control with the goal of supporting the development and maintenance of hardware, software and system components to create an fog computing based ecosystem.

Multi-access Edge Computing was introduced by ETSI MEC ISG in September 2014 with the aim of providing IT resources close to mobile users within the RAN. The focus of this edge computing concept is on the specification of a framework including the components, such as application orchestrator, platform and virtualization management, and their interaction with each other, in order to enable the on-site provisioning of applications close to the user's device.

In summary, the three Edge Computing implementations have the same goal: to provide IT resources at the edge of the network to optimize the network connectivity between user devices and the respective applications. However, they differ in aspects of infrastructure design, structure and components, access mediums, and the level of detail in the specification of functionalities.

6 References

[1] P. Mell and T. Grance. *The NIST Definition of Cloud Computing*. Gaithersburg, MD 20899-8930: National Institute of Standards and Technology (NIST), Sept. 2011. DOI: `https://doi.org/10.6028/NIST.SP.800-145`.

[2] K. Dolui and S. K. Datta. "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing". In: *2017 Global Internet of Things Summit (GIoTS)*. June 2017, pp. 1–6. DOI: `https://doi.org/10.1109/GIOTS.2017.8016213`.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. "The Case for VM-Based Cloudlets in Mobile Computing". In: *IEEE Pervasive Computing* 8.4 (Oct. 2009), pp. 14–23. ISSN: 1558-2590. DOI: `https://doi.org/10.1109/MPRV.2009.82`.

[4] Y. Ai, M. Peng, and K. Zhang. "Edge computing technologies for Internet of Things: a primer". In: *Digital Communications and Networks* 4.2 (2018), pp. 77–86. ISSN: 2352-8648. DOI: `https://doi.org/10.1016/j.dcan.2017.07.001`.

[5] A. Wolbach, J. Harkes, S. Chellappa, and M. Satyanarayanan. "Transient Customization of Mobile Computing Infrastructure". In: *Proceedings of the First Workshop on Virtualization in Mobile Computing*. MobiVirt '08. Breckenridge, Colorado: Association for Computing Machinery, 2008, pp. 37–41. ISBN: 9781605583280. DOI: `https://doi.org/10.1145/1622103.1622108`.

[6] S. Simanta, G. A. Lewis, E. Morris, K. Ha, and M. Satyanarayanan. "A Reference Architecture for Mobile Code Offload in Hostile Environments". In: *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference*

on Software Architecture. 2012, pp. 282–286. DOI: https://doi.org/10.1109/WICSA-ECSA.212.46.

[7] J. Wang, Z. Feng, S. George, R. Iyengar, P. Pillai, and M. Satyanarayanan. “Towards Scalable Edge-Native Applications”. In: *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. SEC ’19. Arlington, Virginia: Association for Computing Machinery, 2019, pp. 152–165. ISBN: 9781450367332. DOI: https://doi.org/10.1145/3318216.3363308.

[8] M. Satyanarayanan, G. Klas, M. Silva, and S. Mangiante. “The Seminal Role of Edge-Native Applications”. In: *2019 IEEE International Conference on Edge Computing (EDGE)*. July 2019, pp. 33–40. DOI: https://doi.org/10.1109/EDGE.2019.00022.

[9] 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*. Helsinki, Finland: Association for Computing Machinery, 2012, pp. 13–16. ISBN: 9781450315197. DOI: https://doi.org/10.1145/2342509.2342513.

[10] OpenFog Consortium Architecture Working Group. *OpenFog Reference Architecture for Fog Computing*. Feb. 2017. URL: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf (visited on 10/30/2023).

[11] M. Patel, Y. Hu, P. Hédé, J. Joubert, C. Thornton, B. Naughton, J. Ramos, C. Chan, V. Young, S. Tan, D. Lynch, N. Sprecher, T. Musiol, C. Manzanares, U. Rauschenbach, S. Abeta, L. Chen, K. Shimizu, A. Neal, P. Cosimini, A. Pollard, and G. Klas. *Mobile-Edge Computing*. Introductory Technical White Paper. Mobile-edge Computing (MEC) industry initiative, Sept. 2014. URL: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf (visited on 10/30/2023).

[12] Multi-access Edge Computing European Telecommunications Standards Institute Industry Specification Group. *ETSI GS MEC 003 V3.1.1 - Multi-access Edge Computing (MEC); Framework and Reference Architecture*. Mar. 2022. URL: `https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_MEC003v030101p.pdf` (visited on 10/30/2023).