P2P Fog Computing: Enhancing Fog-based IoT Scenarios with Distributed Hash Tables

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Ahmad Rabay'a
geboren in
Ta'ef, Saudi Arabien

Düsseldorf, Mai 2022

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Dr.-Ing. Kalman Graffi
Korreferent: Prof. Dr. Michael Schöttner
Tag der mündlichen Prüfung: 05.07.2022

Für Noor, Lora, Amr and Lea

Abstract

Internet of Things (IoT) is here to stay, and has changed the technological landscape where existing legacy technologies started to struggle meeting with the emerging needs of IoT. A vital example is Cloud computing, which for decades, has served the internet backbone with a centralized computing and storage model. With the massive growth of IoT, the centralized computing model of the cloud encountered growing challenges, such as network bandwidth constraints, which is the main focus of this research. Fog computing is a new computing architecture which overcomes the limitations of the centralized cloud with a layer of connected, geographically distributed, heterogeneous fog nodes located at the proximity of the end users. Those fog nodes act like mini clouds which have the computing and storage capabilities to handle part of the edge traffic, reducing the bandwidth load of the cloud. However fog nodes are resource constrained in terms of storage capabilities and computational power. Therefore, any resource intensive requests which cannot be single handed by the fog nodes, the cloud needs to be contacted. The challenges arise in this endeavor are discussed in this research. We propose a new model called peer-to-peer fog (P2P Fog) which combines the capabilities of fog computing with peer-to-peer (P2P) network mechanism in order to support the cloud node meeting the increasing demands of IoT devices. Even though each fog node has a limited computational capabilities, our proposed P2P Fog model combines the computational capabilities of the fog nodes creating a larger pool of resources. Which provides an approach to increase the amount of requests handled by the fog nodes, in order to reduce the requests that will be sent through to the cloud.

As a pre-study, we conduct an exploratory research on implementing cloud computing in business enterprise systems. Since the emergence of cloud computing it has been utilized almost everywhere. However, one area which faces many uncertainties to implement the cloud is ERP systems. This research investigates the main challenges that are restraining the market of implementing ERP systems using cloud computing from the market perspective and introduces solution to these challenges. The goal of this research is to helps the market and researchers get a better view about ERP in the cloud, as well addresses some of the issues and limitations of the cloud computing model.

As mentioned earlier, cloud computing has been utilized, at various extends, almost everywhere, including emerging technologies such as IoT. Our novel P2P Fog computing model aims to overcome the limitations faced by cloud computing and fog computing when dealing with the huge data volume resulted from the emergence of IoT. The P2P Fog model enhances the fog computing model by introducing P2P mechanism between the fog nodes at the fog layer to empower the fog nodes capabilities. This allows the sole fog nodes to collaborate with the other fog nodes at the fog layer using P2P overlays to aggregate their computational capabilities and create a larger pool of resources to meet the IoT devices needs at the proximity of the end user. Hence minimize the requests to the cloud and further reduce the bandwidth consumption of the cloud. The P2P distributed hash table (DHT) based overlays considered for this research are: Chord, Pastry and CAN. With a simulator environment configured to imitate an IoT setup, the different P2P Fog configurations are implemented, evaluated and compared in terms of their bandwidth consumption in order to figure out under which overlay the P2P Fog model performs better. The simulation outputs, which are done using a P2P simulation software called "PeerfactSim.KOM", show that Pastry overlay produces better outcomes in terms of reducing the bandwidth consumption compared to cloud computing, conventional fog computing, Chord

P2P Fog, and CAN P2P Fog models.

In conclusion, in this work, we identify and elaborate on one of the main challenges facing the cloud computing and its complementary fog computing raised with the emergence and massive growth of IoT, namely: network bandwidth constraints. This lead to the proposal of our novel computing model that combines fog computing with peer-to-peer network mechanism to form an enhanced three layered client-P2PFog-cloud model named P2P Fog. We show that our proposed P2P Fog model further reduces the bandwidth consumption of the cloud compared to fog computing. While fog computing is helping to achieve faster processing with less delay, P2P Fog extends the capability of fog computing to include demanding and resource intensive computations, while reducing the bandwidth consumption results from contacting the cloud. With these contributions, we address the open questions raised in this Dissertation and enable P2P Fog to be utilized in use cases found in the real world.

Zusammenfassung

Das Internet der Dinge (Internet of Things, IoT) ist nicht mehr wegzudenken und hat die technologische Landschaft verändert, in der die bestehenden Legacy-Technologien anfingen, mit den neuen Anforderungen des IoT zu kämpfen. Ein wichtiges Beispiel ist das Cloud-Computing, das seit Jahrzehnten mit einem zentralisierten Rechen- und Speichermodell als Internet-Backbone dient. Mit dem massiven Wachstum des IoT stößt das zentralisierte Computing-Modell der Cloud auf wachsende Herausforderungen, wie z. B. Einschränkungen der Netzwerkbandbreite, die im Mittelpunkt dieser Forschung stehen. Fog Computing ist eine neue Computing-Architektur, die die Einschränkungen der zentralisierten Cloud durch eine Schicht verbundener, geografisch verteilter, heterogener Fog-Knoten in der Nähe der Endnutzer überwindet. Diese Fog Nodes fungieren wie Mini-Clouds, die über die Rechen- und Speicherkapazitäten verfügen, um einen Teil des Edge verkehr zu bewältigen und die Bandbreitenbelastung der Cloud zu verringern. Allerdings sind die Ressourcen der Fog Nodes in Bezug auf Speicherkapazität und Rechenleistung begrenzt. Daher muss bei ressourcenintensiven Anfragen, die von den Fog Nodes nicht allein bewältigt werden können, die Cloud kontaktiert werden. Die Herausforderungen, die sich dabei ergeben, werden in dieser Dissertation diskutiert. Wir schlagen ein neues Modell namens Peer-to-Peer Fog (P2P Fog) vor, das die Fähigkeiten des Fog Computing mit Peer-to-Peer (P2P) Netzwerkmechanismen kombiniert, um den Cloud-Knoten bei der Erfüllung der steigenden Anforderungen von IoT-Geräten zu unterstützen. Obwohl jeder Fog Nodes nur über eine begrenzte Rechenkapazität verfügt, kombiniert das von uns vorgeschlagene P2P Fog-Modell die Rechenkapazitäten der Fog Nodes und schafft so einen größeren Pool an Ressourcen. So kann die Anzahl der von den Fog Nodes bearbeiteten Anfragen erhöht werden, um die Anfragen, die an die Cloud gesendet werden, zu reduzieren.

Als Vorstudie führen wir eine Sondierungsuntersuchung zur Implementierung von Cloud Computing in Unternehmenssystemen durch. Seit dem Aufkommen des Cloud Computing wird es fast überall eingesetzt. Ein Bereich, in dem viele Unsicherheiten bei der Implementierung der Cloud bestehen, sind jedoch ERP-Systeme. Diese Studie untersucht die wichtigsten Herausforderungen, die den Markt für die Implementierung von ERP-Systemen mit Cloud Computing aus der Marktperspektive einschränken, und stellt Lösungen für diese Herausforderungen vor. Das Ziel dieser Forschung ist es, dem Markt und den Forschern zu helfen, einen besseren Überblick über ERP in der Cloud zu bekommen, sowie einige der Probleme und Einschränkungen des Cloud-Computing-Modells zu behandeln.

Wie bereits erwähnt, wird Cloud Computing in unterschiedlichem Umfang fast überall eingesetzt, auch bei aufkommenden Technologien wie des IoT. Unser neuartiges P2P Fog zielt darauf ab, die Einschränkungen von Cloud Computing und Fog Computing zu überwinden, wenn es um die Bewältigung der riesigen Datenmengen geht, die durch das Aufkommen des IoT entstehen. Das P2P Fog Modell verbessert das Fog Computing Modell durch die Einführung eines P2P Mechanismus zwischen den Fog Nodes auf der Fog Layer, um die Fähigkeiten der Fog Nodes zu stärken. Dies ermöglicht es den einzelnen Fog Nodes, mit anderen Fog Nodes auf der Fog Layer zusammenzuarbeiten, indem sie P2P-Overlays verwenden, um ihre Rechenkapazitäten zu bündeln und einen größeren Ressourcenpool zu schaffen, um die Anforderungen der IoT-Geräte in der Nähe des Endbenutzers zu erfüllen. Dadurch werden die Anfragen an die Cloud minimiert und der Bandbreitenverbrauch der Cloud weiter reduziert. Die P2P Distributed Hash Table (DHT)-basierten Overlays, die für diese Forschung in Betracht gezogen werden, sind Chord, Pastry und CAN. In einer Simulationsumgebung, die so konfiguriert ist, dass sie ein IoT-Setup imitiert, werden die verschiedenen P2P-Fog-Konfigurationen implementiert, bewertet und hinsichtlich ihres Bandbreitenverbrauchs verglichen, um herauszufinden, unter welchem Overlay das P2P-Fog-Modell besser abschneidet. Die Simulationsergebnisse, die mit einer P2P-Simulationssoftware namens "PeerfactSim.KOM" durchgeführt wurden, zeigen, dass das Pastry-Overlay im Vergleich zu den Modellen Cloud Computing, konventionelles Fog Computing, Chord P2P Fog und CAN P2P Fog bessere Ergebnisse hinsichtlich der Reduzierung des Bandbreitenverbrauchs erzielt.

Zusammenfassend lässt sich sagen, dass wir in dieser Arbeit eine der größten Herausforderungen für das Cloud Computing und das komplementäre Fog Computing, die mit dem Aufkommen und dem massiven Wachstum des IoT einhergehen, identifiziert und näher erläutert haben, nämlich die Beschränkung der Netzwerkbandbreite. Dies führte zum Vorschlag unseres neuen Computermodells, das Fog Computing mit Peer-to-Peer-Netzwerkmechanismen kombiniert, um ein erweitertes dreischichtiges Client-P2PFog-Cloud-Modell namens P2P Fog zu bilden. Wir zeigen, dass unser vorgeschlagenes P2P Fog-Modell den Bandbreitenverbrauch der Cloud im Vergleich zum Fog Computing weiter reduziert. Während Fog Computing dazu beiträgt, eine schnellere Verarbeitung mit weniger Verzögerung zu erreichen, erweitert P2P Fog die Fähigkeit des Fog Computing, anspruchsvolle und ressourcenintensive Berechnungen einzubeziehen, während der Bandbreitenverbrauch durch die Kontaktaufnahme mit der Cloud reduziert wird. Mit diesen wissenschaftlichen Beiträgen gehen wir auf die in dieser Dissertation aufgezeigten offenen Fragen ein und ermöglichen die Nutzung von P2P Fog in Anwendungsfällen, die in der realen Welt vorkommen.

Acknowledgements

I would like to take this opportunity to thank the people who has supported, surrounded, and motivated me during my doctoral studies. To my amazing supervisor Kalman Graffi, without your support, none of these would have been possible. It was a great honor to undertake my research under your supervision. Your encouragement and influence in both my professional and personal life was endless, and your kindness was immeasurable, and for that I could never be thankful enough. I would also like to thank my mentor Michael Schöttner for always being there and supporting me whenever I needed.

I would like to thank as well all my colleagues at the "Technik sozialer Netzwerke" work group / Computer Networks for being supportive and friendly since day 1. Special thanks to Andreas Disterhöft, Andre Ippisch and Tobias Amft. To my first office mate Ahmad Cheraghi, and my last office mate Newton Masinde whom I shared many moments and talks about everything. To Raphael Bialon who always been so amazing. To my brother from another mother Raed Al-Aaridhi thanks a lot for being there whenever I needed you.

I would like to thank as well my colleagues at Uniper SE, who has been extremely supportive and helpful. And special thanks to Robert Daamen and Steven van den Berg for their endless support, which made this journey possible.

I am most grateful to my parents and siblings for their support and encouragement, however, it still sadden me that my dad will not be celebrating this with me. This has been a long journey, which would not have been possible if it were not for my wife, Noor. Your care, love, and support was immeasurable. To my kids, Lora, Amr, and Lea, you bring the joy to my days and make me a better person. Love you.

Last, and most importantly, I am grateful to God Almighty for everything in my life.

Contents

Chapter 1

Introduction

The current technological wave of development is focusing on Internet of things (IoT), and their role of changing the computational needs. In this age of IoT everything is more and more connected and those connected devices are producing a massive amount of data, and require a huge volume of resources. Wearable devices such as smart watches and glasses, smart cities, self-diving vehicles, sensor networks and the like are some of many examples to the ubiquity of connected devices [75]. As reported by Cisco, there will be 5.3 billion total Internet users (66 percent of global population) by 2023. While in 2017 a single person had 2,4 devices and a monthly traffic of 16,2 GB, in 2023 the devices count will increase by 50% but the monthly traffic will more than triple to 49,8 GB. Out of the 29.3 billion devices connected to the Internet by 2023, there will be 14.7 billion IoT devices. Moreover, the annual global IP traffic will reach 4.8 Zettabytes per year by 2022, and the monthly IP traffic will reach 50 GB per capita by 2022, up from 16 GB per capita in 2017 [7, 31]. With the rapid growth in the data volume, the data generation speed is also increasing rapidly. A recent analysis of healthcare related IoT application show that 30 million users generated up to 25,000 tuples data per second [19]. Another report in The Economist describes how cows will be monitored to ensure healthier, more plentiful supply of meat for people to consumer. On average, each cow generates about 200 MB of information a year, and it is estimated that there are around 1.2 billion cattle on the planet. That would estimated around 224 PB of data per year globally for cows monitoring only [75]. Another example, is the Boeing 787 which generates 5 Gigabyte of data every second, or autonomous vehicle which generates 1 Gigabyte every second, and it requires real-time processing to make correct decision [67].

Edge devices, especially smart-phones and IoT, are the two major commodities that have facilitated true ”anywhere, anytime, anyhow” user connectivity. Today, the scenarios of smart cities, smart transportation, and smart homes are no longer in the domain of research, but are becoming the new ”normal” [51]. Since IoT devices are usually inadequate in computation power, battery, storage and bandwidth, many IoT computations are outsourced to strong server ends, which are mostly deployed in the cloud. The cloud is considered a solution to deliver services to end users with scalable resources at low cost [31]. Therefore, cloud computing is becoming the overarching Internet approach for storage and information management, and IoT devices become the major outlets. This means that a key task for the next networks generation is the successful integration of cloud computing and IoT devices.

Over the past decade, moving the computing, storage and network management functions to to centralized data centers hosted in the cloud has been the trend [18]. Cloud computing has already been a major computing infrastructure for Internet, as reported in [49], around 90% of Internet users globally are depending on services offered by cloud directly or indirectly. Cloud computing offers many advantages in terms of reducing administration complexity, increasing flexibility, cost reduction and better reliability [14]. Cloud computing makes computing resources such as hardware infrastructure, development platform and applications available as services over the Internet. The services made available are commonly known as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [34, 17]. Because of its high computation power and storage capability, cloud computing has been the answer to many of computational difficulties in the recent decade.

1.1 Context and Motivation

However, with the emergence of IoT, the centralized computing model of the cloud, where most of the computation happen in the cloud, encounters growing challenges. In cloud computing, all data and requests are transmitted to the centralized cloud, although the data processing speed has increased rapidly, the network bandwidth has not increased much, which is becoming the bottleneck of cloud computing for such a huge amount of data. This may result in long unacceptable transmission latency. Some IoT applications might require a very short response time and mobility support, such as smart transportation systems, self driving cars, smart wearable devices, eHealth, and other latency-sensitive applications [7, 39]. The time cost of long distance transmission in a cloud based architecture cannot guarantee the latency sensitive requirements for many IoT applications [48]. The delay caused by transferring data is unacceptable. Moreover, some decisions can be made locally without having to be transmitted to the cloud. Even if some decision have to be done in the cloud, it is inefficient to send all the data to be processed and stored in the cloud, because not all data is useful for decision making and analysis. These challenges, which are caused by the huge growth of IoT, the related requirements with regard to high capacity between the clients and the cloud cannot be addressed solely in dependence on cloud model [39, 40].

A project such as Google robotic cars generates data approximately over 1 GB/sec from the surrounding via sensors and cameras, and this data must be processed by different components in milliseconds. Currently, an onboard computer is deployed in the Google robotic cars, but it is insufficient and cloud participation has become crucial. However, due to low latency and high bandwidth requirements, if data is delayed in a cloud server, a smart car may lose its intelligence and cause accidents [48]. To sum up, as mentioned earlier, the cloud centralization combined with rising number of IoT devices induces many challenges, which are [19]

- Network bandwidth constraints: The huge and rapidly growing number of connected devices is creating data at an exponential rate. It is estimated that an autonomous vehicle generates about one gigabyte per second [39]. To support such resource hungry applications, it is important to keep a high rate data exchange between cloud and IoT devices. However, with the long-thin connection between the cloud and mobile users, the network bandwidth becomes a bottleneck for cloud computing, as predicted in [67].

- Latency Requirement: many IoT applications, such as vehicle-to-vehicle communications, require latencies below a few tens of milliseconds. Mainstream cloud services cannot guarantee such latencies [75].

- Real-time response: many IoT devices rely on real-time response where time plays a vital role whether the response is valid or not, such as wearable devices like Google glasses [19] is very latency-sensitive to deploy on cloud [31].

Edge devices are now smarter and embedding capacities such as processing and storage, as well as richer in functionality, such as decision making and data collection. This shows the way forward toward a highly distributed and dynamic IoT scenario bringing together heterogeneous edge devices, fog computing, and traditional cloud computing [51]. Furthermore, since 2011, the smart-phones have overtook PCs and become the major outlet of service applications for users, and the successful integration between cloud computing and mobile devices is a must. Unlike traditional PC users, mobile users have predictable service demands mostly subject to their locations [11]. For example, a mobile user at the train station tends to be interested with the trains schedule and traffic information, this information will mostly become useless once he/she leaves the train station.

In order to overcome the limitations of cloud computing to meet the demands of IoT and other emerging computing models and paradigms, as well as overcoming the issues between cloud and mobile applications, new kind of cloud computing model is proposed. In this new model the devices that respond to and process the client requests are hosted at the proximity of the end user rather than far away in an unknown location in the middle of the Internet cloud. This kind of computing model is called ”Fog Computing” [34]. Fog Computing is a cloud-like layer set up just above the edge of the internet at the proximity of the edge devices to meet the challenges of low latency, high reliability, mobility, bandwidth and high performance. This Fog layer has the capability to handle part of the edge traffic, reducing the bandwidth load of the cloud [67]. Fog computing extends the traditional cloud computing by placing a light-weight cloud-like facility at the proximity of IoT devices. Being deployed at the localized sites, Fog computing can serve IoT devices and mobile users with a direct short-fat connection compared to the long-thin cloud connection, and provide customized location-aware services for mobile users [49]. Compared to cloud computing, fog computing can provide better service quality with increased data rate and reduced service bandwidth cost [19].

Fog computing integrates network edge devices and cloud center, and is considered a more effective solution to address the limitations of cloud computing. Fog computing is a geographically distributed computing architecture, in which heterogeneous edge devices at the proximity of the end user are connected to collaboratively provide computation, communication and storage services [81]. Fog computing makes computation, communication, control and storage closer to end users by pooling the local resources. Data is consumed by the geographically distributed network edge devices. Therefore, the data transfer time and the amount of network transmission are greatly reduced [21]. Fog computing can meet the demands for real-time and latency-sensitive applications, and remarkably reduce network bandwidth bottlenecks [39].

As so many new fog servers in the close proximity of the edge devices are envisioned, the fog servers only oversee a fraction of the devices, in comparison to the global overview at the cloud. The fog servers need to forward the queries to the cloud or coordinate themselves in order to still resolve all queries. Which challenges arise in this endeavor are discussed next.

1.2 Aim and Challenges

Fog Computing is already a successful approach in reducing internet bandwidth consumption of the cloud while also providing reduced response time and latency. Having a fog layer with multiple fog nodes could alleviate the cloud node from most of the requests that arrive from the various IoT devices, and better utilize the cloud node resources to solve the more complex problems which could not be met by the sole fog nodes resources. But like the IoT devices, fog nodes are resource constrained in terms of storage capability and computational power. Fog nodes could be routers, access points, or even PCs and super computers that provide service to the end edge devices. However, either way, the fog nodes storage and computational capabilities are limited. Therefore, for many resource intensive requests, they need to contact the cloud because the required amount of storage and computations resources cannot be single handled by a fog node. When a fog node does not have the resources and capabilities to satisfy a client's request, the cloud has to be contacted and the internet backbone has to be used again thus cloud bandwidth is burdened. The more we can fulfill the end users needs at the edge of the network, the better bandwidth consumption results we can achieve. This way we minimize the traffic in the Internet backbone, by reducing the number of cloud contacts, resulting in better bandwidth consumption values compared to simple fog architectures.

Because sole Fog computing has limited computational resources to relieve load from the cloud and the network backbone, a new model called peer-to-peer fog (P2P Fog) is proposed in this research. Our proposed P2P Fog is a new architecture model which combines the capabilities of fog computing with the peer-to-peer (p2p) network mechanism in order to support the cloud node meeting the increasing demands of the end users.

While fog computing extended the traditional two layer (client-server) cloud architecture, by introducing a third intermediate layer between the server (cloud) and the clients, to form a three layer architecture (client-fog-cloud). The proposed model in this thesis enhances the three-layered client-fog-cloud by introducing a peer-to-peer mechanism into the fog layer to become client-P2PFog-cloud, as illustrated in Fig. 5. The proposed P2P Fog computing paradigm will empower the fog node by trying to delegate some of the computation to the peered fog nodes in order to avoid the huge delay and bandwidth consumption of contacting the cloud. In this model, in case the fog node does not have sufficient computing and storage capabilities to handle the clients request, it can, using a P2P overlay, delegate some of the computation tasks to the peered fog nodes in order to fulfill the IoT devices requests near the proximity of end users rather than contacting the cloud. Even though each fog node has a limited computational capabilities, combining the computational capabilities of a number of fog nodes create a larger pool of resources which might provide a promising approach to reduce the requests that will be sent through to the cloud. When we could fulfill as many requests as possible at the proximity of end users we would have much less requests to be sent to the cloud and much bandwidth cost saved. So while fog computing is helping to achieve faster processing with less delay required by the emerging network paradigms, P2P fog would extend the capability of fog computing to include more demanding and resource consuming computations which will further extend the capabilities and benefits of fog computing and reduce the bandwidth consumption that results from contacting the cloud.

The goal of this research is to show that our proposed P2P Fog model can further reduce the bandwidth consumption compared to fog computing. Our approach in achieving that is

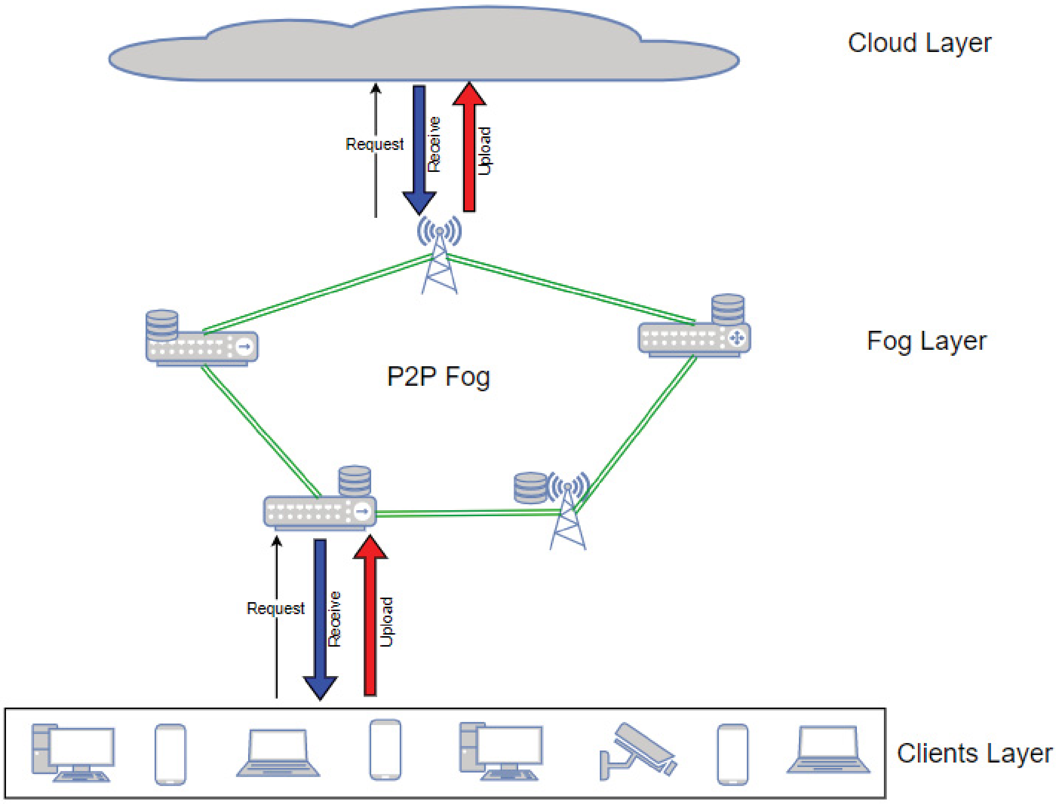

Figure 1.1: Layered Architecture of the proposed P2P Fog model

based on evaluating the simulation outcomes of the proposed P2P Fog model and compare it to the fog computing and cloud computing simulation outcomes. In this research we implement in simulation and evaluate a realistic fog computing workload on different p2p overlays. The simulations are done using a P2P simulation software called ”PeerfactSim.KOM”. This research focuses on an IoT data access setup as a use case to measure and evaluate the bandwidth consumption of the different scenarios. The P2P Fog is a new model motivated by IoT services requirements, user demands, and technology progress that, to the best of our knowledge, has not been addresses before in a coordinated fashion. That said, it is a concept related to some recently proposed approaches, such as it was directly introduced by Chiang et al. in [18] as a research opportunity. As this is a new model, to test this hypothesis, we decided to start with a simple setup, then gradually add new features and details to the model we proposed. The system we aim to achieve in this research, as illustrated in Fig. 1.1, is where the fog nodes at the fog layer are connected with each other using different overlays to see which produces the best results in terms of bandwidth consumption. In this system, the clients can read the files they receive form the fog nodes, and can as well update the files and upload them again to the fog nodes and to the cloud. Another important feature is implementing the best available caching mechanism that fits with the system requirements.

1.3 Research Questions and Approach

In the following, we discuss the research questions which are derived from the aims and challenges described above. Our main goal is to investigate how to reduce the internet bandwidth consumption of the cloud to help it meet the emerging needs of IoT, and how the emerging Fog computing could influence that. The main research questions we addressed in this research are:

1. Can we empower the sole fog nodes to meet the users' requests at the fog layer, in order to reduce the amount of requests that will go to the cloud and avoid the huge bandwidth consumption resulting from that. (Chapter 03)

2. Can fog nodes cooperate with other fog nodes by their proximity using p2p overlays (Chord in this case) by aggregating their computational capabilities to create a larger pool of resources, and can this proposed model reduce the bandwidth consumption for the cloud. (Chapter 03)

3. Does our proposed model perform in the same manner when we change the p2p overlay between the fog nodes from Chord to Pastry and CAN, and which overlay produces the best results in terms of cloud bandwidth consumption. (Chapter 04)

4. Can we understand the major challenges and concerns that are restraining the market from implementing crucial enterprise systems, such as ERP system, into the cloud. As a step in order to propose solutions and specifications to boost the implementation various enterprise systems into the cloud. (Chapter 05)

To answer the previous research questions and achieve the proposed model, this research is conducted in 3 phases:

- Phase 1: is covered in our paper [55], which answers the questions 1 and 2 above. This paper is our first attempt to propose, implement and test the idea of the new P2P Fog model based on Chord overlay. The simulation outcome results of the proposed model are compared against the fog computing and cloud computing outcomes in terms of bandwidth consumption. Our [55] paper contributes to the initial model of the P2P Fog, where requests go from the clients to the fog nodes at the fog layer, and responses go the other way around, from the fog nodes to the clients. The fog nodes are connected together with a Chord overlay to form the P2P Fog layer, and could cooperate together to meet the clients requests. Last but not least, the fog nodes in the P2P Fog layer can send requests and receive responses from the cloud. As explained in details in Chapter 03.

- Phase 2: based on the results achieved in Phase 01, we further implement, adjust and test the P2P Fog model based on Pastry and CAN overlays, in addition to the Chord overlay which is implemented and tested by us in Phase 01. Then, based on the simulation outcome results of the three overlays, we evaluate and compare the three overlays scenarios to decide which scenario produces the best outcomes in terms of bandwidth consumption, as detailed in our paper [54]. In [54] paper we follow the same approach as our previous paper [55], with significance in the P2P Fog layer, as it introduces two different overlays between the fog nodes namely: Pastry and CAN, in addition to Chord, to check which performs better. As explained in details in Chapter 04. This phase answers to the research question 3.

- Pre Phase 1, which is covered in our paper [53]. The driving force behind this paper is to investigate the main challenges that are restraining the market from implementing Enterprise systems, such as Enterprise Resource Planning (ERP), using cloud computing, and introduces solutions to these issues. In this paper we address the market views towards these issues through an exploratory study with the market representatives. The assembled issues and proposed solutions are then validated based on specifications by the Cloud Security Alliance (CSA). Our final results of this research is to help understanding the customers' requirements, protect the customers and reduce their fears toward the cloud. This phase answers to research question 4, and explained in details in Chapter 05.

1.4 Contributions

The contributions of this thesis are in three publications that provide answers to the research questions presented in Section 1.3. These publications form the body of the three chapters (Chapters 3 to 5). In the following, we summarize the contributions of each publication:

- Development of a new novel computing model named P2P Fog which combines both fog computing and p2p technologies to empower the fog nodes capabilities and further reduce the bandwidth consumption of the cloud to meet the emerging needs of IoT by fulfilling the IoT devices requests at the proximity of the end user device and reduce the amount of requests that will be sent through to the cloud node. The P2P Fog is a new model motivated by IoT services requirements, user demands, and technology progress.

- Evaluation of the proposed P2P Fog model, using a simulator environment, under 5 different scenario settings; 3 of which are p2p distributed hash table (DHT) based namely: Chord, Pastry and CAN, and 2 are basic fog computing and cloud computing scenarios. The simulator environment is configured to imitate an IoT setup. The different scenarios are implemented and evaluated using the simulator and compared against each other and the outputs were evaluated and compared based on different configuration vales to figure out under which overlay our P2P Fog model performs better by producing better outcome results in terms of bandwidth consumption.

- Providing a set of simplified issues, best practices, and specifications toward implementing ERP in the cloud from both customers and providers point of views. The results have been then validated based on specifications from the Cloud Security Alliance (CSA), and could help the market and researchers get a better view about ERP in the cloud.

1.5 Outline

In this introductory Chapter 1 we introduce and motivate our proposed peer-to-peer fog computing model (P2P Fog) (Section 1.1), as well as the aim and challenges of the thesis (Section 1.2), the related research questions (Section 1.3), in addition to the contributions of the thesis (Section 1.4).

In the next chapter, we introduce the background and overview of this thesis with a focus on internet of things (Section 2.1), cloud computing (Section 2.2), Fog computing (Section 2.3), peer-to-peer technology (Section 2.4), and the PeerfactSim.KOM simulator (Section 2.5).

Chapter 3 introduces and implements the initial architecture of the proposed P2P Fog model by introducing p2p mechanism (Chord overlay) between the fog nodes at the fog layer. The simulation outcome results of the proposed model are compared against the fog computing and cloud computing outcomes in terms of bandwidth consumption.

Chapter 4 further extend the model proposed in Chapter 3 to include 2 additional P2P DHT based overlays (Pastry and CAN). The 2 newly implemented overlays in addition to the Chord overlay implemented previously in Chapter 3. The simulation outcome results of the three overlays are then evaluated and compared to decide which scenario produces the best outcomes in terms of bandwidth consumption

Chapter 5 investigates the main challenges that are restraining the market from implementing Enterprise Resource Planning (ERP) systems on the cloud, and proposes solutions to these challenges. The assembled challenges and proposed solutions are then validated based on specifications by the Cloud Security Alliance (CSA).

Finally, we conclude our contributions and findings in Chapter 6 and discuss possible future directions.

Chapter 2

Background and Related Work

This chapter presents relevant background overview regarding the technologies that influenced and impacted the idea of P2P Fog computing, staring with Internet of Things (IoT) (Section 2.1) which was relying on Cloud Computing (Section 2.2) to provide the needed network connectivity backbone, but as IoT evolved, cloud computing was not able to meet the massive connectivity requirements of the IoT devices. This is where Fog Computing (Section 2.3) stepped in to fill the gap and meet the emergent connectivity needs of the IoT devices. To complete the picture three other topics are covered in this chapter, which are: Caching in Fog Computing (Section 2.4) which is an important part of out proposed P2P Fog model. DHT-based Peer-to-Peer Networking (Section 2.5) which introduces the technology and the overlays we use to connect the fog nodes together in our proposed model. PeerfactSim.KOM (Section 2.6) which introduces the simulator we use in our research to simulate and plot the different scenarios and findings.

2.1 Internet of Things

In the digital era, the Internet and wireless networks played a major role. Before this digital era, the control and execution of tasks was mainly human to machine oriented and would take much time. Currently, with the advanced network and wireless communications, and intelligent computing, a novel new technology called Internet of Things (IoT) has emerged [46].

IoT is an evolutionary emergent technology that gained huge attention in the science and engineering applications for solving problems without the physical intervention of human-machine contact [46]. The term Internet of Things was first introduced in 1999 [35]. Started in the context of supply chain management, and later evolved to cover wide range of applications like health-care, transportation, education, agriculture, utilities, etc. [71]. Even though with time the 'Things' part of the original term has changed, the main idea of sensing information without a human intervention remains the same [38]. The advancement in internet technologies helped to have a wider and stronger connectivity between the IoT objects [46]. An overview of the IoT offering is illustrated in Fig. 2.1.

IoT is considered by the US National Intelligence Council (NIC) as a "Disruptive Civil Technologies” [20]. NIC expected that "by 2025 Internet nodes may reside in everyday things – food packages, furniture, paper documents, and more” [9]. IoT is transforming the static

Internet into a fully integrated Internet where a network of interconnected objects sense information from the environment and interacts with the physical world, in addition to use Internet standards to provide transfer services, analytics, and communications [38]. Supported by the spread of devices with open wireless technology capabilities such as Bluetooth and Wi-Fi as well as embedded sensor and actuator nodes [16].

IoT is becoming the new normal where smart devices, smart cities and self-driving cars are becoming more and more common. Only in 2011 the number of interconnected devices world wide did overtake the actual number of people on the planet [38]. In 2023 it is expected to reach 29.3 billion devices connected to the Internet, which is almost 3 times the population [7].

IoT is defined by Kranenburg as "a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'Things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network" [74].

IoT architecture consists of a collection of huge number of sensors, actuators, cloud services, communications, users, developers, and enterprise layer [57]. The four main IoT implementation building blocks, include: the Things (Devices), Network Infrastructure, Gateways, and Cloud Infrastructure [46][57]:

- Things (Device): IoT systems are based on devices that provide sensing, actuation, control, and monitoring activities [57]. In addition to collecting data from other devices and exchanging data with other connected devices and application. IoT devices can, based on their computational constraints (i.e. memory, processing capabilities, communication latencies, and speeds), either process their data locally or send it to servers, mostly cloud based, for processing or performing some tasks [57].

- Network infrastructure: such as Routers and Repeaters, which help in providing the control over the provided information and allows smooth and secure flow.

- Gateways (Communication): is used for the connectivity purpose and it performs the communication between devices and remote servers. IoT communication protocols generally work in data link layer, network layer, transport layer, and application layer. However, some IoT devices use Wifi to communicate without the need for a a further gateway such as for ZigBee.

- Cloud infrastructure: equipped with the information storage and computing capabilities, which allows the analytic, logical, and other computing abilities. However, in some IoT scenarios the IoT devices directly communicate with a server not hosted in the cloud.

There are many key utility factors that any IoT must characterized by, such as [80, 57]:

- Dynamic and self adapting: IoT devices are capable to dynamically adapt with the changing operating conditions, user's context, or sensed environment.

- Self-configuring: IoT devices have self-configuring capability where a large number of devices can work together to provide certain functionality (such as weather monitoring).

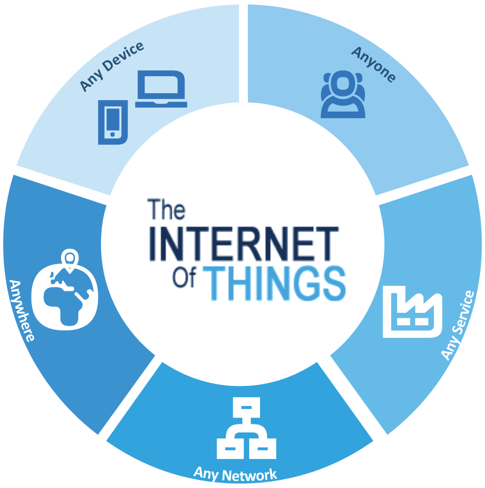

Figure 2.1: Overview of Internet of Things (IoT) [46]

IoT devices have the ability to configure themselves, setup the networking, and get latest upgrades with minimal user intervention.

- Interoperable communication protocols: most IoT devices support a number of communication protocols and can communicate with any different other devices, unless highly proprietary.
- Unique identity: Each of IoT device has a unique identity, and provides an interface which allows users to contact and query the IoT devices, monitor its status, and control it remotely.
- Context-awareness: Based on the sensed information the IoT device gains knowledge about the surrounding physical and environmental context. Therefore, the decisions that the IoT device take thereafter are context-aware.

As introduced, IoT is currently the major trend in technological development, as everything is more and more connected and producing a massive amount of data which requires a huge volume of resources to be stored and processed. Since IoT devices are usually limited in computation power, battery, storage and bandwidth, many of their computations are outsourced to strong server ends, which are mostly deployed in the cloud, which is becoming the major approach for storage and information management in the Internet, as introduced in more details in the next section. However, based on the current pace of IoT advancement and the huge exponential increase of produced data volume, that will put pressure on the cloud, and question the capability of the traditional form of the cloud to keep up with the IoT advancement pace.

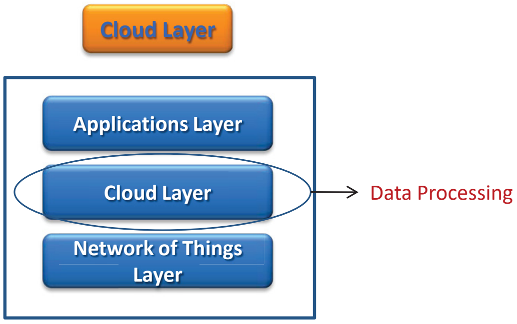

Figure 2.2: Cloud based architectures of IoT [46]

2.2 Cloud Computing

Cloud computing is flexible and scalable techniques which supports IoT systems with various services, include; storage, processing power, tools and analytics, platform, and infrastructure. Cloud based architecture became very popular in IoT systems due to the unclear nature of data sensed and produced by an IoT devices. As illustrated in Fig. 2.2, the IoT system have a cloud centric architecture where the cloud is between the applications and network of things [46].

Over the past decade, moving computing control and data storage into the cloud has been the trend, where computing, storage, and network management capabilities are moved to centralized data centers [18]. Cloud computing has already been a major computing infrastructure for Internet, as reported, around 90% of global Internet users are relying on services provided by cloud [49].

Traditional business enterprise applications were a major source of costs and complexity for companies, as they require the ability to handle large number of users and large amount of data and flexibility to expand quickly. They need a data center with office space, power, networks, servers, and storage. Complicated software, a team of experts to install, configure, and run them. They need development, testing and production environments [29, 43]. One vision of the 21st century computing is to move the computing and data away from desktop and portable PCs into large data centers, where applications can be delivered as services through the Internet, same thing applied to hardware and software infrastructure. Cloud service clients can adjust their infrastructure according to their needs and budget. Clients can add more capacity at peak demands and remove unneeded capacity with less effort, and may reduce costs for clients and increase utilization for service providers [24]. Cloud computing is proposed as a solution for these requirements, which enables sharing of resources in large pool of systems linked together to provide IT services [52]. Cloud computing defined according to Sharif as [66]

"An accessible resource of hardware and software which an organization or individual can harness, anywhere in the world via the Internet."

The general architecture behind cloud computing is a massive network of servers linked together and able to dynamically provide and configure computing and storage resources to achieve the requirements of the running application [52]. The cloud architecture typically involves multiple central and/or distributed computing resources communicating with each other over application programming interfaces, usually web services to provide users with scalable and abstracted IT capabilities, including software, development platforms, and virtualized servers and storage [8, 23]. Applications built on cloud architectures are used only when needed to perform a specific job, and then relinquish the unneeded resources after the job is done. This brings up hundreds of virtual servers on-demand, runs parallel computation using distributed processing framework, then shuts down all the virtual servers releasing all its resources back to the cloud with low effort from the customer and reasonable cost instead of buying these resources. These cloud Applications run in-the-cloud where the physical location of the infrastructure is determined by the provider, and the reliability and scalability logic of the underlying services is hidden inside-the-cloud [76].

Cloud architectures address key difficulties regarding large-scale data processing in traditional data processing, such as: (1) get as many computing resources as an application needs, (2) get the resources when needed, (3) distribute large-scale job on different computing resources and run processes on them, (4) auto-scale up and down resources based on dynamic workloads, and (5) get rid of those resources when the job is done [76]. Customers of cloud computing do not actually own the physical infrastructure they use, instead they rent these resources from a third party provider and pay only for what they use [8]. Cloud computing systems must address four main fundamental characteristics [23]

- Highly abstracted (virtualization): hardware infrastructure is hidden from the users, so no need for the customer users to manage physical hardware or storage.

- Pay-as-you-go: payment is based on the actual consumption of the customer, which reduce the potential cost for the customer and increase the provider resources utilization as unused capacity can be quickly offloaded.

- Immediately scalable: via dynamically (on-demand) provisioning of resources, organizations can scale up or down their IT infrastructure (CPUs, storage, networking or number of users) according to the consumption needs, avoiding costly upgrades, waiting time, and capacity constraints. Organizations no longer need to choose between too much or too little [8, 23, 58]. Usage and cost can be scaled up or down to the right number of users.

- Multi-tenant: refers to a principle in software architecture when single instance of a software application serves multiple customers (tenants), with no or limited customization ability. In contrast with single-tenancy where each customer has their own software instance and wide range of customization options. In a multi-tenancy environment each client organization works with a virtual application instance, where multiple customers share the same application instance, running on the same operating system, on the same hardware, with the same data-storage mechanism. Thus customers do not share or see each other's data. With multi-tenancy customers can share the price of the application and database instances, and no longer need to worry about software upgrades expenses [13].

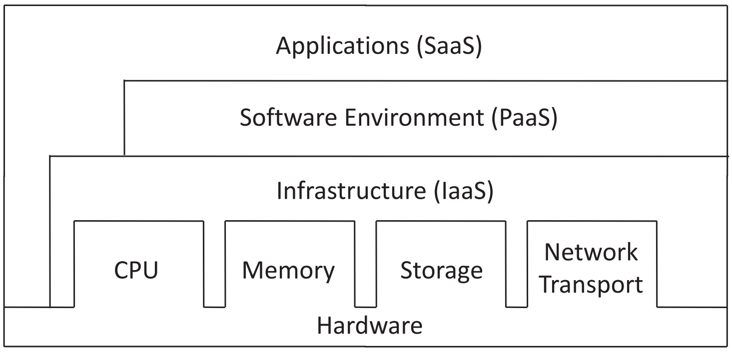

Figure 2.3: Cloud services [58]

Cloud computing taxonomy can be categorized into different families of services, illustrated in Fig. 2.3, which are delivered and consumed in real-time over the Internet [23, 58]:

- Infrastructure-as-a-service (IaaS): Deliver computer infrastructure as a web-based service. Rather than purchasing servers, storage, data center space, or network equipments, clients just subscribe to these resources from a supplier with a usage-based payment.

- Platform-as-a-service (PaaS): Web-based service that provides developers with a platform including all the facilities required to support the complete end-to-end life cycle of building and delivering web applications and services, which include developing, testing, deploying and hosting facilities.

- Software-as-a-service (SaaS): A multi-tenant web-based service that support set of processes. It uses common resources and a single instance to support multiple customers simultaneously. Different application can be delivered as SaaS, these application can either target business users (e.g. CRM/HR/ERP), or private users (e.g. collaboration programs and social networking).

Moreover, cloud services could be categorized according to how public and external the cloud is. This distinction essentially differentiates the degree to which the cloud in externalized from the organization, which includes public, private and hybrid clouds [23]. Where public clouds service can be accessed through the internet via web applications, using systems shared among multiple users, with vary degree of data privacy control. Private clouds are computer architectures built, managed and used internally by an enterprise. The one in between is a hybrid cloud where customer's computing environment consists of multiple resources in-house and others provided externally. Hybrid cloud is used when the organization might use a public cloud service, but does not want to get rid of the existing hardware, this model can be used by the customers to accommodate considerable load, or as a transition first step to using the cloud exclusively [23].

Even though the Cloud contains a massive storage and computing capabilities, it faces many challenges while trying to cope with the development pace and requirements of IoT technological wave. Due to its centralized nature, cloud computing faces challenges in terms of bandwidth, latency, and real-time response when trying to meet the increasing demands of IoT devices. Therefore, a new computing model is needed to overcome the limitations of the Cloud and meet the emerging needs of the IoT wave.

2.3 Fog Computing

In order to overcome the limitations of cloud computing and meet the demands of IoT and other emerging computing models and paradigms, a new kind of cloud computing model has been proposed. In this new model the devices that respond to and process the client requests are hosted at the edge of the local network rather than far away in an unknown location in the cloud. This computing model is called "Fog Computing" [34], which extends cloud computing by introducing an intermediate Fog later between mobile devices and cloud [49].

Fog computing is considered as a more effective solution to address the limitations of cloud computing. [18]. Fog computing is a geographically distributed computing architecture, in which various heterogeneous devices at the edge of the network are connected and collaboratively provide elastic computation, communication and storage services to end users [81]. Data is consumed by the edge devices, therefore, the data transfer time and the amount of network transmission are greatly reduced [21]. The fog paradigm effectively meets the demands of latency-sensitive applications, and remarkably ease network bandwidth bottlenecks [39].

The idea of Fog Computing is placing a light-weight cloud-like layer at the proximity of mobile users. Being deployed at the localized sites such as, shopping centers, airports, train stations, bus terminals, streets, etc. Fog computing can serve mobile users with a direct short-fat connection, and provide customized location-aware services for mobile users, as illustrated in Fig. 2.4 [49]. The term "Fog Computing" was introduced by Cisco as a new model of computing to ease wireless data transfer to distributed devices in IoT [34, 15]. The motivation of Fog computing is to place the content and application services as close as possible to their consumer [49]. This resource-rich layer between end devices and cloud fulfills the needs for low latency, high reliability, mobility, bandwidth and high performance [82, 39]. Fog computing can provide better quality of service in terms of speed and reduced data traffic over the Internet [34].

Some of the important features of IoT include location awareness, wireless connectivity, faster processing requirements, low latency, real time interaction and mobility. In order to meet these requirements the clients can benefit from the processing nodes located very close to their location that can communicate with the clients over low delay wireless links [34]. Fog computing can provide effective ways to overcome many of the challenges introduced with the emerging IoT that cannot be adequately handled by the current cloud computing models, such as: [18]

- Network Bandwidth: The growing number of connected IoT devices is creating data at an exponential rate. With the long-thin connection between cloud and mobile users, the network bandwidth becomes a bottleneck, as predicted in [39].
- Latency Requirement: most IoT applications require latencies below tens of milliseconds.

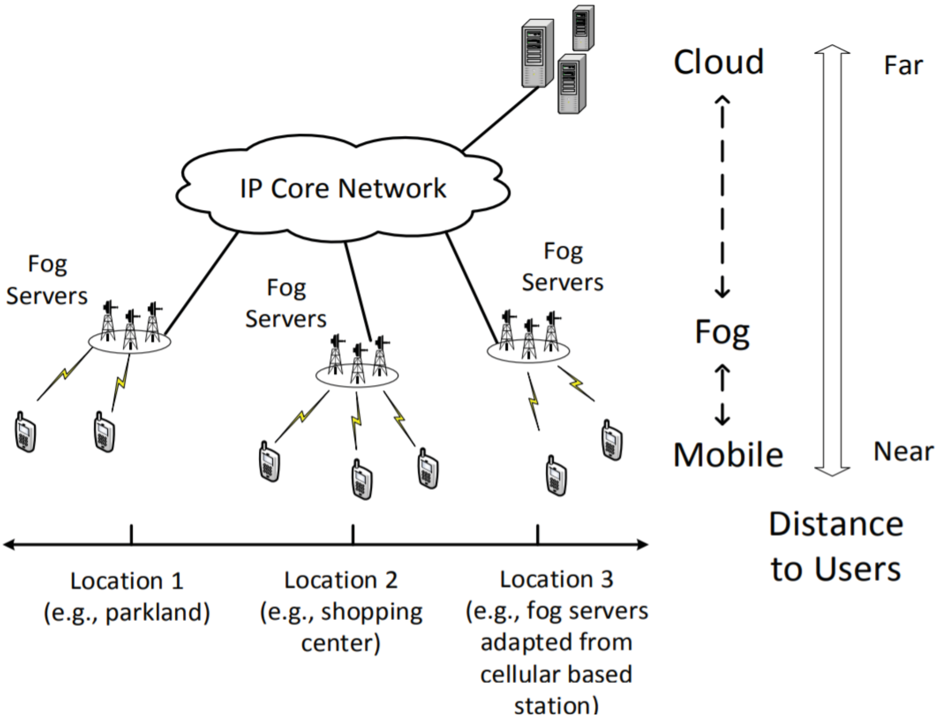

Figure 2.4: Fog computing architecture [49]

Mainstream cloud services cannot guarantee such latencies [39].

- Real-time response: many IoT devices rely on real-time response which is too latency-sensitive to be on cloud [81].

Fog computing fully utilize the computation and storage capabilities of devices at the network edge. The data gathered by end devices can be processed and stored locally. As not all the data will be transferred to the cloud, this will reduce the amount of network transmission, saves bandwidth, as well as accelerates data analysis and decision making [39].

According to Bonomi et al., [15] "fog computing is a highly virtualized platform that provides compute, storage and networking services and typically located between end user devices and the cloud data centers hosted within the Internet." The main feature of fog computing is its ability to support applications that require low latency, location awareness, and mobility, which is made possible by deploying the fog nodes very close to the end users in a widely distributed manner [34]. Another definition is proposed by [75] which encompass all the key ingredients of the fog: "Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties.".

The fog layer is composed of geo-distributed fog nodes which are deployed at the local premises of mobile users, with a purpose to place a handful of computing, storage and communication resources in the close proximity of mobile users. Fog nodes are key component of any fog-based computing, which is considered as in [82] "In fog computing, facilities or infrastructures that can provide resources for services at the edge of the network are called fog nodes." or in [68] "a fog node is the physical device where for computing is deployed." Fog nodes can be adapted from the existing network components, like routers, switches, wireless access points, and cellular base stations by upgrading the computing and storage resources and reusing the wireless interface [49]. Fog computing puts additional computing and storage resources near the edge. Being close to the mobile users, fog nodes provide fast rate services to mobile users via local short distance high rate wireless connections. Fog nodes directly communicate with mobile users within its wireless coverage through single-hop wireless connections [49].

Therefore, a fog node is a generic virtualized equipment with on-board storage, computing and communication capabilities which can intelligent and adaptive to serve the users [49]. The three resources are detailed below:

- Storage: Because of their location awareness, fog nodes in a specific service area predict mobile user's demand on information and pre-cache the frequently requested contents accordingly using a proactive way [11]. According to [11] information demand patterns for mobile users are predictable to an extent and propose to proactively pre-cache the desirable information before users request it.

- Computing: Fog nodes are intelligent computing system, which autonomously and independently serve local computation and data processing requests from mobile users [63]. Fog nodes as well collect the environmental data from mobile users at the deployed spot, and transmit the collected big data to the cloud for in-depth data analysis for more strategic and valuable insights for planning and decision making [49].

- Communication: A fog node is a highly virtualized system that acts as an intermediate networking component which connects mobile users with fog nodes and the cloud.

So a device to become a potential fog node it has to have computing power, storage and networking capabilities, and of course be connected to the Internet [50]. However, fog nodes can be categorized based on the role edge devices play into two categories [50]:

- fog nodes as mini clouds, with dumb edge devices acting as data produces/consumers, such as sensors and actuators.

- fig nodes as mini clouds, with smart edge devices that have significant computing capabilities, such as cars and mobile phones.

Fog computing is usually cooperated with cloud computing to form a three layer service delivery model. The architecture of fog computing is composed of the following layers as illustrated in Fig. 2.4 [39]

- Terminal (end user) layer: The closest layer to the end users, it consists of mobile and IoT devices like; smart phones, smart vehicles, sensors, cameras, etc. These devices are widely geographically distributed in general, and responsible for transmitting the data to the upper layer for processing and storage.

- Fog layer: This layer is located on the edge of the network at the proximity of the end users. The fog layer is composed of a large number of fog nodes, which are widely distributed between the end user devices and the cloud. These fog nodes are capable of computing, transmitting, and temporarily storing data received from the end user devices. The fog layer is connected to the cloud data centers by IP network.

- Cloud layer: which is the top layer of the hierarchy. It consists of multiple high performance servers and storage devices. It has powerful computing and storage capabilities to support extensive computing analysis and permanent storage of an enormous amount of data.

In this architecture, each end device is connected with one of the fog nodes through wireless access technologies, such as Wireless Local Area Network (WLAN), WiFi, 3G, 4G, 5G, Bluetooth, etc. The fog nodes can be interconnected by wired or wireless communication technologies, and each fog node is linked to the cloud by IP network [39].

The most significant characteristics and advantages of fog computing are listed as follows:

- Low latency and real time interactions: To reduce latency, the workload should better be finished in the nearest layer which has enough computation capacity to the end user device [67]. Fog nodes at the network edge acquire the data generated by the IoT devices such as sensors, process and store the data. This significantly reduces data movement across the Internet and provides speedy high-quality localized services. Therefore, it enables low latency and meets the demand of real-time interactions and time sensitive applications [14].

- Save bandwidth: Fog computing provides the computation and storage capabilities at the network edge, to perform the processing and storing closer to the end user. So most of the computations tasks are performed locally, and only part of the useful data is transmitted to the cloud over the Internet, so the bandwidth between the edge device and the cloud is saved [39, 67]. High bandwidth can reduce transmission time, especially for large data like videos. For short distance transmission, we can establish a high bandwidth wireless access to send data from edge device to the fog node [67].

- Support mobility: Fog computing support for mobile devices makes mobility an important characteristic. In addition to the mobility of edge devices, fog nodes cloud be mobile such as deploying fog servers on vehicles or trains [39].

- Location awareness: Being closer to the network edge makes the fog nodes aware of the end device location. This enables applications to provide services better suited to user and device location. The information required by the end device can be specific to their geographical location. Because the cloud offers global and centralized services, it can be difficult for cloud systems to maintain a level of location awareness with devices that are mobile and geographically distributed [72].

- Geographical distribution: Fog computing consists of a large number of widely geographically distributed nodes that have the ability to track the location of end devices in order to support the mobility. Fog computing ensures the processing and storing of information closer to the proximity of end devices rather than in centralized data centers far away from end users. This characteristic supports faster analysis, better location-based services, and more powerful real-time application capabilities [39].

- Heterogeneity: Fog nodes have different forms and are deployed in a wide variety of environments [41]. They usually range from servers, routers, gateways, access points, base stations, etc. These hardware platforms have varying levels of computation and storage capabilities, run various kinds of operating systems (OS), and different software applications [39]. Moreover, the network infrastructure of fog computing is also heterogeneous, which include high-speed links connecting to data centers, and wireless access technologies such as; WLAN, WiFi, 3G, 4G, ZigBee, etc connecting to the edge devices [14].

2.4 Caching in Fog Computing

The main goal of fog computing paradigm is move the services and resources closer to the edge of the network and away from the cloud, and empower the fog nodes to perform the computations and storing services solely without involving the cloud. However, with the significant growth of the data volume from IoT devices, it became extremely important to empower the fog nodes for more efficient storage and computation capabilities [83]. Efficient data cashing technique at the fog nodes is considered vital for fog computing, which plays an important role in reducing the computational complexity of the cloud, and improving the network performance in terms of hit ratio, response time and bandwidth [10, 83, 3].

When an edge device sends a request for data to the fog server, if the data is available in the fog server, by means it has already been cached before, the fog server will respond to the requested edge device with the target data [3]. Otherwise, the fog server forwards the request to the cloud server, which in return responds to the fog server with the requested data. The fog server then caches the data and delivers it eventually to the requesting edge device. However, the emergence of IoT raises the demand for more efficient fog computing caching techniques, otherwise, low quality caching mechanism might add an extra burden to the network by increasing the miss ratio [1, 3]. Hence, fog servers do not have the capabilities to store the entire content of the cloud, it must be empowered with the most relevant content to serve the end users associated with each specific fog server. That is why there are various cache types, and each one could be more "appropriate" for a specific scenario requirements [3].

The appropriate term introduced before varies based on the different situations and user demands which could have a huge impact on the cache content. Conventional caching techniques focus on the location of the fog server in order to predict the demands of the edge devices [49, 2], while other cache techniques consider some content ranking as the base for their criteria to predict what to cache, such as popularity, or most recent [60, 3]. The available fog based caching techniques could be classified into the following 3 categories [83]:

- Caching based on functionality: in this category the cached content is stored in the control level rather than the data level, which provides better latency and consumed bandwidth. However, it is not ideal for fog based scenarios since content would probably be handled more efficiently at the edge of the network.

- Caching based on location: this caching technique is suitable for fog-based scenarios. In this category, the data is cached in the fog node with the highest connectivity based on its geographical position. This caching category is based on combining probabilistic caching and geographical factors. This caching technique enhances the fog computing performance in terms of data retrieval.
- Caching based on content: This caching approach is based on suggesting that the Least Recently Used (LRU) approach would probably be the most suitable caching technique for fog-based scenarios.

However, some other new more complex caching techniques have emerged which consider other criteria such as; clustering, type of application, and data similarity [83, 3].

2.5 DHT-based Peer-to-Peer Networking

In the scope of this work peer-to-peer (P2P) is necessary for routing the communication between the fog nodes. Peer-to-peer (P2P) is a distributed network architecture where all participating devices are independent regarding processing power, storage capacity and bandwidth. The devices have the capability to access each node in the network and are capable of sharing their resources. So in respect to the classic server-client architecture the devices in a P2P network can represent both roles, client and server [64]. In a p2p architecture nodes are able to directly exchange resources and services between themselves without the need for centralized servers [78]. P2P system scan be classified in centralized/decentralized systems, further, the decentralized systems can be classified in structured/unstructured systems [65]. Centralized peer-to-peer system mix features of centralized and decentralized architectures. There are one or more central servers, which peers send messages to in order to determine the address of peers that contain the desired resource. Once a peer has its information, it can directly communicate with other peers, like a decentralized system, without going through the server.

In a decentralized peer-to-peer system, every peer has equal rights and responsibilities. Peers only have a partial view of the P2P network and offer data/services that may be only relevant for some queries/peers. The difference between structured and unstructured P2P networks lies in how queries are being forwarded. In an unstructured P2P system, each peer is responsible for its own data. Peers keep track of a set of neighbors that it may forward queries to. There is no fixed topology in the nodes and no strict mapping between the identifiers of objects and those peers. Joining a unstructured system has a low cost of resources because it simply copies its neighbors links. However request queries can be very inefficient due to the restricted knowledge of the nodes. They have no global knowledge of data placement. A query in an unstructured system does not guarantee a result whether it contains the resource or not [78, pp.40]. Structured P2P systems use a mapping between data and peers. Data placement is under the control of certain predefined strategies [77]. In structured P2P systems the nodes are organized in a fixed topology such as a ring as in Chord or a mesh as in Pastry. Joining a structured peer-to-peer system requires a strict procedure to set up its position. This has a huge maintenance cost but will support a efficient lookup of the queried requests. Nodes in this system have a global knowledge about the resources. The structured system can guarantee a successful lookup if the resource exists [78, p. 50].

Moreover, overlay networks create a virtual topology on top of the physical topology [22]. Requesting specific resources in a large P2P Network can cause flooding because of the amount of participating nodes and the potentially needed bandwidth if many nodes try to reach an unknown node that has the requested resources. A solution for better scaling resource lookups is the Distributed Hast Table (DHT). DHT is a distributed data structure that provides a bijective key to resource-value mapping function to accomplish easier and more determined routing [62]. The P2P overlays Chord and CAN are implementations of those definitions. The participating nodes create a virtual network to share their resources and use DHT to archive better scalability, reduce flooding, and also guaranty requested resources are found if they exist in the network. Furthermore, there are also DHTs that are focused on anonymity, geographical topology, social closeness as explained in [6, 5, 4]. In the following, the three overlays used in this research are briefly explained.

2.5.1 Chord

Chord is a structured, decentralized peer-to-peer system that implements the characteristics of P2P and the distributed hash functionality of DHT using the SHA-1 as a base for hashing [33]. The Chord protocol supports just one operation: given a key, it maps the key onto a node. Chord uses a variant of consistent hashing to assign keys to Chord nodes [70]. Because of the consistent hashing functions (often SHA-1 [30]) each node gets roughly the same amount of keys and chord is therefore relatively balanced [70, p.149]. In Chord every node has a unique overlay identifier. The identifiers are arranged in a ring structure. Every node has a successor and predecessor. The predecessor is the next node in the circle in the counter-clockwise direction. The successor, in contrary, is the next node in the clockwise direction. But since this is inefficient ($\mathcal{O}(n)$), beside the predecessor and successor, Chord is using a bounded routing table with only about $O(\log\ n)$ entries, called fingers, where n is the amount of nodes is the network. Nonetheless resource lookups need just up to $O(\log\ n)$ sent messages [70, 79]. This is achieved by the way the fingers are chosen. Fingers are short-cuts to some nodes further away. Every lookup is sent to the closest finger, which is still preceding the queried identifier [12]. These fingers allow to skip nodes and act as a shortcut to find the key faster. In an n -node network, each node maintains information only about $\mathcal{O}(\log n)$ other nodes, and a lookup requires $\mathcal{O}(\log n)$ messages [70, 27].

Each data is assigned to a key, and storing the key/data item pair at the node, whose corresponding responsibility interval contains the key. To lookup data, the node queries the responding key, this results in a query that is routed towards the peer, whose responsibility interval contains the key [36, 12].

2.5.2 Pastry

Pastry is a decentralized, scalable, and self-organizing network overlay protocol [59]. It falls under the category of the structured peer-to-peer systems. Each node in the Pastry network has a unique numeric identifier (nodeId) and holds routing information to nodes with nodeIds numerically closest to its own. This routing information is called a leaf set. When presented with a message and a numeric key, a Pastry node efficiently routes the message to the node with a nodeId, which is numerically closest to the key among all currently live Pastry nodes [59]. Each node is responsible to keep track of its neighbors and therefore responsible to notify the application of new arrived nodes: Nodes, which are offline and nodes, which reconnect to the overlay [59]. A lookup requires $\mathcal{O}(\log N)$ messages, but because Pastry node's leaf set containing nodeIds numerically closest to its own, it can take locality of the network into account. Pastry tries to minimize the travel time of a queried message based on a given metric (e.g minimize the number of traversed hops), which can have a benefit for certain metrics over other p2p network protocols.

Pastry has a rather simple application programming interface (API) containing only two methods [59]:

- `nodeId = pastryInit()` returns a new nodeId of a node joining or creating the Pastry network with its state.

- `route(msg,key)` routes the given message to the given key, which is numerically closest to the nodeId of all nodes being connected and online.

Furthermore, Pastry expects applications layered on top to implement this API[59]:

- `deliver(msg,key)` is called when the message is received by the respective node with the closest nodeId for the key.

- `forward(msg,key,nextId))` is called just before a message is forwarded to other nodes. Application may modify the message here or returning NULL for the termination of the operation.

- `newLeafs(leafSet)` is called when a change happens in the local leaf set. Application specific modification can be applied to the leaf set here.

2.5.3 Content Addressable Network (CAN)

CAN utilizes a d-dimensional Cartesian coordinate space, a d-torus. The entire coordinate space of that torus can be dynamically partitioned, so each participating node owns a zone of that space. A hash function is used to map any (key,value) pair onto a specific coordinate on the d-torus. A node which owns a zone in the coordinate space only knows its direct neighbors. Routing utilizes the coordinate that is calculated from the key, so a lookup is sent to the neighbor that is nearest to the corresponding coordinates to forward the request. This kind of routing needs $O(dn^{1/d})$ hops for short paths where n is the count of nodes. Additionally

Ratnasamy proposes the use of r realities so that a CAN node has coordinates in r different d-dimensional tori. An open problem of CAN represents the vulnerability to denial of service attacks [56].

CAN is a P2P overlay that implements the characteristics of P2P and the distributed hash functionality of DHT. The design is simple yet offers potential enhancements like increasing dimension and reality. Regarding our research we use a 2 dimensional torus, thus every node has 4 neighbors at maximum. To abstract the torus, we can see the coordinate space of our simulation example as a grid [56].

2.6 PeerfactSim.KOM

In order to evaluate the proposed P2P Fog computing scenarios, we used PeerfactSim.KOM as a simulator for this research. PeerfactSim.KOM is a modular, event-based networking simulator written in Java [45]. Originally developed at the technical university Darmstadt since 2005, later in development at the Heinrich Heine university. PeerfactSim.KOM allows us to do large scale, multi-parameter comparative evaluations of peer-to-peer systems [33]. PeerfactSim.KOM uses an event system and a message system, and is structured in a multiple layers that resemble the multilayered structure of real networks [32]. The multi-layered architecture of PeerfactSim.KOM is shown in Fig. 2.5 contains six layers [33, 32]:

- User Layer: is the top layer to setup configurations and options for the underlying application layer. This layer implements various user specific behaviors (e.g just searching for previously published files, specific online-times).

- Application Layer: This layer implements different applications. Already implemented are file sharing, tag-based search and lookup generators.

- Service Layer: This layer offers services like monitoring or management. This layer has the ability to manage and improve P2P Overlay and serves as an interface between the application and overlay layer.

- Overlay Layer: This layer has already implemented, among others, P2P DHT architectures like Chord, Pastry and CAN, but also a non-P2P DHT Overlay which can be used for server-client situation centralized storage. The implementations for our use cases utilize and modify those overlays to be suited for our fog computing scenarios.

- Transport Layer: offers services such as multiplexing by using the notion of ports, connection-oriented data streams, error-correction, and flow-control, further it offers the transmission control protocol (TCP) and user datagram protocol (UDP) for message exchange.

- Network Layer: allows to use a range of network models that differ in complexity and realism for simulating roundtrip-times, jitter, delay, packet loss or global positioning in the network.

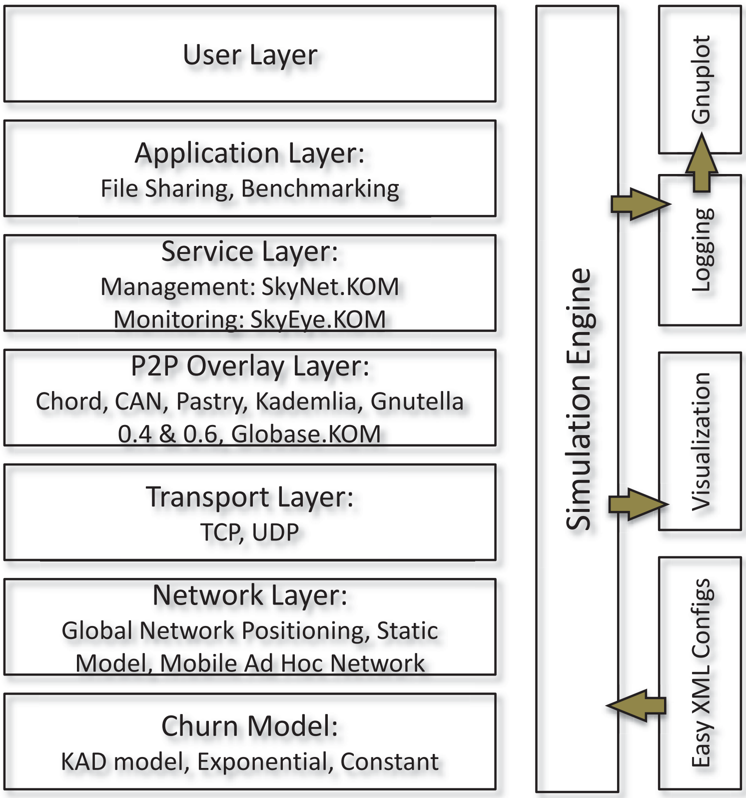

Figure 2.5: PeerfactSim.KOM architecture

- Churn Model: presents options to simulate the join and leave behavior of nodes in the network [33].

Each layer provides well-defined interfaces to express its individual functionalities, operations, and provided services [36]. Remarkable for this research is the overlay layer, application layer, and network layer. The PeerfactSim.KOM simulator is created for evaluation of P2P architectures which also implicates the feasibility to use it on non P2P networks. The simulator can be configured for any purpose by creating an xml configuration file. It can set each layer like the type of application, the overlay and the needed service. It can also set a seed, a time-based length and the number of peers in the simulation. We also have to create an action file to define the time of the peers joining and their behavior. The events that PeerfactSim.KOM engine is based on are queued, processed and logged by the simulation event queue. The logs are used for the visualization graphical user interface (GUI) but also processed into statistic files which are easily usable by the plotting software Gnuplot [44]. PeerfactSim.KOM simulator has a wide range of applicability which is used to simulate a wide range of overlays as in [6, 5, 4] and for P2P network monitoring solutions as in [37, 28, 25, 26].

2.7 Related Work

The emergence of fog computing technology have introduced many computing platforms that aim to serve the main purpose by easing the communications and resources needed for the compute nodes. The research presented in this thesis builds upon some previous researches which explored the idea of applying DHTs in implementing fog computing scenarios. While some research looks at it from communication point of view, others look at it from resource providence or architecture point of views.

One important aspect of the architecture of fog computing is how the compute nodes communicate between themselves in the fog layer and all the way to the cloud. Karagiannis et al. [42] focuses on the role communication plays in fog computing, were two main communication types in fog computing are identified that are related to our research: peer to peer (P2P) and hybrid. The P2P communication type refers to organizing the compute nodes of the fog in a P2P manner, which shown potential for handling fog infrastructures in a scalable manner and for this reason, P2P mechanisms have found use in fog computing platforms. On the same topic, Santos et al. [61] propose a resource discovery service which can be used in fog computing to gain dynamic resource provisioning based on DHT. By using DHT the compute nodes are organized in a ring structure, where the nodes exchange information about the available resources and workload. The proposed approach is assessed by examining three different DHT based P2P protocols, namely: Chord, Kademlia and Pastry. On the other hand, the hybrid communication type combine both hierarchical and P2P types. Where the overall structure the compute nodes are organized in hierarchical layers. The compute nodes within each layer do communicate and exchange information between them using p2p connectivity. On this, Lee et al. [47] propose a framework to utilize the fog network to serve the requests from IoT applications. According to this framework, the fog nodes within each layer communicate using p2p connectivity. However, the overall architecture of the system is organized hierarchically using three layers. The first layer from the bottom contains the IoT resource constrained devices which send their computation requests to the fog layer. Each compute node of the fog layer shares the workload with the neighbor nodes in proximity, but can also outsource workloads to the cloud layer which is located at the top of the hierarchy. While the mentioned researches tackle an important issue about the combination between fog computing and p2p, and the role of communication in the combined architecture, our research focuses in simulating how the utilization of fog computing and p2p in IoT served the whole system better in terms of bandwidth consumption.

IoT has been utilized in various sectors in our daily life, therefore, researchers have been longing for an architecture that could serve the IoT at the best possible way. While fog computing proved to be the most suitable technology available to fulfill the emergent needs of IoT, the basic fog computing architecture might need to be enhanced to get the best of the IoT technology. Tracey et al. [73] focuses on investigating the suitability of combining the p2p network with fog computing to achieve an architecture that is scalable and interoperable to fulfil the requirements of IoT. This paper proposed that with the emergence of IoT has created a need for an architecture such as Fog computing, with abstractions for all components in the entire flow starting from the edge device all the way to the cloud. Moreover, the p2p overlay is proposed to offer more scalability and availability at the edge of the network. On this regard, the paper present an extended Holistic Peer-to-Peer (HPP) application layer protocol that uses a Distributed Hash Table (DHT) based on Kademlia. This paper demonstrates that this architecture could be scaled down to run on constrained devices and scaled up to Cloud services in an overlay P2P network, using a DHT that allowed peers in Wireless Sensor Networks (WSNs) and external networks to exchange information using the same application layer protocol, without requiring proxies or additional middle-ware. P2P overlays differ in terms or performance and efficiency in responding to incoming requests, while the previously mentioned research focuses on Distributed Hash Table (DHT) based on Kademlia combined with fog computing to achieve a scalable and interoperable architecture to fulfil the IoT requirements, our research simulates and compares the proposed p2p fog computing architecture based on three of the most common P2P overlays, namely: Chord, Pastry, and CAN. Then analyze and compare the results to show which overlay provide the best results in terms of network bandwidth consumption.

Another aspect of improvement that could enhance the performance and efficiency of fog computing is how to best utilize the compute nodes resources and storage by distributing the requests to the nodes with most available resources. Steffenel et al. [69] focuses on implementing data locality in p2p fog computing platforms, which proposes a location-aware mechanism for placing jobs on available resources on the edge nodes. This paper demonstrate interesting performance improvements for fog platforms with data locality. Furthermore, data locality scheduling must consider the condition and available resources in terms of computing power and storage on each node before delegating tasks to that node. Some other aspects should be considered such as the remaining battery lifetime. The concept proposed in this paper believed to improve the performance of data-intensive applications such as IoT. Our research does not tackle the data locality when it comes to enhancing the performance of the p2p fog computing, rather we use a simple caching mechanism to cache the data at the fog nodes, and when the cache is full, the data will be replaced randomly. Therefore, this research is a valuable source for our future work.

In summary, the work presented in this paper builds on previous research to explore how availability information relates to people's communication decisions. While earlier work focused on how availability information impacts the people initiating communication, we focus on its impact on the decisions of the recipient. Further, we are able to study this behavior at a much larger scale than previously possible by looking at the users of a popular enterprise communication system that infers its users' availability.

Chapter 3

Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios

This chapter summarizes the contributions and gives a verbatim copy of our paper [55].

Ahmad Rabay'a, Eduard Schleicher and Kalman Graffi. "Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios". In: *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2019. Acceptance Rate: 16.7%

3.1 Paper Summary

P2P Fog computing is a proposed mode which extends the fog computing architectures by introducing P2P Chord mechanism into the fog layer, with the aim to enhance the bandwidth consumption of fog computing, in order to meet the emerging needs of IoT devices. In this research we demonstrate that our proposed Chord P2P Fog model can produce a better bandwidth consumption results compared to fog computing and cloud computing. The simulation outcomes of the three architectures models are evaluated and compared to each other using PeerfactSim.KOM. In the research we use a read-only file-sharing application as a use case to measure and evaluate the bandwidth consumption behavior of the three architectures models. In this file sharing application, file requests go from a client to a server as lookup, and the response publishes the file to the client. Fog servers behave like storage units that pre-cache files, deliver files to clients, and request files from the cloud. For storing and publishing files, the file-sharing application relies on the offered functionality of the Chord overlay.

We have chosen metrics that highlight our research goal, to evaluate and compare the bandwidth consumption of the three scenarios. The three architectures models are evaluated and compared based on their Net Bandwidth and NetPeers Bandwidth metrics. The Net Bandwidth shows the overall bandwidth consumption in bytes per second for the whole scenario.

While the NetPeers Bandwidth shows the bandwidth consumption in bytes per second of every single peer. Furthermore, PeerfactSim.KOM offers many configuration parameters, and for this research, we consider: 100 nodes, 100 minutes, 100 files with both 2 and 1 second lookup distribution period. Based on the simulations of different configuration parameters values, it is clear that the lookup distribution value has the most impact on the simulator outcomes. The lower lookup distribution value is, the closer our scenario gets to an IoT scenario.

Based on the simulations, the cloud computing scenario has the highest bandwidth consumption value compared to the other two scenarios for the all simulations. But on the same time, the cloud scenario manages to finish more operations than the other two in the same period. On the other hand, remarkably, the proposed P2P Fog scenario manages to finish more operations than the fog scenario and still has much less overall bandwidth consumption value. On the other hand, for the peers bandwidth consumption, we notice the huge difference in the bandwidth consumption values for the cloud node between the three scenarios. For the cloud scenario, the cloud node bandwidth consumption is very high compared to the other scenarios, as in this scenario the cloud node is responsible for all the requests that come form the clients. The cloud node bandwidth consumption increases hugely when we reduce the lookup distribution. For the fog scenario, the cloud node bandwidth consumption under 5 and 2 sec lookup distribution is the lowest compared to the two other scenarios. However, it increases sharply under 1 sec lookup distribution. Regarding our proposed P2P Fog scenario, the cloud node bandwidth consumption shows a very significant behavior when we reduce the lookup distribution value from 5 sec to 2 and 1 sec. At the time the other two scenarios show a huge increase in bandwidth consumption when we reduce the lookup distribution, in the P2P Fog scenario the cloud node bandwidth consumption increases rather slightly and eventually produces a better bandwidth consumption results than the fog scenario under 1 sec lookup distribution. When comparing the increase in the bandwidth consumption between fog nodes and the cloud node, under the different lookup distribution values we notice the impact our proposed scenario has by reducing impact on the bandwidth consumption of the cloud node in a huge manner, compared to the other scenarios.

In our simulations, we selected a 100 minutes simulation time, however in reality such IoT systems will run for weeks or even months, meaning that the difference margin will be much bigger. However, even for this small simulation time, compared to realistic IoT scenarios, we notice that our proposed P2P Fog model provides better bandwidth consumption compared to the cloud computing and fog computing.

3.2 Contribution

In this paper, we present the novel peer-to-peer fog computing model (P2P Fog), which is a new architecture model that combines the capabilities of fog computing with the peer-to-peer (p2p) network mechanism. By doing so, we aim to solve the challenge that a sole fog node is limited in computational resources to relieve load from the backbone, thus still highly dependent on the cloud, which means the internet backbone has to be used again thus cloud bandwidth is burdened. The proposed P2P Fog computing model will empower the fog nodes, in case they do not have sufficient computing and storage capabilities to handle the clients request, by trying to delegate some of the computation tasks to the peered fog nodes, using the Chord overlay, in order to fulfill the IoT devices requests near the proximity of end users, and avoid the huge delay and bandwidth consumption of contacting the cloud. The P2P Fog is a new but rather realistic scenario motivated by IoT services requirements, user demands, and technology progress that, to the best of our knowledge, has not been addresses before in a coordinated fashion. That said, it is a concept related to some recently proposed approaches, such as it was directly introduced by Chiang et al. in [18] as a research opportunity.

3.3 Personal Contribution

Ahmad Rabay'a, author of this thesis, has motivated and organized the paper and done most of the writing. Rabay'a introduced the idea of P2P Fog computing, described the system model and designed its architecture. Furthermore, Rabay'a described the simulation methodology and the PeerfactSim.KOM configurations and test cases. In addition to implementing the model skeleton with all the required classes, interfaces, and APIs. And finally he collected and formulated the results. Eduard Schleicher did the detailed implementation of the three scenarios, and ran the various tests. Kalman Graffi highly contributed in the discussions around formulating the architecture of the P2P Fog model. As well as he supervised and reviewed the paper and made valuable suggestions and annotations regarding the P2P Fog in various discussions.

3.4 Importance and Impact on Thesis

The presented paper is important to this thesis since, in addition to answer many of the research questions listed in Section 1.3, it is where the initial architecture of the new P2P Fog model is first proposed, implemented, and tested, which is later the basis for the rest of the thesis. We show with this paper that it is possible to develop a new computing model that combines both fog computing and p2p technologies to further reduce the bandwidth consumption of the cloud to meet the emerging needs of IoT. In this P2P Fog model fog nodes are connected via the Chord overlay and can cooperate by combining their computational capabilities to create a larger pool of resources in order to meet the users' requests at the proximity of the network edge rather than traveling all the way to the cloud, which reduces the amount of requests that will go to the cloud and avoid the huge bandwidth consumption resulting from that.

We identify with this work that it is possible to combine fog computing with p2p to create a cooperative layer between the fog nodes using Chord overlay, as our proposed model produced better outcomes in terms of bandwidth consumption compared to the traditional fog computing and cloud computing models.

Chapter 4

Reducing IoT Bandwidth Requirements by Fog-Based Distributed Hash Tables

This chapter summarizes the contributions and gives a verbatim copy of our paper [54].

Ahmad Rabay'a, Pascal Kudla, Lukas Kubitza, Kalman Graffi and Michael Schöttner.
"Reducing IoT Bandwidth Requirements by Fog-Based Distributed Hash Tables". In: *4th Conference on Cloud and Internet of Things (CIoT)*. 2020. Acceptance Rate: 34%

4.1 Paper Summary

To empower the single fog nodes against their limited computational resources to relieve load from the backbone, a peer-to-peer fog (P2P Fog) model was proposed in Chapter 3. This model allows fog-nodes at the fog layer to cooperate, using the Chord overlay, and reduce the bandwidth requirement to the cloud and the system as a whole by fulfilling the end users requests at the fog layer. In this publication we extend the previously mentioned model by implementing in simulation and evaluate the fog computing workload on two additional p2p overlays, namely: Pastry and content addressable network (CAN), and compare the outcome results with the scenario using the Chord overlay. This research focuses on an IoT data access setup as a use case to measure and evaluate the bandwidth consumption of the different scenarios.

To simulation the different scenarios, we focus on a use case that causes traffic between connected devices without adding undesired complexity to our attempt. For that a basic IoT data access use case, where end devices retrieve relevant data from the network or update information to the network through push/pull based functionality, is considered to measure and evaluate the bandwidth consumption of the different scenarios. The used simulator is configured for: 100 nodes, 100 minutes, 100 data elements with 10, 2 and 1 second lookup distribution period. The configurations and parameters values are chosen to be good enough to illustrate our goal of the evaluation.

Based on the simulation results, different patters are noticed regarding how the scenarios reacted. While the bandwidth consumption value of the simple fog scenario increases massively by reducing the lookup distribution value, the bandwidth consumption value of the P2P Fog scenarios (Pastry fog, Chord fog and CAN fog) increase rather slightly and in a small margin. However, for a high look distribution value the simple fog scenario produces better bandwidth consumption values compared to the P2P Fog scenarios. Applying P2P Fog models shift the traffic load from the cloud to the fog nodes, especially under frequent queries from the clients. This means that the simple fog scenario is suitable for the typical systems that rarely contact the cloud, while the proposed P2P Fog scenarios show an improved saving in terms of reducing bandwidth consumption for the systems that require an intensive resource access and communication with the cloud. Moreover, the Pastry overlay produces better results compared to Chord and CAN because it has a routing algorithm that determines the best routes to store in the routing table, which minimize the travel time of the message between the peered fog nodes by quickly locating the responsible node.

4.2 Contribution

In this paper, we extend our novel P2P Fog model to include two other p2p overlays, namely: Pastry and CAN, in order to figure out under which overlay our P2P Fog model performs better by producing better outcome results in terms of bandwidth consumption. The three p2p based fog computing models are implemented and evaluated using the simulator PeerfactSim.KOM and compared against each other and against a basic fog computing scenario. Even though the proposed P2P Fog model shows an improved output results for IoT similar scenarios, for simple systems that do not require intensive communication to the cloud, it does not make sense to use the proposed model, since the fog nodes cache is empty at the beginning, so the clients requests travel through multiple nodes before they eventually retrieve the information from the cloud. And for a scenario with not much pull requests from the clients, very little requests will go to the fog nodes, therefore the fog nodes cache is filled rather slowly, this causes extra bandwidth consumption compared to retrieving the information from the cloud immediately as in the simple fog scenario. For resource intensive IoT similar scenarios, the P2P Fog scenarios show a very significant behavior in term of reducing the bandwidth consumption compared to the simple fog scenario. The reason for that is the fog nodes caches are filled faster, and more and more requests are fulfilled at the fog layer without the need to contact the cloud.

4.3 Personal Contribution

Ahmad Rabay'a, author of this thesis, has motivated and organized the paper, in addition to writing the whole paper. The paper's main idea was derived from a previous paper published by Rabay'a, therefore, he setup the detailed requirements, sketched the architecture design, and implemented the model skeleton with all the required classes, interfaces, and APIs. And finally collected and formulated the results from the two sources, and combined the results together. Both Pascal Kudla and Lukas Kubitza have each done the detailed implementation of one of the newly implemented p2p overlays, ran various tests, and provided the outcomes to Rabay'a. Kalman Graffi and Michael Schöttner both highly contributed in the discussions around formulating the architecture of the P2P Fog model. As well as supervised and reviewed the paper and made valuable suggestions and annotations regarding the P2P Fog in various discussions.

4.4 Importance and Impact on Thesis

The [54] paper is important for this thesis since it forms a continuum to our proposed P2P Fog model, proving that combining fog computing with p2p technology would have a great potential of solving a major issue facing the current cloud computing in the era of increasing IoT technology. This paper shows that the P2P Fog model would work under any other p2p overlay, but with different output results.

Chapter 5

Implementing Enterprise Resource Planning Systems in the Cloud: Challenges and Solutions

This chapter summarizes the contributions and gives a verbatim copy of our paper [53].

Ahmad Rabay'a and Kalman Graffi. "Implementing Enterprise Resource Planning Systems in the Cloud: Challenges and Solutions". In: *International Symposium on Networks, Computers and Communications (ISNCC)*. 2019. Acceptance Rate: 37%

5.1 Paper Summary

While Enterprise Resource Planning (ERP) in the cloud offers customers many potential benefits, there are some concerns that customers need to consider before implementing the system. The driving force behind this research is to investigate the main challenges that are restraining the market from implementing ERP systems in the cloud, and propose solutions to these challenges. The market views towards these issues are addressed in this paper through an exploratory study. Then the results are validated by the Cloud Security Alliance (CSA). The results of this research are simplified assembling and filtering of issues, proposed solutions, and specifications toward implementing ERP in the cloud based on data from the CSA. The summarized findings of the research are as following:

Regarding Service Level Agreement (SLA), despite the general agreement between all the interviewees that SLA is a necessary contract between customers and providers, customers do not think it is enough. SLA needs to be regularly monitored and measured to make sure it satisfies the customers' requirements. As well as more emphasis penalties must be added to maintain a complete, accurate and relevant SLA, and assure availability and redundancy of the offered service. The CSA has a special attention for the SLA as checking procedure and penalties are suggested to be introduced to the SLA. CSA as well suggests that there should be procedures for maintaining complete, accurate and relevant SLA between providers and customers.

For the ERP in cloud standardization, even though customers and small providers would support an ERP in the cloud standard, bigger providers and market leaders do not believe or welcome such a standard, as they might be at the verge of losing their market share to small and new ERP providers. The proposed solution is to standardize the core of the ERP in the cloud, where all providers might provide the basic standardized core, and offer any distinct customized modules on top of that core. CSA supports such a security and interoperability standard and assures that APIs should be designed, developed, and deployed to comply with leading industry standards, in addition to security and privacy policies to ensure interoperability and migration of data.

The participants' views toward the influence of the cloud resistance are obscured and scattered, which indicates that the market does not have adequate knowledge about the cloud service. It is noticed that the participants' views toward the influence of the customers' resistance are based on individual expectations rather than industrial researches or case studies. To close this gap between the providers and customers views, providers must show real examples for ERP in the cloud using services that makes it possible for customers to try real ERP systems using their own business processes and data. The CSA focuses on how to avoid and mitigate the customers' resistance toward ERP in the cloud. The CSA suggests that providers should provide tenants with documentation describing the Information Security Management Program that includes administrative, technical, and physical safeguard to protect assets and data.

5.2 Contribution

In this paper we explore the market view toward implementing ERP systems using cloud computing. Even though the theory suggests that ERP systems in the cloud would provide solutions to many difficulties encountered by conventional ERP systems, this research aims to further investigates this claim by interviewing both ERP providers and customers about their view regarding the accompanying issues of implementing ERP systems using Cloud computing. The research further introduces solutions to the investigated issues, before validating the investigated issues based on specifications from the Cloud Security Alliance (CSA). The results of this research are simplified assembling of issues, best practices, and specifications toward implementing ERP in the cloud. The results of this work should help the market and researchers get a better view about ERP in the cloud.

5.3 Personal Contribution

Ahmad Rabay'a, author of this thesis, has motivated and organized the paper and done all of the writing. Rabay'a explored the idea of implementing ERP systems in the cloud, conducted the exploratory study with both parts of the market, analyzed and evaluated the output, before validating the results against the recommendations published by the Cloud Security Alliance (CSA). Kalman Graffi highly contributed in the paper organization, as well as supervised and reviewed the paper and made valuable suggestions and annotations regarding the content and structure.

5.4 Importance and Impact on Thesis

The presented paper, even though presented last, is considered the starting point of the thesis, as it explored the applicability of cloud computing which in turn lead to the emergence of fog computing.

Chapter 6

Conclusion and Future Work

In this chapter, we present a summary of our research contents, followed by an abstract overview of some potential future work possibilities.

6.1 Conclusion

With the explosion growth of IoT and its high demands, the centralized computing model of the cloud, where most of the computation happen in the cloud, we encounter a growing challenge. Despite that data processing speed has risen rapidly, the network bandwidth has not increased appreciably. In order to overcome the limitations of cloud computing and meet the demands of IoT and other emerging computing models and paradigms, the Fog Computing model was introduced. Fog computing is a cloud-like layer set up just above the edge of the internet, which makes computation, communication, and storage closer to end users by pooling the local resources. The fog paradigm can effectively meet the demands of IoT devices, and notably ease network bandwidth bottlenecks. Despite that Fog Computing is already a successful approach in reducing internet bandwidth consumption of the cloud, sole fog nodes are limited in computational resources thus still highly dependent on the cloud. In this research a new model called peer-to-peer fog (P2P Fog) has been proposed, which combines the capabilities of fog computing with the peer-to-peer (P2P) network mechanism to form an enhanced three layered client-P2PFog-cloud. Even though each fog node has a limited computational capabilities, combining the computational capabilities of a number of fog nodes create a larger pool of resources. The P2P Fog paradigm empowers the sole fog nodes by delegating part of the computations to the peered fog nodes in order to avoid the huge delay and bandwidth consumption of contacting the cloud. In this research we implement in simulation and evaluate a realistic fog computing workload on different p2p overlays. This research focuses on an IoT data access setup as a use case to measure and evaluate the bandwidth consumption of the different scenarios. Therefore, through a series of simulations, our research shows that the P2P Fog model can further reduce the bandwidth consumption compared to fog computing.

In [53], we investigate the major challenges that are restraining the market from implementing crucial enterprise systems, such as ERP system, into the cloud. We provide a set of simplified issues, best practices, and specifications toward implementing ERP in the cloud from both customers and providers point of views. The results are then validated based on specifications from the Cloud Security Alliance (CSA). Our final results is to help understanding the customers' requirements, protect the customers and reduce their fears toward the cloud. During this research many issues were investigated, and varied results have been acquired. Firstly, in general SLA is a necessary contract between customers and providers, but needs to have more emphasis and clarity on monitoring and penalties mechanisms. Secondly, an ERP in the cloud standard is supported by all parties and encouraged by CSA, despite that big ERP providers who do not believe or welcome such a standard. Finally, it is rather the unmatured knowledge of the technology that caused the scattered view toward the cause of customer resistance toward ERP in the cloud, as CSA focuses of mitigating the potential resistance rather than avoid it.

In [55], is our first attempt to propose, implement and test the idea of P2P Fog model which aims at empowering the sole fog nodes by cooperating with the other fog nodes using p2p overlays to aggregate their computational capabilities and create a larger pool of resources to meet the increasing needs of IoT devices and reduce the huge bandwidth consumption resulting from contacting the cloud. The simulation outcome results of the proposed model are compared against the fog computing and cloud computing outcomes in terms of bandwidth consumption. In this research the fog nodes are connected together with a Chord overlay to form the P2P Fog layer, and could cooperate together to meet the clients requests. In our simulations, we have decided for a 100 minutes simulation time, however in reality such systems will run for weeks or even months, meaning that the difference margin will be much bigger. Despite that, we still could see that our proposed P2P Fog model provides better bandwidth throughput compared to the cloud computing and fog computing. We could notice the impact our proposed scenario will have in the IoT systems by reducing the bandwidth throughput of the cloud node in a huge manner.

In [54], and as a continuum for what we achieved in [55] we investigated the behavior of the P2P Fog model under different P2P overlays, namely: Pastry and CAN overlays, in addition to the Chord overlay which is implemented and tested by us in the previous research. Based on the simulation outcome results of the three overlays, we evaluate and compare the three overlays scenarios to decide which scenario produces the best outcomes in terms of bandwidth consumption. The simulator environment is configured to imitate an IoT setup. The simulation results show that Pasty performs best regarding bandwidth reduction. The simulations have been run for 100 minutes and we expect even better results for real-world setups where applications run for months or years (of course depending on the access patterns and fog node resources).

With these few conclusive statements on the work we present in this thesis, we now focus on possible future contributions based on some of the gaps identified during our work.

6.2 Future Work

Fog computing and IoT are two of the hottest topics in the technology world nowadays. Despite the fact that Fog computing is a relatively new technology, many studies have been focused on foreseeing the upcoming opportunities to improve the service delivery and introducing additional functionalities than what we have in our proposed P2P Fog architecture. However, while working on this research idea we came across some interesting research ideas that can offer many value added to our proposed P2P Fog model. In the following we discuss some of them.

6.2.1 Evaluate the P2P Fog model with latency sensitive applications

Our research focuses mainly on simple IoT data access setup as a use case to demonstrate in a simulator the proposed P2P Fog architecture and how the outcomes of this proposed model will differ from the traditional fog computing and cloud computing models. However, the results might differ when considering demonstrating the P2P Fog architecture using resource intensive applications that are less tolerance to latency, such as streaming applications and surveillance cameras. This is considered an exciting research opportunity, as it allows to cover a wide range of applications to widen the applicability of the potential model resulted from combining fog computing and p2p.

6.2.2 P2P Fog computing – read and write file sharing (file update)

The scenario investigated in our research is a read-only data access setup were requests only flows in one direction upward, from edge devices to the fog nodes, between the fog nodes, and up to the cloud. Furthermore, the responses will flow the other way downward. However, the results might differ when considering read-write scenarios, were any of the involved nodes can modify the content of the file/message, before sending it to the other nodes. This research opportunity is the expected step forward from our research and see if the evaluation outcome from our research stands when considering a more complicated use case. In this use case requests and responses can flow in both directions, therefore, the scenario will get more complicated, and a mechanism needed to be in place to control the versioning of the data to ensure the integrity and consistency between the edge, fog, and the cloud nodes.

6.2.3 Caching mechanism

As the storage capacity of the fog nodes is quite small and limited compared to the cloud, caching plays an important role in making sure that the edge devices' requests get fulfilled in the fog layer. This idea is supported by the fact that fog nodes do have a location awareness, therefore, they could predict the category of the users' requests within their proximity and cache that information, which will reduce the need of contacting the cloud. However, as the storage capacity of the fog nodes are limited, once the fog node's storage is full, and a new information needs to be cached, some old data will be dropped. Deciding what to drop is an important issue, and there are many available algorithms to decide which is the least needed piece of information that if dropped will have the least impact further on. Accordingly, less requests will pass through to the cloud and the bandwidth consumption can be kept as small as possible. In our research we implement a simple random caching mechanism, so when the fog node's cache is full a random piece of information will be replaced by the new cached information. However, we believe that the P2P Fog model outcome would be improved if a more suitable algorithm is utilized to replace the data in the fog node's cache and reduce the backbone bandwidth consumption of contacting the cloud. Further research needs to be conducted to test whether a simple caching mechanisms like First In First Out, Last In First Out, Least Recently Used, Most Recently Used, and Least Frequently Used, could prove sufficient for our proposed model, or a tailored algorithm that is specific for fog computing caching, needs to be developed.

6.2.4 P2P Fog Computing – Capacity Sharing (Capacity based search)

A very interesting area of research is capacity sharing, were the fog nodes in the fog layer not only share the files among them, but share their computing and storage resources capacity. Therefore, when a computation request is sent from a dumb edge device, such as sensors and actuator, to the fog node, there are two scenarios; either the fog node does have the computing capacity to fulfill the request, so it will execute the computation and send the results back to the edge device. Or the fog node does not have enough resources, and in that case, is could delegate part of that computation to the peered fog nodes. After that a mechanism needs to be in place to assemble the result again, and send the final result to the requested edge device.

6.3 Closing Words

While Cloud Computing as a term has been introduced in the 1990s, fog computing is a relatively new technology which was introduced in 2015. Despite being a new technology, in this research we shed the light on an exciting improvement potential which makes fog computing more powerful and more convenient. So, while fog computing will grow in helping the emerging network paradigms that require faster processing with less delay, P2P Fog would extend the capability of fog computing to include more demanding and resource intensive computations which will further extend the applicability and benefits of fog computing and reduce latency, response time and bandwidth consumption that results from contacting the cloud. Furthermore, and despite the improvement we present in this research in terms of reduce bandwidth consumption of the cloud, we expect to have much better bandwidth consumption results when we implement the right caching mechanism and run the system in a real-world setups where applications run for months or years.

Bibliography

[1] Nabeel Al-Milli and Wesam Almobaideen. “Hybrid neural network to impute missing data for iot applications”. In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. IEEE. 2019, pp. 121–125 (Page: 19).

[2] Wesam Almobaideen, Maha Saadeh, Noor Al-Anbaki, Rawan Zaghloul, and Aryaf Al-adwan. “Geographical route selection based on user public transportation and service preferences”. In: *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE. 2015, pp. 144–149 (Page: 19).

[3] Wesam A Almobaideen and Ola M Malkawi. “Application Based Caching in Fog Computing to Improve Quality of Service”. In: *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2020, pp. 20–27 (Pages: 19, 20).

[4] Tobias Amft and Kalman Graffi. “A Tale of Many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks”. In: *Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany, Tech. Rep. TR-2017-001* (2017) (Pages: 21, 24).

[5] Tobias Amft and Kalman Graffi. “Moving peers in distributed, location-based peer-to-peer overlays”. In: *2017 international conference on computing, networking and communications (ICNC)*. IEEE. 2017, pp. 906–911 (Pages: 21, 24).

[6] Tobias Amft, Barbara Guidi, Kalman Graffi, and Laura Ricci. “FRoDO: Friendly routing over dunbar-based overlays”. In: *2015 IEEE 40th conference on local computer networks (LCN)*. IEEE. 2015, pp. 356–364 (Pages: 21, 24).

[7] Hamid Reza Arkian, Abolfazl Diyanat, and Atefe Pourkhalili. “MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications”. In: *Journal of Network and Computer Applications* 82 (2017), pp. 152–165 (Pages: 1, 2, 10).

[8] Pankaj Arora, Rubal Chaudhry Wadhawan, and Er Satinder Pal Ahuja. “Cloud computing security issues in infrastructure as a service”. In: *International journal of advanced research in computer science and software engineering* 2.1 (2012) (Page: 13).

[9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The internet of things: A survey”. In: *Computer networks* 54.15 (2010), pp. 2787–2805 (Page: 9).

[10] Eren Balevi and Richard D Gitlin. “Synergies between cloud-fag-thing and brain-spinal cord-nerve networks”. In: *2018 Information theory and applications workshop (ITA)*. IEEE. 2018, pp. 1–9 (Page: 19).

[11] Ejder Bastug, Mehdi Bennis, and Mérouane Debbah. “Living on the edge: The role of proactive caching in 5G wireless networks”. In: *IEEE Communications Magazine* 52.8 (2014), pp. 82–89 (Pages: 3, 17).

[12] Markus Benter, Mohammad Divband, Sebastian Kniesburges, Andreas Koutsopoulos, and Kalman Graffi. “Ca-re-chord: A churn resistant self-stabilizing chord overlay network”. In: *2013 Conference on Networked Systems*. IEEE. 2013, pp. 27–34 (Page: 21).

[13] Cor-Paul Bezemer and Andy Zaidman. “Multi-tenant SaaS applications: maintenance dream or nightmare?” In: *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*. 2010, pp. 88–92 (Page: 13).

[14] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. “Fog computing: A platform for internet of things and analytics”. In: *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186 (Pages: 18, 19).

[15] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. “Fog computing and its role in the internet of things”. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM. 2012, pp. 13–16 (Pages: 15, 16).

[16] J Buckley et al. *The internet of things: from RFID to the next-generation pervasive networked systems*. 2006 (Page: 10).

[17] Rajkumar Buyya, Rodrigo N Calheiros, and Xiaorong Li. “Autonomic cloud computing: Open challenges and architectural elements”. In: *2012 third international conference on emerging applications of information technology*. IEEE. 2012, pp. 3–10 (Page: 2).

[18] Mung Chiang and Tao Zhang. “Fog and IoT: An overview of research opportunities”. In: *IEEE Internet of Things Journal* 3.6 (2016), pp. 854–864 (Pages: 1, 5, 12, 15, 29).

[19] Rudyar Cortés, Xavier Bonnaire, Olivier Marin, and Pierre Sens. “Stream processing of healthcare sensor data: studying user traces to identify challenges from a big data perspective”. In: *Procedia Computer Science* 52 (2015), pp. 1004–1009 (Pages: 1–3).

[20] N Council. “Six technologies with potential impacts on us interests out to 2025”. In: *Disruptive Civil Technologies* (2008) (Page: 9).

[21] Soumya Kanti Datta, Christian Bonnet, and Jerome Haerri. “Fog computing architecture to enable consumer centric internet of things services”. In: *2015 International Symposium on Consumer Electronics (ISCE)*. IEEE. 2015, pp. 1–2 (Pages: 3, 15).

[22] Doval DD and Donal O’Mahony. “Overlay networks: A scalable alternative for P2P”. In: *Internet Computing, IEEE* 7 (Aug. 2003), pp. 79 –82. DOI: `10.1109/MIC.2003.1215663` (Page: 21).

[23] LLP Deloitte. “Cloud Computing-Forecasting Change”. In: *Market Overview and Perspective Consulting* (2009) (Pages: 13, 14).

[24] Marios D Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis, and Athena Vakali. “Cloud computing: Distributed internet computing for IT and scientific research”. In: *IEEE Internet computing* 13.5 (2009), pp. 10–13 (Page: 12).

[25] Andreas Disterhöft and Kalman Graffi. “Convex Hull Watchdog: Mitigation of Malicious Nodes in Tree-Based P2P Monitoring Systems”. In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE. 2016, pp. 52–60 (Page: 24).

[26] Andreas Disterhöft and Kalman Graffi. “Minicamp: Prototype for Partial Participation in Structured Peer-to-Peer Monitoring Protocols”. In: *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE. 2017, pp. 203–206 (Page: 24).

[27] Andreas Disterhoft and Kalman Graffi. “Protected chords in the web: secure P2P framework for decentralized online social networks”. In: *2015 IEEE international conference on peer-to-peer computing (P2P)*. IEEE. 2015, pp. 1–5 (Page: 21).

[28] Andreas Disterhöft, Phillip Sandkühler, Andre Ippisch, and Kalman Graffi. "Mr. Tree: Multiple Realities in Tree-based Monitoring Overlays for Peer-to-Peer Networks". In: *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2018, pp. 354–360 (Page: 24).

[29] Stefan Dumbrava, Doru Panescu, and Mihaela Costin. "A Three-tier Software Architecture for Manufacturing Activity Control in ERP Concept". In: *International Conference on Computer Systems and Technologies-CompSysTech*. Vol. 2005. 2005, pp. 3–1 (Page: 12).

[30] Donald Eastlake and Paul Jones. *US secure hash algorithm 1 (SHA1)*. 2001 (Page: 21).

[31] Dave Evans. "The internet of things: How the next evolution of the internet is changing everything". In: *CISCO white paper* 1.2011 (2011), pp. 1–11 (Pages: 1, 3).

[32] Matthias Feldotto and Kalman Graffi. "Comparative evaluation of peer-to-peer systems using PeerfactSim. KOM". In: *2013 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE. 2013, pp. 99–106 (Page: 23).

[33] Matthias Feldotto and Kalman Graffi. "Systematic evaluation of peer-to-peer systems using PeerfactSim. KOM". In: *Concurrency and Computation: Practice and Experience* 28.5 (2016), pp. 1655–1677 (Pages: 21, 23, 24).

[34] Mohamed Firdhous, Osman Ghazali, and Suhaidi Hassan. "Fog computing: Will it be the future of cloud computing?" In: The Third International Conference on Informatics & Applications (ICIA2014). 2014 (Pages: 2, 3, 15, 16).

[35] GMDT Forecast. "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022". In: *Update* 2017 (2019), p. 2022 (Page: 9).

[36] Kalman Graffi. "PeerfactSim. KOM: A P2P system simulator—Experiences and lessons learned". In: *2011 IEEE International Conference on Peer-to-Peer Computing*. IEEE. 2011, pp. 154–155 (Pages: 21, 24).

[37] Kalman Graffi and Andreas Disterhöft. "SkyEye: A tree-based peer-to-peer monitoring approach". In: *Pervasive and Mobile Computing* 40 (2017), pp. 593–610 (Page: 24).

[38] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future generation computer systems* 29.7 (2013), pp. 1645–1660 (Pages: 9, 10).

[39] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, and Tie Qiu. "Survey on fog computing: architecture, key technologies, applications and open issues". In: *Journal of network and computer applications* 98 (2017), pp. 27–42 (Pages: 2, 3, 15–19).

[40] Mohammad Manzurul Islam, Sarwar Morshed, and Parijat Goswami. "Cloud computing: A survey on its limitations and potential solutions". In: *International Journal of Computer Science Issues (IJCSI)* 10.4 (2013), p. 159 (Page: 2).

[41] Kang Kai, Wang Cong, and Luo Tao. "Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues". In: *the journal of China Universities of Posts and Telecommunications* 23.2 (2016), pp. 56–96 (Page: 19).

[42] Vasileios Karagiannis. "Compute node communication in the fog: Survey and research challenges". In: *Proceedings of the Workshop on Fog Computing and the IoT*. 2019, pp. 36–40 (Page: 25).

[43] Eric Knorr and Galen Gruman. "What cloud computing really means". In: *InfoWorld* 7 (2008), pp. 20–20 (Page: 12).

[44] Tobias Korfmacher. “Implementation and Evaluation of a Mobility Model in a Peer-to-Peer Simulator”. In: (2015) (Page: 24).

[45] Aleksandra Kovacevic, Sebastian Kaune, Hans Heckel, André Mink, Kalman Graffi, Oliver Heckmann, and Ralf Steinmetz. “PeerfactSim. KOM-A Simulator for Large-Scale Peer-to-Peer Networks”. In: *Technische Universität Darmstadt, Germany, Tech. Rep. Tr-2006-06* (2006) (Page: 23).

[46] Nallapaneni Manoj Kumar and Pradeep Kumar Mallick. “The Internet of Things: Insights into the building blocks, component interactions, and architecture layers”. In: *Procedia computer science* 132 (2018), pp. 109–117 (Pages: 9–12).

[47] Gilsoo Lee, Walid Saad, and Mehdi Bennis. “An online secretary framework for fog network formation with minimal latency”. In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6 (Page: 25).

[48] Yang Liu, Jonathan E Fieldsend, and Geyong Min. “A framework of fog computing: Architecture, challenges, and optimization”. In: *IEEE Access* 5 (2017), pp. 25445–25454 (Page: 2).

[49] Tom H Luan, Longxiang Gao, Zhi Li, Yang Xiang, Guiyi Wei, and Limin Sun. “Fog computing: Focusing on mobile users at the edge”. In: *arXiv preprint arXiv:1502.01815* (2015) (Pages: 2, 3, 12, 15–17, 19).

[50] Eva Marín-Tordera, Xavi Masip-Bruin, Jordi García-Almiñana, Admela Jukan, Guang-Jie Ren, and Jiafeng Zhu. “Do we all really know what a fog node is? Current trends towards an open definition”. In: *Computer Communications* 109 (2017), pp. 117–130 (Page: 17).

[51] Xavi Masip-Bruin, Eva Marín-Tordera, Ghazal Tashakor, Admela Jukan, and Guang-Jie Ren. “Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems”. In: *IEEE Wireless Communications* 23.5 (2016), pp. 120–128 (Pages: 1, 3).

[52] Giuseppe Minutoli, Maria Fazio, Maurizio Paone, and Antonio Puliafito. “Virtual business networks with cloud computing and virtual machines”. In: *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE. 2009, pp. 1–6 (Pages: 12, 13).

[53] Ahmad Rabaya and Kalman Graffi. “Implementing Enterprise Resource Planning Systems in the Cloud: Challenges and Solutions”. In: *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. 2019, pp. 1–8. DOI: `10.1109/ISNCC.2019.8909115` (Pages: 7, 35, 39).

[54] Ahmad Rabaya, Pascal Kudla, Lukas Kubitza, Kalman Graffi, and Michael Schöttner. “Reducing IoT Bandwidth Requirements by Fog-Based Distributed Hash Tables”. In: *2020 4th Conference on Cloud and Internet of Things (CIoT)*. 2020, pp. 25–32. DOI: `10.1109/CIoT50422.2020.9244294` (Pages: 6, 31, 33, 40).

[55] Ahmad Rabaya, Eduard Schleicher, and Kalman Graffi. “Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios”. In: *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2019, pp. 82–89. DOI: `10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00036` (Pages: 6, 27, 40).

[56] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. "A Scalable Content-addressable Network". In: *SIGCOMM Comput. Commun. Rev.* 31.4 (Aug. 2001), pp. 161–172. ISSN: 0146-4833. DOI: `10.1145/964723.383072`. URL: `http://doi.acm.org/10.1145/964723.383072` (Page: 23).

[57] Partha Pratim Ray. "A survey on Internet of Things architectures". In: *Journal of King Saud University-Computer and Information Sciences* 30.3 (2018), pp. 291–319 (Page: 10).

[58] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. "A taxonomy and survey of cloud computing systems". In: *2009 Fifth International Joint Conference on INC, IMS and IDC*. Ieee. 2009, pp. 44–51 (Pages: 13, 14).

[59] Antony Rowstron and Peter Druschel. "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems". In: *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer. 2001, pp. 329–350 (Page: 22).

[60] Huda Saadeh, Wesam Almobaideen, Khair Eddin Sabri, and Maha Saadeh. "Hybrid SDN-ICN architecture design for the internet of things". In: *2019 Sixth International Conference on Software Defined Systems (SDS)*. IEEE. 2019, pp. 96–101 (Page: 19).

[61] José Santos, Tim Wauters, Bruno Volckaert, and Filip De Turck. "Towards dynamic fog resource provisioning for smart city applications". In: *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE. 2018, pp. 290–294 (Page: 25).

[62] Siamak Sarmady. "A Survey on Peer-to-Peer and DHT". In: *arXiv preprint arXiv:1006.4708* (2010) (Page: 21).

[63] Mahadev Satyanarayanan, Zhuo Chen, Kiryong Ha, Wenlu Hu, Wolfgang Richter, and Padmanabhan Pillai. "Cloudlets: at the leading edge of mobile-cloud convergence". In: *2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE)*. IEEE. 2014, pp. 1–9 (Page: 17).

[64] Rüdiger Schollmeier. "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". In: Sept. 2001, pp. 101 –102. ISBN: 0-7695-1503-7. DOI: `10.1109/P2P.2001.990434` (Page: 20).

[65] J Seni. "Peer-to-peer networks:Architectures, Applications and Challenges". In: *2012 3rd National Conference on Emerging Trends and Applications in Computer Science*. IEEE. 2012, p. 1 (Page: 20).

[66] Amir M Sharif. "It's written in the cloud: the hype and promise of cloud computing". In: *Journal of Enterprise Information Management* (2010) (Page: 12).

[67] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge computing: Vision and challenges". In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646 (Pages: 1–3, 18).

[68] Cisco Fog Computing Solutions. "Unleash the power of the Internet of Things". In: *Cisco Systems Inc* (2015) (Page: 17).

[69] Luiz Angelo Steffenel. "Improving the performance of fog computing through the use of data locality". In: *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE. 2018, pp. 217–224 (Page: 26).

[70] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications". In: *ACM SIGCOMM Computer Communication Review* 31.4 (2001), pp. 149–160 (Page: 21).

[71] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. “Vision and challenges for realising the Internet of Things”. In: *Cluster of European Research Projects on the Internet of Things, European Commision* 3.3 (2010), pp. 34–36 (Page: 9).

[72] Madiha H Syed, Eduardo B Fernandez, and Mohammad Ilyas. “A pattern for fog computing”. In: *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs.* 2016, pp. 1–10 (Page: 18).

[73] David Tracey and Cormac Sreenan. “How to see through the Fog? Using Peer to Peer (P2P) for the Internet of Things”. In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT).* IEEE. 2019, pp. 47–52 (Page: 25).

[74] Rob Van Kranenburg. *The Internet of Things: A critique of ambient technology and the all-seeing network of RFID.* Institute of Network Cultures, 2008 (Page: 10).

[75] Luis M Vaquero and Luis Rodero-Merino. “Finding your way in the fog: Towards a comprehensive definition of fog computing”. In: *ACM SIGCOMM Computer Communication Review* 44.5 (2014), pp. 27–32 (Pages: 1, 3, 16).

[76] Jinesh Varia. “Building greptheweb in the cloud, part 1: Cloud architectures”. In: *Amazon Web Services* (2008) (Page: 13).

[77] Q.H. Vu, M. Lupu, and B.C. Ooi. *Peer-to-Peer Computing Principles and Applications.* Springer, 2010, pp. 13–15 (Page: 20).

[78] Quang Hieu Vu, Mihai Lupu, and Beng Chin Ooi. *Peer-to-peer computing: Principles and applications.* Springer Science & Business Media, 2009 (Pages: 20, 21).

[79] Philip Wette and Kalman Graffi. “Adding capacity-aware storage indirection to homogeneous distributed hash tables”. In: *2013 Conference on Networked Systems.* IEEE. 2013, pp. 35–42 (Page: 21).

[80] Geng Yang, Li Xie, Matti Mäntysalo, Xiaolin Zhou, Zhibo Pang, Li Da Xu, Sharon Kao-Walter, Qiang Chen, and Li-Rong Zheng. “A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box”. In: *IEEE transactions on industrial informatics* 10.4 (2014), pp. 2180–2191 (Page: 10).

[81] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. “Fog computing: Platform and applications”. In: *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb).* IEEE. 2015, pp. 73–78 (Pages: 3, 15, 16).

[82] Shanhe Yi, Cheng Li, and Qun Li. “A survey of fog computing: concepts, applications and issues”. In: *Proceedings of the 2015 workshop on mobile big data.* ACM. 2015, pp. 37–42 (Pages: 15, 17).

[83] Hadi Zahmatkesh and Fadi Al-Turjman. “Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies-an overview”. In: *Sustainable Cities and Society* 59 (2020), p. 102139 (Pages: 19, 20).

Curriculum Vitae

Personal information

First name(s) / Surname(s)	**Ahmad Rabay’a**
Address	Wörthstr 19, 47053 Duisburg (Germany)
Telephone(s)	+49 20394173203 Mobile +49 15144893423
E-mail(s)	rabaya.ahmad@gmail.com
Nationality	Jordanian
Date of birth	17 April 1983
Gender	Male

Education

Dates	03 June 2017 - Now
Title of qualification awarded	PhD. Computer Science
Research area	P2P Fog Computing: Enhancing Fog-based IoT Scenarios with Distributed Hash Tables
Name and type of organisation	Heinrich-Heine-Universität Düsseldorf Düsseldorf, Germany
Level	Doctorate

Dates	01 September 2008 - 07 January 2011
Title of qualification awarded	MSc. Software Engineering and Management
Principal subjects / occupational skills covered	- Software Engineering (Basic & Advanced) - Enterprise resource planning (ERP) systems. - IT project management - IT products evaluation and systems usability
Thesis	*Cloud Computing for ERP*: Investigating the possibility of achieving a standard model that is defined technically and accepted organizationally for applying ERP as cloud service.
Name and type of organisation	Linköping University (University) Linköping University, 58183 Linköping (Sweden)
Level	Master

Dates	01 September 2001 - 30 June 2005
Title of qualification awarded	BSc. Software Engineering
Principal subjects / occupational skills covered	- Computer programming and database design - Software engineering - Software analysis and design
Name and type of organisation	The Hashemite University (University) Zarqa, 13115 Zarqa (Jordan)
Level	Bachelor

Work experience

Date	November 2019 – Now
Occupation or position held	Data Engineer

Name and address of employer	Uniper SE Holzstraße 6, 40221 Düsseldorf
Type of business or sector	Energy
Date	June 2017 – May 2022
Occupation or position held	Research Assistant
Name and address of employer	Heinrich-Heine-Universität Düsseldorf Düsseldorf, Germany
Type of business or sector	Educational Institute (University)
Date	30 October 2011 – 31 May 2017
Occupation or position held	Full-time Instructor & E-Learning Specialist
Name and address of employer	Palestine Technical University / Kadoorie Kadoorie, Tulkarm – Palestine
Type of business or sector	Educational Institute (University)
Date	01 March 2006 - 31 August 2008
Occupation or position held	Software Developer & BI Specialist
Name and address of employer	Pioneers Information Technologies Gardens Street, 11953 Amman (Jordan)
Type of business or sector	Business solutions provider

Personal skills and competences

Mother tongue(s) **Arabic**

Other language(s)

Self-assessment	**Understanding**				**Speaking**				**Writing**	
	Listening		Reading		Spoken interaction		Spoken production			
English	C1	Advanced user	C1	Advanced user	C1	Advanced user	C1	Advanced user	C1	Advanced user
German	A1	Basic User	A1	Basic User	A1	Basic User	A1	Basic User	A1	Basic User

Conferences

1. *International Symposium on Networks, Computers and Communications (ISNCC)*, 18-20 June 2019, Istanbul, Turkey.
2. *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 14-17 July 2019, Atlanta, GA, USA.
3. *4th Conference on Cloud and Internet of Things (CIoT)*, 07-09 October 2020, Niterói, Brazil.

Personal Publications

Reviewed conference papers

1. Ahmad Rabay'a and Kalman Graffi. "Implementing Enterprise Resource Planning Systems in the Cloud: Challenges and Solutions". In: *International Symposium on Networks, Computers and Communications (ISNCC)*. 2019. Acceptance Rate: 37%

2. Ahmad Rabay'a, Eduard Schleicher and Kalman Graffi. "Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios". In: *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2019. Acceptance Rate: 16.7%

3. Ahmad Rabay'a, Pascal Kudla, Lukas Kubitza, Kalman Graffi and Michael Schöttner. "Reducing IoT Bandwidth Requirements by Fog-Based Distributed Hash Tables". In: *Conference on Cloud and Internet of Things 2020 (CIoT'20)*. 2020. Acceptance Rate: 34%

Eidesstattliche Erklärung
laut §5 der Promotionsordnung vom 06.12.2013

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Ort, Datum	Ahmad Rabay’a