

MAJID ASHOURI

**TOWARDS SUPPORTING IOT
SYSTEM DESIGNERS IN EDGE
COMPUTING DEPLOYMENT
DECISIONS**



**MALMÖ
UNIVERSITY**

**TOWARDS SUPPORTING IOT SYSTEM DESIGNERS IN
EDGE COMPUTING DEPLOYMENT DECISIONS**

Malmö University,
Studies in Computer Science No 13,
Licentiate Thesis

© Majid Ashouri, 2021
ISBN 978-91-7877-158-5 (print)
ISBN 978-91-7877-159-2 (pdf)
Printing: Holmbergs, Malmö 2021

MAJID ASHOURI
TOWARDS SUPPORTING IOT
SYSTEM DESIGNERS IN EDGE
COMPUTING DEPLOYMENT
DECISIONS

Malmö University, 2021
Faculty of Technology and Society
Department of Media and Computer Science

Studies in Computer Science

Faculty of Technology and Society
Malmö University

1. Jevinger, Åse. *Toward intelligent goods: characteristics, architectures and applications*, 2014, Doctoral dissertation.
2. Dahlskog, Steve. *Patterns and procedural content generation in digital games: automatic level generation for digital games using game design patterns*, 2016, Doctoral dissertation.
3. Fabijan, Aleksander. *Developing the right features: the role and impact of customer and product data in software product development*, 2016, Licentiate thesis.
4. Paraschakis, Dimitris. *Algorithmic and ethical aspects of recommender systems in e-commerce*, 2018, Licentiate thesis.
5. Hajinasab, Banafsheh. *A Dynamic Approach to Multi Agent Based Simulation in Urban Transportation Planning*, 2018, Doctoral dissertation.
6. Fabijan, Aleksander. *Data-Driven Software Development at Large Scale*, 2018, Doctoral dissertation.
7. Bugeja, Joseph. *Smart Connected Homes: Concepts, Risks, and Challenges*, 2018, Licentiate thesis.
8. Alkhabbas, Fahed. *Towards Emergent Configurations in the Internet of Things*, 2018, Licentiate thesis.
9. Paraschakis, Dimitris. *Sociotechnical Aspects of Automated Recommendations: Algorithms, Ethics, and Evaluation*, 2020, Doctoral dissertation.
10. Tegen, Agnes. *Approaches to Interactive Online Machine Learning*, 2020, Licentiate thesis.
11. Alvarez, Alberto. *Exploring the Dynamic Properties of Interaction in Mixed-Initiative Procedural Content Generation*, 2020, Licentiate thesis.
12. Alkhabbas, Fahed. *Realizing Emergent Configurations in the Internet of Things*, 2020, Doctoral dissertation.
13. Ashouri, Majid. *Towards Supporting IoT System Designers in Edge Computing Deployment Decisions*, 2021, Licentiate thesis.

The publication is also available at:
<https://doi.org/10.24834/isbn.9789178771592>

To those who live a life true to themselves

ABSTRACT

The rapidly evolving Internet of Things (IoT) introduces new challenges for system designers. One such challenge concerns efficient deployment of IoT systems to meet quality requirements, such as latency, energy consumption, privacy, and bandwidth utilization. The increasing availability of computational resources close to the edge has prompted the idea of using these for distributed computing and storage, known as edge computing. Edge computing may help and complement cloud computing to facilitate deployment of IoT systems and improve their quality attributes. However, deciding where to deploy the various application components to achieve optimal quality is not a straightforward task.

To support IoT system designers in making the best use of edge computing, we have focused on the quality requirements and provide three main contributions. First, by reviewing the literature, we identified the relevant and most used qualities and metrics. Moreover, to analyse how computer simulation can be used as a supporting tool, we investigated available edge computing simulators, and in particular the metrics they provide for modeling and analyzing IoT systems using edge computing. Finally, we introduced a method to consider multiple qualities in deployment decisions. As a use case, we analyzed the distribution of Deep Neural Network layers and ranked the deployment options by measuring the relevant metrics via simulation experiments.

Keywords: Internet of Things, Edge computing, Decision Support, Quality Attributes, Metrics, Simulation.

PUBLICATIONS

Included Papers

Paper I. Ashouri, M., Davidsson, P., Spalazzese, R.: Cloud, Edge, or Both? Towards Decision Support for Designing IoT Applications. *Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 155-162, 2018. <https://doi.org/10.1109/IoTSMS.2018.8554827>

Paper II. Ashouri, M., Davidsson, P., Spalazzese, R.: Quality Attributes in Edge Computing for the Internet of Things: A Systematic Mapping Study, *Internet of Things*, vol. 11, 100346, Elsevier, 2021. <https://doi.org/10.1016/j.iot.2020.100346>

Paper III. Ashouri, M., Fabian, L., Davidsson, P., Spalazzese, R.: Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics, *Future Internet*, vol. 11, pp. 1–12, 2019. <https://doi.org/10.3390/fi11110235>

Paper IV. Ashouri, M., Fabian, L., Davidsson, P., Spalazzese, R., Svorobej, S.: Analyzing Distributed Deep Neural Network Deployment on Edge and Cloud Nodes in IoT Systems. *IEEE International Conference on Edge Computing (EDGE)*, 2020.

Personal Contribution

For all publications above, the first author was the main contributor regarding the planning and execution of the research as well as the writing of the publications.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and most profound appreciation to my supervisors Paul Davidsson and Romina Spalazzese, for their continuous support, patience, and availability. Thank you for your mentorship, insightful suggestions, the share of knowledge and enormous contributions to my research path and this thesis. I am so grateful to have had you both as my study leaders. I am also deeply indebted to Prof. Johan Holmgren as the head of DM3 unit for his continuous support and understanding.

Furthermore, I would like to extend my sincere thanks to PhD examiner Jan Persson, and the follow-up group, Helena Holmström and Åse Jevinger for reviewing this thesis, assessing my individual study plans. Your suggestions and discussions throughout the process of completing my licentiate helped to accomplish this thesis. I would also like to thank Thais Batista for acting as the opponent in the licentiate defense.

I want to thank the research profile “Internet of Things and People” (IOTAP) funded by the Knowledge Foundation, Malmö University and faculty of Technology and Society to provide me with the opportunity of pursuing my PhD and support my research studies.

I am also thankful to Fabian Lorig and Sergej Sverobej for their support and contributions to my research studies. I gratefully acknowledge the Solveig-Karin Erdal, and Susanne Lundborg’s assistance for the different types of their support throughout my study. Furthermore, I would like to extend my sincere thanks to all my PhD colleagues and especially, Alberto, Fahed, Johan, Josef, Lars, Saied, and Sergej for their friendship and kind support in many ways.

And finally, my biggest thanks to my wife, Sidika. Your love, understanding, sacrifices, and extreme support was a significant help to pave the PhD path. Thank you.

CONTENT

PART I	1
1. INTRODUCTION.....	3
1.1. A Deployment Decision Scenario	5
1.2. Background	7
1.3. Research Questions	12
1.4. Outline.....	14
2. RESEARCH METHODOLOGY	16
2.1. Literature Review	16
2.2. Design Science Research.....	17
3. CONTRIBUTIONS	19
3.1. Key Qualities and Metrics in the Literature	19
3.2. Qualities and Metrics Supported by Edge Computing Simulators	21
3.3. Towards a Decision Support Framework	23
4. CONCLUSIONS AND FUTURE WORK	25
4.1. Conclusions.....	25
4.2. Future Work.....	26
REFERENCES	28
PART II.....	33
Paper I: Cloud, Edge, or Both? Towards Decision Support for Designing IoT Applications	35
Paper II: Quality Attributes in Edge Computing for the Internet of Things: A Systematic Mapping Study.....	53
Paper III: Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics	101
Paper IV: Analyzing Distributed Deep Neural Network	

Deployment on Edge and Cloud Nodes in IoT Systems..... 121

PART I

COMPREHENSIVE SUMMARY

1. INTRODUCTION

Cloud computing has been one of the main solutions for application deployment in the last decade. This usually provides an efficient centralized architecture for data processing and storage. At the same time, we are witnessing the fourth digitization wave, often referred to as the Internet of Things (IoT). IoT systems include devices residing at the edge of the network that may generate huge amounts of data, which poses new requirements challenging cloud solutions to provide efficient service provisioning [1]. For instance, vehicular safety applications demand for hard real-time responses with just a few milliseconds latency [2]. Considering energy systems, it is estimated that the US smart grid will generate 1000 petabytes of data each year [3]. This motivates the development of new solutions alongside cloud computing [4].

The idea of utilizing distributed resources at the “edge” of the network, known as *edge computing*, and providing the required capabilities closer to the source of data has been proposed in recent years [5]. Edge computing may help to address the challenges of cloud computing for hosting IoT applications by lowering latency, handling privacy issues, reducing data communication, and reducing energy consumption. Moreover, distributed edge computing can be integrated with centralized processing in the cloud, resulting in a hybrid edge-cloud model that may combine the strengths of both approaches [6]. However, this impacts the way of processing data compared to the traditional cloud architecture and adds more complexity when designing IoT applications.

The software of IoT systems can typically be divided into a set of software components, and often it is possible to distribute these components across the available infrastructure, which may result in a hybrid edge-cloud solution. However, having more options for the deployment of software components poses additional challenges to the system designer regarding how to choose the best solution to deploy the components among all possible combinations.

The deployment problem has been already studied, but mainly from the infrastructure provider's point of view [7][8]. The main focus of that research is on how to efficiently allocate the available resources in the edge or cloud to satisfy the received requests from IoT applications in order to maximize the performance of the infrastructure resources with the minimum cost. In other words, the main goal of those studies is to optimize the usage of the available infrastructure resources at run-time. Instead, in this thesis we study the deployment decision problem from another perspective, i.e., from the IoT system designer's point of view. In particular, we aim to provide support for the designers when facing deployment decisions.

To achieve this goal, we focus on the desired system qualities to evaluate deployment alternatives and provide quality-oriented guidelines for the designers. Accordingly, one of the initial steps is to identify the relevant qualities and metrics for IoT systems, which then can be used for evaluating the different design options. Previous studies have mostly focussed on a few qualities, like time behaviour and resource usage, and used limited metrics, such as energy consumption and response time [9][10]. Based on our observations of some real-world IoT systems, we found that there are usually more qualities and metrics that impact the decision (an example of system is discussed in Section 1.1). To this end, we investigated the relevant qualities and metrics used in the literature.

Moreover, suitable tools for measuring the key metrics should be provided to the designers to facilitate the process of finding an appropriate deployment option at design time. Computer simulation has been proven to be a suitable technique for investigating different aspects of complex systems, such as service configuration, resource placement, or management strategies in IoT systems. To this end, several simulators has been proposed for the simulation of edge and cloud computing [11]. However, an analysis of the metrics supported by the simulation tools is missing while it could help designers to choose the best tool for evaluation. This is the second area studied in this thesis. We investigated some edge computing simulators, and in particular the metrics they provide for and analyzing IoT systems which use edge computing.

Finally, to support the designers, a method is needed for measuring the qualities and providing deployment preferences to the designers. Therefore, the third objective of this thesis is to provide a first step towards a decision support tool for IoT system designers that considers

multiple qualities/metrics in the decision. Finally, we use the domain of Deep Neural Networks as a case and propose a method for distributed deployment of the different neural layers by considering multiple relevant metrics in the decision.

To better explain the problem of supporting deployment decisions in a hybrid edge-cloud architecture, in the following we introduce an example scenario that is also used in the papers included in this thesis.

1.1. A Deployment Decision Scenario

We will use a Smart Traffic Light (STL) system to illustrate the decision-making challenges involved in application component deployment. As in most cases, there are multiple conflicting criteria that impact the decision.

An STL system typically is a part of a more complex Intelligent Transportation System possibly including Connected Vehicles. Abstractly, it contains several distributed components: traffic sensing that detects the vehicles and generate the sensed data, data aggregation that collect and combine the sensed data, real-time cycle scheduling that determines the cycle phase and its duration for a traffic light, cycle display which actuates by displaying the cycle signs such as red, green, and yellow, and overall system long-term optimization that optimize scheduling policies by collecting data from all the traffic lights, traffic statistics and feedbacks from the users. The considered case concerns scheduling and cycle management of traffic lights, in which the traffic data are collected form the traffic sensors, and the phase of a traffic light (e.g., red or green) and its duration are adaptively determined according to the predefined policies as well as the current traffic situation. It includes different software components that can be distributed in different ways. Let us assume there are three deployment options as depicted in Figure 1 for distributing an application: Things which consists of devices with very limited computational and storage resources, Edge nodes with more computational, storage, and communication capabilities than the Thing nodes, and Cloud with massive computational, storage and communication capabilities (the terms are explained in more detail in Section 1.2.2). In addition, let us assume the following deployment possibilities for the application components: Traffic sensing and cycle display components are executed on Thing nodes such as magnetic sensors and traffic light display; data aggregation and real-time cycle management components can be deployed on Edge nodes like traffic light

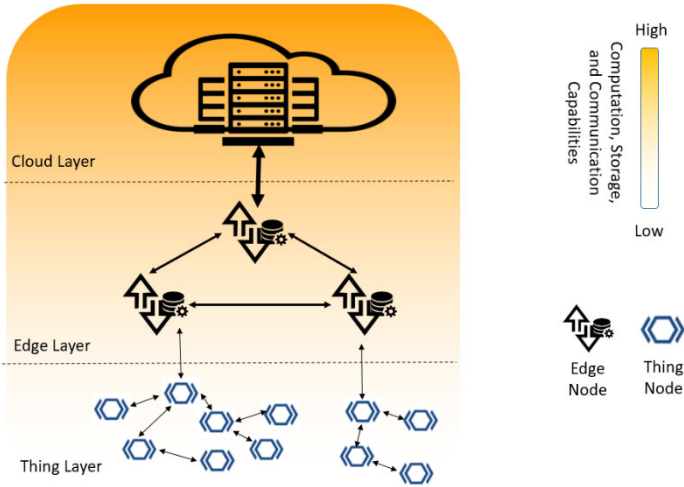


Figure 1. Hybrid edge-cloud reference architecture for IoT systems.

controllers and local servers, or Cloud; finally, long-term optimization components can be deployed on Cloud. Therefore, for this case, the designer should decide about the data aggregation and real-time cycle management components and decision support to find the best option is required.

To show the role of qualities and metrics in the decisions let us consider the traffic light controller device as an edge node and candidate to host the real-time cycle management and data aggregation components. To make accurate decisions and to provide a green wave, it is necessary to collect information from other intersections and controllers in addition to the data from the local sensors. Thus, in this case the accuracy and availability of the system should be evaluated since different level of data is accessible in the Edge compared to a centralized solution. If energy consumption is also considered as an important factor, the amount energy by propagating the collected data of a traffic light node to several other nodes should be measured, since it may even increase the total energy consumption. Latency as response time is another important metric, but there is not a hard real-time constraint for this service. The available computation resources and data synchronization between adjacent controllers influence the latency. The number of STL nodes and their density also impacts decision in terms of the amount resource usage and latency. For example, if there are several isolated STL with a few crossing

cars, it might be an option to run the components in a STL node without communication with other entities. Moreover, employing dependable security mechanisms is critical, and if sensing is performed by image processing methods, privacy is also a quality that should be considered.

1.2. Background

In this section we will describe the central concepts used in this thesis.

1.2.1. Internet of Things

With the new advances in information technologies, the Internet of Things (IoT) is rapidly merging into our daily life. After three digital waves, i.e. computer systems and products, internet enabled computers, and mobile internet, IoT is referred as the fourth wave of digitization that has made major impacts on our life. In IoT, many of our surrounding objects such as smart mobile phones, home appliances, vehicles, and different types of sensors and actuators will be connected, and different data can be collected and exchanged among IoT devices through modern communication network technologies.

Generally, IoT refers to collections of various sensors, objects, and smart nodes, which have the capability to communicate with each other [12]. During the last decade different definitions and taxonomies of Internet of Things (IoT) systems have been proposed [13]. For example, the International Telecommunication Union illustrated IoT as enabling the “connectivity for anything, from anytime and at anyplace” [14] or IEEE Internet Initiative has characterized IoT as “envisions a self-configuring, adaptive, complex network that interconnects ‘things’ to the Internet through the use of standard communication protocols” [15].

Similarly, different elements have been mentioned as the building blocks of IoT. Al-Fuqaha et al. [16] introduce six categories of elements: identification, sensing, communication, computation, service, and semantic. Identification element is responsible to provide naming and addressing of the IoT resources. Examples of identifications protocols are IPv6, Electronic Product Codes (EPC), Ubiquitous Codes (uCode). The sensing element is responsible for gathering data from the objects within the network and sending it back to a data warehouse or database. The IoT communication element connects heterogeneous objects together. Examples of communication protocols in IoT are WiFi, Bluetooth, IEEE 802.15.4, RFID, and LTE-Advanced. The Computation element contains

both software and hardware units providing computational facilities for IoT. Arduino, UDOO, FriendlyARM, Intel Galileo, Raspberry PI are some examples of hardware platforms. Operating systems such as Real-Time Operating Systems (RTOS), TinyOS and LiteOS can be used as the software part of computation elements. Moreover, platforms with high computational capacity such as cloud solutions form another important computational part of IoT. The Service element is mainly responsible to collect and summarize raw sensory measurements, to make decision and react accordingly. The services provided by IoT in application domains such as smart healthcare, smart grids, smart home, smart buildings, intelligent transportation systems (ITS), and industrial automation lies under service element category. And finally, the semantic element in the IoT refers to the ability to extract knowledge smartly by different mechanisms. Resource Description Framework (RDF) and the Web Ontology Language (OWL), and Efficient XML Interchange (EXI) format are examples of technologies in the semantic element.

1.2.2. Cloud Computing

As mentioned above, one of the key challenges in IoT is to provide required infrastructure to store and process collected data from end devices. Using cloud computing has been introduced as a promising technology to host IoT services [24]. Cloud computing provides services to the end-users through pooling of resources including processing, storage, and network [24][25]. National Institute of Standard and Technologies (NIST) defined cloud computing as [26]: “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”.

IoT can benefit from unlimited capabilities and resources of the cloud to compensate technological constraints of IoT (e.g., storage, processing, communication) [24]. Cloud can be an effective solution for implementing IoT applications. There are several motivations derive to integrate cloud and IoT. Cloud solutions can provide services to cover IoT issues such as high heterogeneity of devices and technologies, scalability, interoperability, flexibility, reliability, efficiency, availability, and security. Ease of use and the reduced cost are two other derives for using cloud

[24]. However, IoT systems have devices residing at the edge of the network and generate huge amounts of data. This may cause challenges for cloud solutions that are usually located far from the edge such as more latency and energy consumption, high bandwidth usage, and privacy issues. This led to proposing the idea of utilizing distributed resources at the edge of the network, known as edge computing.

1.2.3. Edge Computing

Edge computing aims to move processing and storage of data away from cloud servers to the edge of the network. The application developers can use the edge computing by offering the users services closer to them [25]. Adding the edge for hosting IoT applications to the cloud would be beneficial for several reasons: it enables cloud providers to provide their services with a higher quality, e.g., like reducing latency; it enables application designers to develop more efficient and cost effective applications addressing end user requirements; it reduces the amount of data communication through the network thus supporting network infrastructure providers; finally, it supports privacy and provides a better user experience to the end-users.

The terms edge and fog computing are often used interchangeably in the literature [27]. The concept fog computing was originally proposed by Cisco, and it is defined as an “extension of the cloud computing paradigm that provides computation, storage, and networking services between end devices and traditional cloud servers” [28]. Others (e.g.[29][30]) have also proposed the idea of extending cloud computing and supporting multi-tenancy. NIST [31] has defined fog computing as “a horizontal, physical, or virtual resource paradigm that resides between smart end-devices and traditional cloud or data centers.” They propose six characteristics to distinguish fog computing from other computing paradigms: contextual location awareness and low latency, geographical distribution, heterogeneity, interoperability and federation, real-time interactions, and scalability and agility of federated fog-node clusters.

Similarly, the term edge computing is used with various meanings in the literature. Narrowly defined, it refers to local processing excluding both fog and cloud computing [30]; in a broader definition, it is considered as any computing and network resources along the path between data sources and cloud data centers with the rationale of happening at the proximity of data sources [27]. In this thesis, we use the term edge computing

in its broader scope and include any computing and storage resources (including fog computing) along the path between sensors/actuators and cloud servers.

1.2.4. IoT Reference Architectures

A number of reference architectures have been proposed for IoT systems [16]. For example, Pereira [17] proposed a five-layer architecture for IoT consisting of: edge layer, access gateway layer, internet layer, middleware layer, and application layer. The architecture was designed for stand-alone applications, in which data communication is possible only within the same application. To address the problem of interoperability, Tan [18] proposed a new layer called the coordination layer to provide a unified structure that can be used by different applications. An abstract and simplified a three-layer architecture of IoT was proposed by Wu et al. [19]. It consists of three layers, the perception layer, the application layer, and the network layer. However, the three-layer architecture did not cover some features of IoT. To complete the three-layer architecture, a five-layer IoT architecture was proposed by Khan et al. [20], adding the middleware layer and the business layer to the three-layer architecture. To integrate IoT with cloud computing, the CloudThings architecture was proposed by Zhou et al. [21]. CloudThings is a centralized cloud-based approach and distributed deployment was not considered in the architecture. To support distributed fog-cloud computing with a centralized control unit, Salman et al. [22] proposed an IoT architecture for distributed data processing. This architecture consists of four layers: the device layer, the network layer, the control layer, and the application layer. The idea of this architecture was to distribute the processing tasks between cloud and fog nodes. The architecture considers both physical devices, and logical and control elements. To focus on the deployment perspective and physical resources, a hybrid edge-cloud architecture based on physical resources for deploying IoT applications was proposed by Ashouri et al. [23]. This architecture and the details of its layers which shows a physical view for deployment of IoT systems is illustrated in Figure 1. We refer to this architecture throughout the thesis.

Figure 1 shows the Hybrid edge-cloud reference architecture that includes three layers: The Cloud Layer, the Edge Layer, and the Thing Layer. In the following, we describe the different roles and responsibilities of each layer of the architecture in more detail. The

separation among the layers is based on basic capabilities, i.e., sensing and actuation, storage, processing, and communication.

Thing Layer: The Thing Nodes included in this layer are devices that typically have very limited computational and storage resources that enable them to host very simple applications components, such as sensing, actuation based on certain events, and sending and receiving data.

Edge Layer: This layer consists of nodes with more computational, storage, and communication capabilities than the Thing Layer (but still limited) that enables them to manipulate, analyse, and dispatch data, interact with the Cloud and Thing Layers, and host IoT applications or components located at the network edge. The role of Edge Nodes can be played by various devices, e.g., gateways, access points, routers, switches, local servers, smartphones and modern vehicles.

It is worth to notice that some devices can play different roles. For example, a traffic light can act at the same time both as a Thing Node that senses the environment and actuates, and as an Edge Node that aggregates data, processes it, and hosts application components.

Cloud Layer: The Cloud Layer consists of several high-end servers and data centres with massive computational, storage and communication capabilities that enables them to host various kinds of IoT applications with high physical resources demand.

1.2.5. Quality Attributes and Metrics

One of the important factors that should be taken into the account by IoT system designers is the relevant quality attributes (qualities) of the system to be designed, such as performance, security, and reliability [32]. According to ISO/IEC 25010, the quality of a system is defined as the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value [33].

In developing a system, it is often not an easy task to fulfill all requirements simultaneously. A major difficulty is how to manage the conflicts and handle the trade-offs between different requirements. This is particularly obvious when addressing the requirements related to different qualities [34]. In the context of edge computing, different qualities have been investigated in research papers. For instance, considering a trade-off between time behavior and resource utilization has been used frequently by the papers [35][36]. A few papers consider more qualities.

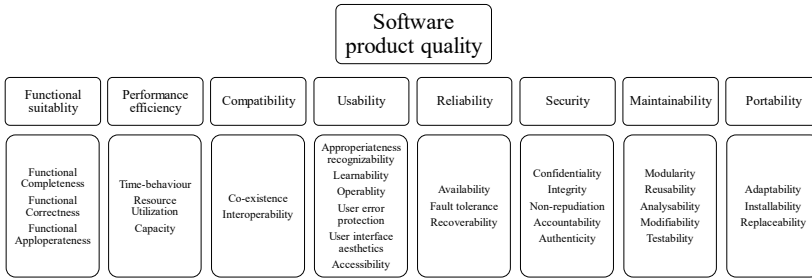


Figure 2. ISO/IEC 25010 quality model, including eight root qualities and their sub-qualities.

For example, in [37][38] delay, fault avoidance, capacity and accuracy were studied, or in [38] capacity, availability, energy consumption and execution time were measured for better management of task offloading.

To perform systematic research on system qualities, the use of quality models is an acceptable means to support quality management [39]. A quality model is "the set of characteristics, and the relationships between them that provides the basis for specifying quality requirements and evaluation" [40]. In order to study qualities of systems and software products, several quality models have been proposed in the literature [39]. ISO/IEC 25010 standard [33] is one of the models used frequently in the literature. It can be used to determine which quality characteristics may be taken into account when evaluating the properties of a system and software product. The standard categorizes the product qualities into characteristics and sub-characteristics, and comprises the eight quality characteristics shown in Figure 2.

Moreover, to measure the quality of a system, different metrics can be defined. The ISO/IEC 25023 standard [41] provides a set relevant metrics for each sub-characteristic of ISO/IEC 25010. For instance, for time-behavior quality, mean response time, response time adequacy, mean turnaround time, turnaround time adequacy, and mean throughput are listed as relevant metrics in the standard.

1.3. Research Questions

The general objective is to support IoT system designers in making the best use of edge computing. In order to meet this objective, we will in the thesis address the following research questions:

RQ1. What are the most relevant quality attributes and metrics for IoT systems using edge computing?

As mentioned above, to support IoT system designers in evaluating the different design and deployment options in hybrid edge-cloud architectures, the key quality attributes that influence the decision-making should be identified. Previous work mostly focuses on just a few qualities, like energy consumption and latency, while IoT applications are diverse with different requirements. Moreover, evaluating a fixed and a few numbers of qualities may lead to inaccurate decisions. Thus, the objective of RQ1 is to investigate what qualities and metrics that are the most relevant for making design decisions for IoT systems using edge computing.

RQ2. What quality attributes and metrics are supported by current edge computing simulators?

Computer simulation has proven to be a suitable technique to support decision makers, especially for better understanding complex systems. For edge computing, several simulators have been developed to enable and facilitate modeling of IoT systems. An initial step towards understanding how simulation can support IoT system designers in case of software deployment problems is to analyze the available simulator with respect to supported qualities that they can measure. Therefore, the objective of RQ2 is to investigate currently available edge computing simulators specifically, studying the support they provide for modeling and analyzing qualities that might be relevant for IoT system designers. This may guide researchers and practitioners to select a suitable simulator for their experiments. In addition, the results will clarify the gaps in the current tools and provide suggestions regarding how to develop more comprehensive simulators.

RQ3. How can IoT system designers be supported when making deployment decisions considering multiple qualities?

After identifying the relevant qualities and metrics, the next step to support the IoT designers is to measure and integrate the relevant qualities. In RQ3, the long-term objective is to provide a decision support tool for the IoT designers. Current approaches consider only a few

qualities and use metrics such as latency, accuracy, and energy consumption, for assessing the performance of deployment alternatives. For this research question, we aim to consider multiple qualities and metrics to find the most preferable design option. In this case, the system designer will model the scenario for simulation and by measuring the required aspect identified by the user it should provide preferable options to the system designer. To partially answer RQ3, a concrete case will be implemented and tested.

1.4. Outline

The thesis is divided into two parts: the first part describes a comprehensive introduction and the second part includes four papers that constitutes the core of the thesis. To address RQ1, we introduce in Paper I the idea of design-time decision support for IoT designers in order to efficiently distribute application components. We also identified and described the relevant key metrics influencing the decision by studying some example systems. To extend this idea, we conduct a systematic mapping study in Paper II to investigate what qualities and metrics have been considered in previous studies. To address RQ2, we investigate in Paper III currently available edge computing simulators and specifically the qualities and metrics they provide for the designers. Finally, to address RQ3, we identify in Paper IV a set of relevant metrics for Deep Neural Networks, and then propose a multi-criteria decision-making method for the rating of the deployment alternatives. Figure 3 illustrates the relationship between the papers, and how they address the research questions.

An outline of the remainder of the first part is presented below:

Chapter 2: Research Methodology. Describes the methodologies that have been applied to answer the research questions.

Chapter 3: Contributions. Presents the main contributions to the research community and their relation to the research questions.

Chapter 4: Conclusions and Future Work. Draws some general conclusions and identifies possible future work directions.

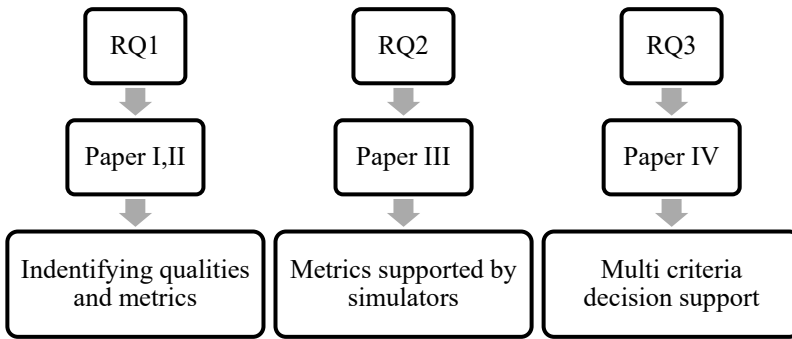


Figure 3. The relationships between the research questions, included papers and their contributions

2. RESEARCH METHODOLOGY

This chapter describes the research methodologies used to address the research questions presented in Chapter 1. We used two research methods, Literature Review and Design Science Research, which are presented in Sections 2.1 and 2.2, respectively.

2.1. Literature Review

To address RQ1 and RQ2, we conducted literature reviews. The main aim of performing a literature review is to find out what is already known about a specific topic in order to summarize the current knowledge, generate and refine the research ideas and to provide a critical review [43]. To address RQ1 that have similar objectives, we chose to do a literature review (in Paper I) in order to deepen our knowledge about how the current approaches have addressed decision support to IoT system designers in edge computing, and to critically analyze the state of the research. Additionally, to further study RQ1 and also address RQ2 (in Paper II and III respectively), we used Systematic Mapping Study (SMS) [44] to extract the key qualities and metrics used in the research. SMS is known as an appropriate methodology to extract and structure a research problem from the researcher's point of view [44]. Below we describe in more detail the literature review methods used in the papers.

In Paper I, we conducted a critical literature review [43]. We first reviewed the pioneering edge computing papers, and then, we analyzed the papers addressing deployment decisions with focus on the qualities studied. This process resulted in identifying some relevant qualities and research gaps for future studies. We continued and completed this study by conducting a systematic mapping study in Paper II.

Systematic Mapping Studies were performed in Paper II and Paper III. We followed the guidelines proposed by Peterson et. Al . The guideline suggests performing following steps: research question, search, study selection, data extraction, analysis and classification, reporting the results, and validity evaluation.

For RQ2 in Paper III, we used the SMS methodology to analyze the qualities and metrics used by the edge computing simulators in two rounds. In first round, the goal was to generally identify the qualities and metrics used in the literature. In this round, by clarifying research questions, we formulated a systematic search, and downloaded the results from four well known literature databases. Then in several rounds by removing duplicates, reading abstract, and reading the full paper we selected the relevant papers for the study. In the next round, we used the identified and classified qualities and metrics in the first round to extract and classify supported metrics by edge computing simulators in the second round. In the second round we did not used systematic search and study selection steps from the methodology, since there were limited number of simulators listed in [11]. Finally, we represented the findings and discussed the results and research gaps.

In Paper II, we conducted another systematic mapping study similar to Paper III, but with different research questions and data extraction categories, and covering more papers for data extractions. The same method was used in here for formulating research questions, search study selection, data extraction and classification and results representation. We also discussed different aspect of the research validity in the paper.

2.2. Design Science Research

We applied the design science research methodology [45] to address RQ3 in Paper IV. According to this methodology, the main goal is to create an innovate artifact to solve practical problems. We chose this method, since the aim of RQ3 is to propose a novel artifact that support IoT system designers in solving the problem of finding the best deployment option according to multiple criteria. There are five stages in the design science research methodology: problem explication, requirements definition, artifact design and development, artifact demonstration, and artifact evaluation [45].

In problem explication stage, we identified different aspects of the problem through the literature studies and group discussions among the authors. The output and knowledge gained from answering RQ1 and RQ2 were also used to formulate the problem. Then, we identified three main gaps (which were the lack of considering multiple metrics for evaluation, user inputs, and scalability) to be answered by the research. In requirement definition stage, we identified the necessary components and

tools required for the research. We explored some of the edge computing simulation tools. Moreover, the relevant metrics for the evaluation and the appropriateness of the DNN approaches were investigated through analyzing the related work and group discussions among the authors. In the artifact design and development stage, we proposed and implemented an AHP-based method for rating the deployment options in design time by considering several metrics for the measurements. Moreover, we extended RECAP [47] simulator to be able to model the IoT computation and communication infrastructure and the DNN application. Then to demonstrate the artifact we implemented an IoT system, automatically scheduling traffic lights by detecting the number of cars going in different directions. We modelled the system in the RECAP simulator for performing the experiments and evaluation. Finally, in the evaluation stage, we conducted several experiments using the simulator. By performing the experiments, we could show the ability of the proposed method for ranking the deployment scenarios based on different input parameters.

3. CONTRIBUTIONS

This chapter presents the contributions of the thesis. The contributions are described and grouped according to the research questions, starting with RQ1.

3.1. Key Qualities and Metrics in the Literature

The existing research studies usually investigate the problem of allocating infrastructure resources from infrastructure providers' point of view and not the IoT system designers' [23]. In other words, the focus is mostly on how to optimize the use of available infrastructure resources at run-time in order to benefit infrastructure providers. We found a gap regarding the investigation of the application distribution problem at design-time and from IoT designers' point of view. To fill this gap, in Paper I, we introduced the idea of design-time decision support for IoT designers in order to efficiently distribute application components.

Moreover, previous research has mostly focused on just a few qualities and metrics, such as energy consumption and latency, to assess the deployment options in edge computing. But, based on our observation in some real-world use-cases, we found that there are more quality attributes that significantly may impact deployment decisions. Thus, in Paper I, we also identified and described the relevant key qualities influencing the decision. Finally, we also proposed a series of potential research areas to be investigated in future to pave the way toward design-time decision support.

As mentioned, identifying the key quality attributes is an important step towards supporting IoT system designers. The output of Paper I, was based on a few case studies. To further explore this idea, we conducted a systematic mapping study in Paper II to investigate what qualities have been considered in previous studies of IoT systems using edge computing. The search and paper selection process resulted to 98 papers for final data extraction. Moreover, we used ISO/IEC 25010 and ISO/IEC 25023 as a base to categorize the qualities and metrics used in the papers.

The results indicate that, the most investigated qualities are *time behavior* and *resource utilization*. Only a few studies considered security, and usability qualities. No study was found that investigated compatibility, maintainability, and adaptability, while we believe they are also relevant in edge computing. The majority of the papers consider just one or two qualities for quantitative evaluation. Regarding metrics, the results indicate that the most used metric categories are response time, turnaround time, throughput, functional correctness, processor utilization, bandwidth utilization, energy consumption, edge/cloud device utilization of devices, and transaction processing capacity.

Moreover, we investigated the trade-offs among the attributes, and the validation types. The results suggest that the trade-off between *time behavior* and *resource utilization* is the most commonly studied, and that trade-offs among two or three qualities are usually considered. Regarding the validation methods, simulation is by far the most used.

Another main contribution in Paper II is a list of suggestions for future research directions that can guide researchers to fill the current knowledge gaps. The suggested areas for further studies are: (i) Metrics corresponding to the less measured qualities, i.e., security, usability, compatibility, maintainability and portability; (ii) General frameworks for considering measurements of multiple qualities and the trade-off between them; (iii) Considering qualities other than time behavior and resource utilization for making deployment decisions; (iv) Extension of the available IoT simulators to support additional measurements.

In summary, the contributions of Paper I and II are as follows:

- the concept of design-time decision support for IoT application designers in an edge-cloud architecture,
- a research roadmap to achieve a design-time decision support framework,
- identification of the qualities and metrics, and their trade-offs considered in the design of IoT applications using edge computing,
- specification of the knowledge gaps and future research directions in relation to qualities for IoT systems using edge computing.

3.2. Qualities and Metrics Supported by Edge Computing Simulators

As mentioned above, our investigation indicated that simulation is currently the most used method to validate research results in edge computing for IoT systems. In order to investigate the potential of simulation in supporting IoT system designers, an initial step is to analyze the available edge computing simulators with respect to what qualities and metrics they support. Accordingly, and to address RQ2, we do this in Paper III. To achieve this, and in order to reduce the risk of bias for identifying the list of qualities and metrics, first we systematically investigated state of the art papers to derive the key qualities and related metrics. This was based on a subset of the results from RQ1. Accordingly, we extracted the qualities and metrics used by other researchers in edge computing based IoT systems. Then, from the list, the relevant and feasible metrics for simulation were derived and categorized based on a set of criteria defined in Paper III. Also, in this case, the ISO/IEC standards were used for classifying the identified metrics.

Then, we analyzed the available edge computing simulators with respect to the derived metrics and presented the results in a comparison table of the simulators (see Table I). The table provides a guide for IoT designers for the selection of an appropriate simulator, given what qualities and metrics are relevant for the case at hand. Moreover, our results show that none of the simulators cover more than 5 of the 14 identified metrics, and we argue that there is still a need for simulators that can assess a wider set of qualities. Among the qualities, time-behavior is the most popular one, and which is measured by different metrics. Mean response time, processing delay and network delay are the most common time-behavior metrics used by the simulators. Among resource utilization metrics, network usage is the most common. Processor utilization and energy consumption are also supported by some simulators. However, some of the identified metrics, like throughput and availability, are not supported by any of the simulators. Moreover, capacity metrics, which can be considered as a measure for system scalability, are usually neglected by current tools.

In summary, to support IoT designers to take deployment decisions in Paper III, we:

- identified a set of relevant qualities for edge computing simulation, and

- performed an analysis of the edge computing simulators with respect to the identified qualities and related metrics they support.

Table I: Supported qualities and metrics by edge computing simulators

		iFogSim	Fog-Net-Sim++	Edge-Cloud-Sim	IoT-Sim	Fog-Torch II	Emu-Fog	FogBed
Time behavior	Response time	Y	-	Y	-	Y	Y	Y
	Processing delay	Y	Y	Y	Y	-	-	-
	Network delay	-	Y	Y	-	Y	Y	-
	Storage read/write delay	-	-	-	-	-	-	-
	Throughput	-	-	-	-	-	-	-
Resource Utilization	Bandwidth utilization	Y	-	Y	Y	Y	-	-
	Processing utilization	Y	-	-	Y	-	-	-
	Memory utilization	-	-	-	-	Y	-	-
	I/O devices utilization	-	-	-	-	Y	-	-
	Energy consumption	Y	Y	-	-	-	-	-
Capacity	Transaction processing capacity	-	Y	Y	-	-	-	-
Availability	System availability	-	-	-	-	-	-	-
Fault tolerance	Failure avoidance	-	-	-	-	-	-	-

3.3. Towards a Decision Support Framework

To address RQ3, we propose a multi-criteria method to support IoT system designers in Paper IV. We use simulation as the main measurement tool.

As a use case, we investigated how Deep Neural Networks (DNNs) can be deployed on the hybrid edge-cloud architecture. Using DNNs for image recognition tasks is common in IoT systems, e.g., for monitoring traffic or crowds. When executing DNNs in IoT systems, the computational tasks can be distributed and deployed on things, edge and cloud nodes. This is often referred to as Distributed Deep Neural Networks (DDNN) [46]. However, in the previous studies on DDNN, only a few metrics have been used to assess the deployment alternatives. Also, as the importance of the metrics is different for the different applications, they should be considered as a designer input. This has not been considered in the existing literature. Moreover, the current DDNN approaches usually work only on a single or a few sources of data and do not consider large scale deployments.

To address these gaps, we first identified a set of potentially relevant metrics for DDNNs, and then proposed a multi-criteria decision-making method based on Analytical Hierarchy Process (AHP) for the comparison of the deployment alternatives. We used simulation to measure the relevant metrics and study the deployment of DDNNs for large scale systems. We extended the existing RECAP simulation framework [47] to model and assess DDNN deployment alternatives, while varying the size of the system. By using AHP we were also able to link human judgement (about the importance of the metrics) to the simulation measurements, integrate several metrics, and provide a ranking of deployment alternatives for different system sizes.

In RECAP, we implemented a DDNN application behavior model and additional performance metrics, such as resource capacity and drop rate, to provide a more comprehensive assessment of deployment scenarios. The usefulness of the approach was shown through a use case concerning a smart city scenario where road intersections are equipped with cameras for the estimation of the number of vehicles using DDNN. We simulated AlexNet layers and analyzed and compared different deployment scenarios of this DNN. Then, the deployment suggestions for two specific cases were represented. It was shown how the proposed method is able to

suggest appropriate deployment solutions based on the user preferences and different scales of the system.

In summary, the main contribution of Paper IV (in relation to supporting deployment decisions) was a decision support method, which ranks deployment solutions based on the user input on multiple relevant quality metrics.

4. CONCLUSIONS AND FUTURE WORK

4.1. Conclusions

The idea of utilizing edge computing has been proposed to perform computation and storage closer to where the data is generated. This may help addressing the traditional cloud computing challenges such as latency and privacy by using a hybrid edge-cloud architecture. However, using such a hybrid model adds more complexity when designing IoT applications, e.g., by allowing more options for the deployment of software components across the network. Deciding how to best distribute the components while meeting the system's requirements is challenging. The problem is mostly due to the trade-offs between the relevant qualities for the application. However, in most previous studies only a few qualities have been addressed and investigated. In this thesis we have investigated how IoT system designers could be supported while considering the relevant qualities in making deployment decisions.

In this thesis, we have addressed the problem of supporting the IoT system designers in selecting an appropriate deployment solution by considering multiple qualities. In particular, we tried to answer three research questions. The first concerns identifying the relevant qualities and metrics in edge computing as an initial step for supporting deployment decisions. By systematically investigating the literature, we found the most used qualities in edge computing are *time behavior* and *resource*. We also found a knowledge gap related to qualities like compatibility, maintainability, and adaptability, which are also important for many IoT systems. Moreover, we found that simulation is a very popular tool for measuring qualities and validating the research results.

The second objective (RQ2) was to study the usefulness of edge computing simulators as a key supporting tool for the IoT system designers. In particular, we first investigated what are the relevant qualities and metrics to be measured by simulators, and then we analyzed to what extent the current edge computing simulators support those

metrics. To guide IoT designers for the selection of an appropriate simulator, we provided a comparison table of the simulators with the supported metrics. Moreover, we found that none of the simulators cover more than 5 of the 14 identified metrics, and there is still a need for further development of the edge computing simulators.

In answering the third research question we addressed the challenge of considering multiple qualities for ranking deployment options. We proposed and implemented a multi criteria decision support approach to rank deployment options. We applied the method to deploy Deep Neural Networks in IoT systems, and used simulation to study a concrete case. By measuring multiple relevant metrics in simulation, we showed how the approach can help the designers to rank their deployment choices.

4.2. Future Work

Based on the research presented in this thesis, there are several directions that are interesting to pursue as described below.

4.2.1. Industry Perspective on Qualities

Qualitative methods such as interview and survey can be used to obtain the designers' opinion to provide deeper and richer understanding of the case study [48]. To explore the current deployment of IoT systems, investigating the process and decisions made by practitioners when deploying IoT software components could be a useful research study. Such a study would complement the literature review studies (Paper I, II, and III) and clarify the industry's needs.

4.2.2. Unexplored Qualities and Metrics

Based on the findings from Paper II, there was a lack of metrics for measuring some of the relevant qualities e.g., security, privacy, portability, and maintainability, even though they can be influential in the decision-making process. Thus, quantitatively formulating these qualities and providing metrics for their measurement is an important direction for future work. Moreover, conducting more studies on the impact of these qualities for the deployment of IoT systems will provide a deeper understanding of their role in this domain.

4.2.3. Extending the Decision Support Framework

Having metrics at hand would not be sufficient to support designer decisions. To offer a comprehensive solution, a general decision support framework containing the relevant metrics is required. In Paper IV, a method for ranking DNN deployment options was introduced. The method can be extended and generalized to support various IoT applications. The framework should support evaluating the relevant qualities by different metrics and provide ranking of the appropriate deployment solutions based on the user inputs. The framework should also be flexible enough to work with new qualities or new metrics defined by a designer.

REFERENCES

- [1] M. Chiang, T. Zhang, “Fog and IoT : An Overview of Research Opportunities,” *IEEE Internet Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijnen, B. Jarupan, K. Lin, and T. Weil, “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.
- [3] N. Cochrane, “US smart grid to generate 1000 petabytes of data a year.” <<http://www.itnews.com.au/news/us-smart-grid-to-generate-1000-petabytes-of-data-a-year-170290#ixzz458VaITi6>> [Published: March 23, 2010]
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [5] W. Yu et al., “A Survey on the Edge Computing for the Internet of Things,” *IEEE Access*, vol. 6, pp. 6900-6919, 2017.
- [6] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G. J. Ren, “Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems,” *IEEE Wirel. Commun.*, vol. 23, no. 5, pp. 120–128, 2016.
- [7] J. Du, L. Zhao, J. Feng, and X. Chu, “Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee,” *IEEE Transactions on Communications*, vol. 66, pp. 1594-1608, 2017.
- [8] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, “Multi-objective Optimization for Computation Offloading in Fog Computing,” *IEEE Internet Things Journal*, vol. 5, no. 1, pp. 283-294, 2017.
- [9] Y. Liao, Y. Han, Q. Yu, Q. Ai, Q. Liu, M.S. Leeson, “Wireless Body Area Network Mobility-Aware Task Offloading Scheme,” *IEEE Access*, vol. 6, pp. 61366–61376, 2018.
- [10] Di.B.C. Lima, R.M.B. Da Silva Lima, D. De Farias Medeiros, R.I.S. Pereira, C.P. De Souza, O. Baiocchi, “A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT,” *IEEE 10th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON*, pp. 76–81, 2019.
- [11] S. Svorobej, P. Takako Endo, M. Bendechache, C. Filelis-Papadopoulos, K. Giannoutakis, G. Gravvanis, D. Tzovaras, J. Byrne, T. Lynn, “Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges,” *Futur. Internet*, vol. 11, no. 3, pp. 55, 2019.

- [12] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 544–546, 2018.
- [13] F. Alkhabbas, R. Spalazzese and P. Davidsson, "Characterizing Internet of Things Systems through Taxonomies: A Systematic Mapping Study", *Internet of Things*, vol. 7, pp. 100084, 2019.
- [14] I. Peña-López, "Strategy and Policy Unit of International Telecommunications Union," Geneva, Switzerland: ITU Internet Reports, 2005.
- [15] R. Minerva, A. Biru, "D. Rotondi Towards a definition of the internet of things (IoT)," *IEEE Internet Initiat.*, vol. 1, pp. 1-86, 2015.
- [16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [17] J. Pereira, "From autonomous to cooperative distributed monitoring and control: Towards the Internet of smart things," in *ERCIM Workshop on eMobility*, 2008.
- [18] L. Tan, "Future internet: The Internet of Things," *3rd Int. Conf. Adv. Comput. Theory Eng.*, pp. 376–380, 2010.
- [19] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, "Research on the architecture of Internet of Things," *3rd Int. Conf. Adv. Comput. Theory Eng. Proc.*, vol. 5, pp. 484–487, 2010.
- [20] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," *10th Int. Conf. Front. Inf. Technol. FIT 2012*, pp. 257–260, 2012.
- [21] J. Zhou et al., "CloudThings: A common architecture for integrating the Internet of Things with Cloud Computing," *Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2013*, pp. 651–657, 2013.
- [22] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "An architecture for the Internet of Things with decentralized data and centralized control," *IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, 2016.
- [23] M. Ashouri, P. Davidsson, R. Spalazzese, "Cloud, Edge, or Both? Towards Decision Support for Designing IoT Applications." *Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 155-162, 2018.
- [24] A. Botta, W. De Donato, V. Persico, A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684-700, 2016.
- [25] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, "Edge computing: a survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219-235, 2019.
- [26] P. Mell, T. Grance, The NIST definition of Cloud computing, *Natl. Inst. Stand. Technol.*, vol. 53 no. 6, pp. 50, 2009.

- [27] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.
- [28] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," *ACM MCC Work. Mob. cloud Comput.*, pp. 13–16, 2012.
- [29] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," *Springer Big Data Internet Things A Roadmap Smart Environ.* pp. 169-186, 2014.
- [30] M. Chiang; T. Zhang, "Fog and IoT : An Overview of Research Opportunities," *IEEE Internet Things J.* vol. 3, no. 6, pp. 854-864, 2016.
- [31] Feldman, R. Barton, M.J. Martin, N.S. Goren, and Ch. Mahmoudi, "NIST Fog Computing Conceptual Model," *Special Publication (NIST SP)*, pp. 500-325, 2018.
- [32] M. Barbacci, M. H. Klein, T. A. Longstaff, C. B. Weinstock, "Quality Attributes," 1995.
- [33] ISO/IEC-25010, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Technical report, 2010.
- [34] K. Henningsson, and C. Wohlin, "Understanding the Relations between Software Quality Attributes - A Survey Approach," *Proceedings 12th International Conference for Software Quality*, 2002.
- [35] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, J. Wang, "DEBTS: Delay Energy Balanced Task Scheduling in Homogeneous Fog Networks," *IEEE Internet Things J.* vol. 5, no. 3, pp. 2094–2106, 2018.
- [36] A. Mebrek, L. Merghem-Boulahia, M. Esseghir, "Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing," *IEEE 16th International Symposium on Network Computing and Applications*, 2017.
- [37] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Industrial IoT Data Scheduling Based on Hierarchical Fog Computing: A Key for Enabling Smart Factory," *IEEE Trans. Ind. Informatics.*, vol. 14, no. 10, pp. 4590–4602, 2018.
- [38] Y. Liao, Q. Yu, Y. Han, and M. S. Leeson, "Relay-enabled task offloading management for wireless body area networks," *Appl. Sci.*, vol. 8 no. 8, 2018.
- [39] J.P. Miguel, D. Mauricio, and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products," *International Journal of Software Engineering & Applications*, vol. 5, no. 6, pp. 31-53, 2014.
- [40] ISO/IEC IS 9126-1. Software Engineering - Product Quality – Part 1: Quality Model. *International Organization for Standarization*, Geneva, Switzerland, 2001.
- [41] ISO/IEC 25023. Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Measurement of System and

- Software Product Quality; *International Organization for Standardization*, BSI Group: Geneva, Switzerland, 2016.
- [42] IEEE Standard for a Software Quality Metrics Methodology, in *IEEE Std 1061-1992*, March 1993.
- [43] J.K. Jesson, F.M. Lacey, “How to do (or not to do) a critical literature review” *Pharm Educ*, vol. 6, no. 2, pp. 139-148, 2006.
- [44] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.
- [45] P. Johannesson and E. Perjons, “An Introduction to Design Science,” Springer, 2014.
- [46] S. Teerapittayanon, B. McDanel, and H. T. Kung, “Distributed Deep Neural Networks over the Cloud, the Edge and End Devices,” *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 328–339, 2017.
- [47] P. O. Ostberg *et al.*, “Reliable capacity provisioning for distributed cloud/edge/fog computing applications,” in *2017 European Conference on Networks and Communications (EuCNC)*, , pp. 1–6, 2017.
- [48] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, p. 131, 2009.

ISBN 978-91-7877-158-5 (print)

ISBN 978-91-7877-159-2 (pdf)

MALMÖ UNIVERSITY
205 06 MALMÖ, SWEDEN
WWW.MAU.SE