MSc thesis

Master’s Programme in Computer Science

Edge cloud architectures: a survey

Jamilya Shestovskaya

July 6, 2020

FACULTY OF SCIENCE
UNIVERSITY OF HELSINKI

Supervisor(s)

Prof. J. Kangasharju

Examiner(s)

Dr. W. Wong

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Study programme Master's Programme in Computer Science

Tekijä — Författare — Author

Jamilya Shestovskaya

Työn nimi — Arbetets titel — Title

Edge cloud architectures: a survey

Ohjaajat — Handledare — Supervisors

Prof. J. Kangasharju

Työn laji — Arbetets art — Level MSc thesis	Aika — Datum — Month and year July 6, 2020	Sivumäärä — Sidoantal — Number of pages 46 pages

Tiivistelmä — Referat — Abstract

Nowadays the number of connected devices is growing sharply. Mobile phones and other IoT devices are inherent parts of everyday life and used everywhere. The amount of data generated by IoT devices and mobile phones is enormous, which causes network congestions. In turn, the usage of centralized cloud architecture increases delay and cause jitter. To address those issues the research community discussed the new trend of decentralization – edge computing. There are different edge compute architectures suggested by various researchers. Some are more popular and supported by global companies. Most of those architectures have similarities. In this research, we reviewed seven edge compute architectures. This thesis is a comparative analysis carried out by using key attributes and presentation of the Venn diagram to select the right edge compute architecture.

ACM Computing Classification System (CCS)
General and reference → Document types → Surveys and overviews
Networks → Network services → Cloud computing
Computer systems organization → Architectures → Distributed architectures → Cloud computing

Avainsanat — Nyckelord — Keywords

edge compute, cloud computing, edge compute architectures

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

Contents

1 Introduction

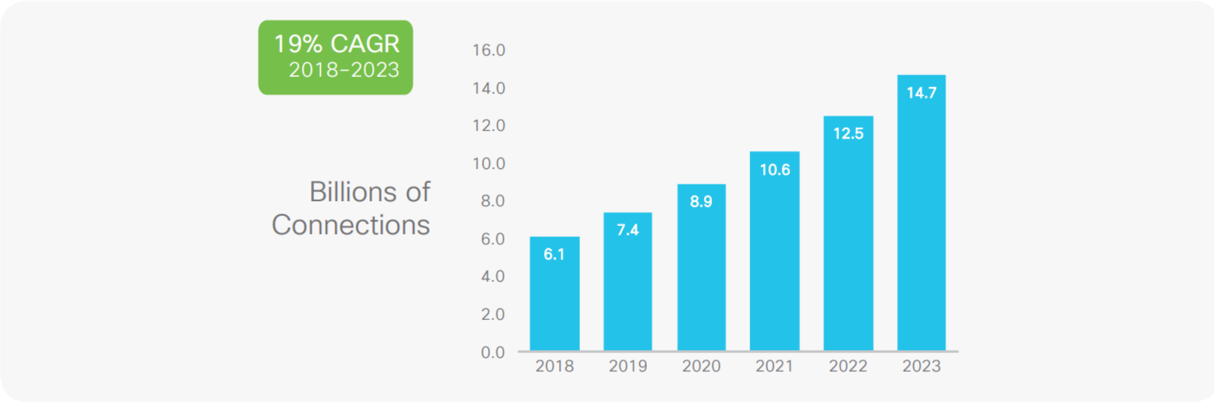
Trends in computing architectures are changing all the time. The huge mainframes were replaced by personal computers. Limited performance power of PCs caused the development of the opposite concept – cloud architecture. Companies such as Google, Amazon, Rackspace, etc. support cloud servers, which are actually huge centralized data centers facilities. Recent trend in scientific community is decentralization from clouds closer to the source of the data – to edges, without transferring the whole data to the remote cloud server. Multiple drivers push edge compute research. We can highlight key items that accelerate the development of edge technology:

- growing number of IoT devices
- applications require low latency and jitter for smoother user experience
- necessity to make quick decisions without transmission delays (bandwidth issues)
- limited performance power of the IoT devices
- possible battery savings when uploading to the edge compared to the cloud
- privacy issues while sharing the information with centralized cloud servers
- development of 5g network.

Growing number of IoT devices. People nowadays are more and more using IoT devices, such as mobile phones, smartwatches, medical sensors, multiple home sensors, kitchen appliances, etc. With the development of 5G networks, the amount of data generated by those devices and stored in the cloud is enormous. According to Cisco research [CIS18] by 2023, the number of IoT devices connected to the Internet will reach 14.7 billion. The figure below shows the predicted growth of IoT devices from 6.1 billion in 2018 to 14.7 in 2023. Moreover, data generated by all devices connected will increase significantly. Authors in [SD16] show an exponential increase in data in the latest years.

Latency, jitter, and bandwidth issues. First, the increased number of devices and data generated can cause delays and jitter in data transmission from end-users to cloud facilities. The delay could be critical for some applications such as augmented reality applications,

Figure 1.1: Global IoT growth [CIS18])



autonomous vehicles, and real-time game streaming. Edge compute concept brings computation closer to end-users, which allows saving the transmission time and saving the network traffic. Placing edge servers closer to the users allows solving delays in reply and getting smoother experience for the users. In the article [Tal+17a] authors presented two scenarios when edge technologies helped users to improve the quality of services received from IoT devices. The results of the implementation showed an improvement in quality and reply. Secondly, such a huge amount of devices connected to the internet will generate an enormous amount of data, which can cause network congestions.

Limited performance power of IoT and other mobile devices. The performance power of IoT devices is limited, so to process the data the device needs to upload it to remote facilities. Even though the performance power of edge servers by definition is lower than the performance power of cloud servers, the edge solution can still improve the quality of services provided to users.

Saving battery. Authors in [SD16] showed the device battery savings while using edge cloud compared to the usage of the classic cloud paradigm. Moreover, authors in [Shi+15] reviewed the battery life and computation time for three augmented applications: SharpLens, AReader, and Ubii. The computation time was at least 50 percent faster by using offloading to the cloudlet. The battery capacity was twice better when using offloading for Google Glass.

Privacy. Usage of the centralized cloud means sending and sharing all the data collected on the mobile device. The data may include videos from cameras, geolocation, and other private information. Users of mobile applications may prefer to process data closer on the edge servers rather than uploading all personal information to a centralized cloud. Users may prefer data processing locally due to privacy restrictions and laws.

5g network. A high-speed 5g network will push the development of edge compute. The speed over the 5g network will be faster, so it would justify sending data to nearby edge servers rather than to remote cloud.

There are different edge-cloud paradigms or architectures currently presented. Examples of edge compute architectures include but not limited to fog computing, mobile edge computing, mobile cloud computing, superfluid cloud, edge-centric computing, etc. As a basis of this work, we took the paper published by R. Rodrigo et al. [RLM18]. The publication presents a very good overview of the different cloud architectures with the explanation of key edge cloud architectural elements. In this survey, we present an overview of edge cloud architectures, compare them, and highlight similarities and differences.

This work is organized in the following order: next chapter 2 depicts several scenarios where edge compute can solve latency and jitter issues. Chapter 3 presents an edge compute paradigm. We will introduce different architecture examples of edge compute. In chapter 4 we will specify similarities and differences of different edge architectures. And finally, chapter 5 is a conclusion where we summarise obtained results and overview of possible future research in the field.

2 Edge Cloud Scenarios

In this part, we will present a few scenarios where edge compute can help to solve the issues with latency, jitter, and transition delays.

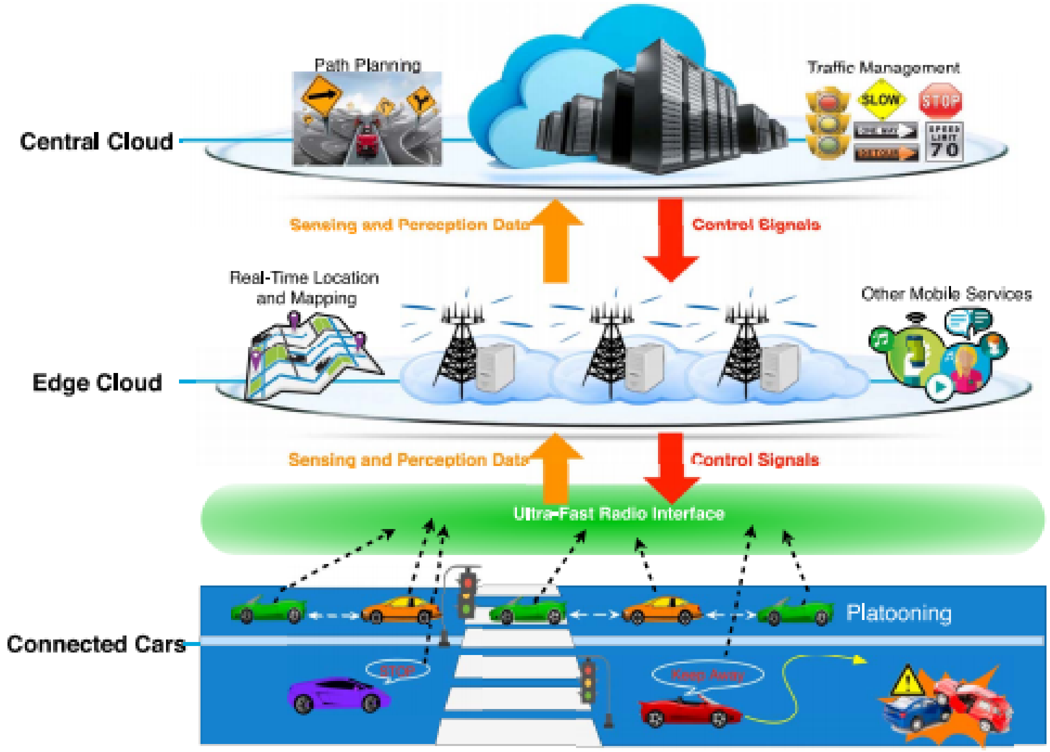
Augmented reality (AR) applications give users the possibility to observe physical reality with some augmented object placed in it. The usage of aforesaid technologies can vary significantly. Imagine a furniture store. A user of said furniture AR application can use the mobile phone and put a new piece of furniture right into the room. A customer of the furniture store then can evaluate whether the selected piece of furniture fits the interior. Another example of AR is recently launched by Google 3D animals [Goo]. People with modern mobile devices can search for animals and put them into the room (see picture 2.1). Currently, people use 3D animals just for fun, however, this technology in the future can allow users to study different things, such as animal behavior or human body anatomy. Generally, AR technology allows interacting with augmented objects by chasing different goals. As the device needs to analyze information from the camera, such as the location of objects in the room the usage of edge compute is important to process data quickly and closer to users.

Another widely discussed edge solution is **autonomous vehicles**. The autonomous vehicle concept is extensively studied by researchers and companies all over the world [Bim15]. To recognize objects and make a fast decision the vehicle computer needs to process data provided by numerous sensors quickly. The audio and video streams need to be processed and handled closer to the vehicle. Due to bandwidth constraints and sensitivity to delay, the audio and video streams cannot be fully uploaded to the cloud and processed in the cloud. The concept of automated driving is depicted in the picture 2.2 below. Multiple vehicles send the data about the traffic, obstacles, traffic jams, pedestrians, etc. to the edge server. The edge server analysis incoming information and send back the information to improve path planning. The information can also be used to control traffic lights.

Figure 2.1: 3D Google Lion standing in a room

Recently cloud gaming or **gaming-as-a-service** become ex-

Figure 2.2: edge compute for automated vehicles [Mao+17]



tremely popular. The game is running on a server. The server streams the video and audio to mobile devices. Users control the gameplay. The user interaction is sent back to the server, which synchronizes the game between multiple players. Such technology removes mobile operating system (OS) constraints and allows users to play single-player games together with friends. Naturally, cloud gaming requires a very good internet connection that is why edge compute can be the best solution to solve issues with latency, bandwidth, and jitter. The authors et al. [Cho+12] stated that mobile players start to notice the delay when the overall latency exceeds 100 ms.

$$T = t_{client} + \overbrace{t_{access} + t_{isp} + t_{transit} + t_{datacenter}}^{t_{network}} + t_{server} \tag{2.1}$$

See the formula 2.1 for better understanding the statement above. The definition of t_{client} is a time spent by the client device to send interaction, get and run the video. On the right side of the formula, t_{server} is the time spent by the server to handle the request coming from the player, create the video stream, and send the outgoing stream back to the client. For simplicity $t_{client} + t_{server}$ is estimated to be 20 ms. In such a situation the rest of the formula depicted as $t_{network}$ should be less than 80 ms to allow users to play cloud games without interruption. Thus, the network delay should be less than 80 ms. The measures taken in the research showed that implementation using Amazon Cloud (EC2) gives the delay over 80 ms by more than one-quarter of the research population. The servers placed on the network edge can improve the situation. The same article displayed user coverage increased for more than 28 percent by using servers placed on the network edges.

Another production use-case scenario is an **automated loading machine** in a warehouse. This example is similar to the autonomous vehicle above. Additionally, the machine can update stock counts according to goods received, or goods consumed. Also, such a machine can help to save labor costs for making inventorization.

In the energy sector, we presume edge compute can be useful in **wind plants**. Already multiple sensors send information about the wind, monitor the rotation of the blade, the air pressure, and other indicators. According to the article [PJ09] the typical wind turbine has rotor speed sensors, anemometers, wind vane, power measurement device, strain gauges on the tower and blades, accelerometers, position encoders, and torque transducers. The amount of information is huge and decisions should be made quickly. Moreover, such information might be sensitive from a security point of view, that is why edge compute could be an excellent solution for the developers of wind plants. Especially, when we consider offshore wind platforms. Due to the remote location of the offshore facility, the delay transmitting the data to the centralized cloud could be numerous, which make edge technology more suitable for offshore wind farm control system.

Edge compute is also important when we talk about **places of natural disasters** or places with bad connections to the cloud such as **ships at the sea or offshore platforms**. There is a stream of data and metrics that need to be collected and stored. In such cases putting the edge-server can help to collect and process metrics, aggregate them, and send them to the cloud for further analysis after the connection is restored and available again.

3 Related Work

Lately, we see a sharp increase in the number of IoT devices connected to the web. IoT devices connected generates an enormous amount of data, which in turn raise the network traffic. Unfortunately, classic cloud computing or client-server architecture cannot handle the workload. Moreover, some applications require fast response, should be free of delays and jitter. Besides, there is a necessity to pre-process the data closer to the end-user. Thus edge compute is a shift from *classic* cloud computing toward end-users. Scientific community understand different things under the definition of edge compute: cloudlets [Sat+09], edge-centric computing [Gar+15], mobile edge computing [Ins14], [Mao+17], fog computing [Bon+12], edge-fog clouds [MK16] etc. In this chapter, we will look closer to the definitions of different edge compute architectures.

3.1 Cloudlets

M. Satyanarayanan and his colleagues defined the *cloudlet* as a trusted, resource-rich computer or cluster of computers that are well-connected to the Internet and available for use by nearby mobile devices [Sat+09]. The cloudlet network should be available and connected with the high-speed wireless network within one-hop. The proximity of the cloudlet network is particularly important. Physically the cloudlet is a box that represents a cluster of internally connected multicore computers - “data center in a box”. To be widely used such cloudlet boxes should be easily managed, with little or no effort from the host. That is why the cooperation between cloudlet developers and service providers is especially critical. The cloudlet can be considered as a tier between mobile devices and real cloud, such as Amazon cloud. The concept of cloudlet is vital when access to the cloud is slow. On the other hand, if the cloudlet is not available the mobile device should have the possibility to roll back and use a traditional cloud connection until the nearest cloudlet will not be discovered and used.

Figure 3.1 below displays the definition of cloudlets as a network of self-managed devices or “data center in a box” which can be placed at a network edges and in a physical places such as coffee shops [Sat+09].

M. Satyanarayanan used the Virtual Machine (VM) technology to implement the cloudlet

Figure 3.1: What is cloudlet? [Sat+09]

concept. The VM in the initial state was deployed in the cloudlet box. When mobile users need resources they can request the cloudlet (VM) to process the data. When resources of VM are no longer required the VM is returned to the initial state. The proof of concept was implemented and the working code was published by M. Satyanarayanan later [Sat].

The initial ideas of M. Satyanarayanan were developed further. The key drawback of the "data center in a box" approach is the dependency on the service providers, who implement the infrastructure. However, T. Verbelen et al. [Ver+12] expanded the definition of cloudlet to all devices in the LAN network. At the same time, the implementation of the cloudlet is not dependant on the manufacturer of the box, but all the devices can be connected forming peer-to-peer communication models to the cloudlet and potentially share available resources to solve high computational tasks. The picture 3.2 below illustrates the view.

Fesehaye et al. [Fes+12] showed that the delay of data transfer is longer in the cloudlet network with the number of hops more than two. Thereby, the key requirement of physical proximity and availability is still valid and cloudlets usually are one-hop networks. In such peer-to-peer cloudlets, the device which requests the network resources is called the initiator. Other devices that share computational power are called cloudlet nodes. Figure 3.3 illustrates one hop (a) and multi-hop (b) mobile cloudlets.

Figure 3.2: Extended definition of the cloudlet network [Ver+12]

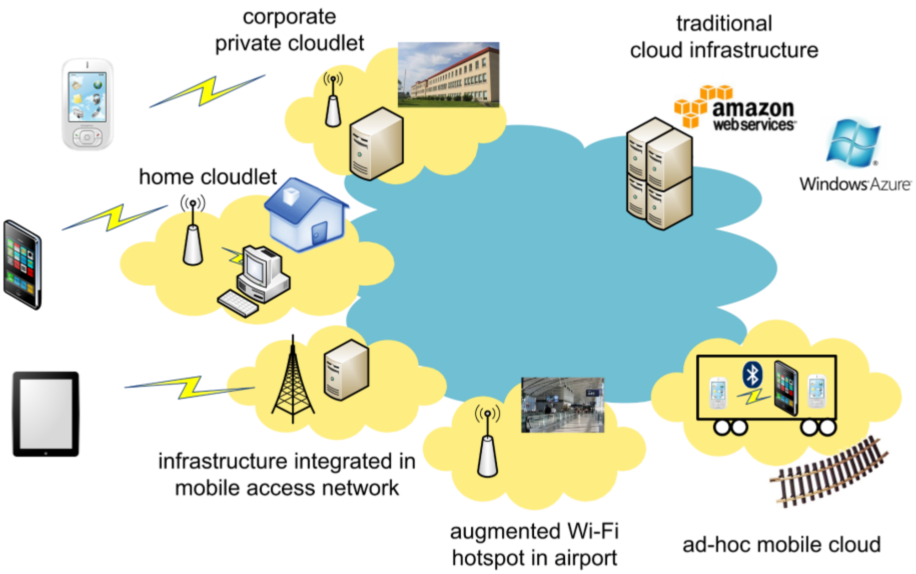
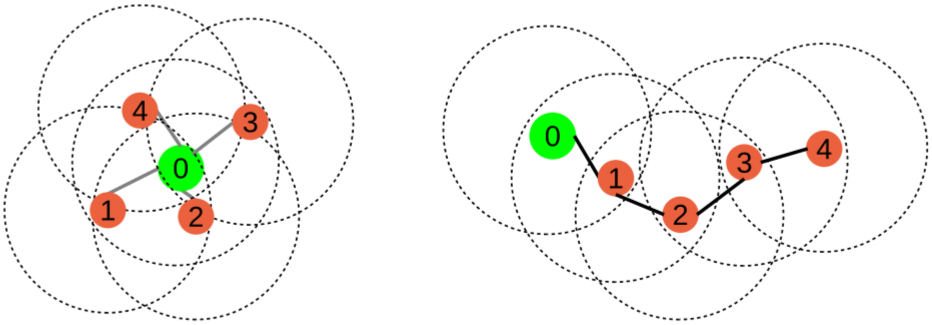


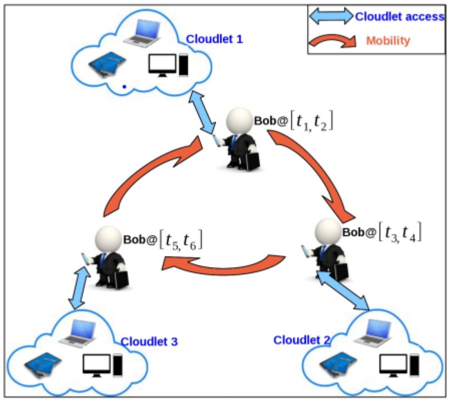
Figure 3.3: One hop and multi-hop mobile cloudlet [LW14]



(a) One-hop mobile cloudlet (b) Multi-hop mobile cloudlet

Due to nodes mobility, devices can join and leave the network. Thus the lifetime and reachable time of nodes, as well as the size of the cloudlet, is essential and defined in the article [LW14]. Li Y. et al. [LW13] reviewed the impact of the user mobility and calculated access probability between the user device and the cloudlet. The picture 3.4 below shows that mobile user Bob is using the nearest cloudlet. During time $[t_1, t_2]$ Bob is near Cloudlet 1. Bob is moving in the area and at time $[t_3, t_4]$ he is near Cloudlet 2, thus Bob is using Cloudlet's 2 resources to offload mobile tasks. At the time $[t_5, t_6]$ Bob moved from Cloudlet 2 and is using Cloudlet 3 instead. Authors defined the *cloudlet access probability* as the probability that the mobile user can find and connect to at least one cloudlet in the network at any given time [LW13]. Moreover, Li Y. proved that access probability is equal to $\mu T_C/(\mu T_I + \mu T_C)$. Where μT_C is the chance of connection time T_C when the device is connected to the cloudlet. And respectively, μT_I is a chance of interconnection time T_I when the device is not connected to the cloudlet.

Figure 3.4: User mobility in the cloudlet [LW13]

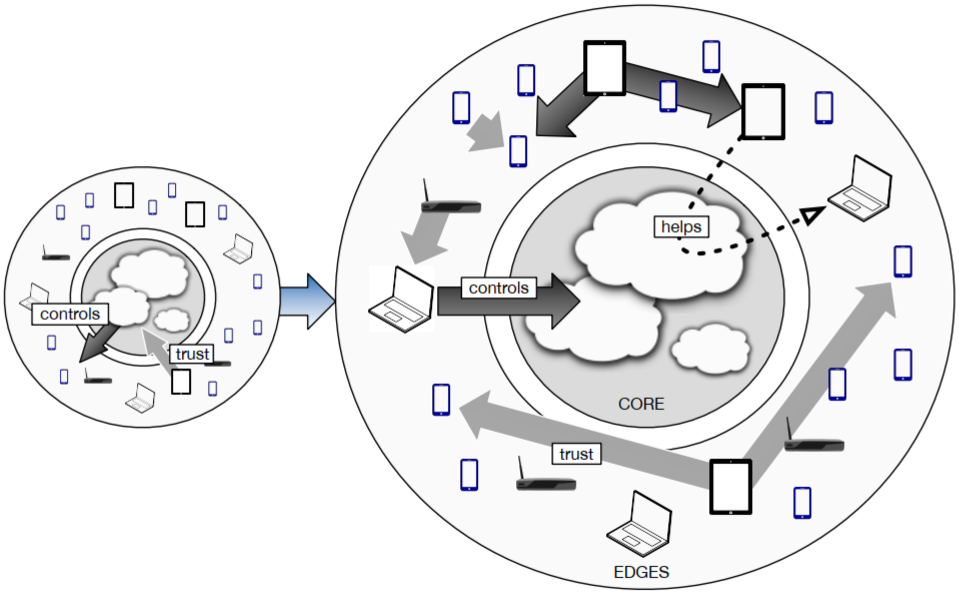


3.2 Edge-Centric Computing

The concept of edge-centric computing (ECC) was first presented by Garcia Lopez P. et al. [Gar+15]. The work was driven by the same trend from central cloud-centric models to edge-centric models. P. Garcia Lopez brought attention to key elements of ECC which are: proximity, intelligence, trust, control, and human-centric design. The ECC intends to merge heterogeneous devices such as smartphones, nano data centers, routers, and media centers to the edge-centric ecosystem. Garcia Lopez P. emphasized that humans are the source of the data, and the data need to be analyzed close to humans rather than sent to centralized cloud services. Humans should carry a key role in the data control loop. See figure 3.5 that depicts how ECC (on the right) differs from cloud approach (on the left).

The concept was further developed and Li H. et al.[LOD18] proposed edge content-centric networking (ECCN). ECCN architecture is based on the ECC framework with the implementation of content-centric networking (CCN).

Figure 3.5: Cloud computing (on the left) versus edge-centric computing (on the right) [Gar+15]

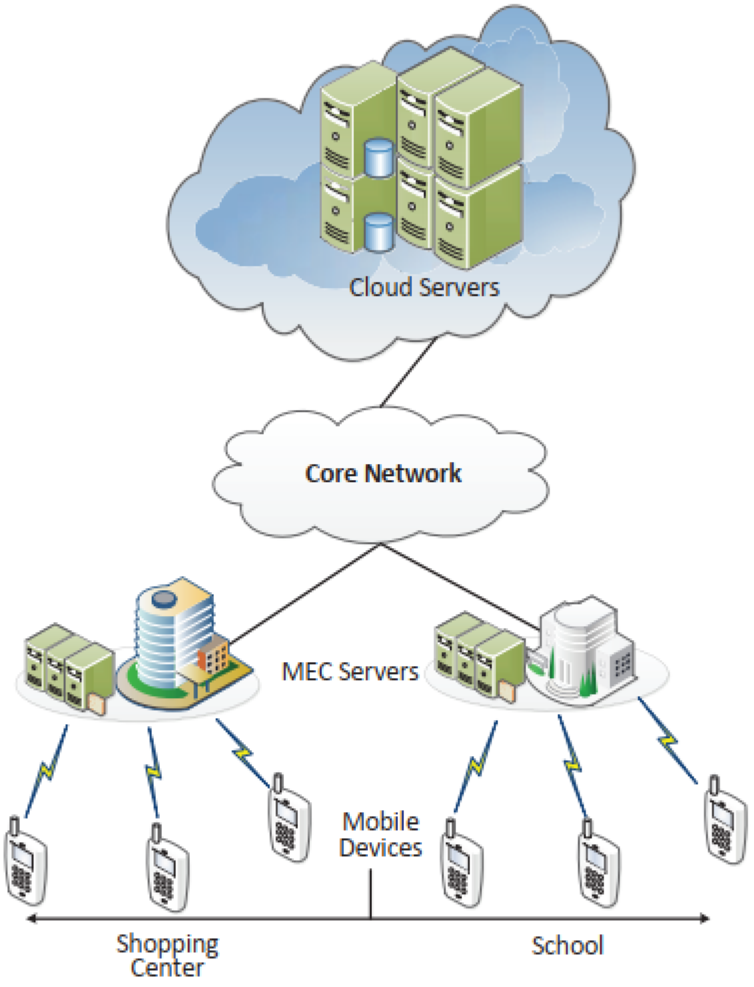


3.3 Mobile Edge Computing

European Telecommunications Standard Institute (ETSI) started the work to standardize mobile edge computing (MEC) concept in 2014. The main goal of the standardization process is to create an open environment over heterogeneous platforms on the network edges. The standardization process promotes advantages of mobile edge computing and gives different parties such as developers, mobile network operators, and other service providers the possibility to access the MEC framework. According to ETSI *"Mobile-edge compute provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers"* [Ins14]. Components of MEC architecture are mobile or IoT devices and mobile edge servers or small data centers placed by telecom operators on base stations. MEC servers are, in turn, connected to the Internet and the cloud. The mobile devices are connected to MEC servers through WAN networks. Furthermore, devices form a Device-to-Device (D2D) connection. See picture 3.6 that depicts MEC architecture components.

Following MEC characteristics presented by Ahmed A. and Ahmed E. in [AA16]: proximity, geographical distribution, low latency, location awareness, and network context

Figure 3.6: MEC architecture [AA16]



information. We will describe each characteristic in the upcoming paragraphs.

Proximity. By definition, mobile devices access MEC servers using RAN connection. Additionally, devices in the MEC network create a D2D connection. Due to the high geographical distribution of servers and speed of RAN connection the proximity of the MEC network is high. Without losing generality we can assume that devices are in one-hop distance to the MEC server.

Geagraphical Distribution. MEC servers are widely geographically distributed. This is caused by the ETSI effort to standardize the MEC concept. The availability of programming platform and high promise of the concept, make the whole idea of developing applications for MEC very encouraging.

Low Latency. Compared to the classic cloud paradigm the latency in MEC architecture is low by definition. The low latency figures are achieved due to the close placement of the MEC servers.

Location Awareness. As MEC servers are usually placed on the base station MEC application developers can use the user location information to promote better algorithms for resource offloading and mobility management. The MEC server can trace and create future mobility patterns to predict future movements of mobile users.

Network Context Information. MEC servers have network context information. The MEC servers have information about network bandwidth and congestions.

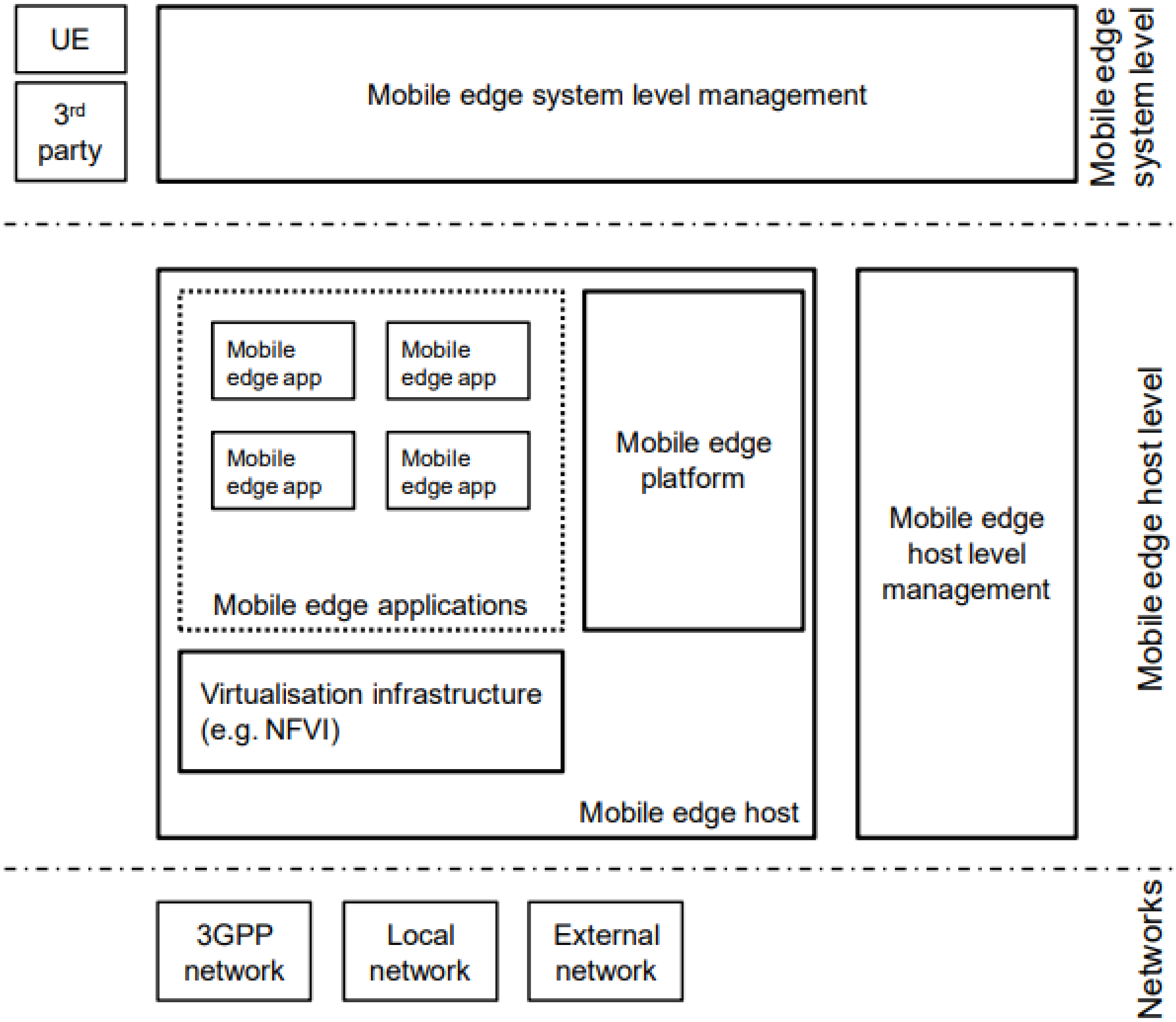
The logical structure of the MEC framework is as follows: mobile edge system level, mobile edge host level, and networks level. See figure 3.7. The main level of the MEC framework is the mobile edge host level. The level contains a mobile edge host, which supports mobile edge platform, mobile edge applications, and virtualization infrastructure. The mobile edge platform facilitates edge service discovery and allows the consumption of the edges service. The platform forms virtualization infrastructure. Additionally, the level has a mobile edge host level management agent. The top level is a mobile edge system level. This level contains user equipment and third-party devices. Additionally, the level includes system level management. The lower level is a network level. The network level supports the whole MEC framework.

The following models of MEC were presented in different research papers: computational task models, communication models, computation models of mobile devices, computation models of MEC servers [Mao+17]. From the resource management point of view, MEC architecture can be divided into single-user MEC systems, multiuser MEC systems, and MEC systems with a heterogeneous server. Each model tries to solve the problem of how to offload tasks to the MEC server and minimize the usage of mobile resources, such as battery usage.

The below applications can be implemented on the MEC platform: computation offloading, distributed content delivery and caching, web performance enhancement, IoT, and Big Data applications [Tal+17b].

Computation offloading is an approach when the end-user mobile device offloads the computation to the MEC server. Program developers use this technique to speed up the time spent to complete the task when the end-user device does not have sufficient processing power. The technique is used to save battery lifetime, save energy, and other limited

Figure 3.7: MEC framework [Ins16]



resources of the mobile device. Different research papers investigate and suggest computation offloading applications. As an example, CloneCloud et al. [Chu+11] converts single-machine execution into distributed execution over the MEC platform. Another system – Cuckoo et al. [Kem+12] is a system or a programming platform to write and manage applications that can offload tasks. Cuckoo system targets the Android platform. Developers of the Cuckoo platform implemented component-level partitioning.

Distributed content delivery and caching. The video streaming is extremely heavy. Mobile operators store the data in the database, which is usually centralized and sometimes located far away from the end-users. Delivering video streaming to a user can cause delays and jitter due to network congestions. Several research papers investigate caching on network edges as a type of extension of the content delivery network. For example, the Media Cloud framework [JWW15] introduces precaching on network edges.

Web performance enhancement makes the web surfing faster, loads the web pages by optimizing the network. The proposed systems can use cookies and web browser history to preload web content [SAD11].

IoT and Big Data applications. With a growing amount of connected IoT devices, MEC infrastructure can preprocess data requests, etc. IoT devices can be grouped to lower the amount of signaling as it is shown in [CZ16].

ETSI MEC Industry Specification Group (ISG) launched 13 Proof of Concepts topics. By the time when this thesis was done all 13 PoCs were completed [ISG].

1. Video User Experience Optimization via MEC - A Service Aware RAN PoC. The PoC team included Intel, China Mobile, and iQiYi. The idea of the case is to identify paid video streams and prioritize those paid video streams to give subscribers better experience watching videos.

2. Edge Video Orchestration and Video Clip Replay via MEC. Researched by Nokia, EE, Smart Mobile Labs. Through MEC servers users can watch video streams from professional cameras, moreover, users can switch from one camera to another, changing the angle of the video.

3. Radio aware video optimization in a fully virtualized network. Such companies as Telecom Italia, Intel UK Corporation, Eurecom, and Politecnico di Torino participated in the PoC. The information about radio conditions helps the content provider to change the video stream and improve user experience.

4. FLIPS – Flexible IP-based Services. Such companies as InterDigital, Bristol is Open, Intracom, CVTC, and Essex University participated in the research. The PoC is designed to advance the IP-based content.

5. Enterprise Services. Such companies as Saguna, Adva Optical Networking, and Bezeq International are among the PoC team. The MEC servers are used in the Enterprise networks to improve the quality of services for enterprise users.

6. Healthcare – Dynamic Hospital User, IoT, and Alert Status management. The team includes Quortus Ltd, Argela, and Turk Telecom. The research shows the use-case

for a hospital when the network is sliced to open access to different systems based on access rights.

7. Multi-Service MEC Platform for Advanced Service Delivery. Participants are Brocade, Gigaspaces, Advantech, Saguna, Vasona, and Vodafone. This PoC demonstrates the MEC ecosystem which contains not only one MEC platform but combines multiple MEC platforms sitting on common computing infrastructure. The ecosystem provides a cloud orchestration system to switch between MEC servers.

8. Video Analytics. The team of this PoC includes Nokia, Vodafone Hutchison Australia, and SeeTec. The system is analyzing raw video data from cameras in a City, catches predefined relevant incidents, and provides reports to the control center.

9. MEC platform to enable low-latency Industrial IoT. The research team is Vasona Networks, RIFT.io, Xaptum, Oberthur Technologies, Intel Corporation, and Vodafone. The PoC simulates the Industry. The idea is to speed up data transfer and real-time analytics for IoT devices and cloud-based industrial applications.

10. Service-Aware MEC Platform to Enable Bandwidth Management of RAN. The team consists of Industry Technology Research Institute, Linker Network, FarEasTone, and Advantech. The idea of the PoC is to demonstrate the current features of the ETSI MEC framework. For demonstration purposes, two applications were selected Enterprise Video Call/VoIP and Tele-Drone.

11. Communication Traffic Management for V2X. The team comprises KDDI Corporation, Saguna Networks Ltd., and Hewlett Packard Enterprise. The idea of the research is to use data generated by connected vehicles and manage road traffic congestions.

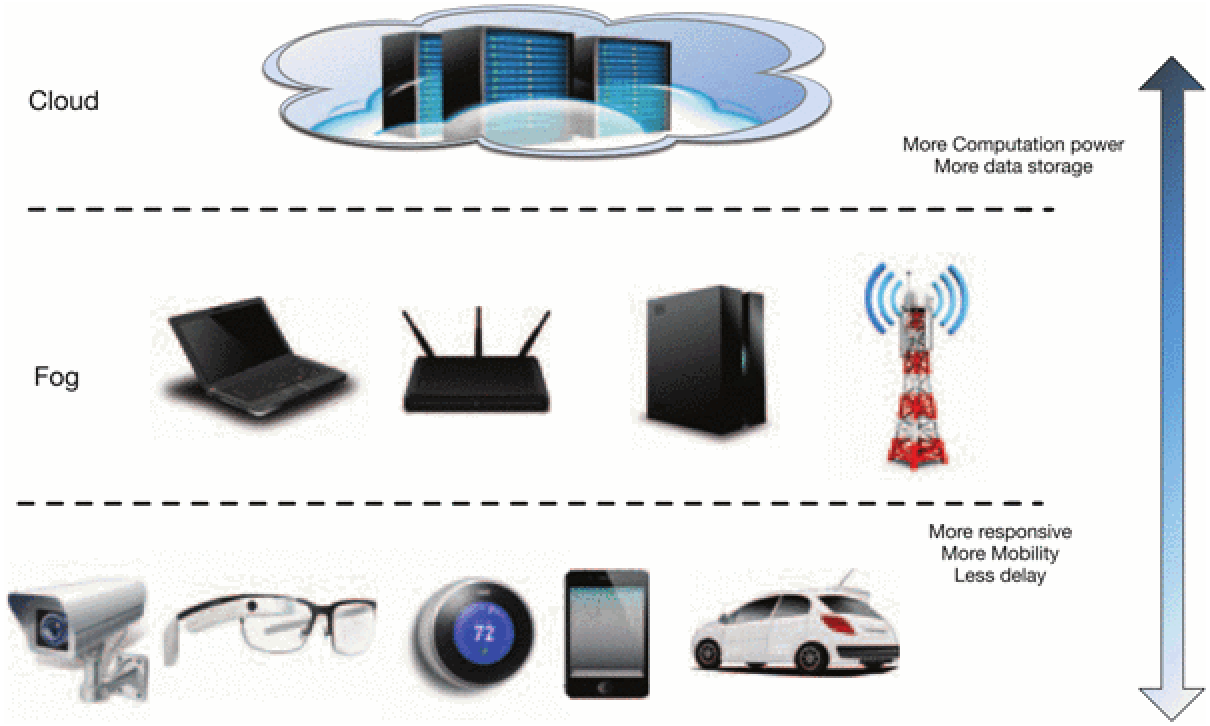
12. MEC enabled Over-The-Top (OTT) business. The researchers are China Unicom, ZTE, Intel, Tencent, Wo video, and UnitedStack. The goal is to distribute OTT content among China Unicom subscribers.

13. MEC infotainment for smart roads and city hot spots. Participants include TIM, Intel, Vivida, ISMB, and City of Turin. The idea is to show 4G/5G services for pedestrians and car drivers.

3.4 Fog Computing

The initial definition of *fog computing* was presented by Bonomi et al [Bon+12]. Fog computing is a virtualized platform that is located between end-devices and cloud computing data centers. Physically fog computing consists of heterogeneous fog nodes which can be anything: routers, switches, gateways, etc. End-users see the uniform abstract platform. The fog platform hides the heterogeneity of the fog nodes and provides a set of characteristics such as low latency, geographical distribution, support for a large number of nodes, mobility maintenance, real-time interactions, a preponderance of wireless access, interoperability and federation, online analytic support. Fog platform support non-IP communications to connect the end-devices. Functions of the fog platform are managed by Service Orchestration Layer which receives user requests and gives distributed resources. The picture 3.8 below depicts fog computing architectures.

Figure 3.8: Fog architecture [Yi+15]

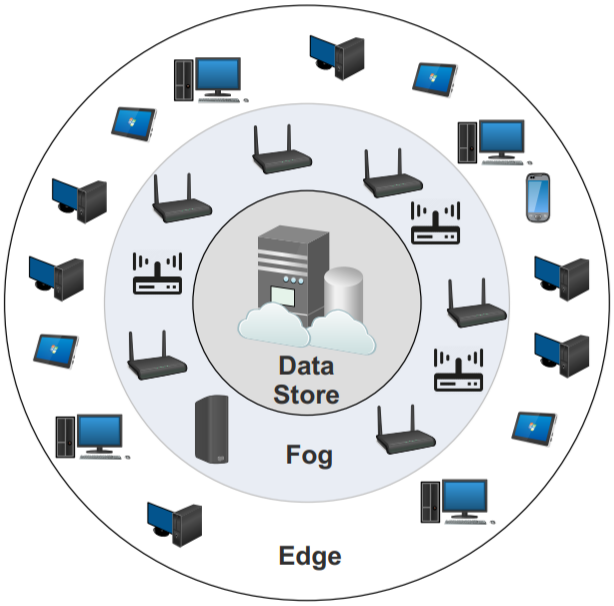


The idea of fog computing becomes so popular that Cisco presented Cisco Fog Computing solution [CIS15] as a realization of fog computing idea. Hong, K. et al [Hon+13] suggested a programming model for application development in the fog. Hong K. showed that the fog-based approach outperforms a cloud-based approach for applications such as vehicle tracking systems and traffic monitoring with a relatively short distance between devices.

3.5 Edge-Fog Cloud

Edge-fog cloud defined in [MK16]. Edge-fog cloud is a hybrid of edge-centric cloud and fog cloud models. Figure 3.9 shows the edge-fog cloud. The edge-fog cloud model consists of three different layers: edge layer, fog layer, and data store. Below we will briefly describe each layer of the edge-fog architectural model.

Figure 3.9: Edge-Fog Cloud Architecture [MK16]



Edge Layer. This layer consists of nano data centers, desktops, laptops, etc. IoT devices can connect the edge and create a one or two hops network. This condition guarantees the proximity of edge resources to the devices. The devices in the edge layer have device-to-device connectivity.

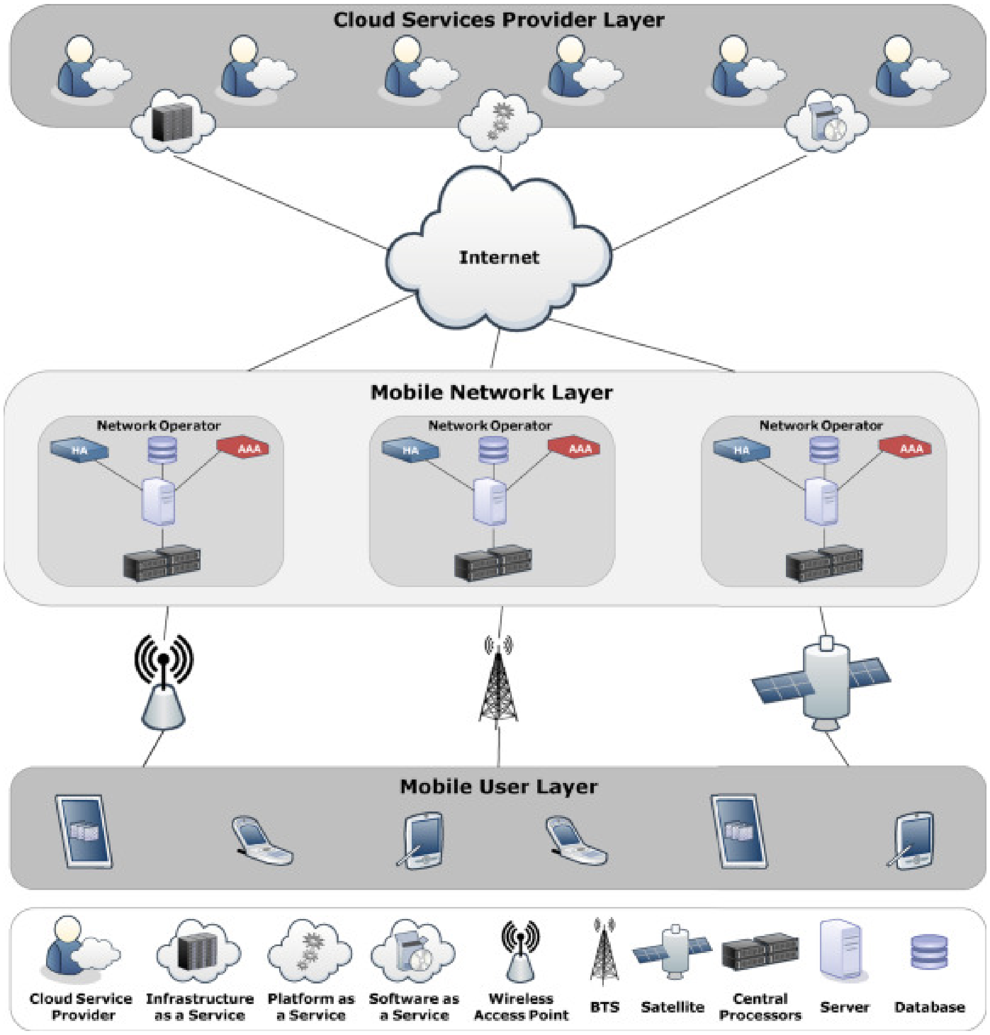
Fog Layer. The layer placed on top of the edge layer is a fog layer. The layer is physically represented by routers, switches, and other high computing resources. As the fog layer is placed closer than the central cloud, the edge layer can offload high computational tasks to the fog layer. The fog layer plays the role of middleware or backbone in the edge-fog cloud model.

Data Store. The data store layer stores the data in the cloud. Both the edge layer and fog layer can access the data store layer to read or write the data. The data store layer acts as a repository for the data.

3.6 Mobile Cloud Computing

Mobile cloud computing (MCC) was defined as "*an infrastructure where both the data storage and data processing happen outside of the mobile device*" [Din+13]. Figure 3.10 demonstrates the concept in detail. The lowest layer is *mobile user layer*. The layer consists of heterogeneous mobile devices (smartphones). The mobile user layer is connected to the next layer – mobile network layer.

Figure 3.10: Mobile Cloud Computing Architecture [Noo+18]



Mobile network layer consists of mobile operators or base stations. Mobile operators provide Authentication, Authorization, and Accounting (AAA) services to mobile users and handle mobile user requests. The service is administered by the Home Agent (HA). Mobile operators can pass mobile users' requests to the cloud. So that cloud services can be accessed by mobile users.

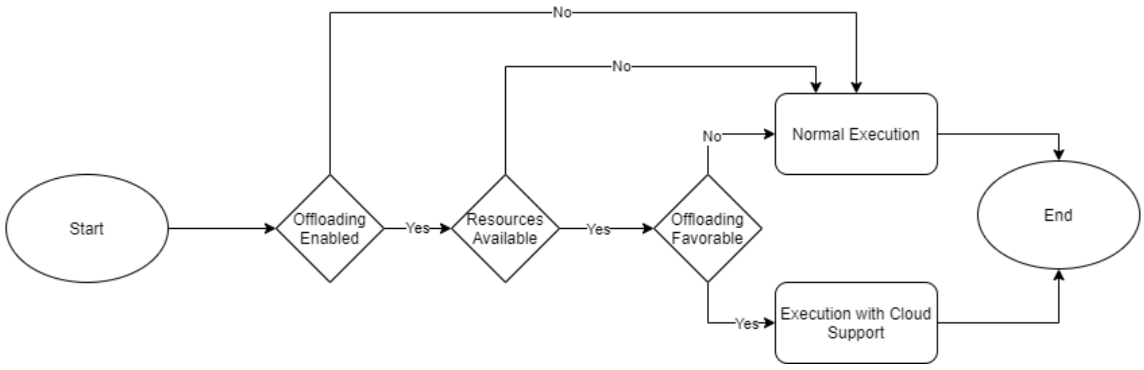
The upper layer is the *cloud service provider* layer. This layer represents cloud providers that grant access to cloud services including Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS).

Thus, mobile cloud computing is a paradigm similar to a cloud, however, MCC servers are placed closer to the user, for example on the base station. Such a closer distance allows mobile devices to offload part of the computation to the server. The main advantages of MCC development are listed in [Din+13]: extending the battery life of the mobile devices, improving data storage capacity and processing power, improving reliability, dynamic provisioning, scalability, multitenancy, and ease of integration.

However, there are several research areas which are investigated in MCC. Such issues like mobile offloading, cost-benefit analysis, mobility management, security, privacy and trust, and data management are still open for thorough research.

Offloading is not a simple task. Several research papers investigate which tasks should be offloaded to minimize battery usage. In the article [R K+14] authors described the simple process of computational offloading (see figure 3.11 below). When the application starts it checks whether the offloading is enabled on the mobile device. If enabled the application checks the availability of MCC resources. The next decision point is to check whether the offloading is favorable or not. This depends on user preferences. If yes, the application is executed in MCC. On any step, the application may instead use local mobile resources. As offloading is not so transparent, here are several entities such as user, connection, smartphone, application nature, etc. that can affect the offloading decision.

Figure 3.11: Process of computational offloading [R K+14]



It is important to analyze costs and expected benefits when deciding to use mobile cloud resources. Such approaches as cost models using resource monitoring and profiling (history-based), cost models using a stochastic method, cost models using parametric analysis were suggested in [FLR13].

MCC should support the mobility of devices. The mobile cloud should be able to determine if mobile devices are willing to join the network in a particular area. Or if the device left the location of the mobile cloud. To provide mobility management the useful information is the user current location. Location can be determined using such technologies as GPS, RFID, and IR. However, GPS is not working in buildings, and the usage of GPS requires high battery utilization, which is unacceptable for mobile devices. That is why peer-based location techniques are used to identify the location of the device relative to other devices. Another method to solve mobility issues is fault tolerance and component and proxy migration [FLR13].

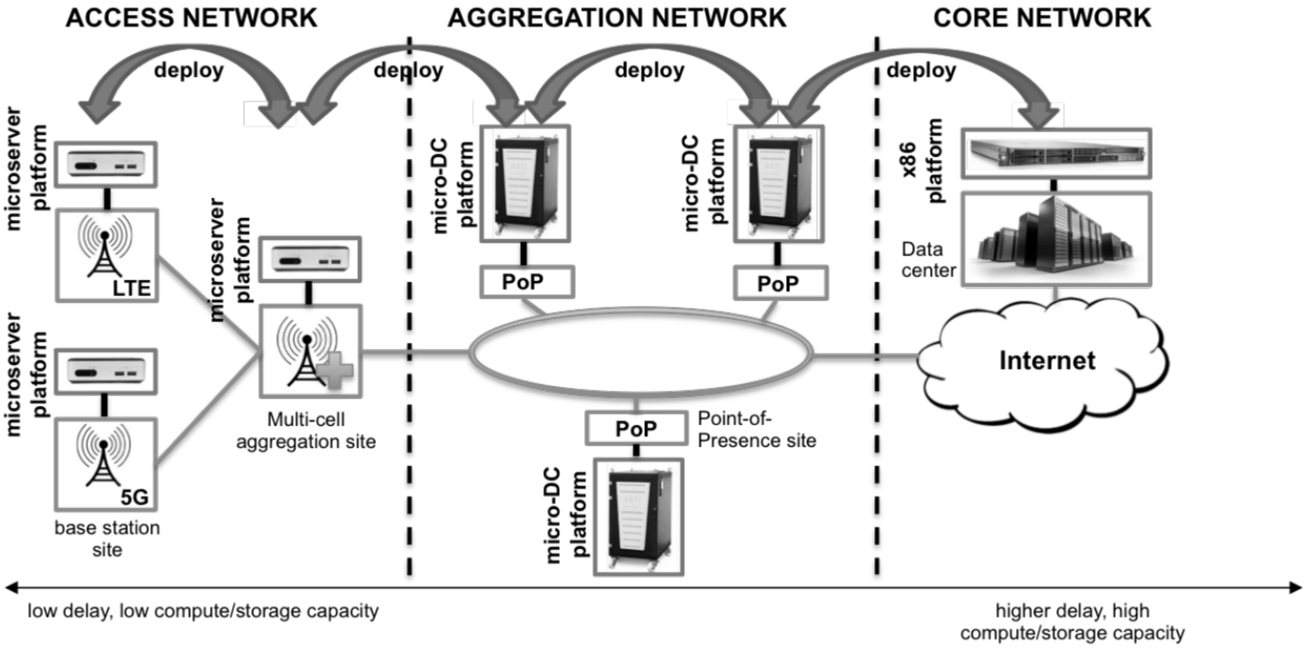
Security, privacy, and trust concerns that apply to the cloud environment are also crucial for the mobile cloud. Such issues as protecting the sensitive data offloaded to the cloud, backup and recovery, regulatory compliance, etc. need to be considered in the mobile cloud computing environment as well.

Data management in mobile cloud computing is a hard task. As data from mobile devices is potentially shared with other devices, and cloud service providers, this may increase privacy concerns. Additionally, the mobile cloud consists of heterogeneous devices, which are using different operating systems. This may raise compatibility and interoperability issues.

3.7 Superfluid Cloud

The concept of the superfluid cloud was defined in [Man+15] as *a model where multi-tenant, virtualized software-based services run on common, shared commodity hardware infrastructure deployed throughout the network.*[Man+15] The main idea is to create a virtualized cheap platform to give third parties, such as end-user and network operators the access to cloud services which are placed not on the centralized platform, but in micro data centers deployed by telecommunication operators. The architecture of the superfluid cloud is depicted in the figure 3.12. The superfluid cloud is deployed on diverse hardware such as base stations, multi-cell aggregation sites, point-of-presence sites, and even in the big data centers. Each of these nodes can provide services on-the-fly and only when those services are requested and needed. The end-user or other third parties who are using the superfluid cloud can use either low delay, but low compute/storage capacity (on the left) or high delay, but high compute/storage capacity of the cloud servers (on the right).

Figure 3.12: Superfluid Cloud Architecture [Man+15]



4 Comparison of Edge Architectures

Nowadays the interest in edge compute is tremendous. The advantages are clear for users and service providers. However, the availability of different edge compute architectures introduces complications by selection of the proper one. In this chapter, we will compare different edge architectures presented in the chapter 3. In the first section 4.1 we have selected the first set of criteria. Based on the next set of criteria we built a Venn diagram to select the appropriate edge compute architecture 4.2. In the last section 4.3 we will go through application scenarios presented in the chapter 2 and map application scenarios to the criteria selected.

4.1 First set of criteria selected for comparison

We have selected key properties below to compare different edge solutions. Those properties are broadly reviewed in the observed research papers. However, it is possible that some criteria are not explicitly defined in the papers, but there is a technical possibility to implement the feature in the edge architecture.

- latency
- architecture
- mobility support
- availability
- scalability
- ownership
- hardware
- security
- privacy
- distance to users.

Similarities

Latency. We define latency as a roundtrip time spent by the packet to travel from the destination to the server. Latency is a key property that drives the development of edge architecture solutions. Latency is low for all observed edge compute architectures if we compare them with the classic cloud computing. Low values of latency dictated by the shorter distance from the end-devices to the cloud.

Architecture. All proposed edge architectures are n-tier decentralized and distributed. This is due to the distributed nature of IoT devices and willingness to place servers or edge nodes as close to end-users as possible.

Mobility support. The mobile devices forming edge cloud are continuously connecting and leaving the network. That is why mobility support is embedded in all proposed architectures and is high for all of the edge paradigms.

Availability is a probability that the system is up and running at any selected time. Availability is integrated in all reviewed edge compute solutions as the architecture is distributed and have algorithms to ensure high availability of distributed architecture.

Scalability is a possibility to extend the operation by adding additional resources if it is required. The system is considered to be scalable if resources can be added to meet growing demand without the loss of availability. Scalability is another default characteristic of edge cloud. Scalability is high for all observed edge compute technologies.

Differences

Ownership is the rights and control over the deployed edge compute equipment. It defines which entity is deploying the solution and will be maintaining the software and hardware. Because of the nature of devices forming the edge cloud the ownership can belong to private entities and individuals as for cloudlets and reside with telco companies as it is for mobile edge computing.

Hardware represents the actual difference between the presented architectures. Such edge architectures as edge cloud computing, the superfluid cloud consists of mobile or IoT devices connected together to share the resources. Cloudlet hardware is a resource-rich computer or data center in a box. The fog cloud hardware is routers, switches, access points and gateways, while the mobile edge computing is servers usually co-located on the base stations. The mobile cloud computing concept is very similar to a centralized cloud, where the hardware is represented by big servers.

Security. A centralized cloud storage facility may be more secured, because centralized cloud facilities are maintained by high skilled information security employees, have a bigger budget for security measures, more rigorous prevention, and more thorough detection of the security threats. At the same time, a centralized cloud is more attractive for intruders as it stores a bigger amount of data. Edge architectures as cloudlets, edge-centric computing, and other edge architectures owned by small private entities and individuals are more susceptible to security attacks due to human factors. However, we have to emphasize, that edge architectures managed by telecommunication operators as, for example, MCC are more secure.

Privacy. IoT sensors and mobile devices can collect private data such as personal information, location, track user movements, information about the house, vehicle, etc. Sending private information over the network to a centralized cloud can be undesirable. In such cases, the user and edge cloud architects may prefer to keep the data on the network edges and not share the private information with the centralized cloud service providers. An alternative solution could be sharing anonymized or partial data.

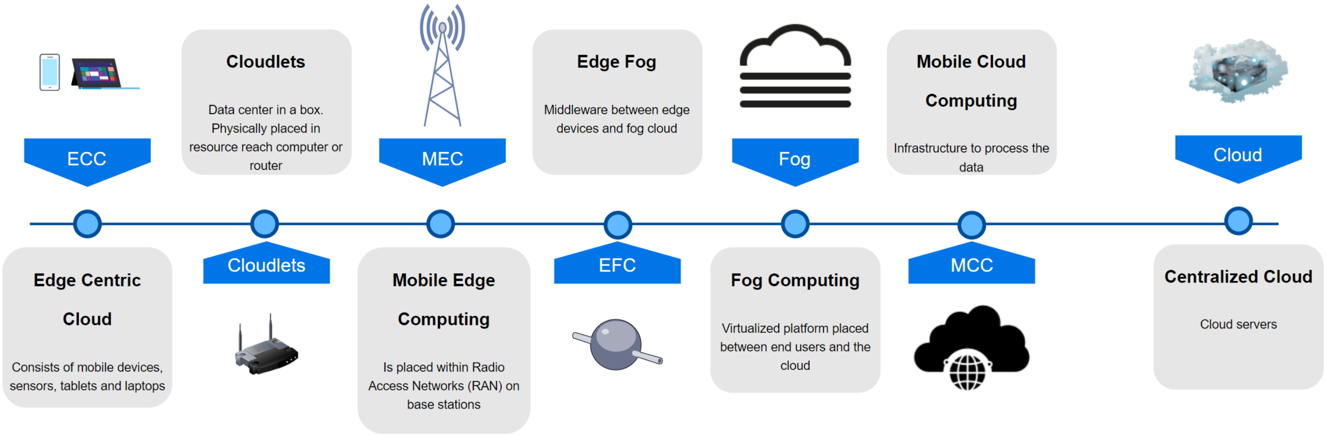
Table 4.1 summarises the criteria described above for different edge compute architectures.

Table 4.1: Summary of first set of criteria

	Cloudlets	**ECC**	**MEC**	**Fog**	**Edge-Fog**	**MCC**	**Superfluid**
Latency	Low						
Architecture	Decentralized and distributed						
Mobility support	Yes						
Availability	High						
Scalability	High						
Ownership	Private entities, individuals	Individuals	Telco companies	Private entities, individuals	Private entities, individuals	Telco companies	Individuals
Hardware	Data center in a box	Connected mobile devices	Servers running on base stations	Routers, Switches, Access Points, Gateways	Hetergeneous servers	Servers and user devices	Connected mobile devices or connected servers
Security	Less secure		More secure	Less secure		More secure	Less secure
Privacy	The data is stored closer to user					Shares the data with centralized cloud	Users may share the data with centralized cloud

We reviewed the distance to the user as other characteristic that differentiate edge architectures presented in this work. The picture 4.1 depicts the placement of different cloud architectures from the closest to the user (on the left) to the furthest (on the right). The ECC is closest to end-user architecture as it consists of connected user devices to form the edge cloud. On the other hand, MCC is the farthest architecture because it consists of heterogeneous network servers. The distance between end-devices and edge network nodes can be called physical proximity. We will use this term in the next section 4.2 to introduce the diagram for edge architecture selection.

Figure 4.1: Distance to different edge compute architectures: from closest to farest



4.2 Second set of criteria: Venn diagram

Dolui K. et al. [DD17] presented a decision tree and compared three edge solutions: cloudlets, fog computing, and mobile edge computing. The authors took the following parameters to create a multi-entry decision tree: proximity, the power consumed, context awareness, and computational time. We took a similar approach and created a diagram for seven edge architectures presented in the chapter 3 above (see figure 4.2).

Table 4.2 summarises the second set of criteria for the edge compute architectures. We will describe each of the selected parameters below.

Physical proximity is an actual distance from the end-devices to the edge node. Whereas, the *logical proximity* is a number of hops from the device to the nearest edge node. Both definitions have a significant impact on latency as many hopes and higher distances can cause congestions. Physical proximity is high for cloudlets, edge-centric computing, and fog computing. For mobile-edge computing, edge-fog cloud, mobile cloud computing, and superfluid cloud physical proximity is low. Logical proximity is ensured for cloudlets. We can also mention that in other edge compute architectures logical proximity is possible, but not always ensured.

Table 4.2: Summary of the second set of criteria

	Cloudlets	**ECC**	**MEC**	**Fog**	**Edge-Fog**	**MCC**	**Superfluid**
Physical proximity	High	High	Low	High	Low	Low	Low
Logical proximity	Ensured	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe
Power consumed	Low	High	High	Low	Low	High	High
Computation time	High	High	Low	High	High	Low	High
Context awareness	Low	Low	High	Low	Low	High	Low
Non-IP support	No	No	No	Yes	Yes	No	No

The *power consumption* is how quickly the device is using the battery. When an IoT device is searching for a network or tries to establish a geolocation position the power consumption is higher. Thus, power consumption depends on the network technology used to establish a connection between a user device and an edge node. For example, fog computing and edge fog cloud support such technologies as WiFi, LTE, ZigBee, Bluetooth Smart, etc. In the cloudlet, architecture devices are connected through WiFi technology. Authors in [Hua+12] showed that devices consume more energy when using LTE networks. Based on this we conclude that power consumption is lower for architecture decisions that are connected through WiFi networks.

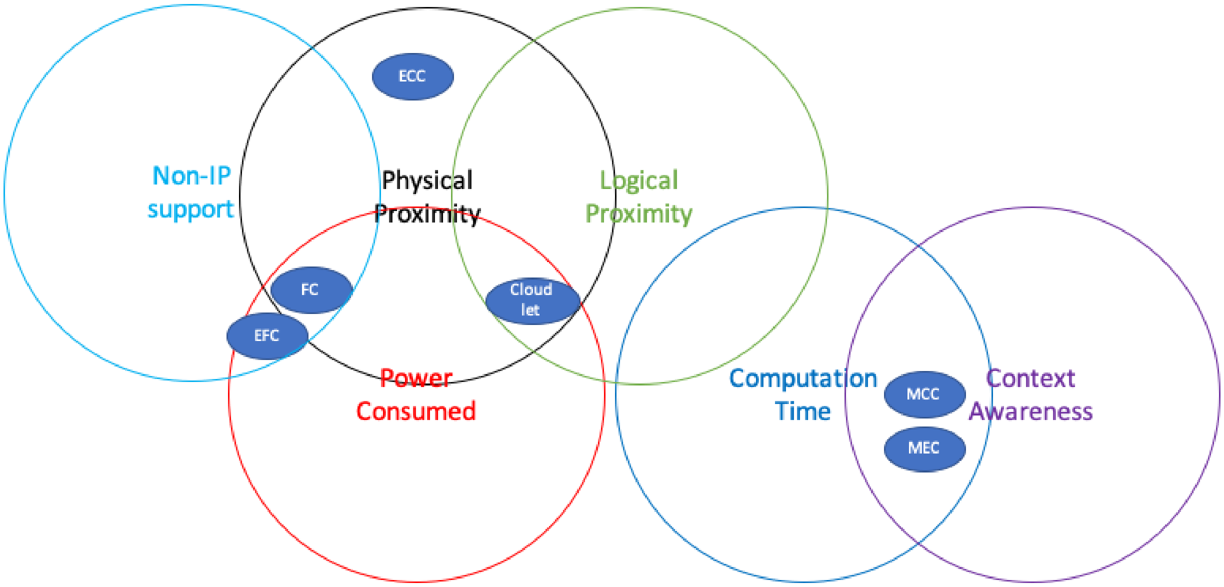
We define *computation time* as time spent to complete the task requested by the end-device. Computation time is another important feature. The parameter is lower for mobile-edge computing, and mobile cloud computing architectures due to highly virtualized resources. On the other hand, computation time is higher for cloudlets, edge cloud computing, fog computing, edge-fog cloud, and superfluid cloud due to the limited power of the devices forming the cloud.

Context awareness is knowledge about the surrounding environment. The knowledge allows edge compute servers to adapt accordingly and improve the quality of services. For example, the MEC server uses a radio access network and can re-route traffic to avoid congestions. The same applies to mobile cloud computing. Cloudlets, fog compute, edge-fog cloud, and edge-centric computing do not have context-awareness property.

Non-IP support. This characteristic describes the ability of the network to support the HTTP stack. Due to the heterogeneity of fog devices and virtualized platform, the fog cloud supports non-IP protocols. The same is applicable for edge-fog cloud as the edge-fog cloud is built on the fog cloud architecture. For other presented architectures non-IP support is not explicitly defined in the reviewed papers.

Based on the results summarised in the table 4.2 we decided to present more visualised diagram to show parameters above. See the picture 4.2 for more details.

Figure 4.2: Selection of edge architecture: venn diagram



4.3 Mapping of application scenarios to the criteria reviewed

For the research purposes we will re-visit application scenarios listed in the chapter 2. We will compare each application using the criteria defined in sections 4.1 and 4.2. We have selected grading scale from below to measure the importance of each criteria:

1. not important
2. slightly important
3. moderately important
4. important
5. critical

Summary of the application scenario evaluation presented in the table 4.3 below. Latency is critical for application scenarios where decisions should be made quickly otherwise the delay will be dangerous for human life: AR, autonomous vehicles, gaming-as-a-service, automated loading machines. Latency is also important for wind plants. Such application scenarios as a place of natural disaster, ship at the sea or offshore platform are more tolerant to delay. We have evaluated them as important, moderately important, and slightly important respectively.

In terms of architecture, we assessed application scenarios as NOT important when the application can use a centralized cloud architecture without losing the key functionality. For example for such application scenarios as augmented reality, gaming-as-a-service, wind plants the decentralized architecture is moderately important. The decentralized architecture is important for automated loading machines, ships at the sea/offshore platform scenarios. Whereas, for AR, and place of natural disaster, the existence of a decentralized edge node is critical.

Mobility support is not important for wind plants, because wind plants are static and do not need to move from one edge network to another. Similarly, mobility support is slightly important for a ship at the sea/offshore platform scenario. At the same time mobility support is critical for autonomous vehicles, automated loading machines, and the place of a natural disaster. This is due to the moving nature of devices forming edge cloud.

Network availability is critical for autonomous vehicles, automated loading machines, wind plants, and the place of a natural disaster. The loss of the network can cause a risk to human life.

The scalability criterion is critical for autonomous vehicles, and gaming-as-a-service. Potentially such applications should support a fast-growing number of end-devices connected to the edge cloud. As an example, the number of users come and go following the daily routine patterns the applications should be scalable and provide resources on a need basis. In addition, during the morning commute increase load would be applied to autonomous vehicle edge services. During the evening leisure period, people tend to play games more often, hence the increased load on gaming-as-a-service services. To reduce the overall usage bill and be more efficient, having a highly scalable system is a must in those scenarios.

In terms of ownership of the edge network, we think it is important for the government to control automated vehicles, in other cases, the control can be given to telecom operators or even to private entities.

We believe that due to the heavy impact on human lives for autonomous vehicles, automated loading machines, and places of natural disaster hardware is critical. Hardware should be highly redundant, compact, and placed on the device that forms the edge node. In other cases, the hardware is not so important.

We divided security and privacy to two different criteria. That is why from the security point of view the implementation of AR is less critical. Whereas for such scenarios as autonomous vehicles, automated loading machines, wind plants, place of natural disaster security issues are critical as it can impact human lives.

In terms of privacy, such applications as autonomous vehicles have previously visited addresses, home addresses, and other critical personal data. AR applications can have personal data of users, credit cards, etc. The same applies to gaming-as-a-service. That is why we have evaluated those application scenarios as critical.

Distance to user, logical, and physical proximity are closely connected to latency. That is why we have evaluated presented application scenarios in the same way.

Consumed power is less important for AR applications because end-users usually can connect AR devices such as AR helmet to an electrical socket. On the other hand in place of natural disasters, it is critical to save the battery of the device and keep the device working, as electricity may not be available.

Computation time is critical for application scenarios not tolerant of the delays. Also, we evaluate computation time as critical in the place of a natural disaster.

Context-awareness is critical for autonomous vehicles, gaming-as-a-service, automated loading machines, and a place of the natural disaster. As it is very important to have information about other devices in the network to predict the movement and create patterns if necessary.

The support of non-IP protocols is critical for autonomous vehicles, automated loading machines, and places of the natural disaster. As it allows to create device-to-device connections and separate protocols to support edge networks.

We used the grading scale presented in the table 4.3 to assess which application is winning for each scenario (*column weigth*). Additionally, we assigned 1 for each criterion if criteria are presented in the edge architecture and 0 otherwise. To get the winning edge architecture we multiplied scale to the criteria. $total\_weight = weight * criteria$.

See tables 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 below which summarize approach for each application scenario.

Table 4.3: Table: evaluation of application scenarios

	Augmented Reality	**Autonomous vehicles**	**Gaming as a service**	**Automated loading machine**	**Wind Plants**	**Place of natural disaster**	**Ship at the sea or offshore platform**
Latency	Critical	Critical	Critical	Critical	Important	Moderately Important	Slightly Important
Architecture	Moderately Important	Critical	Moderately Important	Important	Moderately Important	Critical	Important
Mobility support	Important	Critical	Moderately Important	Critical	Not Important	Critical	Slightly Important
Availability	Not Important	Critical	Not Important	Critical	Critical	Critical	Moderately Important
Scalability	Important	Critical	Critical	Moderately Important	Not Important	Moderately Important	Not Important
Ownership	Private entities	Government	Private entities	Private entities	Telecom. Operators	Telecom. Operators	Telecom. Operators
Hardware	Slightly Important	Critical	Slightly Important	Critical	Slightly Important	Critical	Important
Security	Slightly Important	Critical	Moderately Important	Critical	Critical	Important	Important
Privacy	Critical	Critical	Critical	Moderately Important	Slightly Important	Moderately Important	Moderately Important
Distance to user	Important	Critical	Critical	Critical	Slightly Important	Moderately Important	Moderately Important
Physical proximity	Critical	Critical	Critical	Critical	Important	Moderately Important	Slightly Important
Logical proximity	Critical	Critical	Critical	Critical	Important	Moderately Important	Slightly Important
Power consumed	Slightly Important	Important	Important	Important	Not Important	Critical	Moderately Important
Comp. time	Important	Critical	Critical	Critical	Slightly Important	Critical	Slightly Important
Context awareness	Slightly Important	Critical	Critical	Critical	Slightly Important	Critical	Not Important
Non-IP support	Not Important	Critical	Not Important	Critical	Not Important	Critical	Slightly Important

Table 4.4: Augmented Reality: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Architecture	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Mobility support	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
Availability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Scalability	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
Ownership	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1
Hardware	2	1	2	1	2	0	0	1	2	0	2	0	0	0	0
Security	2	0	0	0	0	1	2	1	2	0	0	1	2	0	0
Privacy	5	1	5	1	5	1	5	1	5	1	5	0	0	0	0
Distance to users	4	1	4	1	4	1	4	1	4	1	4	0	0	0	0
Physical Proximity	5	1	5	1	5	0	0	1	5	0	0	0	0	0	0
Logical Proximity	5	1	5	0	0	0	0	0	0	0	0	0	0	0	0
Power Consumed	2	1	2	0	0	0	0	1	2	1	2	0	0	0	0
Comp. Time	4	0	0	0	0	1	4	0	0	0	0	1	4	0	0
Context Awareness	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0
Non-IP support	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
total:		12	41	10	34	10	34	13	44	10	30	8	25	6	18

Table 4.5: Autonomous vehicles: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Architecture	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Mobility support	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Availability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Scalability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Ownership	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Hardware	5	1	5	1	5	0	0	0	0	0	0	0	0	0	0
Security	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Privacy	5	1	5	1	5	1	5	1	5	1	5	0	0	0	0
Distance to user	5	1	5	1	5	1	5	0	0	1	5	0	0	0	0
Physical proximity	5	1	5	1	5	0	0	1	5	0	0	0	0	0	0
Logical proximity	5	1	5	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	4	1	4	0	0	0	0	1	4	1	4	0	0	0	0
Comp. time	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Context awareness	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Non-IP support	5	0	0	0	0	0	0	1	5	1	5	0	0	0	0
total:		11	54	9	45	11	55	9	44	9	44	9	45	5	25

Table 4.6: Gaming-as-a-service: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Architecture	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Mobility support	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Availability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Scalability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Ownership	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1
Hardware	2	1	2	1	2	0	0	0	0	0	0	0	0	0	0
Security	3	0	0	0	0	1	3	0	0	0	0	1	3	0	0
Privacy	5	1	5	1	5	1	5	1	5	1	5	0	0	0	0
Distance to user	5	1	5	1	5	1	5	0	0	1	5	0	0	0	0
Physical proximity	5	1	5	1	5	0	0	1	5	0	0	0	0	0	0
Logical proximity	5	1	5	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	4	1	4	0	0	0	0	1	4	1	4	0	0	0	0
Comp. time	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Context awareness	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Non-IP support	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
total:		12	44	10	35	10	40	10	33	10	33	8	30	6	18

Table 4.7: Automated loading machine: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Architecture	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
Mobility support	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Availability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Scalability	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Ownership	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1
Hardware	5	1	5	1	5	0	0	0	0	1	5	0	0	0	0
Security	5	0	0	0	0	1	5	0	0	1	5	1	5	0	0
Privacy	3	1	3	1	3	1	3	1	3	1	3	0	0	0	0
Distance to user	5	1	5	1	5	1	5	0	0	1	5	0	0	0	0
Physical proximity	5	1	5	1	5	0	0	1	5	0	0	0	0	0	0
Logical proximity	5	1	5	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	4	1	4	0	0	0	0	1	4	1	4	0	0	0	0
Comp. time	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Context awareness	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Non-IP support	5	0	0	0	0	0	0	1	5	1	5	0	0	0	0
total:		12	50	10	41	10	45	10	40	12	50	8	37	6	23

Table 4.8: Wind Plants: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
Architecture	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Mobility support	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Availability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Scalability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ownership	3	0	0	0	0	1	3	0	0	0	0	1	3	0	0
Hardware	2	1	2	1	2	0	0	0	0	0	0	0	0	0	0
Security	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Privacy	2	1	2	1	2	1	2	1	2	1	2	0	0	0	0
Distance to user	2	1	2	1	2	1	2	0	0	1	2	0	0	0	0
Physical proximity	4	1	4	1	4	0	0	1	4	0	0	0	0	0	0
Logical proximity	4	1	4	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
Comp. time	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0
Context awareness	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0
Non-IP support	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
total:		11	29	9	24	11	30	9	22	9	20	9	26	5	14

Table 4.9: Place of natural disaster: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Architecture	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Mobility support	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Availability	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5
Scalability	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Ownership	3	0	0	0	0	1	3	0	0	0	0	1	3	0	0
Hardware	5	1	5	1	5	0	0	0	0	0	0	0	0	0	0
Security	4	0	0	0	0	1	4	0	0	0	0	1	4	0	0
Privacy	3	1	3	1	3	1	3	1	3	1	3	0	0	0	0
Distance to user	3	1	3	1	3	1	3	0	0	1	3	0	0	0	0
Physical proximity	3	1	3	1	3	0	0	1	3	0	0	0	0	0	0
Logical proximity	3	1	3	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	5	1	5	0	0	0	0	1	5	1	5	0	0	0	0
Comp. time	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Context awareness	5	0	0	0	0	1	5	0	0	0	0	1	5	0	0
Non-IP support	5	0	0	0	0	0	0	1	5	1	5	0	0	0	0
total:		11	43	9	35	11	44	9	37	9	37	9	38	5	21

Table 4.10: Ship at the sea: edge solutions

	Weight	**Cloudlet**	**Weight* Cloudlet**	**ECC**	**Weight* ECC**	**MEC**	**Weight* MEC**	**FOG**	**Weight* FOG**	**EFC**	**Weight* EFC**	**MCC**	**Weight* MCC**	**Superfluid**	**Weight* Superfluid**
Latency	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Architecture	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
Mobility support	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Availability	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
Scalability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ownership	3	0	0	0	0	1	1	0	0	0	0	1	1	0	0
Hardware	4	1	4	1	4	0	0	0	0	0	0	0	z	0	0
Security	4	0	0	0	0	1	4	0	0	0	0	1	4	0	0
Privacy	3	1	3	1	3	1	3	1	3	1	3	0	0	0	0
Distance to user	3	1	3	1	3	1	3	0	0	1	3	0	0	0	0
Physical proximity	2	1	2	1	2	0	0	1	2	0	0	0	0	0	0
Logical proximity	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0
Power consumed	3	1	3	0	0	0	0	1	3	1	3	0	0	0	0
Comp. time	2	0	0	0	0	1	2	0	0	0	0	1	2	0	0
Context awareness	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0
Non-IP support	2	0	0	0	0	0	0	1	2	1	2	0	0	0	0
total:		11	29	9	24	11	26	9	22	9	23	9	20	5	12

Based on the anlysis above we created a summary table with the winning edge architectures (see table 4.11).

Table 4.11: Table: winning edge architectures for each scenario

	Augmented Reality	**Autonomous vehicles**	**Gaming as a service**	**Automated loading machine**	**Wind Plants**	**Place of natural disaster**	**Ship at the sea or offshore platform**
Rank 1	Cloudlet	Mobile Edge Computing	Cloudlet	Cloudlet	Mobile Edge Computing	Mobile Edge Computing	Cloudlet
Rank 2	Fog Cloud	Cloudlet	Mobile Edge Computing	Edge-Fog Cloud	Cloudlet	Cloudlet	Mobile Edge Computing
Rank 3	Edge-Centric Computing	Edge-Centric Computing	Edge-Centric Computing	Mobile Edge Computing	Mobile Cloud Computing	Mobile Cloud Computing	Mobile Cloud Computing

Based on the results from comparison above, we can conclude that the cloudlet is a winning technology for most of the application scenarios. The idea of putting edge nodes within one hop reach sounds extremely promising. However, the implementation of such technology depends on the hardware manufacturers and telecom operators, which could be expensive to operate cost wise. Thus real world implementation could be problematic.

The other winning technology is MEC. The implementation of MEC is highly supported by the telco companies. With the overall development of high speed networks the speed from end-device to the nearest MEC server can be fast.

5 Summary and Conclusion

The aim of this thesis was to formulate an overview of current state-of-the-art in edge compute paradigms. We introduce, describe and compare the most popular edge compute architectures. This research highlights seven most popular architectures: cloudlets proposed by Satyanarayanan M., Edge-Centric computing introduced by Garcia Lopez P., Mobile Edge Computing led by European Telecommunications Standard Institute (ETSI), Fog computing presented by Bonomi F., Edge-Fog cloud researched by Mohan N., Mobile Cloud computing and Superfluid cloud by Manco F.

In the chapter 1 we explore the reasons behind the recent trend in edge compute research in scientific community. We have defined seven key drivers that push the development of edge research. Additionally, we have presented several scenarios to show how edge technology can be used in everyday life and in the production environment. In following chapter 3, we have reviewed existing edge compute publications and research articles.

Based on the key drivers from chapter 1, chapter 4 shows two sets of criteria to compare the described architectures. First, based on the criteria selected we noted similarities and differences of proposed architectures to solve latency and jitter issues. Second, we have introduced the Venn diagram that can help to select appropriate edge architecture based on such parameters as proximity, the power consumed, context awareness, and computational time. Finally, the chapter is concluded with mapping of application scenarios to both sets of criteria and evaluated which architectures are winning based on the application scenarios.

As edge computing paradigm is still under heavy development, what could be ahead of us? What kind of architecture might be suitable for end-users?

We could envision that the line between cloud and edge would slowly disappear. The architecture will move towards a set of well defined APIs and parameters by which applications, devices, or end-users could request relevant server deployment locations. For example, the API would accept criteria we have defined in chapter 4 as an input and provide end-users with a relevant resource.

We believe that this research will help to understand the edge approach better and gives an overview of different edge cloud paradigms.

Bibliography

[AA16] A. Ahmed and E. Ahmed. “A survey on mobile edge computing”. In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. 2016, pp. 1–8.

[Bim15] K. Bimbraw. “Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology”. In: *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Vol. 01. 2015, pp. 191–198.

[Bon+12] 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*. MCC ’12. Helsinki, Finland: Association for Computing Machinery, 2012, pp. 13–16. ISBN: 9781450315197. DOI: 10.1145/2342509.2342513. URL: https://doi.org/10.1145/2342509.2342513.

[Cho+12] S. Choy, B. Wong, G. Simon, and C. Rosenberg. “The brewing storm in cloud gaming: A measurement study on cloud to end-user latency”. In: *2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*. 2012, pp. 1–6.

[Chu+11] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. “CloneCloud: Elastic Execution between Mobile Device and Cloud”. In: *Proceedings of the Sixth Conference on Computer Systems*. EuroSys ’11. Salzburg, Austria: Association for Computing Machinery, 2011, pp. 301–314. ISBN: 9781450306348. DOI: 10.1145/1966445.1966473. URL: https://doi.org/10.1145/1966445.1966473.

[CIS15] CISCO. *Cisco Fog Computing Solutions: Unleash the Power of the Internet of Things*. Tech. rep. 2015. URL: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-solutions.pdf.

[CIS18] CISCO. *Cisco Annual Internet Report (2018–2023) White Paper*. Tech. rep. 2018. URL: https://www.cisco.com/c/en/us/solutions/collateral/

executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[CZ16] M. Chiang and T. Zhang. “Fog and IoT: An Overview of Research Opportunities”. In: *IEEE Internet of Things Journal* 3.6 (2016), pp. 854–864.

[DD17] 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)*. 2017, pp. 1–6.

[Din+13] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. “A survey of mobile cloud computing: architecture, applications, and approaches”. In: *Wireless Communications and Mobile Computing* 13.18 (2013), pp. 1587–1611. DOI: 10.1002/wcm.1203. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcm.1203. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.1203.

[Fes+12] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang. “Impact of Cloudlets on Interactive Mobile Cloud Applications”. In: *2012 IEEE 16th International Enterprise Distributed Object Computing Conference*. Sept. 2012, pp. 123–132. DOI: 10.1109/EDOC.2012.23.

[FLR13] N. Fernando, S. W. Loke, and W. Rahayu. “Mobile cloud computing: A survey”. In: *Future Generation Computer Systems* 29.1 (2013). Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures, pp. 84–106. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2012.05.023. URL: http://www.sciencedirect.com/science/article/pii/S0167739X12001318.

[Gar+15] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. “Edge-Centric Computing: Vision and Challenges”. In: *SIGCOMM Comput. Commun. Rev.* 45.5 (Sept. 2015), pp. 37–42. ISSN: 0146-4833. DOI: 10.1145/2831347.2831354. URL: https://doi.org/10.1145/2831347.2831354.

[Goo] Google. *First AR objects launch in Google Search with 3D animals*. https://9to5google.com/2019/05/31/google-search-ar-animals/. Online, Accessed: 2020-04-08.

[Hon+13] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe. "Mobile Fog: A Programming Model for Large-Scale Applications on the Internet of Things". In: *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*. MCC '13. Hong Kong, China: Association for Computing Machinery, 2013, pp. 15–20. ISBN: 9781450321808. DOI: 10.1145/2491266.2491270. URL: https://doi.org/10.1145/2491266.2491270.

[Hua+12] 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*. MobiSys '12. Low Wood Bay, Lake District, UK: Association for Computing Machinery, 2012, pp. 225–238. ISBN: 9781450313018. DOI: 10.1145/2307636.2307658. URL: https://doi.org/10.1145/2307636.2307658.

[Ins14] E. T. S. Institute. *Mobile-Edge Computing – Introductory Technical White Paper*. Tech. rep. Online, Accessed: 2020-04-19. 2014. URL: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf.

[Ins16] E. T. S. Institute. *Mobile Edge Computing (MEC); Framework and Reference Architecture*. Tech. rep. Online, Accessed: 2020-05-22. 2016. URL: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf.

[ISG] E. M. ISG. *Ongoing PoCs*. https://mecwiki.etsi.org/index.php?title=Ongoing_PoCs. Online, Accessed: 2020-05-27.

[JWW15] Y. Jin, Y. Wen, and C. Westphal. "Optimal Transcoding and Caching for Adaptive Streaming in Media Cloud: an Analytical Approach". In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.12 (2015), pp. 1914–1925.

[Kem+12] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. "Cuckoo: A Computation Offloading Framework for Smartphones". In: *Mobile Computing, Applications, and Services*. Ed. by M. Gris and G. Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 59–79. ISBN: 978-3-642-29336-8.

[LOD18] H. Li, K. Ota, and M. Dong. "ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network". In: *IEEE Wireless Communications* 25.3 (June 2018), pp. 88–93. ISSN: 1558-0687. DOI: `10.1109/MWC.2018.1700315`.

[LW13] Y. Li and W. Wang. "The unheralded power of cloudlet computing in the vicinity of mobile devices". In: *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013, pp. 4994–4999.

[LW14] Y. Li and W. Wang. "Can mobile cloudlets support mobile applications?" In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. Apr. 2014, pp. 1060–1068. DOI: `10.1109/INFOCOM.2014.6848036`.

[Man+15] F. Manco, J. Martins, K. Yasukata, J. Mendes, S. Kuenzer, and F. Huici. "The Case for the Superfluid Cloud". In: *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*. Santa Clara, CA: USENIX Association, July 2015. URL: `https://www.usenix.org/conference/hotcloud15/workshop-program/presentation/manco`.

[Mao+17] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. "A Survey on Mobile Edge Computing: The Communication Perspective". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2322–2358.

[MK16] N. Mohan and J. Kangasharju. "Edge-Fog cloud: A distributed cloud for Internet of Things computations". In: *2016 Cloudification of the Internet of Things (CIoT)*. Nov. 2016, pp. 1–6. DOI: `10.1109/CIOT.2016.7872914`.

[Noo+18] T. H. Noor, S. Zeadally, A. Alfazi, and Q. Z. Sheng. "Mobile cloud computing: Challenges and future research directions". In: *Journal of Network and Computer Applications* 115 (2018), pp. 70–85. ISSN: 1084-8045. DOI: `https://doi.org/10.1016/j.jnca.2018.04.018`. URL: `http://www.sciencedirect.com/science/article/pii/S1084804518301504`.

[PJ09] L. Y. Pao and K. E. Johnson. "A tutorial on the dynamics and control of wind turbines and wind farms". In: *2009 American Control Conference*. 2009, pp. 2076–2089.

[R K+14] A. u. R. Khan, M. Othman, S. A. Madani, and S. U. Khan. "A Survey of Mobile Cloud Computing Application Models". In: *IEEE Communications Surveys Tutorials* 16.1 (2014), pp. 393–413.

[RLM18] R. Roman, J. Lopez, and M. Mambo. "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges". In: *Future Generation Computer Systems* 78 (2018), pp. 680–698. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2016.11.009. URL: http://www.sciencedirect.com/science/article/pii/S0167739X16305635.

[SAD11] G. Simmons, G. A. Armstrong, and M. G. Durkin. "An exploration of small business Website optimization: Enablers, influencers and an assessment approach". In: *International Small Business Journal* 29.5 (2011), pp. 534–561. DOI: 10.1177/0266242610369945. eprint: https://doi.org/10.1177/0266242610369945. URL: https://doi.org/10.1177/0266242610369945.

[Sat] M. Satyanarayanan. *Elijah Cloudlet-based Edge Computing.* http://elijah.cs.cmu.edu/. Online, Accessed: 2020-03-24.

[Sat+09] 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: 10.1109/MPRV.2009.82.

[SD16] W. Shi and S. Dustdar. "The Promise of Edge Computing". In: *Computer* 49.5 (May 2016), pp. 78–81. ISSN: 1558-0814. DOI: 10.1109/MC.2016.145.

[Shi+15] B. Shi, J. Yang, Z. Huang, and P. Hui. "Offloading Guidelines for Augmented Reality Applications on Wearable Devices". In: *Proceedings of the 23rd ACM International Conference on Multimedia.* MM '15. Brisbane, Australia: Association for Computing Machinery, 2015, pp. 1271–1274. ISBN: 9781450334594. DOI: 10.1145/2733373.2806402. URL: https://doi.org/10.1145/2733373.2806402.

[Tal+17a] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck. "Mobile Edge Computing Potential in Making Cities Smarter". In: *IEEE Communications Magazine* 55.3 (Mar. 2017), pp. 38–43. ISSN: 1558-1896. DOI: 10.1109/MCOM.2017.1600249CM.

[Tal+17b] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella. "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration". In: *IEEE Communications Surveys Tutorials* 19.3 (2017), pp. 1657–1681.

[Ver+12] 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*. MCS ’12. Low Wood Bay, Lake District, UK: Association for Computing Machinery, 2012, pp. 29–36. ISBN: 9781450313193. DOI: 10.1145/2307849.2307858. URL: https://doi.org/10.1145/2307849.2307858.

[Yi+15] S. Yi, Z. Hao, Z. Qin, and Q. Li. “Fog Computing: Platform and Applications”. In: *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. 2015, pp. 73–78.