Aalto University
School of Engineering

Samu Viitanen

Creation of a Building Operating System: Holistic Approach

Master's Thesis
Department of Built Environment
School of Engineering
Aalto University

Espoo, 1 December 2017

Bachelor of Science in Technology
Samu Viitanen

Supervisor: Professor Seppo Junnila
Instructor: Professor Timo Seppälä

Aalto-yliopisto
Insinööritieteiden korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Tekijä Samu Viitanen

Työn nimi Rakennuksen käyttöjärjestelmän luonti: kokonaisvaltainen lähestymistapa

Koulutusohjelma Kiinteistötalous

Pää-/sivuaine Kiinteistötekniikka **Koodi** M3007

Työn valvoja Professori Seppo Junnila

Työn ohjaaja(t) Professori Timo Seppälä

Päivämäärä 1.12.2017 **Sivumäärä** 85 **Kieli** englanti

Tiivistelmä

Tämän diplomityön tarkoituksena on tutkia rakennuksen käyttöjärjestelmän holistisia vaatimuksia. Laaja kirjallisuuskatsaus tehtiin aiheen ymmärtämiseksi, joka tutkii käyttöjärjestelmien evoluutiota rinnakkain tietojenkäsittelyn historian kanssa, tarkoituksena hahmottaa käyttöjärjestelmän käsitettä. Lisäksi, eri rakennusten tietojärjestelmiä, mukaan lukien rakennusautomaatiojärjestelmiä ja esineiden internet -järjestelmiä käytiin läpi ymmärtääkseen nykyisiä ja tulevia trendejä rakennusteknologiassa. Edelleen kirjallisuuskatsaus tutkii televiestintää ja sähköistä tunnistautumista niiden kehityksen ja standardisoinnin kautta kohti yhteentoimivuutta, tarjoten tietoa siitä, miten yhteentoimivuutta voitaisiin kehittää rakennusjärjestelmissä.

Haastattelututkimus tehtiin diplomityön empiirisenä osuutena, jonka tarkoituksena oli laajentaa työn teoreettista viitekehystä. Tusina rakennusten digitalisaation asiantuntijaa haastateltiin, joilta kysyttiin rakennusjärjestelmien nykytilasta ja tulevaisuudesta. Lähemmin, keskustelut käsittelivät avoimia järjestelmiä, avointa dataa, alustan omistajuutta, disruptiota, menestyssovelluksia, käyttäjäkeskeisyyttä sekä Suomen kansainvälistä potentiaalia rakennuksen käyttöjärjestelmän näkökulmasta.

Rakennuksen käyttöjärjestelmä vaatii rakennuksen sisällä olevien eri teknologioiden yhteenliittämisen, sekä yhteistyötä rakennusta käyttävien ja hallinnoivien osapuolten välillä. Järjestelmän pitäisi hyödyntää avoimia standardeja ja mahdollistaa avoimen datan käytön. Käyttäjäkeskeistä suunnittelua pitäisi kannustaa loppukäyttäjien etuja suosien. Järjestelmän täytyy levitä globaalisti saavuttaakseen kriittisen massan ja ottaakseen käyttöön sen koko potentiaalin. Jokaisella samankaltaisella rakennuksella täytyisi olla käytössään yhtäläiset ominaisuudet, mahdollistaen samojen palveluiden ja sovellusten käytön missä tahansa käyttöjärjestelmää käyttävässä rakennuksessa, täten mahdollistaen siirrettävyyden. Järjestelmä vaatii sopivat ohjelmointirajapinnat, abstraktiot ja ohjelmistokehykset sovellus- ja palvelukehittäjien tarpeita varten. Laaja kehitysyhteisö vaaditaan alustan levittämiseksi ja sovellustarjonnan laajentamiseksi.

Avainsanat digitalisaatio, käyttöjärjestelmä, kiinteistö- ja rakentamisala, älykäs rakennus, rakennusautomaatio

Aalto-yliopisto
Insinööritieteiden korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Author Samu Viitanen		
Title of thesis Creation of a Building Operating System: Holistic Approach		
Degree programme Degree Programme in Real Estate Economics		
Major/minor Real Estate Planning		**Code** M3007
Thesis supervisor Professor Seppo Junnila		
Thesis advisor(s) Professor Timo Seppälä		
Date 1.12.2017	**Number of pages** 85	**Language** English

Abstract

Purpose of this thesis is to examine requirements for a building operating system from a holistic perspective. To understand the context of the subject, an extensive literature review was carried out which explores the evolution of operating systems alongside the history of computing, unravelling the concept of an operating system. In addition, various building information systems, including building automation systems and internet of things systems are reviewed in order to understand modern and future trends of building technology. Furthermore, literature review investigates telecommunications and digital identity authentication through their evolution and standardisation towards interoperability, to provide knowledge on how to achieve interoperability in building systems.

An interview study was conducted as the empirical part of the study in order to complement the theoretical framework of the thesis. A dozen building digitalisation experts were interviewed, inquiring their insights on the current and future situation of building systems. More closely, open systems, open data, platform ownership, disruption, killer applications, user-centredness, and Finland's opportunities were discussed in respect of the building operating system.

Building operating system requires connection between various technology inside a building, and collaboration between various parties who use and manage the building. The system should exploit open standards and enable open data. User-centred development should be encouraged for the benefits of end users. The system needs to expand globally to achieve critical mass and unleash its full potential as a platform. Each building with similar properties should have the same features, being able to use same services and applications in any building with an operating system, thus enabling portability. The system requires convenient software development kits, application programming interfaces and abstractions for the needs of software and service developers. A vibrant developer community is required to expand the platform and enable a wide range of services and applications.

Keywords digitalisation, operating system, real estate, construction, smart building, building automation

Foreword

Biggest thanks belong to my thesis advisor, Timo Seppälä, who suggested this interesting thesis topic last spring. Idea of a building operating system is an intriguing one which has not received much discussion before. Also, thank you for enduring my super-tight schedule and guiding me throughout the whole process. Furthermore, I thank my supervisor Seppo Junnila for giving his busy time to review my thesis.

Thank you to everyone who agreed to be interviewed for this thesis. I would have hoped for more of you, but circumstances did not allow for it this time.

Another big thank you goes to my boss and advisor behind the scenes, Teemu Lehtinen at KIRA-digi project, who has given great advice and insights about the thesis and the whole digitalisation scheme of built environment sector. KIRA-digi project has been a real eye-opener, giving me a spectacular outlook of the whole built environment sector and its development. I also want to thank other members of KIRA-digi, especially Minna and Virve at the Ministry of Environment, who have been utmost kind towards me and my hurries with this thesis. Our journey with KIRA-digi will continue for another year at least.

Entire journey towards graduation has been a blast. For half a year, I had the chance to visit University of Tokyo as an exchange student, where I learned Japan's language, culture and land use planning, played in student orchestras and travelled around the country. In Otaniemi, I have had the chance to play in Retuperän WBK and Polyteknikkojen Orkesteri, participating in dozens of concerts and gigs during the past five years. I have also had the honour to be a founding member of the Aalto University mahjong club. Thank you to everyone whom I have met during these five and a half years. Hopefully, our journey continues in the future.

Finally, I want to thank my family and dearest friends.

In the Southern Border of Rainy and Gloomy Vantaa, December 2017

Samu Viitanen

Table of Contents

Acronyms

AIOTI	The Alliance for Internet of Things Innovation
API	Application Programming Interface
ARPA	Advanced Research Projects Agency, USA
ATIS	Alliance for Telecommunications Industry Solutions
BA	Building Automation
BAS	Building Automation System
BOS	Building Operating System
CLI	Command Line Interface
CRM	Customer Relationship Management
DDC	Direct Digital Controller
DOJ	United States Department of Justice
ERP	Enterprise Resource Planning
ES	Enterprise System
ETSI	European Telecommunications Standards Institute
EU	European Union
FCC	Federal Communications Commission, USA
1/2/3/4/5G	Wireless mobile telecommunications technology generations
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HVAC	Heating, Ventilation and Air Conditioning
ICT	Information and Communications Technology
IE	Internet Explorer
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IT	Information Technology
ITU	International Telephone Union
JCL	Job Control Language
MRP	Material Requirements Planning
MRP II	Manufacture Resource Planning
OS	Operating System
PC	Personal Computer
RECS	Real Estate and Construction Sector
SDK	Software Development Kit
SOA	Service-Oriented Architecture
SRI	Stanford Research Institute
UCLA	University of California, Los Angeles
UCSB	University of California, Santa Barbara
UI	User Interface
UX	User Experience
W3C	World Wide Web Consortium
WWW	World Wide Web

1 Introduction

1.1 Background

Digitalisation has taken industries, governments and consumers by storm during the past two decades after the emerge of ubiquitous internet. Companies in all industries are trying to cope with the demands digitalisation is posing. Especially now, the Industry 4.0 trends are driving the shift from product- to service-oriented business, and the necessity of organisational development in addition to technological has been realised (Lasi et al. 2014). Geissbauer et al. (2016) point out in their Industry 4.0 survey that the annual digital revenue increases 2.9 % on average. Digitalisation is hoped to bring cost reductions enabling shorter operational lead times, higher asset utilisation and maximal product quality. However, promised cost reductions are not the only factors why digitalisation is rapidly expanding into every aspect of life. Value co-creation, enhanced user experience, user-driven development and other customer-centred eases and enhancements are also in the core of digitalisation paradigms (Lenka et al. 2016; Jungner 2015; Geissbauer et al. 2016).

More closely in the global construction sector, digitalisation is still taking its baby steps. McKinsey made a research on the levels of digitisation in different industries, where construction was in the last place (Gandhi et al. 2016). The research included 22 industry sectors, including media, oil & gas, mining, government and healthcare. Other digitalisation surveys imply the same, including a paper on the Industry 4.0 related technologies pursued in the construction industry, an older PwC digitisation survey and a recent Nordic survey on digitalisation (Oesterreich & Teuteberg 2016; Friedrich et al. 2011; Magenta Advisory 2017). As for global real estate sector, McKinsey's research placed it on the 12th place out of 22 industries, thus having a mediocre digitalisation level.

Puhto et al. (2016) discovered that the traditionally conservative sectors of real estate and construction are still taking their first steps in digitalisation in Finland. The average grade that the companies gave themselves was 5,2 (in a scale from 1 to 10). Moreover, digitalisation is seen mostly as a tool to boost current business, not create new business. Organisations do not see themselves enough agile and competent to develop digitality, and the advantages of digitalisation are not sufficiently explicit to them yet. Likewise, Snellman (2016) concludes with similar results in her thesis. Piikkilä et al. (2016) point out how important digital information management is for the success of construction projects, considering their complexity and intricateness. A recent report about the quality of built environment in Finland gave a grade of 7- (on a scale of 4 to 10) to digital solutions in the built environment sector, mentioning that the digitalisation of the industry has progressed, yet there are many opportunities in digitalisation that are still utilised limitedly. Furthermore, report expresses special critique for the lack of process evolution towards digitalisation, demanding the utilisation of joint data sources and the use of building information models in facility management for example (ROTI 2017).

Like digitalisation, construction sector has also been a laggard in productivity development. According to McKinsey and The Economist, construction industry's productivity has not risen since 1995, while manufacturing productivity has almost doubled (The Economist 2017; Changali et al. 2015). In Finland, construction productivity has remained the same for over 40 years (Lohilahti 2017). Data from Statistics Finland (2016) implies similar productivity shortcomings in Finland, offering even worse figures for real estate sector than construction sector, shown in Figure 1.

Compared to 1975, construction productivity has remained the same and real estate productivity has dropped, while manufacturing has over doubled, and finance, insurance, agriculture, forestry and fishery has over tripled.

Reasons for productivity stagnation in construction industry include fragmented structure of the industry, dominated by a majority of small, independent and specialised contractors; low investment in research and development, poor organisation, inadequate communication, missed connections, poor short-term planning and contractual misunderstandings (The Economist 2017; Changali et al. 2015). In Finland, one reason for the situation has been the overly confidence of builders, believing that the customers will pay for the construction in any case. Fragmented structure of the industry has also fragmented construction projects, carried out by separate designers, contractors, subcontractors and consults which easily leads to quarrelling (Lohilahti 2017). Clearly, a solution is needed to connect and unite the industry in order to boost its productivity.

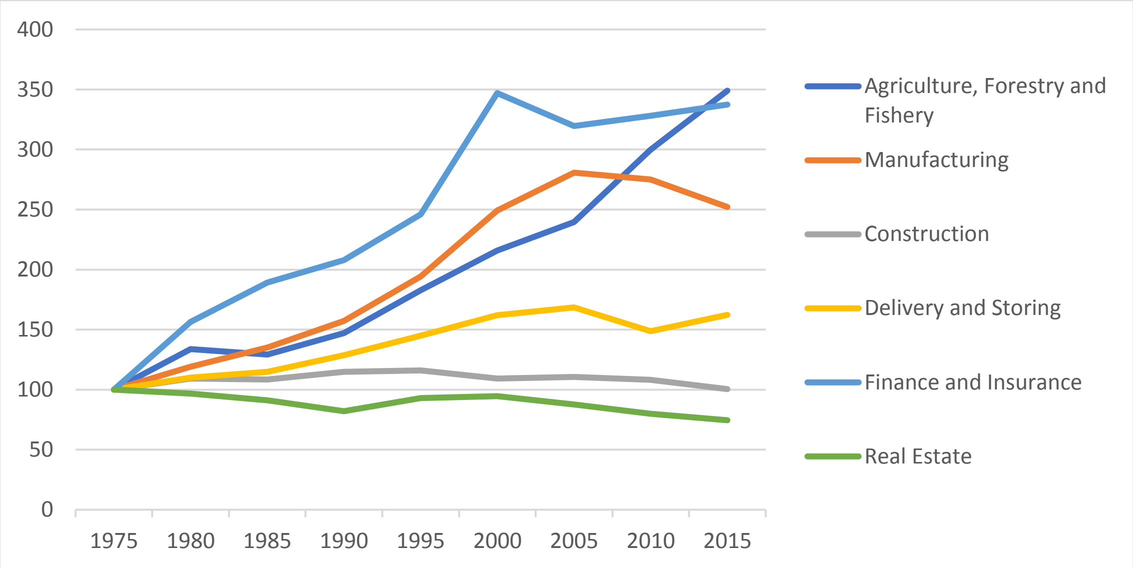

Figure 1. Productivity of Different Industries Based on Added Value (Statistics Finland 2016).

Consequently, it appears that RECS is in the beginning of its digital revolution. This a major opportunity for digitalists and digital disruptors, considering the facts that built environment in Finland is 80 % of national wealth, 13 % of employment, 20 % of GDP and 99 % of time spent (ROTI 2017). We are all heavy users of built environment, but its digital potential still remains mostly in hiding.

Perhaps the most prominent digitalisation trend currently in RECS is smart buildings. Global smart building market size is expected to reach $36 billion by 2020, expanding beyond fivefold from 2014 to 2020 (GlobeNewswire 2017). Likewise, couple concepts related to smart buildings are found from the hype cycle, including Connected Home, IoT Platform and Smart Workspace (Gartner 2016). The world is already dreaming, developing and creating smart buildings, considering the substantial amount of newspaper articles found on the internet discussing smart buildings and its hyponyms. Also, major companies are moving up in smart buildings, including Cisco, Schneider Electric, IBM, Panasonic and Hitachi. Dozens of smart building start-ups are also emerging around the world, for example in Finland companies like Cozify, 720°, Houm and Leanheat are creating sensor, IoT and building automation solutions.

In building automation and internet of things, the lack of interoperability and integration has remained as a problem (Al-Fuqaha et al. 2015; Domingues et al. 2016; Kastner et al. 2005; Dawson-Haggerty et al. 2013; Al-Fuqaha et al. 2015; Borgia 2014; Li et al. 2014). Various building automation systems have continued to exist in silos, with the difficulty of managing them together. Thus, an operating system is needed to knit various IoT and building automation systems, abstracting the details of their hardware devices and communication protocols. Furthermore, this type of operating system enables the eased production of applications and services for smart buildings which can truly yield added value to users, residents, tenants and owners of buildings.

Built environment aside, ICT has reshaped nearly every aspect of our life in the past three quarters of a century. Computers have evolved from warehouse-sized industrial machines into day-to-day commodities of digital comfort and pleasure. Since the 1990s, the internet has shaped ICT even further with its disruptive potential in communicating and connecting. Especially after the so-called second phase of smartphones when the iPhone was released in 2007 (Sarwar & Soomro 2013). Today, nearly half of the people in the world use the internet (Internet World Stats 2017; Internet Live Stats 2017).

Telecommunications industry have succeeded to develop several universal open standards, in order to enable interoperability and communication between various heterogeneous telecommunication devices. Most notably, in the 2G era, GSM became the first global universal standard, initially expanding through joint efforts between European countries and later diffusing across the world. GSM was one of the first standardisation schemes to utilise open standardisation and modular problem solving, enabling competition between service providers and mobile manufacturers (Funk 2009). Currently, LTE used in 4G technologies have followed a similar path of standardisation development as GSM, driven by inter-governmental support and open standardisation policies.

Another major revolutioniser of computing in addition to the internet has been operating systems. Operating system in short is a software that enables applications to interact with a computer's hardware (Deitel et al. 2004). In other words, operating system provides user applications clean abstract of the resources instead of having to deal with messy and complex hardware directly (Tannenbaum 2009). Operating systems started to universalise after the introduction of GUI-enabled and more user-friendly OS starting from the 1980s. 1980s and 1990s were times of intense PC competition between Apple, Microsoft, IBM and Commodore. In the PC market, IBM PC compatibles claimed victory during the 1980s (Dediu 2017). Biggest competitors in the IBM PC operating system market were IBM OS/2 and Microsoft Windows. Microsoft Windows cornered OS/2 quickly in the 1990s, thus becoming the most popular operating system on desktop computers. Today, Microsoft is the third largest public company by market value globally (Forbes 2017).

Another major step in consumer computing was the emerge of mobile computing and their operating systems. In the early 1990s, the recently emerged cellular phone started to evolve from a mobile telephone into a mobile computing device (Hall & Anderson 2009). Mobile phones started to feature text messages and small built-in applications such as calendar and address book in addition to voice connectivity (Sorensen et al. 2015). Bit by bit, more applications and gadgets were connected to the mobile phone which facilitated the evolution of mobile telephones into mobile computing devices. The smartphone era escalated with the release of iPhone in 2007, after which phones became personal computers on a mobile device. Currently, two major operating system platforms for mobile devices exist, namely Apple's iOS and Google's Android, which have succeeded to build large and successful application marketplaces with vibrant developer communities and numerous value-adding applications (Sorensen et al. 2015). Smartphones, their mobile operating systems, digital distribution platforms and applications have completely disrupted the whole ICT industry and computer usage.

1.2 Research questions and scope

The object of this study is to determine the holistic requirements of a building operating system. To find these requirements, it is necessary to investigate the purpose and evolution of various operating systems such as computer operating systems and mobile operating systems. Additionally, enterprise resource planning systems are reviewed which function as operating systems for enterprise management.

Literature review also covers various information systems used in buildings, including building automation systems and IoT systems, in order to understand modern ICT used in buildings. Furthermore, previous research on building operating system prototypes is reviewed. Purpose of the building ICT review is to give an understanding of how contemporary building systems function and how do they need to be developed further in order to create a building operating system.

In addition to computing and building ICT literature, standardisation development in telecommunications and electronic authentication is reviewed. Current building information systems lack interoperability; thus, an examination of telecommunications and electronic authentication standardisation is carried out to give insights about their possible key success factors in interoperability, integration and universal diffusion through standardisation. Furthermore, successful standardisation schemes in other industries can give suggestions on how to achieve interoperability and universality in building operating systems.

With this framework of literature research, scope of the study is framed into one main research question and four auxiliary research questions:

I. **What are the holistic requirements for a building operating system?**

A. What is an operating system, what are their purpose and how have they evolved?
B. What information systems are currently used in buildings?
C. What kind of development does the contemporary building ICT require in order to create a building operating system?
D. How can interoperability be achieved with a building operating system?

This study will find answers to these research questions in a holistic fashion. In order to answer the questions, the thesis will use literature from the fields of information and communications technology, building automation, communication protocols, construction and real estate, digitalisation, identity authentication, platform economy and computer architecture.

1.3 Methods

This thesis provides answers to the research questions through a literature review and an interview study. Literature review provides answers to all research questions by contemplating evolvement of computers, computer operating systems, enterprise resource planning systems, building information systems, telecommunications protocols and electronic authentication. More closely, auxiliary question A is examined through a literature review on computer and operating system evolution, question B and C is examined by reviewing contemporary and prospective building ICT and question D is explored through a standardisation development review of telecommunications and electronic authentication. Main research question is investigated through the whole literature review.

To assist the literature review, an interview study is carried out where both present and future state of building ICT is discussed with building digitalisation experts. Primarily, interview study provides additional answers to auxiliary questions B, C and D and the main research question. With the combination of literature review and interview study, discussion and suggestions for the research questions are conducted and a framework for the main research question is compiled.

1.4 Structure of the Thesis

This thesis is divided into eight chapters. First chapter introduces the study in question, laying out a background for the research and defines research questions and methods for the purpose of the thesis. Second chapter covers the evolution of computers and operating systems, going through the development of computers, mobile devices, the internet, mobile and desktop operating systems and enterprise resource planning systems. Third chapter gives an overview of building automation systems, including their structure, architecture, purpose and standardisation. Furthermore, third chapter discusses internet of things, market structure of building ICT and building operating systems depicted in other studies. Fourth chapter discusses interoperability in telecommunications and electronic authentication with the aid of standardisation, following the evolvement of NMT, GSM, TUPAS and Mobiilivarmenne, among other things. Fifth chapter gives background on the real estate and construction sector and lays out the interview study methodology, objective and implementation. Sixth chapter reviews the results of the interview study and reflects it toward the theoretical framework. Seventh chapter discusses and suggests answers for the research questions and the eighth chapter concludes the thesis.

2 Evolution of Operating Systems and Computing

Operating systems, computing devices, the internet and mobile devices have changed profoundly in the last century. In this chapter, cruxes along the evolution of computers including mobile devices and operating systems are examined. Also, evolution of internet and enterprise resource planning systems are discussed.

This chapter lays out an extensive overview of operating systems and computing devices from an evolutional perspective. Chapter discusses how operating systems have developed historically, why do they exist, what are their purpose, what types of operating systems there are, which operating systems have prospered, and what benefits do they offer. In order to comprehend the evolution and context of these operating systems, an overview on the history of computing and software engineering is laid out alongside the operating system literature. Additionally, enterprise resource planning systems are discussed in the last subchapter as the operating systems of enterprise management. Altogether, this chapter gives context to operating systems and provides implications on what an operating system would stand for in a building.

2.1 Early History of Computing

Computer-like machines have been built since the 17th century when Blaise Pascal built a mechanical calculating machine to aid his father in his tax collecting (Tanenbaum 2006). The next major milestone happened in the 19th century when Charles Babbage, a mathematics professor at University of Cambridge built a so-called difference engine which could compute tables of numbers useful for naval navigation. This machine could only run a single algorithm. Later, Babbage started to build the analytical engine which could read computing instructions from punched cards and carry them out. In other words, the machine was general purpose and it was programmable with simple software. For this purpose, Babbage hired a young woman named Ada Augusta Lovelace who became the first computer programmer.

George Boole introduced **Boolean logic** in 1847, a fundamental paradigm in the development of digital electronics, and Claude Shannon who showed that Boolean algebra could be used to make simpler circuits and telephone routing switches (O'Regan 2016). Virtually all digital circuits today use electronic representations of Boolean logic functions in their logic gates.

The Second World War brought the first wave of large general-purpose **digital computers**. Before digital computers, computers were already being built from levers and gears (mechanical computers) and switches and relays (electro-mechanical computers). Eventually, first digital computers built from **vacuum tubes** were created during the Second World War. Quickly after the war, first commercial digital computers came to the market.

A major turning point towards modern computing in the post-war era was the introduction of **von Neumann architecture**, name after the interdisciplinary scientist John von Neumann. The architecture was uncovered in the middle of the 1940s, when J. Presper Eckert and John Mauchly started to envision the **stored-program concept** to ease the use of computers. Idea of the stored-program concept is that both machine instructions and data are stored in a single store i.e. in single memory (O'Regan 2016). In 1945, John von Neumann wrote a report on their stored-program concept computer where the so-called von Neumann architecture was described for the first time, carrying the design ideas of stored-program concept further. Eckert and Mauchly had been designing hardware constructions of the idea but von Neumann described the machine in terms of its logical structure which clarified and gave form to the von Neumann architecture. (Ceruzzi 2003, Stallings 2010.)

The key advantages of von Neumann architecture over the previous computer architectures were that it was far simpler to reconfigure the computer to perform a different task. Before von Neumann architecture, computers were fixed program machines that were designed to do one specific task. If the machine was required to run a different algorithm, it was required to rewire physically, evidently being an arduous task. With the use of von Neumann architecture, machines were reprogrammable so that if the computer was needed to operate other instructions, it was only required that new machine instructions were entered to computer memory, whereupon no physical rewiring was required. (O’Regan 2016, Ceruzzi 2003.)

The legacy of von Neumann architecture is substantial in making computing more flexible and efficient. Both von Neumann architecture and stored-program concept have led to the establishment of programming and software design and making them separate from hardware design. The basic design of von Neumann architecture has persisted until the middle of the 1990s. (Ceruzzi 2003.)

2.2 Age of Transistors, Mainframes, Minicomputers and Chips

Transistor was invented at Bell Labs in 1948 by John Bardeen, Walter Brattain and William Shockley. Within ten years of invention, transistors revolutionised computers completely and in the 1960s vacuum tube computers were already obsolete (Tanenbaum 2006). Compared to vacuum tubes, transistors are smaller, cheaper and they dissipate less heat but still can be used similarly as vacuum tubes to construct computers. Vacuum tubes require wires, metal plates and a glass capture, whereas transistors are solid-state devices which are made from silicon (Stallings 2010). Vacuum tubes could burn similarly to light bulbs, which meant that vacuum tube computers were non-functional as long as there were burnt tubes in the computer (O’Regan 2016). Therefore, transistor computers brought many advantages over vacuum tube computers.

Transistors did not reach the commercial computer market until the end of the 1950s. Early transistor computers manufactured were still prototypes used and tweaked by research institutes and the government. Then in the end of the 1950s, first commercial transistor computers came into the market. IBM’s turning point in the transition from vacuum tubes to transistors was the introduction of the often-noted classic mainframe computer, IBM 7090 (Ceruzzi 2003).

In 1958, scientist named Jack Kilby at Texas Instruments invented the **integrated circuit** which allowed many transistors to be combined on a single chip, aiding the evolution of computing. Original IC created by Kilby was made of germanium but later in 1960 Robert Noyce of Fairchild Semiconductors built an IC with silicon which is the material used today for semiconductors. (Ceruzzi 2003, O’Regan 2016.)

The IC is extremely compact, especially compared to a circuit made of independent components, and it further accelerated the reduction of size and cost of electronics (Ceruzzi 2003). Throughout the years, ICs have gone through several generations, starting from small-scale integration in the early 1960s when chip had less than 30 transistors. Today, an integrated circuit can have more than ten billion transistors (see 24-core AMD EPYC 7401P microprocessor). The growth of the number of transistors on a chip was observed in 1965 by Intel’s co-founder Gordon Moore who predicted that the number of transistors on a chip would double every 18 months for the next ten years. In fact, this phenomenon called **Moore's Law** still stands today, but it is likely that the growth in transistor density will slow to a doubling of density every 3 years by 2015 (O'Regan 2016). Moore's Law is also affiliated with the exponential growth of processor speed, memory capacity and other capability advancements of digital electronic devices.

It took a while before IC computers started influencing the commercial computer market, as they were an expensive technology prior to mass production. Initially, IC computers made its largest commercial impact on the minicomputer market and on embedded systems. United States aerospace community was especially interested in IC, and its projects of Minuteman missiles and Apollo Guidance Computer launched the volume production for ICs (Ceruzzi 2003). Likewise, minicomputers paced the march of ICs with the launch of the first IC minicomputer PDP-8 by DEC in 1965.

Minicomputers arose from the traditional massive-sized mainframes during the 1960s. As was mentioned, the development of minicomputers was hastened by the introduction of integrated circuits. Minicomputers differed from the original mainframes that they were smaller (hence the name) and cheaper. In the 1960s, they were described as the smallest general-purpose computers (O'Regan 2016). Minicomputers were not direct competitors to the mainframes, instead they opened new markets as a personal interactive device which gave the chance to directly interact with a computing machine for people, particularly engineers and scientists in the early days. Later, minicomputers paved road for the personal computers in the 1970s (Ceruzzi 2003).

2.3 Early Operating Systems

Primitive Operating Systems and IBM System/360

In the inception of digital computers in the 1940s, no operating systems were used. Machines at the time were such rudimentary that programmers generally typed their programs one bit a time first with mechanical switches and later with punched cards (Tanenbaum 2009). Because writing pure bits into computers was rather difficult, programming languages started to emerge; first low-level assembly languages which were converted to machine code via an assembler (O'Regan 2016). Assembly languages uses short abbreviations to represent the basic operations of a computer.

In the 1950s, first "operating systems" were created. First operating system was implemented for the IBM 701 computer by General Motors Research Laboratories (Deitel & Deitel et al. 2004). 1950s computers typically executed one **job** at a time which meant running a certain computational task with a set of program instructions. Changing through jobs with the mainframe was a tedious task, so it was streamlined with the **batch-processing system**. Batch system essentially collected a certain amount of jobs into one batch (hence the name) by a computer and then the batch of jobs was executed in one go by another computer. The software interpreting the batches was the operating system which could distinguish jobs and their instructions from each other in the batch.

Batch-processing systems evolved in the 1960s by running several jobs at once. For this it was needed to develop a concept called **multiprogramming**, in which many jobs were saved into memory at once and they used computers resources in a particular order (O'Regan 2016). Furthermore, one job could use the computer's processor and another job could use the input and output devices at the same time (Deitel & Deitel et al. 2004). This timesharing of computer resources was operated by a primitive operating system called the **multi-batch system**. Addition to multi-batch systems, an operating system called Compatible Time-Sharing System (CTSS) was introduced in 1961 at MIT, which was a variant of multiprogramming where each user has an online terminal (Tanenbaum 2009). This created the concept of interactive computing because timesharing computers could provide fast, interactive service for several users through online terminals beside running large batch jobs.

In 1964, the IBM **System/360** family of computers were introduced. The various computer models in the 360 family were designed to be hardware compatible and they all used the same OS/360 operating system. The OS/360 was a multi-batch system which used multiprogramming and it had three variants for different types of program processing. They all used the same application programming interface (API) and job control language (JCL).

Historically, OS/360 was very ambitious operating system and ultimately it spent half billion dollars of IBM (Moumina 2001). It was the biggest and most complex program built at the time, consisting of hundreds of program components and including more than million lines of code. The operating system was introduced in 1964 but it took three years before it was launched into the market. At most, more than 1000 people were working on the operating system. OS/360 was later redeveloped for the IBM 370 and 390 series.

Roots of Software Engineering and the Garmisch Conference

Through the early post-war era, software engineering was still taking its first steps. After the introduction of assemblers, other more high-level programming languages were introduced, most notably **FORTRAN** to the IBM 704 in 1957. It was a major success among IBM customers and the language has continued to live to the present time. FORTRAN had a relatively easy-to-use syntax resembling ordinary algebra, it had functional and distinct user manuals and its compiler generated machine code rather efficiently. Aside from FORTRAN, a business-oriented programming language **COBOL** was introduced in the US a couple of years after FORTRAN. Also, a more rigorous alternative for FORTRAN named **ALGOL** was introduced in Europe in the turn of the decade. (Ceruzzi 2003.)

Software engineering hit a critical point in 1968, when a conference called "Software Engineering" was held in Garmisch, Germany. The idea of the conference was to bring software creation to the level of engineering, in a way that programmers would have the theoretical foundations and disciplines found in traditional fields of engineering such as program users could trust the program in the same way as we trust a skyscraper to keep standing because of the trusted civil engineers. This manner of rigid and certificated programming was never actually implemented but it gave the surge for the rise of software industry and the necessity of trusted and functional software. Quickly after the conference, IBM decided to unbundle their software and services from hardware which issued the need to distinguish software and hardware. Other prominent proposals were made at the conference, including the development of more structured programming regimes and more sophisticated programming languages. (O'Regan 2016, Ceruzzi 2003, Moumina 2001.)

UNIX – First Portable Operating System

The Turing Award winning operating system **UNIX** was designed and developed at Bell Labs by Ken Thompson and Dennis Ritchie in the early 1970s. UNIX has influenced every aspect of computing in the 1990s (Salus 1994). There are several reasons suggested why UNIX was as popular, including that it has a simple user interface, it is written in high-level programming language (C), it uses a hierarchical file system, it uses a consistent format for files, it provides a simple consistent interface to peripheral devices, it is a multi-user, multiprocess system and it hides the machine architecture from the user (Bach 1986). Key reasons why UNIX became so popular stems from its simplicity and consistency.

UNIX's roots go back to the Multics operating system which came to design also at the Bell Labs in the middle of the 1960s. Multics was originally designed for the mainframe, and eventually it was discovered that Multics would be a colossal and costly system to develop, thus the project was cancelled in 1969. From its ashes, a less ambitious system rose, namely UNIX. Funding for the Multics operating system ended with its cancellation, so UNIX was initially developed on a smaller-scale minicomputer PDP-7. First version of UNIX was written in assembly language by Thompson but later Ritchie joined the project and helped rewriting UNIX in the **C programming language** in 1973. Interestingly, the C programming language was designed by Ritchie and was first released in 1972 for the purpose of programming UNIX with it. C programming language is another success story which became widely used by the industry and influenced later programming languages such as C++ and C#. (Bach 1986, Moumina 2001, O'Regan 2016, Ceruzzi 2003.)

The use of C made UNIX portable to various computers, thus becoming a widely-used operating system. Initially, the operating system was sold at a nominal price and without advertising, marketing nor supporting the system because it was owned by AT&T, a government monopoly back then, which prohibited its marketing of computer products (Bach 1986). Nonetheless, UNIX began to rise in popularity, first in universities and the US government, and later in all sides of business and industry. By 1984, there were approximately 100,000 UNIX systems installed in the world (Bach 1986). UNIX started to surface many UNIX-like operating systems, including Berkeley UNIX (BSD) in the early days, and later on it has evolved into numerous operating systems, including two prominent operating systems: Linux and macOS. The original inventors did not make real money with UNIX, including their mother-company AT&T, because it was an "open" system in a sense and was essentially given away (Ceruzzi 2003).

2.4 The Tidal Wave of Personal Computers

Launch of the Microprocessors and the first PCs

The rise of the personal computers (PC) started with the emerge of **microprocessors**. Perhaps the greatest breakthroughs in the microprocessor market were made by Intel which was established in 1968. In 1971, Intel created the world's first microprocessor, the Intel 4004 which was the first chip to contain all the components of a CPU on a single chip (Stallings 2010). Intel 4004 had the same computing power as the 1940s room-sized ENIAC computer, thus the size difference was immense. After 4004, Intel released 8008 in 1972 and 8080 in 1974. Intel 8080 became the industry standard which made Intel the industry leader in 8-bit microprocessor market. Intel 8080 also played an important role in starting personal computer development, especially because the hobbyist-friendly Altair 8800 was built with it (O'Regan 2016). IBM PCs and their clones used mostly Intel microprocessors which made Intel the market dominator by the end of the 1980s.

The timesharing mainframes and minicomputers, especially the PDP computers had brought the concept of interactive computing. Finally, in the middle of the 1970s, waves from the shrinking and power-growing microprocessors, more-capacious memory chips and the rise of interactive computing with timesharing computers had begun the revolution of personal computing. Often noted as the first real personal computer, **Altair 8800** was launched in 1974. Altair 8800 became a big hit amongst computer scientists and hobbyists because it was ten times cheaper than computers that day and it was easily tweakable. In fact, the Altair 8800 was not very useful at its own but computer hobbyists could design and build all they wanted with the "starting kit" it supplied, giving the possibility for computer enthusiasts to develop add-on features, attach extra memory and peripherals and write own personalised software for the PC. Altair 8800 was sold in two different versions: build-it-yourself kit which was cheaper and aimed for hobbyists and an assembled version. (Ceruzzi 2003, O'Regan 2016.)

Altair 8800 was also the igniter for the establishment of **Microsoft** in 1975. Back then, Bill Gates was still studying in Harvard University when his high school friend Paul Allen noticed a cover representing the new Altair 8800 after which they decided to write a BASIC compiler for the machine. Before BASIC, Altair could only be programmed with certain switches on the frontboard. BASIC is an easy-to-learn high-level programming language which had not been used on else than mainframes back then. Ultimately, Bill Gates and Paul Allen developed an interpreter and 4k/8k versions of BASIC for the machine and it was released in July 1975. The Altair BASIC product was sold separately and it cost an additional $60/$75. With this first product by Allen and Gates, Microsoft was established. (Ceruzzi 2003, O'Regan 2016.)

Another important initiation in the personal computer industry was the **Xerox Alto** which was introduced in early 1973. The Alto PC was the first computer to work on a graphical user interface (GUI) and it later influenced the creation of Apple Lisa and Macintosh. Before and also quite long after the Alto, computers used command line interface (CLI). Alto cost $18,000 to build so it was never marketed for personal use because of its grand price. Alto used a mouse and windows similarly as a modern GUI desktop computer. Xerox later launched a commercialised version of Alto named Xerox Star which was also too expensive compared to other PCs at the time, thus failing to have a significant impact on the markets (Ceruzzi 2003, O'Regan 2016, Deitel & Deitel et al. 2004.)

CP/M and DOS, Forerunners of PC Operating Systems

In the end of the 1980s, the Intel 8080 based PC operating system industry was led by **CP/M**, an OS created by Gary Kildall in 1974. CP/M was originally made as an operating system for the Intel 8080 PCs when Kildall was a consultant for Intel. In the beginning, Intel lost interest of the OS and it was used only by few customers (Cringely 1996). After, Kildall resigned from Intel and continued developing CP/M in his own company named Digital Research. The critical year for CP/M was 1977, when Kildall was confronted by IMSAI who wanted to license CP/M for its PCs. For this purpose, Kildall added a specialized code called BIOS (Basic Input/Output System) into the system. Depending on the hardware, the specialized BIOS code was only required to be customised and the else of the OS was common code. Consequently, the BIOS addition standardised the OS making CP/M the choice of operating system for Intel 8080 PCs (Ceruzzi 2003).

Through these developments, the computer industry was ripe for the emerge of the IBM PC ecosystem. Right before, Intel was offering its Intel 8080 for PC manufacturers and most of them were packed with the CP/M operating system. Many computers also offered

Microsoft's BASIC on a read-only-memory from which Microsoft collected a royalty fee. Software development was also starting to thrive and they were developed and distributed through CP/M operated eight-inch floppy disk drives. After Altair, PCs also started to come with standardised ports in order to plug various peripheral devices, including printers, keyboards and cassette tape recorders. (Ceruzzi 2003.)

The arrival of **DOS** was stimulated by the preparation of IBM PC. In the end of the 1970s, Microsoft was still mostly famous because of its BASIC implementation marketed in various PCs, including Altair, Commodore and many CP/M PCs. Then in 1980, IBM finally decided to enter the PC market with its IBM PCs. Traditionally, IBM was a bureaucratic company with long passage times for projects. Exception for this was the IBM PC project. The project was also exceptional, because the IBM PC was not to be equipped with hardware and software made only by IBM. First, the microprocessors for the IBM PCs were chosen to be Intel 8088s. As for software, IBM wanted Microsoft to offer their BASIC implementation to use for their machine powered with Intel 8088. (O'Regan 2016, Ceruzzi 2003, Cringely 1996.)

Microsoft was initially approached for the operating system of IBM PC, but in fact Microsoft did not yet have an operating system at the time. Because of this, Gates led IBM to Kirdall's Digital Research company and its CP/M operating system. However, Digital Research and IBM did not reach an agreement on the licensing. There are a few stories why Digital Research never succeeded to seal the deal, one telling that when IBM was visiting Digital Research to license the operating system for their PC, Kirdall's wife (who handled the company's administrative work) was the only one available, who refused to sign IBM's nondisclosure agreement. (O'Regan 2016, Ceruzzi 2003, Cringely 1996, Moumina 2001.)

In the end, Gates seized the opportunity and acquired their own operating system which they offered for IBM. This system named DOS (Disk Operating System) was launched on the IBM PC in 1981 under the name PC-DOS. Microsoft held the rights to market their DOS operating system elsewhere under the name MS-DOS. DOS was based on a system called 86-DOS or QDOS (Quick and Dirty Operating System) which was written by Tim Paterson of Seattle Computer Products on the Intel 8086 chip (Ceruzzi 2003). IBM wanted certain modifications for the original DOS, so Gates hired Paterson to make the changes to the system (Tanenbaum 2009). The first version of DOS was compatible with CP/M and they resembled each other (O'Regan 2016). MS-DOS became highly lucrative for Microsoft later on when IBM PC clones started to show up in the markets, due to DOS being the dominating operating system on the IBM PC (Moumina 2001).

The Beginning of Apple

Slightly before IBM PCs and after Altair 8800, **Apple** was established by Steve Jobs and Steve Wozniak in 1976[1]. Goal for Apple was to develop a user-friendly computer that would be an alternative for the mainframes and minicomputers of the era. Wozniak did the development of hardware and software, whereas Jobs oversaw marketing. Quickly after the formation, **Apple I** was released in 1976. Apple I was still a computer for computer hobbyists and engineers because it was only an assembled motherboard without

[1] Ronald Wayne was also one of the founders, although he sold his shares when Apple Computer Inc. was established in the beginning of 1977 (Luo 2013).

peripherals, monitor or a case. Inexpensive MOS Technology 6502 processor was used for the computer.

In 1977, **Apple II** was released which was significantly more elegant than its predecessor. Apple II came with a monitor, keyboard and a case and was one of the earliest computers to come preassembled. It also had a colour display. The aspects of Apple II's user-friendliness, flexibility, consumer-aimed attitude and relative performance made it a high-selling personal computer (Ceruzzi 2003; Cringely 1996). It achieved a considerable market share between 1977 and 1981, and continued to sell moderately until the end of the 1990s (Dediu 2017). Initially, Apple II only supported cassette tapes but later a 5 ¼-inch floppy drive was introduced (Ceruzzi 2003).

An important step in the success of Apple II was the introduction of the **VisiCalc** application in 1979. VisiCalc was initially released on the Apple II and it was the first spreadsheet application on any computer platform (Cringely 1996). VisiCalc was created by Daniel Bricklin and Robert Frankston, who formed the software company Software Arts in 1979. Ultimately, VisiCalc became so popular that it received the **killer application** status, which in other words meant that the computer itself was often sold because it had VisiCalc (Ceruzzi 2003). Thus, VisiCalc helped Apple II to become a profitable PC (O'Regan 2016).

IBM PC and Its Clones

The greatest market disruptor for PCs came with the **IBM PC** which was released in late 1981. Amazingly, IBM PC (and its clones) captured over 75 per cent of the PC market share by 1989 (Dediu 2017). Nobody could expect the massive popularity of IBM PCs but some key elements can be captured which made the IBM PC and its compatibles extremely popular: open architecture (and platform), cheapness and vast software collection. Arguably, IBM's status on the computer market also had an effect on the positive reception of the IBM PC. Ironically, in the end IBM lost its grasp on its own product to IBM PC cloners such as Compaq and Dell. By the 1990s, the old IBM PC was overcome by the PC cloners, whose processors were mainly made by Intel, and the clones ran Microsoft's operating system and other software.

IBM PC was initially planned as a proprietary PC with IBM's own hardware and software, as was the case for IBM products before with mainframes and minicomputers. This became a problem because the PC market was evolving rapidly and the IBM projects usually took several years to complete. Because of the pressures from the fast-growing PC market, IBM had to find a way to build their own PC fast. Thus, they assembled a small team which objective was to get a personal computer to the market as fast as possible, and without the hassle of IBM bureaucracy. Eventually, IBM PC was actualised in one year. (O'Regan 2016, Ceruzzi 2003, Tanenbaum 2006.)

Because of the timeframe, IBM had to make compromises so they decided to build the machine with open architecture. This meant, that the computer was assembled from commercial hardware from several manufacturers. As for the microprocessor, Intel was chosen as the supplier. Similar act was made with software and peripherals which meant that other manufactures could produce them for the IBM PC. IBM also published the IBM PC Technical Reference Manual which included complete circuit schematics, IBM ROM BIOS source code and other information on the machine. The use of open architecture later led to the rise of IBM PC clones. (Cringely 1996, O'Regan 2016, Tanenbaum 2006.)

When IBM PC was announced in August 1981, it was not a major advancement on the existing technology. IBM PC employed Intel 8088 (a successor of 8080), integrated BASIC, ASCII character encoding, three options of OS (Pascal-based system by University of California, PC-DOS by Microsoft and CP/M-86 by Digital Research) and similar peripherals used before. PC-DOS quickly became the dominating OS because the Pascal system was only bought by few customers and CP/M-86 was not available until 1982. (Ceruzzi 2003.)

Many important software concerning word processing, accounting and games were made available for the IBM PC. The killer application of Apple II, spreadsheet software VisiCalc also received its rival on the IBM PC. This software called **1-2-3** by Lotus Development was released in 1982, and the combination of both the system and the software quickly overtook Apple II on the PC market (Ceruzzi 2003). IBM PC collected a considerable application base in its first year, having four times more applications than Macintosh after its first year of release (Watt & McGeever 1985).

Since IBM PC had most of its hardware, software and peripherals made by other manufacturers, the only proprietary part of the whole system was the basic input-output system (**BIOS**) firmware. The idea of IBM's BIOS was same to Kildall's BIOS in CP/M, meaning that the BIOS linked the generic operating system for specific hardware. This gave an opportunity for to build an IBM compatible PC if they had their own BIOS. A company called Compaq was the first company to put it into practice, reverse-engineering the IBM BIOS and with legal tricks creating their own IBM PC compatible with "their" BIOS, offering it cheaper than the original IBM PC. Subsequently, Phoenix Technologies started selling BIOS chips separately which made the building of an IBM-compatible PC even easier. This began the wave of IBM PC compatibles which later took over the whole PC market. (Cringely 1996, Ceruzzi 2003, O'Regan 2016.)

IBM PC vs Macintosh vs Commodore vs Atari

In 1982, Time chose PC as their Person (this time Machine) of the Year. At the time, various PCs were competing on which would be the number one PC, including IBM PC and its compatibles, Apple II, Commodore 64, Atari 400/800 and TRS-80. Relatively fast the answer was clear; IBM PC and its compatibles had taken over 50 % of market share in 1987, over 75 % in 1990 and over 90 % by the turn of the millennium (Dediu 2017).

Commodore and Atari also produced highly popular PCs in the late 1970s and early 1980s. Most notably, the Atari 400 and 800 introduced in 1979 and Commodore 64 introduced in 1982. All of these machines incorporated the MOS 6502 chip and were popular for playing games and producing advanced graphics and sound (O'Regan 2016). Today, Commodore 64 retains its popularity in demoscene and retro culture (Swenson 2017).

Apple's answer to conquer IBM PC was the Apple **Macintosh** released in 1984. Key advantage for the Macintosh was its GUI which had not entered the consumer PC market yet at the time. Development of Macintosh had already begun in 1979, two years after the release of Apple II. Reportedly, inspiration for the GUI came from Xerox PARC GUI-enabled computers, where the aforementioned Xerox Alto was introduced six years before. (Ceruzzi 2003, O'Regan 2016.)

Before Macintosh, Apple released another GUI-enabled computer named Lisa in 1983 which failed in the market mainly because of its high price (around $10,000). Finally in 1984, the Macintosh was released with a flashy commercial. The Macintosh was more user-friendly with its GUI, thus being a far easier to use than IBM PC. However, Macintosh failed to capture the market, mainly because it was more expensive and it had fewer software than the IBM PC and its compatibles. (Ceruzzi 2003, O'Regan 2016.)

OS/2 vs Windows

In the 1980s, Microsoft's DOS was the leading operating system for the IBM PC compatibles which in turn reigned the PC industry later on. In 1985, Microsoft launched its **Windows** operating environment which was a graphical operating system shell made on top of the MS-DOS operating system. Windows was initially a response to Macintosh's GUI operating system. The first versions of Windows were not much of a success until the launch of Windows 3.0 in 1990.

Interestingly, at the same time as Microsoft was developing its own GUI-enabled operating system Windows, IBM announced their own operating system called **OS/2** in 1987. The operating system was designed by IBM yet it was programmed by Microsoft. In effect, Microsoft was developing both their rival OS and their own. (Cringely 1996.)

OS/2 1.0 was released in 1987. Microsoft continued to develop OS/2 jointly with IBM until their breakup in 1990. The breakup of Microsoft and IBM is a controversial story which will not be discussed deeply here. Consecutive versions of OS/2 continued to disappoint and sell poorly, whereas Windows thrived and led on as the successor of DOS. Windows and OS/2 fought until the release of Windows 95 which virtually blew OS/2 out of the market. Many reasons for the failure of OS/2 have been presented, including the already dominative status of DOS with its abundance of software (which were later inherited by Windows); and for this reason, OS/2 was made compatible with DOS which in turn did not encourage the development of OS/2 applications resulting in the lack of software made exclusively for OS/2. Another reason suggested has been the inability of IBM to market OS/2 properly, whereas Microsoft marketed Windows fiercely and had them bundled with many IBM PC compatibles.

Since the 1990s, Microsoft has continued to launch successive operating systems of Windows every couple of years. Latest Windows release is Windows 10, introduced in 2015. Since the 1990s, Windows has been the most used operating system on desktop PCs. Today, it is estimated that Windows is used in over 80 % of desktop PCs around the world. The trend seems to be going down, since in 2010 Windows was estimated to be used in over 90 % of PCs. Main competitor for Windows on desktop PCs is the **MacOS** (previously Mac OS X) which succeeded the classic Mac OS in 2001. MacOS has slightly over 10 % of market share. Linux has two to three per cent market share on desktop PCs (StatCounter Global Stats 2017a; Statista 2017a; NetMarketShare 2017).

Linux from Finland

In 1991, **Linux** was created as a hobby by Linus Torvalds, a student at the time at the University of Helsinki. Linux was initially developed as an improvement to the Minix Unix-like operating system which was an educational OS with public source code (Deitel & Deitel et al. 2004). Interested in creating an improved Minix, Torvalds coded his Linux version 0.01 and published it at a Minix newsgroup in September 1991, with all its source code public. First version of Linux had 9300 lines of C and 950 lines of assembler code

(Tanenbaum 2009). The name of Linux is derived from Torvalds first name and aforementioned OS UNIX.

After launch, Linux quickly began to generate a community around the operating system, gathering people around the world testing the operating system and investigating its source code. Computer enthusiasts began sending bug reports, feedback and suggestions to Torvalds, who then reviewed them and made proper improvements. In October 1991, version 0.02 was published with the help of community feedback and suggestions. (Deitel & Deitel et al. 2004.)

Over time, Linux and its community continued to grow ever more and it began to phase out from its initial hobby stage. Linux gradually grew into a full UNIX clone with modern features. Linux originally ran on the Intel 80386 chip but it was promptly ported to other platforms as well. Linux version 1.0, the first viable alternative for UNIX, was released in 1994 which had 165,000 lines of code. Furthermore, the new version attracted even more people into the Linux development community. (Deitel & Deitel et al. 2004; Tanenbaum 2009.)

By 1996, version 2.0 grew Linux kernel to over 400,000 lines of code (Deitel & Deitel et al. 2004). The developer community of Linux was also growing and there were already thousands of developers debugging and modifying the operating system. Quickly after in 1998, many major companies such as IBM, Compaq and Oracle announced their support for Linux. Afterwards in the turn of the millennium, Linux continued to grow through its active development community with several 2.x versions.

In the beginning, Linux suffered from its complicatedness which made it difficult for normal users to install and utilise it on their computers. To ease the problem, different organisations and companies started to create Linux distributions (distro in short), which included Linux kernel, system applications, user applications and tools to ease the installation process (Deitel & Deitel et al. 2004). Today, there are over 300 Linux distributions for the PC systems (DistroWatch 2017).

In the present day, Linux is used in various computer systems, including PCs, supercomputers, server machines, embedded devices and mobile devices. Most notably, the most popular mobile operating system Android uses Linux kernel. Linux also dominates the market share of server and supercomputer OS.

Special for Linux is that it is open software (Tanenbaum 2009). Most of the market dominating operating systems in computing devices, including Microsoft Windows and Apple macOS are proprietary OS. Linux is not owned by any company and it is developed by a volunteering community. In formal sense, Torvalds does not have any authority over Linux, although he still has the influence and leadership to keep Linux intact. All in all, Linux has been a cornerstone of open source movement which has gotten more and more steam in the present day.

Naturally Linux has received criticism. Especially Bill Gates has stated in the past that free or open software does not properly reward developers for their hard work (Ceruzzi 2003). Same can be asked about Linux; if it is free who pays for its development? Although Linux itself is free, it is utilised by many businesses who benefit from it greatly. In fact, Linux is supported by various companies financially but also through development. There are several large companies such as Intel, Red Hat, IBM, Oracle and Google developing Linux (Brodkin 2015).

2.5 Surfing the Web

Before the Boom

Internet and the Web have a crucial role in today's technology and society. Arguably, most people have PCs and smartphones today mainly for web use. Especially the smartphone and social media era since the late 2000s has brought internet even more into our lives. Today, almost half of the world uses the internet, there are over 1 billion websites, and over 2 billion Google searches are made and YouTube videos are watched daily (Internet Live Stats 2017).

Before diving to the massive expansion of the Net in the 1990s, it is essential to briefly review the early history of internet, Ethernet, ARPANET and World Wide Web. First developments concerning the internet started in the US in the Cold War period. In 1962, ARPA determined to interconnect main computers in the Department of Defense through a global, dispersed network (Cohen-Almagor 2011). Later, the same idea evolved into connecting ARPA research investigators with few large mainframes across the country, which enabled cost-effective sharing of hardware and software. In 1965, **ARPANET** project was officially initialised by an initial funding of $1 million.

ARPANET was launched by Bolt Beranek and Newman in 1969 (Cohen-Almagor 2011). The network was created with Interface Message Processors (IMP) which were minicomputers handling all switching and communications functions between mainframes in the network. First IMPs were delivered to UCLA, SRI, UCSB and University of Utah. For now, the ARPANETs network had been deployed but communicating properly between the computers was still unsolved and required appropriate protocols (Kleinrock 2010). Quickly in September 1969, the first host-to-host message was sent on October 1969 using the Network Control Program (NCP) protocol. NCP was the first protocol stack to run on the ARPANET and was later replaced by TCP/IP in the first half of the 1980s. TCP/IP is still used as a fundamental protocol in the contemporary internet.

In the first years of the internet, another important part of the modern internet ecosystem, **Ethernet** was invented at Xerox-PARC in 1973 by Robert Metcalfe and David Boggs. Whereas internet connects computers around the nation and globe, Ethernet was meant for local networking, connecting single-user/personal computers, printers, mass storage et cetera to each other. In the early implementation of Ethernet, different computers were connected through one cable or bus. Computers in the Ethernet were listening to the cable, in other words computer sent its data while the cable was free and if the cable was occupied, computer backed off and tried to resend the data later after a random interval. Ethernet had its initial impact on workstations, and later it expanded to the PC market. (Ceruzzi 2003.)

Local area networks and their connection to the internet were first available in office environments and later they expanded to home computers. Similarly, how workstations replaced mainframes and their terminals in the 1970s, workstations were replaced by networked PCs in the office in the 1980s. Then in the following years, locally networked PCs were connected to other locally networked PCs, creating a network of networks, commonly called the internet. (Ceruzzi 2003.)

A major part of the internet history is the development of **World Wide Web** (WWW) started in 1989 by Tim Berners-Lee. Initial purpose of WWW was to allow communication between computers and software of various types, and to have an information space on the internet where people and machines could share information with one another (Ceruzzi 2003). For these purposes, Berners-Lee created Universal Resource Identifier (URI) which acts as a reference to a resource on the network; Hypertext Transfer Protocol (HTTP) which enables hyperlink-included data exchange in the Web; and Hypertext Markup Language (HTML) which provides a simple way for creating web page documents. With the definition of URI, HTTP and HTML, Berners-Lee wrote the first WWW client (browser) and most of the communications software (Cohen-Almagor 2011).

Components for the modern and commercialised internet were now laid out; ARPANET, TCP/IP (among other internet protocols), Ethernet and WWW served as the backbone for modern internet. ARPANET supplied the basis and initial infrastructure, TCP/IP enabled understandable communication between parties, Ethernet provided numerous LAN networks to be interconnected and WWW helped the navigation and viewing of internet resources.

Internet Becomes a Commodity

Internet witnessed a massive expansion in the 1990s. ARPANET project was officially terminated in 1990 and the control of the public internet was given to the National Science Foundation. In 1991, the global Internet Society was formed. In these years, internet started to expand immensely. Cohen-Almagor (2011) argues that the reasons for the remarkable growth were internet's accessibility, flexibility, technological creativity, multi-application and decentralised nature. People were also curious and wanted to be part of the fast-expanding scene now available for ever more people. From 1992 to 1997, number of hosts in the internet grew from 1 million to 19.5 million and the number of websites from 50 to 500 000 (Gromov 2002).

After the WWW was introduced, there was an imminent need for an easy-to-use and visually appealing browser. The first commercially successful browser Netscape had its roots in a browser named Mosaic which began development in 1992 amongst Marc Andreessen and Eric Bina in joint with the University of Illinois. Later in 1994, Jim Clark persuaded Andreessen to commercialise the browser yet Illinois University objected this. Consequently, Andreessen and Clark founded their own company which later became Netscape Communications Corporations. They released the first version of **Netscape** in 1994. Mosaic continued development under Illinois University, nevertheless Netscape quickly overthrew its position as the most popular web browser. (Ceruzzi 2003.)

In 1995, the American internet was privatised. Before, commercialisation of internet became a major debate which was discussed in a series of conferences by National Science Foundation. Consequently, Merit Network, IBM and MCI Communication received a contract to manage and modernise the internet backbone, and three additional contracts were given to Network Solutions, AT&T and General Atomics. Finally in 1995, National Science Foundation was shut down completely and the privatisation was fulfilled. (Cohen-Almagor 2011.)

In 1995, Microsoft was thriving in the software market with its operating system Windows and Office bundle[2]. The ecosystem of Windows and Office equipped in cheap Intel-equipped IBM PCs was an important factor for Microsoft in competing with IBM and Apple at the time (Ceruzzi 2003). Even though Apple had the same package of hardware and software with more elegant design, Microsoft had cheaper software inside cheaper and more widely-used hardware which quickly turned the market in favour for Microsoft. By 1995, IBM PC compatibles with Windows (and Office) and Intel was already the definite PC standard. Expectedly, this persuaded software developers to create their applications for the most widely-used and accepted system, regardless of it was better or not.

Microsoft also entered the internet market in 1995. A famous memo was sent about it by Bill Gates in May 26th, 1995 called the *The Internet Tidal Wave*. In this 9-page memo, Gates (1995) proclaims that Microsoft will focus going online and "assign the Internet the highest level of importance". Gates continues that the internet is "the most important single development to come along since the IBM PC was introduced in 1981." Indeed, Microsoft did enter the internet boom with their own browser **Internet Explorer** which was initiated five months before the Gates memo. Microsoft paid a web browser called Spyglass for a license to use it for Microsoft's own web browser. Internet Explorer 1.0 was introduced in 1995 as an accessory (Ceruzzi 2003).

Internet Explorer became an arduous competitor for Netscape. Often referred to as the beginning of the end for Netscape was the introduction of IE 4.0 in the autumn of 1997 (Ceruzzi 2003). Because IE 4.0 was integrated with Windows, it quickly started gaining popularity, thus Netscape began to lose its usage share. Eventually, IE became more popular in the turn of the millennium and Netscape gradually perished, eventually discontinuing in 2008 (Cooper 2014).

Concerning the browser wars of IE and Netscape, there was a major antitrust law case between US Department of Justice and Microsoft between 1998 and 2001. DOJ accused Microsoft of abusing its monopoly-like position in operating systems to manipulate browser sales. More closely, the bundling of Windows and IE allegedly restricted the market of web browsers, *cutting off the air supply* of Netscape particularly (Ceruzzi 2003). In 2000, it was ruled that Windows had been unlawfully tied with IE and a few months later it was required that Microsoft would be needed to split up. Later in 2001, the judge was removed from the case and the splitting of Microsoft was avoided. Ultimately, it was ruled that the parties agreed to settle without breaking up the company, requiring Microsoft to abide by a consent decree for five years. The consent decree forbid Microsoft from creating exclusive deals with PC manufacturers, in other words obliging them to work only with Microsoft software developers. Also, Microsoft was required to open access to its application programming interfaces which are necessary in order to make applications work under Windows. Decree was extended twice, and it expired in 2011 (Chan 2011).

Addition to the DOJ case, the European Commission of the European Union also had a similar case against Microsoft for the abuse of its dominant position. Preliminary verdict for the case was given in 2003 where Microsoft was ordered to offer a Windows version without Windows Media Player and to reveal low-end servers' technical details for Microsoft competitors, in order for them to achieve full interoperability with Windows

[2] Office bundle includes several Microsoft applications, such as word-processing software Word, spreadsheet application Excel, presentation software PowerPoint and database management system Access.

(Fried 2003). Later in 2004, the final verdict was given, which included the aforementioned remedies and a €497 million fine (European Commission 2004). During 2005 and 2006 EU fined Microsoft an additional 281€ million because they did not comply with the remedies in a timely manner as was requested. Also in 2008, EU fined Microsoft with an extra €899 million because Microsoft failed to comply in providing sufficient information for other companies to interoperate with Windows. Later in 2012, the General Court reduced the fine to €860 million (Kanter 2012). In total, Microsoft was fined a total of €1.7 billion.

Roughly from 1997 to 2002, the **dot-com bubble** period occurred. During this period, the internet market expanded immensely after its commercialisation in the middle of the 1990s. In order to profit from the swiftly growing internet userbase, numerous internet businesses were established and many of them were taken public. Ultimately, the internet companies acquired excessive speculation for their pricing until the bubble burst after the beginning of the millennium. Many major IT companies, including Microsoft, Apple, Intel, Amazon and eBay suffered from the crash. Nasdaq index was record high in March 2000 at 5048.62, after it dropped to 1114.11 by October 2002, losing almost 80 % of its index value (EconStats 2017).

After the dot-com bubble burst, a new concept of **Web 2.0** emerged. O'Reilly (2007) summarises Web 2.0 to seven core competencies:

- Services, not packaged software, with cost-effective scalability
- Control over unique, hard-to-recreate data sources that get richer as more people use them
- Trusting users as co-developers
- Harnessing collective intelligence
- Leveraging the long tail through customer self-service
- Software above the level of a single device
- Lightweight user interfaces, development models and business models.

Essentially, Web 2.0 does not depict the technical advancement of the Web, rather it lays out the canvas of how it should be used and designed so that it would deliver rich user experiences.

Continuing with web browsers, IE received another competitor from the ashes of Netscape, namely **Mozilla Firefox**. The legacy of Mozilla begins in 1998, when Netscape was released as free software or in modern terms as open source (Krishnamurthy 2009). The Mozilla Organization was established then to create an internet suite, supporting the Netscape web browser ecosystem. Later in 2003, Netscape's parent company AOL started to back off from Netscape, after which the Mozilla Foundation was established in July 2003 in order to assure the continuity of Mozilla without Netscape (Goodger 2006). In 2003, Mozilla began to focus on creating a standalone browser Firefox and the e-mail client Thunderbird instead of Mozilla Suite. Firefox was initially named Phoenix and Firebird but the name was changed promptly to Firefox. Firefox version 1.0 was released on November 9, 2004.

From 2004 to 2010, Firefox grew steadily competing with IE. Firefox reached its peak in web browser usage in 2010, having approximately 30 per cent of the usage share, compared to IE having 50 to 60 per cent of the usage share (StatCounter Global Stats 2017b). In 2008, first version of **Google Chrome** was released which quickly gained steam in the web browser market. In the middle of 2012, Chrome became the most used web browser overcoming IE. Currently, Chrome has 55-57 %, Firefox 5-9 % and IE 3-11 % usage share of web browsers on any platform (StatCounter Global Stats 2017b; W3Counter 2017). Evidently, IE has gradually lost its leader position in the web browser market since 2004. Reasons for the continuing fall of IE included its outdatedness and lack of features. As for why Chrome became the most successful browser, possible reasons can be Google's highly positive image, rich ecosystem and up-to-date features (Jones 2015). Similarly to Firefox, Most of Chrome's source code is open and it is released in the Chromium open-source project.

West & Mace (2009) has depicted the development of **mobile internet** from the perspective of iPhone in detail. According to them, iPhone was the first mobile device to deliver the real internet for consumers instead of building a separate and independent mobile internet, aiding the initial success of iPhone. Such mobile internet has subsequently become the norm in mobile devices, giving momentum to the overall expansion of the internet. In 2016, mobile web browsing traffic surmounted desktop browsing for the first time (Gibbs 2016). Mobile internet has also amplified the principles of Web 2.0 demanding more serviceability and value creation for example.

Social media is an important driver of the contemporary internet movement. Since the beginning of internet, it has been a tool for communication and socialisation; at first by e-mail, IRC and BBS, and later by contemporary social media including Facebook, Twitter, Instagram, Snapchat, Reddit, WhatsApp and YouTube. Due to its high popularity within consumers, social media has amassed major markets with several billion dollar companies.

2.6 Mobile Devices

History of mobile computing devices starts from the early days of telecommunications, sending information over a distance to a receiver. Earliest telecommunication has been archaic fire, smoke and drum signals used before contemporary civilisations. The first postal system was created in the Persian Empire in the sixth century BCE, later which Egyptian and Roman civilisations created similar systems. First semaphore system was devised by the Greeks in the fourth century BCE. (O'Regan 2016.)

Telegram and later telephone were invented during and slightly after the Industrial Revolution. First telegraph, initially optical system was built in France by the Chappe brothers in the late 18^{th} century. It was used to send information from one high tower to another and it reminded of a ship semaphore system. Later in the early 19^{th} century, Samuel Morse constructed the Morse code which allowed letters to be represented in on-off tones. This built the foundation for the commercial transatlantic electrical telegraphs which sent Morse code through electrical lines. (O'Regan 2016.)

Telephone was invented by Alexander Graham Bell in 1876. First telephones were hardwired, connecting only two telephones, for example from home to the office. First commercial telephone exchangers, i.e. switching callers were introduced in the late 1870s. First transcontinental call was made by Bell in 1915. Phone brought a paradigm shift in communication, providing direct communication between two people instantaneously. Telephones were first used by business and affluent people of the society, but it quickly expanded to all social classes in the Western world. (O'Regan 2016.)

Marconi introduced a wireless transmission system of sounds in 1896 which began to enable swift communication between ships and coastal radio stations. Later, the technology evolved into radiocommunication. Later, image transmission through radio signals were developed, enabling long-distance wireless transmission for electronic television. (O'Regan 2016.)

First **mobile phones** were introduced for automobiles in 1946. Some commercialised automobile mobile phone systems existed, including AT&T's Mobile Telephone Service introduced in 1946 and its successor Improved Mobile Telephone Service after 1964 (Farley 2005). Finland also had its own automobile mobile phone system called Autoradiopuhelin (ARP), launched in 1971 and expanded to the whole country by 1978. Similarly to other mobile systems of the day, ARP was only operational in its own domain, Finland (Heikkilä et al. 1999).

Later in 1973, first handheld **cellular** mobile phone call was made by Martin Cooper of Motorola and Joe Engels at Bell Labs. This was based on a cellular system which consists of a network of small geographical areas called cells. Each cell has one base station transmitter and central switch controlling cell traffic. Later, commercial cellular deployment expanded rapidly in the late 1970s and continued into the early 1980s, bringing cellular networks for handheld mobile phone usage. First cellular service in Europe was introduced in 1981, when the Nordic Mobile Telephone System began operations in Finland, Sweden, Denmark and Norway, also being the first multinational cellular system. (Farley 2005.)

Systems mentioned in the previous paragraph were using **analogue transmission**, but the need for **digital transmission** emerged in the 1980s. Analogue cell phone period is often referred to as first generation or 1G era. For transmitting digital radio signals, Global System for Mobile Communications (GSM) standard was developed in Europe in a new radio band. The first commercial GSM network was launched in Finland by Radiolinja consortium in July 1991. Nokia supplied the phone equipment for GSM applications. GSM system included short messaging services (SMS) and subscriber identity module (SIM) cards. GSM also incorporated increased level of security, encrypting communication between the subscriber and base station. Afterwards, GSM became the global standard for digital cellular mobile phone radio service. GSM had several rivals, including CDMA, D-AMPS, TDMA and PDC. (Farley 2005; Heikkilä et al. 1999; O'Regan 2016.)

Digital cellular era, often called as the second generation or 2G era was widely dominated by GSM standard and its applications. Later, GSM networks evolved into GPRS (2.5G) which became available in 2000 (O'Regan 2016). Later, GPRS further evolved into broadband multimedia communication standards of 3G, 4G and the upcoming 5G. At present, G-terms are often used in spoken language for the speed varieties of mobile internet, for example 4G being faster than 3G. O'Regan (2016) expresses this evolvement of mobile technology, how it has transformed the earlier paradigm of *communication between places* to *communication between people*. Today, transferring large amounts of data is one of the main goals of telecommunications.

As for mobile phone development, a fundamental change happened in the middle of the 1990s. First Nokia Communicator was introduced in 1996, providing a typical GSM mobile phone with extended computing abilities. The device had a QWERTY keyboard and several applications, including word processing, calendar, e-mail and limited-access internet (Farley 2005). Nokia Communicator is often thought as the first influential personal digital assistant (PDA). PDAs were the predecessors of modern smartphones which extended the capabilities of normal mobile phones.

In the middle of the 1990s, mobile phones were still optimised only to transfer speech, not data. This quickly changed and data transmission became the top priority in tele industry (O'Regan 2016). Especially after the emerge of commercial Web, solutions for mobile internet were sought for. First, mobile internet was mainly built based on new mobile-specific solutions especially in the western countries during 1997-2007. Such technologies as Wireless Application Protocol (WAP) were used to create a separate mobile internet with the limitations of mobile data transmission of the time. However, Japan developed mobile internet services that exploited the existing wired internet but with limited capabilities, tackling the problems of limited data transmission. Mobile internet was later revolutionised by iPhone and its killer app; the web browser which could access the same internet as was already available on PCs (West & Mace 2010).

With the fast evolution of data speed and mobile devices, the era of smartphones was dawning. Smartphones arose from the existing mobile device and PDA technology. Smartphones became more than just calling devices, essentially being touch-based computers on a phone with operating system, internet access and third-party software (O'Regan 2016). Features that smartphones offer include camera, maps, games, word and spreadsheet tools, social media, video streaming and more.

One of the forerunner devices including PDA and mobile phone features was the IBM Simon introduced in 1993. Simon offered fax, e-mail, address book, calendar and calculator applications. In the same year, Apple Newton was introduced as the first PDA. Xerox PARC had also created a prototype PDA in the 1970s called Dynabook. Other smartphone predecessors included the aforesaid Nokia Communicator, Qualcomm pdQ and Blackberry. (O'Regan 2016.)

2.7 Mobile Operating Systems

First common mobile device operating systems started to emerge in the early 2000s. Before, mobile devices had simple hard-coded operating systems that were only usable on a specific device. Finally, hardware manufacturers Nokia, Ericsson, Panasonic and Samsung agreed to collaborate on one operating system running on their devices, namely **Symbian**. Symbian's roots were in EPOC32 operating system, developed by British company Psion since the end of the 1980s. The first Symbian device, Nokia 7650 smartphone was launched in 2002, four years after Symbian started its development. (Tilson et al. 2011; Hall & Anderson 2009.)

Symbian continued development through the 2000s, while Nokia produced several smartphones with Symbian OS, most notably Nokia E61, N95 and 5800. With the support of hardware manufacturers, Symbian achieved a dominating position in the mobile OS industry, covering over 50 % of the market share until 2008 (West & Mace 2010). Despite the popularity, Symbian was suffering from serious deficiencies, mainly in application development. Different Symbian devices had their own unique UIs tailored by the manufacturer which fragmented the UI compatibility. Furthermore, Symbian was difficult for developing applications, using a difficult-to-program non-standard derivative of C++ (Ocock 2010).

In 2007, a notorious competitor emerged in the smartphone industry: **iPhone**. iPhone had distinct features compared to other mobile and smartphones of the era, including a large touchscreen for video viewing and a universal web browser for browsing the entirety of the Web in the same way as on a desktop device. Also, iPhone included music streaming similarly to an iPod and it did not have a user-changeable battery or memory card. iPhone also required a purchased mobile data service plan, yet iPhone did not have 3G support which was already supported by many Japanese and European smartphones. (West & Mace 2010; West & Mace 2007.)

The first iPhone was highly praised for its user friendliness and product design, collecting a desirable number of users. Criticism was also presented for the high price ($600), slow data speed without 3G network, operator lock and the prevention of third-party applications. Hacker community bypassed these problems, creating a grey market for unlocked iPhones. (West & Mace 2010.)

Although the first iPhone did not allow third-party software, Apple encouraged web-based application development for the iPhone browser. There was a critical problem however, because the mobile web browser did not support Flash and could not run many standard web applications. Later in 2008, Apple started allowing third-party software. (West & Mace 2010.)

Situation in the smartphone application sector started to change in 2008, when Apple allowed third-party software on iPhone and launched the iPhone App Store right before iPhone 3G. App Store initially had 500 applications, which were built with the software development kit released four months prior to release. Quickly after, app store began gathering steam swiftly. After first six months, App Store had more than 15,000 applications and 500 million downloads, and three months after that the numbers had doubled to 30,000 applications and 1 billion downloads. (West & Mace 2010.)

iPhone's App Store quickly started to gather rivalry. Nokia stepped up in 2008, when it purchased the Symbian company to itself. In response, Symbian UIs were unified and operating system was turned into open source software. In 2009, Nokia launched its own application marketplace called Ovi Store. Software development was also improved in 2010, when the old SDKs were replaced by standard C++ using Qt. Nevertheless, Symbian started losing its market share. Tilson et al. (2011) argues, that Symbian failed because Nokia did not pursue an app centric ecosystem, resulting in the lack, difficultness and low quality of applications. Obviously, strategic actions and technological development influenced the outcome of Symbian, having to deal with its legacy in architecture and accumulated complexity. In 2011, CEO of Nokia declared that the company was battling with "burning platform" and had to make drastic moves. A few days later, Nokia announced liaison with Microsoft, adopting Windows Phone 7 as Nokia's main smartphone operating system.

Since 2008, another major competitor in the OS market emerged, namely **Android**. Android was a mobile OS project started in 2003 by Andy Rubin and other associates. Later in 2005, Android was acquired by Google. Finally in 2008, first Android version was released. Android initially had problems finding device manufacturers to deploy Android, but later Motorola and Samsung started using Android as their operating system (Tilson t al. 2012). Today, Android devices are manufactured by Samsung, ZTE, Huawei, ASUS, Acer, Motorola, Dell and many more.

Android is based on a modified Linux kernel. Essentially it is the same as a Linux kernel, but several drivers and libraries have been modified or added to allow Android run more efficiently and effectively on mobile computing devices. Android also utilises an Android specific application framework, therefore, Android is a complete solution stack, including the OS, middleware components and applications (Heger 2012). Linux kernel acts as the hardware abstraction layer in Android. Android's source code is open under an open source license, although most Android devices ship with open and proprietary software.

Quickly, Android started to seize the mobile OS market. In 2009, Symbian still had over half of the market share, but already in the early 2011, Android had overthrown Symbian with 36 % of the market share. In 2013, Android already had 80 % of the market share, dominating the markets. Respectively, iOS has kept a 10-20 % market share since 2009. (Statista 2017b.)

Other mobile OS have also been introduced, most notably Windows Phone and BlackBerry OS. Yet, their market share has continued to decrease, whereas Android and iOS have captured the users with their wide platform ecosystems. That is to say, Apple and Google's smartphone app platforms captured the eyes of developers, attaining the active, sizeable and diverse application marketplaces they incorporate today. Evidently, a key success factor for both have been their ability to mobilise software developers with serviceable SDKs and APIs (Sorensen et al. 2015). Yoo (2012) argues, that companies like Google, Apple and Facebook have created platforms which can be used to build products and services that grow beyond their realm of business.

2.8 Enterprise Resource Planning

Enterprise resource planning (ERP) means the management of functional areas in a business. Functional areas include product planning, production planning, manufacturing, marketing and sales, materials management, inventory management, retail, shipping and payment and finance. This thesis concentrates in the software applications available for enterprise resource planning, which ease the management of company resources.

ERP systems or enterprise systems (ES) are software systems for managing business. Management can tackle areas of business such as planning, sales, marketing, distribution, accounting, human resource management, project management, e-business, transportation et cetera. These different modules ought to be able to share information between one another, allowing horizontal integration of processes inside an enterprise. In a sense, enterprise resource planning systems are operating systems for enterprise management. Several definitions of ERP have been collected in Rashid et al. (2002) paper.

The term ERP was initially suggested in the early 1990s by the Gartner group. Enterprise systems did exist before ERP but the new term added a notion that the software should be integrated through different functional domains of an enterprise. The roots of ERP stem from 1960s and the first material requirement planning (MRP) systems. In the 1960s, lowering costs in product-based manufacturing was the key success factor in market competition. In the late 1960s, an MRP software was created for the planning and scheduling of materials for complex product manufacture. The first solutions of MRP in the 1960s and 1970s were large, unwieldy and expensive, similarly to computers of the time. MRP software ran on a mainframe computer, requiring significant amount of human labour. Related to early enterprise systems was also IBM's COPICS (Communications Oriented Production Information and Control System) software which enabled integrated computer-based manufacturing. Also in the 1970s, software companies which later conquered the ERP market were already established, namely SAP, Lawson and Oracle. (Jacobs & Weston 2007.)

In the 1970s, enterprise software was still feebly integrated. When a need for a new IT application appeared, it was built as a separate discrete system. If it had common ground with other formerly implemented enterprise systems, it was possibly interfaced with them. As such, combining data from two IT systems was troublesome and error prone. Analysing data from the early enterprise systems was also rudimentary, enabling only summary level analysis. Creating an advanced analysis required specific ad hoc programming and laborious effort. Likewise, separate systems included similar same data elements, resulting in incoherent data and irregular overlaps. As such, “one-company, one-system” was not achieved. (Markus & Tanis 2000.)

Later in 1980s, MRP II was introduced. The MRP term was also rephrased to *manufacturing resource planning* and with the roman two in the end, it identified the improved features of the newer systems. Similarly, competitive factors in the markets shifted from lowering cost to quality improvement. MRP II systems included novel features such as enhanced shop floor reporting and detailed cost reporting features. COPICS system was also updated at the time, setting off a new term of CIM (Computer Integrated Manufacturing). CIM framework supported three levels of enterprise functions, including functions such as marketing, engineering and research, production planning, business management, administrative support, application development support, database and communications and presentation tools. Integration of several enterprise functions was already occurring in the enterprise software market, giving a lead to the soon emerging ERP. (Jacobs & Weston 2007.)

As for the interoperability of enterprise systems, 1980s and 1990s brought improved solutions. Several major ES companies, including SAP, PeopleSoft and Baan, were building integrated software packages which shared a common database, enabling the passage of a single transaction between enterprise functions. These software suites acquired the name ERP, coined in the 1990s. In the 1990s, ERP software packages started replacing old, often in-house developed legacy enterprise software. Especially the Y2K problem boosted large companies to start implementing ERP. Accordingly, ERP systems began integrating all domains of business functions. (Markus & Tanis 2000; Rashid et al. 2002; Jacobs & Weston 2007.)

In 2000, ERP II term was coined. The improved ERP II was to answer for the needs of enterprises transforming from vertically integrated into agile and core-competency focused organisations. New-age organisations are increasingly engaging in B2B and B2C electronic commerce but also in collaborative commerce. ERP II addresses these transforming trends, providing “a business strategy and a set of industry-domain-specific applications that build customer and shareholder value by enabling and optimizing enterprise and inter-enterprise, collaborative operational and financial processes”. (Bond et al. 2000.)

By 1999, IBM’s dominance in the enterprise software market had been taken by J.D. Edwards, Oracle, PeopleSoft, Baan and SAP. Later in 2002, Baan dropped out of the competition. In 2003, PeopleSoft and J.D. Edwards merged, and later in 2005 they were bought by Oracle (Jacobs & Weston 2007). After 2005, two major competitors Oracle and SAP have remained in the ERP market, although recently they have faced new competitors. In 2013, SAP had 24 %, Oracle 12 %, Sage 6 %, Infor 6 %, Microsoft 5 % and other vendors including IBM had 57 % of the market share (Columbus 2014). In 2017, SAP had 19 %, Microsoft 16 %, Oracle 13 %, Infor 13 % and other vendors 39 % of the market share (Panorama Consulting 2017). Interestingly, Microsoft's ERP software Dynamics has overtaken Oracle in the ERP market. Seemingly, ERP market does not have a single dominating vendor currently, although SAP is still slightly in the lead.

ERP implementation has changed substantially in the 2000s. In the beginning of the millennium, ERP systems replaced the old unintegrated legacy systems with a single backbone system. In the middle of the 2000s, enterprises implemented one vendor ERP suites, in order to achieve higher levels of integration, improve customer care and improve supply chain efficiency. Also in the middle of the 2000s, ERP evolved into extended ERP which included add-ons such as planning and scheduling (APS), customer relationship management (CRM) and supply chain management (SCM). Recently, alternative approaches for ERP deployment have been emerging, including cloud-based solutions, subscription-based pricing and Software as a Service. Expansion of mobile computing has also expanded ERP applications to smartphones. Nevertheless, vision of ERP has remained the same. (Shaul & Tauber 2013; Rashid et al. 2002.)

ERP systems have been infamous for their long implementation times and failure stories. Implementation problems can result in additional costs, utilisation hindrances and maintenance problems. According to Chang (2004), 90 % of ERP implementations are late or over budget, 67 % of enterprise initiatives fail in achieving corporate goals and are considered negative or unsuccessful and more than 40 % of all large-scale projects fail. Factors why ERP implementations fail include unrealistic expectations, over-customisation of software, timeline flexibility, corporate culture and inherent complexity of ERP implementation (Barton 2001). Umble et al. (2003) mentions that the biggest reason for ERP implementation failure is poor planning or management. Shaul & Tauber (2013) argue that the planning phase of ERP implementation is often disregarded, albeit important decisions are made in this phase, concerning business cases, user requirements, usage scenarios, operational requirements and system requirements. Also, the selection process of an ERP system should be made comprehensively in order to find the best-fitting system. Likewise, ERP implementation projects should understand one's complexity, size and scope, enterprise leaders should be fully committed to the project, extensive training should be afforded and ERP users should be involved in the implementation to ensure that the system works for their needs (Shaul & Tauber 2013).

ERP implementation projects cost considerably, ranging in one or two million. Implementation of an ERP system takes over a year, often nearly two years or even more. Benefits that ERPs fulfil include availability of information, improved productivity and efficiency, integration of business operations, reduced direct operating or labour costs and improved lead time. (Panorama Consulting 2017.)

Companies invest in ERP because they hope it will increase efficiency and productivity, and will help in answering to customers' needs (Conteh & Akhtar 2015). Critical success factors of ERP have been researched broadly (Shaul & Tauber 2013). Two major benefits of ERPs are a unified view of the enterprise comprising of all business functions and departments, and a common database of the company where all business activities are entered, recorded, processed, monitored and reported. In order to implement ERP successfully, enterprises require clear understanding of strategic goals, commitment by top management, excellent project management, organisational change management, a great implementation team, data accuracy, extensive education and training and focused performance measures (Umble et al. 2003).

Interoperability is also an issue often addressed in enterprise management. Enterprise information systems, including enterprise resource planning are key features for enterprise interoperability (Panetto & Cecil 2013). Indeed, in the new environment of IT enterprises ought to collaborate (Panetto et al. 2016). Future of enterprise systems also require information integration and interoperability, allowing business of all sizes share data with suppliers, distributors and customers. Furthermore, further enterprise application integration within the company and across companies would enable a seamless exchange of information within a whole supply chain of several companies (Xu 2011). As for technology, service-oriented architecture helps with the coordination of heterogeneous information systems, further improving interoperability by transitioning enterprises systems from silo-based to service-oriented. (Serrano et al. 2014).

3 Overview of Building Automation Systems

Building automation (BA) has existed in one form or another since the early 20th century. Traditionally, BA has been used to control heating, ventilation and air conditioning (HVAC) of a building. Building automation controls were based on pneumatics which were later replaced by electric and analogue electronic circuits and finally by microprocessors (Kastner et al. 2005). In the recent days, building automation systems has been associated more and more with smart buildings, connecting and automating all the intelligence inside a building, including HVAC systems, central control and monitoring systems, lighting and shading, life safety, IoT devices and more. Originally, BA has been the purpose of optimising indoor conditions and comforting building management but today, building automation is giving tools for connected and controllable smart buildings.

This chapter gives an overview of different building information systems, including building automation systems, building operating system prototypes and IoT systems, reviewing their purpose, architecture, market structure and use cases. Chapter focuses on the management level of building automation, exploring different building management systems including building operating systems.

In the context of this thesis, the chapter lays out current status of building ICT systems and also depicts their anticipated future. Altogether, overview and trends of building ICT systems are examined in order to apprehend what kind of development a building operating system would require in the status quo.

3.1 Structure and Purpose of Building Automation Systems

A building automation system controls and monitors mechanical and electrical equipment inside a building, including HVAC, security, fire, power, water supply and elevator systems. As the name states, BAS automates building equipment and device management, coordinating various electrical and mechanical devices interconnected through a control network (Domingues et al. 2016). BAS can be utilised in various kinds of buildings, including commercial, factory, warehouse, office and residential buildings.

The typical promise of building automation is increased user comfort at a reduced operation cost (Kastner et al. 2005). BAS uses optimised control routines to have the perfect conditions for HVAC, lighting, shading and so on. Thus, energy efficiency and management has had a large role in BAS ensuring that the building is environmentally friendly. Installing BAS results in a higher construction cost, but considering its savings in the whole life cycle of the building it can bring significant savings in energy and water consumption.

In order to make building conditions supportive for human comfort or industry requirements, building services are required. Building services include climate control, visual comfort, safety, security and transportation. Building services can be perceived as passive technical infrastructure such as water management and active controllable systems such as HVAC (Kastner et al. 2005). Evidently, different buildings have different requirements for certain building services. For example, residential and office buildings need adequate climate control whereas storages might not need any climate control at all. Also, more complex buildings usually require more sophisticated building automation to enable sufficient and functional building services.

A good example of a BAS is the automated HVAC. Heating and cooling can be managed with convectors running hot or cold water. If a convection is fan-assisted, it is called a fan-coil unit (FCU). Different electric heating elements can also be used alternatively to convectors. If a building uses forced ventilation, heating and cooling is usually accompanied with the supply air system. In this type of system, central air handling units (AHU) regulate and circulate air in the building, controlling the temperature, purity and humidity of the air. Goal of the HVAC systems is to enable comfortable indoor climate with minimal energy consumption.

In addition to HVAC systems, buildings often include other separate building automation systems including lighting, safety alarm, security alarm, water management and elevator systems (Kastner et al. 2005). Main problem lies with the separation of all these systems. The systems are built and maintained by separate companies, most of them having their own proprietary systems controlling them. Thus, different automation parts are not interoperable with each other and are difficult to be used in cooperation. The increase of interoperability, interdependency and integration of currently diverged building automation segments is highly supported in the building automation literature (Dietrich 2010, Dawson-Haggerty et al. 2013, McGibney & Ploennings 2016, Domingues et al. 2016, Salo 2017). Kastner et al. (2005) suggests that the number of interaction points at the highest system level should be kept at the necessary minimum, while keeping the system flexible for future integrations of additional subsystems.

3.2 Hierarchy and Structure of Building Automation Systems

The hierarchy of building automation can be constructed in three levels, as seen in Figure 2. On the top is the **management level** which includes human interface devices such as workstations and other user access devices, building monitoring units and webservers. Information from the whole system can be accessed from the management level, for example, automation level values can be accessed and modified from the management level. System data presentation, forwarding, trending, logging and archival can be handled from management level. (Kastner et al. 2005; Domingues et al. 2016.)

On the middle is the **automation level**, consisting of autonomously executed sequences. It consists of direct digital controllers (DDC) and unit controllers which receives inputs from the field level systems and returns outputs to the field level according to their control logic. Automation level devices processes the data and measurements given by the field level, executes control loops and activates alarms. (Kastner et al. 2005; Domingues et al. 2016.)

On the bottom is the **field level**, where the building automation system interacts with the physical world through sensors and actuators. Field level collects sensor data such as measurements, counting and metering and transforms it to machine-readable data. Likewise, field level is also controllable using actuators to change the speed of airflow pushed by a variable air volume device or control the lights. (Kastner et al. 2005; Domingues et al. 2016.)

As the architecture suggests, BAS is a distributed system, consisting of various devices working in cohesion. In it, sensors and actuators are connected to controllers via standardised interfaces or field network. Controllers, such as DDC control the processes, handling inputs and outputs of field level devices. In the management level, a server station supervises controllers, logs and trends their data flow. Finally, a human interface device such as a workstation displays the server station data from a convenient user interface.

Kastner et al. (2005) suggests that a trend toward a flatter hierarchy is arising. Automation level functions are often executed by adjacent levels, for example data aggregation and supervision is done in the management level whereas continuous control is operated in the field level.

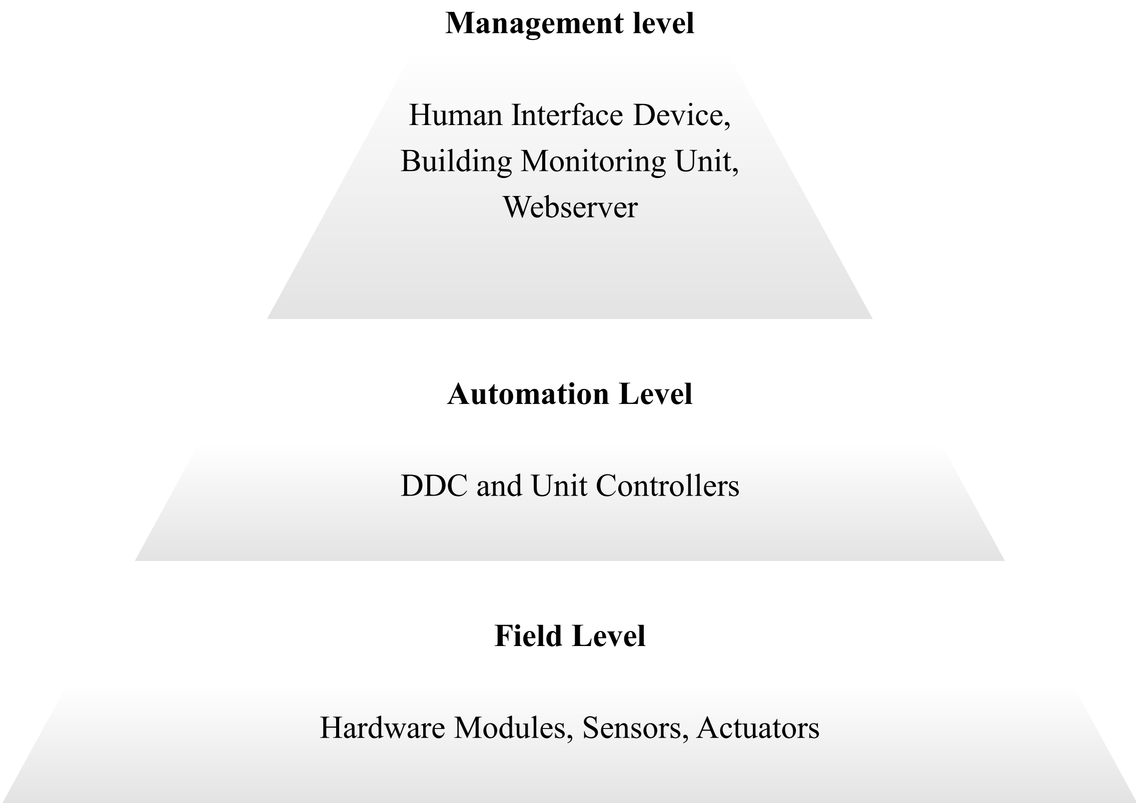

Figure 2. Building Automation Hierarchy (Kastner et al. 2005).

General setup of BAS field level consists of actuators, sensors and hardware modules which are connected by a backbone fieldbus. *Actuators* react to signals with closing circuits or differentiating electric loads, to control lighting for example. *Sensors* collect data from the physical reality, converting physical phenomena to measured signals. *Hardware modules* have actuators and sensors connected to their input and output ports, receiving measurements and readings from them and giving commands to them. *Fieldbus* enables the connections and communication between field level devices, including controllers, sensors and actuators. A similar data channel exists in the management level aggregating data via a common (often IP-based) backbone. The combination of field level fieldbuses and the backbone is often referred to as control network. Main standards today used for the control network include BACnet, KNX, LonWorks and Modbus. (Domingues et al. 2016.)

Controllers manage the interaction between devices through a certain control logic. In BAS, control typically consists of an application-specific hardware with embedded software which controls actuators according to sensor data or received commands (Domingues et al. 2016). For example, a movement sensor detects movement which is registered by the controller, which afterwards gives a command to the light actuators to turn on the lights. Controllers can vary in their complexity, some being able to run more than one control program and being able to communicate with other controllers via fieldbus. According to control function logic, controllers can often be divided to Programmable Logic Controllers (PLC) or Direct Digital Controllers (DDC). Control functions can also be generated in the management level through a server software, which can collect data from various segments of the system in order to create aggregations and sophisticated solutions for controlling the system.

Efficient building automation hardware and software stack enables the abstraction of low-level hardware devices, so they can be utilised effortlessly in software applications. Abstractions hide low-level details, offering a simpler model of low-level functions to higher levels. Especially in computer architecture, abstraction allows programmers to concentrate on software without having to know how the hardware operates (Hennessy & Pattersson 2014). Domingues et al. (2016) provides a comprehensive illustration of the full stack of BAS. Briefly described here, hardware segment of the stack consists of physical devices (lighting, air ventilators, sensors) and their simple logic-based interfaces which are connected to a grouped hardware module. Hardware module controls the physical devices through a microcontroller. Software stack consists of various drivers, including the fieldbus network drivers (implements network protocol stack and provides abstraction) and hardware module drivers.

Device drivers, including the hardware module drivers are parts of the system that controls the devices and provides a layer of abstraction on top of them. Higher in the stack, hardware module drivers are connected to software device drivers which in turn consists of application programming interfaces, enabling the interaction of applications and devices. Software device drivers converts electric signals into suitable data structures that can be read by software applications, and vice versa. The software device drivers act as the software abstraction layer, providing data from the devices in a software-utilisable form. In the top of the stack, high-level abstractions of devices are provided as objects which are more easily understood and manipulated by applications. (Domingues et al. 2016.)

Datapoints are a set of elementary data elements that the BAS represents itself to the software layer. Datapoints represent the physical process of devices in a logical way. Datapoint can be a room temperature, state of a lighting switch or something more abstract. Datapoints can be categorised into physical and abstract points. Every datapoint has their own metadata which lays out the set of rules how the datapoint is accessed and interpreted, including access type, datatype, installed location, influence zone et cetera. Datapoints typically offer three ways of access: read, write or both. (Kastner et al. 2005; Domingues et al. 2016.)

Devices are often grouped or zoned in the BAS. Device groups are a set of devices identified in a logical fashion. *Device group* can be a device collection, for example all devices in a room. Device group can also be a command group which is a set of devices with compatible interfaces recognising same commands or providing the same type of datapoints. Devices are grouped for the purpose of commanding them together, for example turning all the lamps on in one floor. (Domingues et al. 2016.)

Spaces can be arranged to sub-spaces, namely *zones*. A zone can be a room, floor or any other convenient area of a space. Zones can also be part of larger zones, in other words parent and child zones can exist. Zone's metadata consists of zone name, space usage profile, boundaries and location. Zones help perceiving BAS spatial structure, and it is also useful for users to understand, recognise and navigate the controlled area. (Domingues et al. 2016.)

Other functionalities in BAS in addition to grouping and zoning include event notification, alarm notification, historical data access, scheduling and scenarios (Domingues et al. 2016). An event occurs whenever the system's state changes, for example, when a light switch is turned on or a door is opened. Alarms are exceptional events that often announce a malfunction or another special event that the system must be notified of. Historical data access or logging provides data from the system. Logs can contain information about events, processes, alarms and user interactions. Logs are crucial for the purposes of data analysis. Schedules are for executing certain tasks based on a timetable, for example turning on the heaters in the morning or turning off the lights after working hours. Scenarios are desired statuses of devices or device groups, which conduct a desired outcome for a particular purpose. For example, a studying scenario can be a situation where lights are brightened in the workspace or a party scenario would increase the amount of ventilated air.

3.3 Building Automation Communication Protocols

Several building automation communication protocols exist in the building automation market. Proprietary and ad hoc solutions have traditionally dominated the industry, yet open systems have become more demanded (Kastner et al. 2005). Most protocols today are open (Schneider Electric 2015). Open protocols include BACnet, LonWorks and KNX, which are the three most used open protocols in building automation. Standard technologies employ common concepts together such as datapoints, grouping and zoning, however their functionalities are often defined and implemented differently. Domingues et al. (2016) argues that current protocols do not include all the functionalities required from a BAS, therefore BAS vendors fill the missing functionalities with custom made, proprietary and often ad hoc solutions. These issues supposedly are the main reasons for the heterogeneity problem occurring in BAS market.

Open building automation protocols include BACnet, LonWorks, KNX, ZigBee, Modbus, DALI, OPC UA and several web services. From the aforementioned, only ZigBee is a wireless protocol and the rest are wired protocols. Web service protocols, including REST API, KNX WS and BACnet/WS are mainly used in the management level. Domingues et al. (2016) groups web service protocols into management service frameworks, which are used in BA to integrate various devices from various vendors and to offer their abstracted form to external applications. Ultimately, management service frameworks including web services connect various building automation technologies to client applications. Web services connect BA to the internet which also enables the coordination of BA and IoT.

The spread and variety of different BA protocols are not examined in depth here. Several comparison analyses can be found from BA literature (Kastner et al. 2005; Domingues et al. 2016; Salo 2017; Schneider Electric 2015). It should be noted, that BA protocols are rarely interoperable with each other. Nonetheless, devices and systems that can employ a mutual protocol are interoperable. This gives pressure to BA vendors to develop devices that support all or most of the widely-used protocols (Schneider Electric 2015).

3.4 BAS 2.0: Building Operating Systems

In building automation research, concept of a building operation system has already been proposed (Dawson-Haggerty et al. 2013; Weng et al. 2013; Fierro & Culler 2015; Dixon et al. 2012 McGibney et al. 2016). Notably, Dawson-Haggerty et al. (2013) have proposed a new kind of building automation system which makes buildings programmable. According to them, existing building systems do not offer enough abstraction, resulting in complicated software development for building automation technology. Furthermore, they suggest that the control systems (or automation systems) of buildings should be fundamentally re-architected into secure, modular, extensible and networked systems.

In Dawson-Haggerty et al. (2013) report, a prototype called Building Operating System Services (BOSS) was built. The prototype was used in a 13,000 sqm building at the UC Berkeley campus. BOSS uses hardware presentation and abstraction layer to abstract the hardware stack for the use of applications. Furthermore, the system enables application portability. Field level devices are accessed through query requests which are expressed in metadata tags. The metadata-enabled query interface makes it possible to select objects based on their type, attributes and functional or spatial relationships. Different field-level devices have general functions addressed to them. For example, a variable air volume device can have functions such as “set airflow” or “set temperature”. Furthermore, specific vendor-dependent devices can have additional functions. A couple of applications were developed for BOSS, including a HVAC optimization, personalized control system and an energy saving auditing application. BOSS application developers felt that the system gave them power to concentrate on actual problem solving, such as how to visualise building data, i.e. they did not need to dwell on low-level hardware specifications and communication issues because of the provided abstraction.

Other building operating systems have also been designed and implemented primarily for research purposes. Fierro & Culler (2015) have a comparison analysis of such systems, including BOSS, BeMOSS, BAS, HomeOS, SensorAct, Building Depot v2 and XBOS. The building operating systems are evaluated in six main categories:

- Hardware presentation layer (HPL)
- Canonical metadata definition, storage and usage
- control process management
- building evolution management
- security
- Scalable user experience (UX) and application programming interfaces (API)

Hardware presentation layer hides the complexity of BA hardware device spectrum, presenting them in uniform datapoints, such as thermostat temperature readings. Canonical or standardised metadata with widely-used naming conventions gives context to devices, informing their location, zone, type, function and other properties. Control process management enables the control of BA controllers in a scenario-like manner. Building evolution management takes care that the software operating on the building recognises changes and adapts based on them. Security is a critical aspect of building automation communication, so it cannot be tampered by cyber criminals. Lastly, UI and APIs provide the interfaces to connect with the building system, thus providing the ultimate user experience and holistic view of the building automation. (Fierro&Culler 2015; Dawson-Haggerty et al. 2013.)

3.5 Internet of Things

Internet of Things means the connection of physical objects to the internet. These objects can be BA devices, refrigerators, toasters, doors, watches, cars or virtually anything that can be connected to the internet with themselves or through add-ons. Through internet, such objects can communicate and interact together without human intervention. Formally and generally IoT has been defined by the European Research Cluster on the Internet of Things (2017) 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”.

IoT can transform traditional objects to smart objects, enabling them to see, hear, think and perform tasks together. Use cases for IoT have been expected in home and business especially, improving the quality and easiness of life. Smart home applications include automatic coffee and food preparations, indoor climate improvements, TV management et cetera. In order to achieve the real potential of IoT, different applications and devices are required for these purposes. Also, efficient and compatible protocols are required for the communication of numerous heterogenous IoT devices. (Al-Fuqaha et al. 2015.)

Several applications are depicted and already implemented for IoT. IoT applications can be divided to three domains: industrial, health well-being and smart city (Borgia 2014). Industrial domain includes applications in industrial processing, agriculture and logistics, where specific use cases can be industrial plant monitoring, luggage management, irrigation monitoring, warehouse management and shopping operation. Health well-being domain includes medical & healthcare and independent living, drawing use cases in smart hospital services, elderly assistance and medical equipment tracking. Smart city domain consists of smart mobility & smart tourism, smart grid, smart home/building and public safety & environment monitoring. Examples of smart city applications are traffic management, road condition monitoring, energy management, sustainable mobility, comfortable living, video surveillance, building automation management and personnel tracking. Altogether, IoT has an immense amount of use case potential.

IoT can be depicted through certain building blocks and characteristics. Al-Fuqaha et al. (2015) categorises IoT elements to six domains: identification, sensing, communication, computation, services and semantics. *Identification* is a key element for IoT devices to be able to recognise them in the wide network. Identification is typically performed with object IDs (name) and addresses (IP). Second key element for IoT is *sensing* which means the collection data from the physical objects, such as a thermometer reading or a refrigerator content. *Communication* is needed for the interaction and data transmission of devices in the network with one another or a server. Third element is *computing* which means the processing ability of IoT, being able to make computations in IoT devices or in the server or cloud. Evidently, IoT is not just devices but it requires *applications* for creating added value for users. Last element is *semantics* which means that the knowledge and data of various devices must be extracted in a convenient way so it can provide required services.

Architecture of IoT can be depicted in several ways. Al-Fuqaha (2015) presents four different IoT architectures collected from previous IoT literature, including three-layer, middleware based and service-oriented architecture. Li et al. (2015) depicts a service-oriented architecture with four layers, including sensing, network, service and interface layer. Addition to these, Borgia (2014) shows a three-layer architecture of collection, transmission and process, management and utilisation phase.

Borgia's (2014) three-layer representation of IoT will be described in more depth here, since it best depicts the IoT architecture in a conceptual level. As was mentioned, this architecture consists of three layers or phases: collection (1), transmission (2) and process, management & utilisation (3). Collection phase consists of sensing devices, such as RFID, sensors, actuators, thermostats or GPS terminals. On top of them are short-range communication technologies, including ZigBee, Bluetooth and NFC. Overall, collection phase is the phase of sensing physical data with sensors and collecting it with short-range transmittance.

The transmission phase consists of mechanisms that deliver the collected data of collection phase devices to applications and servers. The data is transmitted through various gateways and heterogeneous technologies, including wired, wireless and satellite. Transmission phase is divided to gateway access and network. (Borgia 2014.)

Last phase is the process, management and utilisation (PMU) phase which processes and analyses the data coming from IoT devices. After crunching, the data can be sent to applications and services for external use. The phase also enables two-sided communication, providing feedback to control applications which give commands to IoT devices. Other management work is also done by the PMU phase, including device discovery and management, data filtering and aggregation, semantic analysis and information utilisation. PMU phase supplies the abstraction of lower-level hardware and protocol specifications which enable high-level data and control management needed for relieved application development (Borgia 2014.)

IoT incorporates an abundant amount of different communication protocols and standards. Several organisations and consortia have tried to acquire more harmonised IoT communication and operation, including IEEE, W3C, ETSI, ITU, AIOTI and Open Group. Close to the physical devices, various communication protocols and technologies are used, including IEEE 802.15.4 (ZigBee), IPv4/v6, Wi-Fi, broadband cellular (2.5G, 3G, 4G), Z-Wave, NFC and Bluetooth. Several application or management level protocols are also used, including Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT) and Advanced Message Queuing Protocol (AMQP). (Borgia 2014; Al-Fuqaha et al. 2015; Li et al. 2014; Robert et al. 2016.)

Recently, two IoT messaging standards have been defined, namely O-MI (Open Messaging Interface) and O-DF (Open Data Format). They are published by Open Group. The purpose of these standards has been to fill the interoperability gap present in IoT. These standards tackle the high-level requirements of IoT standardisation. O-MI and O-DF combination compares itself to HTTP and HTML protocols of internet, where HTTP enables the transmission of data and HTML defines how the content should be formatted. O-MI and O-DF have a similar goal in IoT as HTTP and HTML in internet, where O-MI enables universal communication between IoT devices and O-DF gives a format for presenting data and payload. Both O-DF and O-MI are specified in XML Schema. (Robert et al. 2016; Open Group 2017.)

More closely, O-MI follows peer-to-peer communication model, where an O-MI node can act as both server or a client. O-MI messages can be transported by any low-level communication protocols that can handle XML documents or strings. O-MI messages have four operations: read, write, subscribe and cancel. O-DF represents payload received from IoT objects in a standardised fashion, that the data could be understood and exchanged universally in all information systems. O-DF only defines how information should be expressed of IoT objects, not how to communicate it (O-MI covers communication). O-DF is specified in XML Schema. O-DF can be used with the O-MI protocol, but O-MI can transport payloads in other formats also. (Robert et al. 2016; Open Group 2017.)

IoT wave has just began and several issues need to be solved in the future. Al-Fuqaha et al. (2015) addresses eight key challenges: availability, reliability, mobility, performance, management, scalability, interoperability and security & privacy. *Availability* means that IoT should be able to provide services anytime and anywhere for users. *Reliability* is required for IoT, so that the devices and their systems communicate, response and deliver with certainty. *Mobility* must be taken into account, so that IoT implementations can be used on the go through mobile devices. *Performance* is required to achieve an efficiently working system of networked devices. Considering *management*, the abundant amount of IoT devices requires light-weight management protocols that can handle the whole spectrum of IoT networks. *Scalability* is required for the purpose of IoT spectrum expansion. Similarly, *interoperability* is highly demanded for IoT, so that the wide range of heterogeneous devices, their systems, platforms, protocols and conventions can be used in harmony. Last but not least, *security and privacy* are and will be tremendous challenges for IoT, keeping personal information and crucial devices out from the hands of criminals and third parties.

3.6 Commercial Markets

Currently, commercial building automation devices and systems are produced by various vendors, including Honeywell, Tridium, ABB, Schneider Electric, Siemens, Johnson Controls, Bosch, Cisco, Delta Controls, Distech, Echelon and Philips. Commercial systems and devices often support most of the conventional BA communication protocols. Similarly, IoT business has an abundant number of vendors in both software and hardware, including Amazon Web Services, AT&T, Bosch, Cisco, Dell, Google, Huawei, IBM, Intel, Microsoft and Samsung. The competition is fierce in IoT markets right now, attracting major ICT companies. It should also be noted, that numerous start-ups around the world are rivalling in IoT and smart building business.

Contemporary building automation solutions enable IoT and cloud solutions as well, following a similar trend that building operating system research has compiled. For example, Intel markets a building management platform which connects cloud-based building applications with traditional building automation devices and also IoT devices. Similarly, Tridium develops their Framework infrastructure which enables the creation of web-enabled applications that can be used to access, automate and control BA and IoT devices. Furthermore, integration and better UX have evidently become key selling points for BA companies when inspecting their products.

In 2016, total market value of building automation and controls market was approximately $50 billion. It has been estimated, that the market would nearly double to the vicinity of $100 billion by 2022 (Atkinson 2017). Focusing on the smart building market of BA, it is expected to grow from $5.73 billion in 2016 to $24.73 billion by 2021. Likewise, in the IoT market, a major growth has been anticipated. Forecasts include $933.62 billion by 2025 (Industrial IoT), $267 billion by 2020 (Business to Business IoT) and $724.2 billion by 2023 (global) (Grand View Research 2017; Columbus 2017; Research Nester 2017).

4 Interoperability in Telecommunications and Authentication

This chapter gives an overview of telecommunications standards and electronic authentication. More closely, electronic authentication is discussed on mobile and non-mobile computing platforms, giving an insight on Finnish e-authentication methods; specifically TUPAS, a strong digital authentication method supported by Finnish banks and Mobiilivarmenne, a mobile signature solution supported by all major Finnish mobile network operators. Telecommunication standards are discussed on general level, focusing on their evolution and provided benefits. Altogether, chapter concentrates on how interoperability has been achieved through standardisation and collaboration in these fields.

In the context of the thesis, this literature chapter attempts to discover solutions for achieving interoperability in a building operating system. As was discovered in the last chapter, current building automation systems are fairly closed systems with weak integrability. Thus, answers for enhancing interoperability are sought from telecommunications and electronic authentication standardisation development.

4.1 Telecommunications Standards

Evolution of telecommunications, specifically mobile phones and the internet has been generally described in chapters 2.5 and 2.6. This chapter tackles the standardisation aspect of telecommunications, trying to find implications between harmonisation of such technologies and collective benefits through interoperability.

In the manufacturing industry, standard can be interpreted as a mutual set of quality norms or criteria. When considering ICT standards, it is important to perceive the characteristics of information goods. First, ICT depends highly on heterogenous technology systems built from various devices by various manufacturers. Connecting this device entity requires compatibility and interoperability between components. Second, switching of goods and services in the ICT is costly. For example, transferring from 2G to 3G technologies require an immense amount of investment in labour and money. Third, network effects are a key factor in ICT standard policy and they play a substantial role in industry competition. Network effects mean that when more people adopt a certain technology, benefits for the whole user base grow greater. Also, ICT encompasses new-emerging multi-sided markets, where two or more users interact and perform business through a platform. These platforms can obtain significant market share and become de facto standards, for example Android and iOS have become de facto standards in mobile phone operating system market. (Shin et al. 2015.)

Historically, ICT standards have stemmed from government regulative endeavours. Governments have seen a need to assign regulative standards for suppliers in order to prevent monopoly abuse and permit the widest possible choice of providers and features. Without standards and regulations, full potential of technology might have been lost with fragmented and non-interoperable technologies, scattering the markets. Development of competition usually consists of initial phase of monopoly dominance, followed by the rise of many small competitors, and ending in the perseverance of a few strong companies. (Maeda et al. 2006.)

In the United States, Europe and Japan, government presence in ICT standardisation has varied. In Japan, NTT exercised a domestic-oriented policy in telecommunications industry, implementing proprietary domestic standards which hampered the export of Japanese telecommunications technologies but also made difficult for other technologies to enter the Japanese markets. In the US, market-induced standardisation has been the primary driver. US telecommunication carriers have not been government owned, although they have been highly regulated by the government. The Federal Communications Commission (FCC) encouraged market-based competition, permitting the resale and sharing of monopoly services and fighting against the AT&T monopoly. FCC also took into account of the possible issues of fragmented markets with the AMPS system, requiring interoperability between providers. Overall, FCC policies were effective and stimulated competition, resulting in dropped prices for mobile services. In Europe, both market- and government-driven strategies have been implemented in individual countries, however a trend towards privatisation can be observed. In general, less pressure has occurred for the deregulations of telecommunications standardisation in Europe, possibly because co-European governmental efforts have succeeded, for example with GSM standard. (Maeda et al. 2006.)

Funk (2009) has collected a historical framework of standard setting in mobile phone industry. According to him, changes in technology have brought new problem-solving methods, and the evolution of problem solving and standard setting has been circular rather than unidirectional. Traditionally, problem solving in telecommunication was integral, where regulated monopolies determined all standards and specifications in order to conserve integrity. Similarly, standardisation was quasi-vertical, developed by a single administration and implemented vertically in their own domain. This type of quasi-vertical, integral and government monopolised situation occurred until the late 1970s.

Later, problem solving has evolved into modular style and standard setting has become an open process. In the 1970s and 1980s, governments and corporations started to realise that specifications for technologies could be provided by others than government monopolised service providers and that the markets could be opened for competition. In 1978, telecommunications markets were opened in the US, where consumers can buy phones from any approved manufacturers, and the manufacturers were allowed to determine their own specifications and designs for the phones, as long as they conformed to the open interface assigned by FCC. Later, FCC introduced AMPS which followed similar modular strategy in analogue mobile solutions. (Funk 2009.)

Aside from the US, Europe and particularly the Nordic countries have also exploited modular problem solving and open standard setting successfully similarly to FCC in the 1970s. Nordic countries were the forerunners in such methodologies when the Nordic Mobile Telephone (NMT) system was erected in the 1980s. In other European countries, analogue mobile systems did not obtain a similar growth as NMT, because of the lack of open standards and competition. In Japan, Germany, France and Italy for example, mobile systems were quasi-vertically integrated, where the standards were controlled by national service providers, and there was a lack of competition between service providers or mobile manufacturers.

Later in the digital mobile era, similar policies were deployed in the GSM system. Scandinavian countries, including their service providers and manufacturers such as Ericsson and Nokia had an important role in the successful implementation of GSM, convincing Germany, France and other European countries to adopt an open standard

European-wide. Evolutional climax occurred in 1988, when the European Telecommunications Standards Institute (ETSI) was established. Together with the mobile industry, ETSI started developing the GSM standard which started as a de jure standard in Europe and later became the global standard (Funk 2009). Overall, ETSI combined the former fragmented market of country-level government monopolies and opened the markets with an inter-European standardisation system (Robin 1994).

The success of GSM became a prime example of how open standardisation and modular problem solving could succeed in the global telecommunications market. A GSM alliance was established for the globalisation of GSM standard, and later it was integrated with the International Telephone Union (ITU). In the turn of the millennium, the evolution also made an impact on the isolated Japan which was still using national proprietary standards in the 2G era. Finally, when 3G era arrived, Japanese government demanded that their dominant mobile operator NTT Docomo would either adopt or create a global standard. (Funk 2009.)

In the third generation, two noteworthy technology standards have been implemented, namely UMTS and CDMA2000. UMTS is developed and maintained by the 3rd Generation Partnership Project (3GPP) and is mainly used in Europe, Japan and China. UMTS has evolved from GSM, following a similar path of development and implementation as GSM. UMTS was not considered as a replacement of GSM at first, but as a complement. Later, ETSI took an initiative to make UMTS the 3G universal standard with similar steps as GSM was adopted: make UMTS the European standard, use same frequency bands in the Euroarea, and promote inter-European policies for acquiring network diffusion and its coverage (Fuentelsaz et al. 2008). Furthermore, UMTS incorporates the whole 3GPP group of seven telecommunication organisations from all over the world, including ETSI, TTC and ATIS. UMTS's competitor CDMA2000 is developed and maintained by the 3rd Generation Partnership Project 2 (3GPP2) and is mainly used in North America and South Korea.

Fourth-generation telecommunication technologies are dominated by 3GPP's Long-Term Evolution (LTE) standard. LTE technology enables 100 Megabit per second mobile data rates. LTE formerly competed with the Ultra Mobile Broadband (UMB) standard managed by 3GPP2, but UMB's development was discontinued in 2008 when its main sponsor Qualcomm announced that they would cease its development and switch to LTE technologies (Reuters Staff 2008). Recently, LTE's roadmap has been extended to develop LTE-advanced (LTE-A), which will offer data rates beyond 1 Gigabit per second. Furthermore, 5G technologies are already discussed, requiring a paradigm shift that includes higher bandwidths, expanded base station and device densities, and more antennas in order to support the enormous volume of data traffic in the future (Alsharif & Nordin 2017).

Different conventions in telecommunications standard development have been researched widely. David & Steinmuller (1994) have compared US and European conventions in standard development, where US standards have developed through more pure competition, whereas in Europe more harmonised, central-coordinated and consolidated standardisation has been strived via collaborative organisations. Blind & Gauch (2008) suggest that both formal standardisation by standardisation organisations and informal standardisation by industry consortia is recommended, since they can complement and synergise each other and offer dividual benefits. Informal standardisation enables more flexible and grassroots standardisation, whereas formal standardisation actuates wider deployment. Likewise, Leiponen (2008) argues that informal standardisation matters for standard setting in telecommunications. Gruber & Verboven (2001) concludes, that competing and singular standards can both have advantages and disadvantages, but a single analogue standard helps develop markets significantly faster which is consistent with the presence of network effects and the economies of scale. Similar conclusions are implicated to digital standards, including GSM. Similarly, Shin et al. (2015) concludes that standardisation supports both technology or market-based competition, achieving interoperability of complementary products and services. Standards enable integration of telecommunication commodities, and they can help promote innovation and enhance market expansion.

4.2 Electronic Authentication

Electronic authentication is the process of verifying an individual's identity electronically. Different information systems can use the authenticated identity to qualify if a person is authorised to do a specific electronic transaction (Burr et al. 2013). Identifying personalities remotely in an open network, namely internet is a major challenge. Benefits that reliable electronic authentication offer are greater security, reliability, integrity and transparency (Spyrelli 2002). Also, time and costs can be saved, when signing papers and performing public office errands is possible online. Electronic authentication systems rely on cryptographic methods, for the purpose of retaining messages in secret when communicating in the open internet.

In the United States, electronic authentication is an important part in digitalising the government, making it more efficient, flexible and easier to access. Nationwide, the Federal Chief Information Officer of the United States leads the electronic authentication. Guidelines for electronic authentication are provided by the National Institute of Standards and Technology (NIST). NIST categorises electronic authentication in four levels based on the required assurance: Level 1 requires no identity proofing, Level 2 provides single factor remote network authentication, Level 3 provides multi-factor remote network authentication (at least two) and Level 4 provides the highest practical remote network authentication assurance. Level 4 requires special cryptographic tokens, for example the Personal Identity Verification (PIV) authentication key. (Burr et al. 2013.)

Furthermore, the Office of Management and Budget (OMB) in the US gives guidance in a 5-step process on how agencies should meet their electronic authentication assurance requirements (Bolten 2003):

1. Conduct a risk assessment of the government system
2. Map identified risks to the appropriate assurance level
3. Select technology based on e-authentication technical guidance
4. Validate that the implemented system has met the required assurance level
5. Periodically reassess the information system to determine technology refresh requirements

In the EU, electronic authentication is instructed by the EU regulation no. 910/2014 on electronic identification and trust services for electronic transactions in the internal market. More commonly this regulation is called the eIDAS regulation. eIDAS conducts the following (European Union 2014):

a. lays down the conditions under which Member States recognise electronic identification means of natural and legal persons falling under a notified electronic identification scheme of another Member State;

b. lays down rules for trust services, in particular for electronic transactions; and
c. establishes a legal framework for electronic signatures, electronic seals, electronic time stamps, electronic documents, electronic registered delivery services and certificate services for website authentication.

EU used to have another regulation for electronic authentication, the Electronic Signatures Directive 1999/93/EC. eIDAS was decreed in order to amend the old legislation, where two main problems were addressed. First, citizens of EU cannot use their electronic ID in another member state, since the e-ID solutions are national, thus providing access only in its own domain. Accordingly, this problem was addressed and cross-border identification was suggested. Second, the new eIDAS legislation included other trust services than signatures to be used in electronic authentication, including electronic seals, stamps, documents registered delivery services and certificate services for website authentication. (Cujipers & Schroers 2014.)

According to Huhnlein (2014), eIDAS guidelines are well suited for service-oriented identification system, i.e., providing Identification as a Service. Furthermore, Morgner et al. (2016) conclude that the eIDAS transaction system is an easy and practical method in authenticating transactions, and it provides strong assurance with powerful cryptography technologies.

In Finland, eIDAS was carried out in national legislation by amending the law *Act on Strong Electronic Identification and Electronic Signatures 7.8.2009/617*. The amended electronic ID legislation came into force in July 2016.

In practice, strong electronic identification (e-ID) is carried out in a couple of ways in Finland. The most successful and used e-ID method is the TUPAS strong digital authentication. TUPAS comes from Finnish words *Tunnistuspalvelu Standardi* which means identification service standard in English. TUPAS is provided and maintained by Finance Finland, which is an interest group of Finnish banks, insurance companies, finance companies, investment fund companies and other finance sector employers. For identification, TUPAS uses bank customers' electronic user identifiers such as a username and a PIN code, and also a transaction authentication number (TAN). Most commonly, TANs are provided in paper passcode lists of one-time use PIN numbers. With both PIN and TAN, two-factor authentication is achieved and it can be used as a strong electronic identification method in Finland. TUPAS method can be described as PIN/TAN scheme (Kerttula 2015).

Currently, TUPAS is the de facto standard of strong electronic identification and it covers over 90 % of Finnish electronic identification activities, being used approximately 70 million times per year for identification. Recently, TUPAS authentication cost 50 cents per use for the service provider, but since May 2017 authentication cost was reduced to 10 cents per use by a legislative demand. No definitive promises have been made, that the reduced costs from the service provider are not transferred to bank customers (Parviala 2017). Indeed, TUPAS provides a strong electronic identification method but using it requires effort and costs.

Other strong electronic identification methods in Finland include electronic ID card and mobile certificate. Electronic ID cards have not managed to diffuse successfully, mainly because their deployment and use have been cumbersome. Especially, electronic ID cards need a separate card reader which undoubtedly complicates its use; however, in 2017, the situation changed when the cards became readable by NFC-enabled smartphones (Väestörekisterikeskus 2016). Finnish electronic ID cards can be described as a certificate-based authentication scheme, and currently they are the only qualified certificates available for electronic authentication in Finland, however Finnish mobile certificate meets all the requirements to become a qualified certificate, only missing the official status (Kerttula 2015).

Mobile certificate or *Mobiilivarmenne* in Finnish enables strong electronic identification via a mobile phone. Mobile certificate is supported by all the major Finnish operators, DNA, Elisa and Telia. These three operators work in a Circle of Trust, a cooperative framework, where operators accept electronic IDs created by each other, allow co-accepted IDs to roam on their networks and let anyone in the Circle of Trust to utilise each other's agreements in collaboration. This Circle of Trust is based on ETSI standards and standardised Web-service interfaces, and is unique globally as a mobile signature scheme. (Kerttula 2015.)

Guidelines for mobile certificate implementation is nationally given by the FiCom union, which is an organisation promoting interests of telecommunications and IT businesses. FiCom guidelines are based on ETSI MSS TS 102 204, TR 102 206 and TS 102 207 standards. Other technologies include XML Schema, SOAP, XMLSignature, WSDL, PKCS#7 and SAML. Certificate is located in the phone's SIM card, and it has to be activated by the network operator. Currently, mobile certificate works in over 1000 Finnish online services. Mobile certificate does not require a username or a password to be used for authentication, only your mobile phone and a PIN number. (Mobiilivarmenne 2017; FiCom 2014; Pulkkis et al. 2013.)

Finnish mobile certificate uses a two-factor, two-channel model. Each user has their tamper-proof universal integrated-circuit card (SIM card) and a personal secure PIN, i.e., a two-factor authentication is utilised. Two-channel means that service channel is separated from that used for signature. Authentication can be done in two-terminal model, where the authentication occurs on a mobile device and the service is accessed from a desktop device. One-terminal model is also possible, where both service and authentication is processed on a mobile device. (Kerttula 2015.)

Kerttula (2015) argues, that mobile signature scheme is superior to PIN/TAN (TUPAS), one-time password (OTP) token, EMV Card+Reader, Mobile SMS-OTP and Public Key Infrastructure (PKI) Smart Card+Reader authentication schemes. Features that were compared include security, easy-to-use, mobility, usage cost, distribution cost, and maintenance.

Finnish mobile certificate system has been tried two times in the past. In 1999, a SIM-card based pilot was carried out, but the time was too early, mainly because sufficient amount of capable devices were not utilised yet and the legislation was inadequate. Later in 2005, another proper attempt was put out, but the law at that time required police department to issue strong identification, which resulted in difficult registration process and the attempt failed again. The latest effort started in 2011, providing better coordination, adequate legislation, diffusion of compatible mobile devices, better standards and wider network of service providers (Kerttula 2015). Today, mobile certificate is still lagging in user rates, only being used by approximately 200 000 people in Finland. A reason for the slow deployment of mobile certificate might be that its use costs on some operators (Niskanen 2016).

5 Interview Study

5.1 Synthesis of Literature

Theoretical background for the thesis is reviewed in chapters 2, 3 and 4. Chapter 2 discusses the evolution of computers and operating systems, including the general history of mechanical computers, general-purpose computers, digital computers, mainframes, minicomputers, transistors, integrated circuits, personal computers, the internet and mobile devices. Chapter concentrates in the evolution of operating systems. Last segment of 2nd chapter gives an overview of enterprise resource planning systems and their evolution. Second chapter provides definition, purpose and context for operating system and gives implications what it would be in the context of a digitalised building.

Chapter 3 gives an overview of building automation, building automation systems, building operating systems and internet of things. More closely, purpose of building automation and the architecture and structure of building operating systems are covered, review of building operating systems depicted in previous research is carried out, overview of internet of things is laid out, and a brief report of building automation and IoT market is given. Third chapter reviews building ICT's contemporary status and anticipated trends, attempting to find the developmental path and requirements for a building operating system.

Chapter 4 goes over telecommunications and electronical identity authentication standardisation. Telecommunications section follows the generations of telecommunications history from 1G to 5G, describing how different technologies have developed through standardisation. Furthermore, standardisation policies and their evolution in Europe, USA and Japan are discussed, especially concentrating in the evolution of Nordic and European telecommunications standards. Electronic authentication section gives an overview of electronic authentication in USA and Europe, concentrating on legislation, guidelines and standardisation. Also, electronic identification standards developed and utilised in Finland are described in depth. Fourth chapter strives to find resolution for the interoperability issue of current building automation systems.

5.2 Overview of Real Estate and Construction Industry

Construction industry accounts for a considerable portion of global GDP, being approximately 3.1 % in 2016 (Turner & Townsend 2017). Global GDP in 2016 was approximately $75 trillion, thus construction industry accounted for over $2 trillion of global GDP (Statista 2017c). Furthermore, global real estate is estimated to value $217 trillion in 2015 (Hackett 2016). In Finland, built environment (includes real estate and construction industry) accounts for 20 % of GDP, employs 13 % of workforce, is 80 % of Finland's national wealth and consumes 42 % of total energy (ROTI 2017).

Built environment encompasses both construction and real estate industry. These industries are often categorised through the life cycle of built environment, roughly divided to land use, design, construction and facility management & operation. Land use includes such tasks as land acquisition, land policy, planning, zoning and surveying. Design includes the design of buildings and infrastructure with blueprints and more recently with building information modelling. Design can also include bill of quantities and the general planning of a construction project. Designing is usually separated to architectural, structural, mechanical and electrical engineering. Mechanical engineering more closely consists of HVAC and plumbing engineering. Infrastructure encompasses roads, streets, airports, harbours, railways, underground railways, tramlines, sewerage and all other infrastructure required to connect and facilitate built regions within and between them.

Construction encompasses the construction activities of a building. When a building is erected for the first time, new construction occurs. Later in the life cycle of a building, various additions, alterations and renovations are typically made to redevelop the building. Like designing, construction is often categorised to building construction and infrastructure construction. Construction projects are often led by a prime contractor chosen by the client through a competitive tendering. Prime contractors often have several subcontractors especially in large construction projects. Specialised contractors are often used in various construction tasks, including earthworks, structural works, mechanical works, electrical works, plumbing, tiling, painting, masonry, roofing, insulating, carpentry, wet room works, et cetera. Construction products industry also belongs to the construction industry.

More collaborative contracting models have been developed recently, namely alliance contracting. Alliance model is a construction implementation model, where different parties of the project, including the client, designers, contractors and consultants agree on a shared contract and form an alliance. With a shared contract, alliance members share the risks and benefits of the project in a predetermined way. Basic principles of an alliance project are transparency, trust, joint and several liability and joint decision making. (Yli-Villamo & Petäjäniemi 2013.)

Last, facility management & operation oversees the operating time of a building, consisting of building management, building maintenance and building operation. Building operation can include living, office use, shopping use, storage use, building automation utilisation, smart building utilisation et cetera. At the end of the life cycle the building is demolished and redeveloped. Real estate development encompasses the whole life cycle. Evidently, real estate and construction industry go hand in hand throughout the built environment life cycle.

Megatrends being targeted at built environment include user-centredness, aging, global warming, globalisation, urbanisation and digitalisation. Users of buildings and built environment should be in the core of the whole industry. Needs and hopes of residents, office workers, shopping centre customers and public building visitors should be taken into account. Furthermore, serviceability should be emphasised in the user-centredness of built environment. (Kiinteistö- ja rakentamisfoorumi 2011.)

To combat against global warming, actions need to be taken in the built environment, considering that it encompasses 42 % of total energy consumption in Finland. Energy efficient and eco-friendly solutions should be introduced by decreasing energy consumption of heating, lighting and HVAC, and by making building structures more environmentally friendly. People should also be enlightened to use their buildings more energy efficiently.

Urbanisation is also pushing the boundaries of built environment. Globally, urbanisation is increasing every year and since 2008 over half of the global population is living in cities. By 2050, two-thirds of world population is expected to live in urban areas (United Nations 2014). Considering demands of the industry, urbanisation requires functional and serviceable infrastructure, infill development and denser cities. In recent debates, mobility has been a key driver for city development, for example in Finland subway in the Helsinki metropolitan area is being extended and a tramroad is being built in Tampere city. Construction activity is believed to boost economic growth and development (Crosthwaite 2000). This attributes to the reason why many large infrastructure projects are being carried out globally, striving to boost the stagnant economic development. In Finland, almost 43 000 new apartments are expected to start construction in 2017 which is the largest volume achieved since the 1990s (Pakarinen 2017).

Construction industry is often depicted as a traditional, capital-intensive and labour-intensive industry. Recently, digitalisation has begun gaining steam in real estate and construction, breaking the barriers of old traditions and conventions in the industry. Puhto et al. (2016) survey tells that the digitalisation of real estate and construction sector is only beginning in Finland. It is still mostly seen as a tool to enhance old business, instead of creating new digital business. Furthermore, in the survey, one fourth of companies participated did not believe that digitalisation would matter to their companies' strategies and 9 % saw that digitalisation cannot produce value to their clients. It remains to be seen how digitalisation will change the industry, but several initiatives on digitalisation have begun globally, including KIRA-digi in Finland, Smart Built Environment in Sweden, e-difice in Estonia, Bloxhub in Denmark, Catapult Future Cities in United Kingdom and AEC Hackathons all over the world. Globally, construction and real estate industries have been laggards in both digitalisation and productivity development, as is covered in chapter 1.1.

Construction industry is generally plagued with fragmentation (Nawi et al. 2014; Scott 2014). First of all, construction design and projects are separated to several expert fields as was mentioned above. Therefore, construction projects include several stakeholders, including clients, designers (structural, mechanical, electrical), contractors and consultants. Furthermore, contractors often use several subcontractors for different specialised tasks, including excavation, infrastructure, structures and building services. Thus, construction projects constitute of several parties that require to cooperate with each other, necessitating collaboration and integration between various parties of design and construction (Scott 2014). Unfortunately, construction projects often provoke quarrelling and in the worst case end up in court. Also, construction industry is dominated by small companies, encompassing a large amount of sole traders and small businesses (Rakennusteollisuus 2017).

5.3 Methodology

Empirical part of this thesis is carried out through an interview study which is analysed through a thematic analysis. Purpose of the interviews is to expand the framework collected from the literature review by interviewing building environment experts about the present and future state of building ICT.

For the sake of this interview survey, a semi-structured qualitative interview method was chosen. Other qualitative interview methods include unstructured and structured. Structured interviews are often used for acquiring quantitative data. Structured interview usually follows a rigid questionnaire of definite questions, asked in the same order and in a similar way from all the interviewees. Purpose of structured reviews is usually to acquire a fairly large comparable data stack that can be interpreted with statistical methods (Edwards & Hallond 2013).

Qualitative research encompasses a broad spectrum of research methodology, concentrating on qualitative data. Commonly, qualitative research is compared with quantitative research. Quantitative research transforms data of measurements, interviews, activities or any phenomenon to numbers which can be analysed with statistical methods and simplify them into mathematical models. Thus, quantitative researchers often ask the questions “How much?” and “How often?”. Qualitative research in turn does not often transfer data into numbers but tries to find implications in the wholeness of the subject. Flick (2009) lays out four main characteristics of qualitative research:

- *Gestalt*. Meaning the essence of form or shape in German. Gestalt-type research approaches data holistically, connecting various pieces and knitting them together to form an entity.
- *Bricolage*. A pieced-together set of representations that is fitted to the specifics of complex situation. Qualitative research uses various types of possibly fragmentary and mismatched data, combining them into a meaningful, aesthetically pleasing and useful research synthesis.
- *The Funnel metaphor*. Qualitative research begins from a broad research question and gradually its focus starts to narrow based on collected data and knowledge, guiding a more explicit path for the research. Thus, the research evolves metaphorically through a funnel, starting off from a wider hole towards a smaller hole.
- *Sensitising concepts*. Theories or interpretive devices that serve as background ideas based on previous literature or experience that can be used as a framework through which researcher can see, organise and experience the research problem.

Broadly, qualitative research can be defined as any kind of research that does not typically use statistical methods or quantification to create results. Qualitative research often has a more naturalistic approach towards the research question and it strives to find answers on a more contextual and phenomenal level. Furthermore, researchers in qualitative research get involved and take a clearer role in the research, whereas quantitative researchers strive to disassociate themselves from the process. Nevertheless, both qualitative and quantitative research ought to assert their credibility. In qualitative research, credibility is measured by the ability and effort of a researcher, whereas quantitative research is measured by the operability and reliability of the research instrument. (Golafshani 2003.)

A qualitative and semi-structured interview often includes three main characteristics: dialogue, a thematic, topic-centred, biographical or narrative approach and contextualism for the purpose of situated knowledge. A qualitative interview requires both interviewer and interviewee to exchange thoughts and opinions on the subject through a dialogue. Interview is approached in themes, topics, biography or narrations. Also, it is important that the interviewee understands the subject matter and its context, so that the interviewer can acquire the answers he or she is looking for. (Edwards & Hallond 2013.)

Brinkmann (2013) divides the process of a qualitative interview study into four phases: preparation, interviewing, analysis and reporting. These phases are not discrete, but they can overlap one another and go in cycles. Qualitative research design can be loose or tight, but it is important to take advantage of the flexible and inductive research logics that qualitative research offers. This four-phase process has been used in this thesis. Furthermore, Brinkmann (2013) suggests three main goals of qualitative interviewing: discovery, construction and understanding. With these three goals, interviewers should *discover* something that interviewees do not know, *construct* something that interviewees would like to see happen and *understand* something that interviewees do not understand.

As in this thesis, a semi-structured qualitative interview study is often the sole data source in qualitative research study. Semi-structured in-depth interviews are the most widely used format for qualitative research. Furthermore, these semi-structured interviews can be divided to individual and in group interviews (DiCicco-Bloom & Crabtree 2006). In this sense, this study uses a qualitative semi-structured in-depth individual interview method.

Edwards & Hallond (2013) describe a semi-structured interview as a method, where the researcher has a set of questions or topics that want to be covered in the interview, i.e., an interview guide. Nonetheless, interviewer has flexibility on when and how the questions are given to the interviewee and in what way and from what standpoint they should be responded. Context and content is important, so that the interviewees understand the topics and they can provide adequate answers for the interviewer. Semi-structured interviews certainly give room for interviewees to shape their answers in a custom they desire, but it also gives an adequate structure for the dialogue, providing answers for topics which the interviewer is seeking replications to.

Regarding the adequate number of interviewees for a study, a few elements have to be noted. Often, the concept of saturation is used to determine if an interview study has enough interviewee data. This means that interviewing should be continued with more people until saturation is achieved, which means that the interviewees do not dispense new information that is essential for the research. Thus, in principle, there is no exact amount required for a qualitative interview study to ensure its rigorousness and trustworthiness. Furthermore, other reasons that control the number of interviewees include epistemological, methodological, practical and epistemic community issues. Some research also gives definite numbers for a reliable study, which range from 1 to 100. Overall, the number of interviewees depends on the issues mentioned before, pursuing a sufficient saturation level. (Edwards & Hallond 2013.)

Qualitative interviewing poses certain strengths and challenges. As for strengths, qualitative interviewing can give insight to interviewees experiences, social processes, practices and events. Furthermore, qualitative interviewing can explore the following aspects (Mason 2002):

- The texture and weave of everyday life
- The understandings, experiences and imaginings of research participants
- How social processes, institutions, discourses or relationships work
- The significance of the meanings that they generate.

Qualitative interviews are especially useful for acquiring a general picture of a subject. Richness, depth, nuance, context, multi-dimensionality and complexity does not hinder qualitative research but instead it enriches its power (Mason 2002). Thus, qualitative interviewing serves as a powerful instrument in holistic research.

Clearly, qualitative interviewing also has its challenges. Criticism that qualitative research and interviewing often receives include, that it is anecdotal, illustrative, descriptive, rigour-less, unsystematic, biased, impossible to replicate and not generalisable (Edwards & Hallond 2013). In order to confront these challenges, qualitative interviewing should be conducted through a comprehensive research etiquette that ensures the trustworthiness and rigorousness of a research.

A thematic analysis is used to report the results of the interview study. Thematic analysis "is used to analyse classifications and present themes (patterns) that relate to the data" (Alhojailan 2012). These shared themes have been acquired with an adequate interview process, where certain beforehand outlined topic have been brought into discussion with mostly all interviewees. From the topics, certain patterns or themes can be interpreted, for example if interviewees believe that a building operating system will exist in the future. Overall, a comprehensive view can be obtained from the whole interview data, categorised into several themes (Aronson 1994).

Alhojailan (2012) argues that thematic analysis is appropriate for data interpretation, deductive and inductive approaches, analysing two different phased of data, and coding and categorising. Furthermore, Braun & Clarke (2006) argue that thematic analysis gives flexibility for qualitative data analysis.

It is important to understand when making a qualitative interview and a thematic analysis, that the researcher should not bluntly give voice to the interviewees (Braun & Clarke 2006). Of course, the researcher needs to guide the interviewee to acquire the desired answers, but the interviewee should have the freedom to answer according to his or her opinion. Thus, interviewer should not shackle interviewee with a loaded or biased question to defend research's hypothesis. Similarly, in the analysis phase, researcher should reflect the interviewees' thoughts as accurately as possible.

5.4 Objective and Implementation

In this study, semi-structured interviews were carried out through individual face-to-face sessions. As this thesis concentrates on the future of digitalised buildings, a semi-structured method was seen as an adequate method to give further insight on the topic. Futurological research often requires subjective answers related to the outlook of the future. With a semi-structured qualitative interview, the topic can be flexibly addressed, and the method gives freedom for interviewees to express their perception of a future.

Furthermore, experts in the building digitalisation were chosen for interviewing. Digitalisation experts have a comprehensive conception of the current situation in technology, research, development and often business. They have considered the future image of the topic before, and therefore can provide answers on how they vision the future of building digitalisation. Experts should be interviewed from many sectors of the industry, including service providers, building owners, researchers and industry innovators.

Discussed topics for the interview study were chosen based on the framework of literature. Questions were compiled in order to test the findings of the literature review and acquire additional information to the research questions. In the interview study, supplementary answers were sought for the main research question and auxiliary questions B, C and D.

In order to compile the interview questions, paradigms which were relevant to the research questions were collected as the basis of the question framework. The concept of an operating system was constructed in the second chapter which was used to define an operating system for the interviewees and give examples of how operating systems have revolutionised the use of computing devices with abstractions, GUIs and APIs. Furthermore, the platform aspect of several operating systems was discussed with the interviewees, for instance going over the success of smartphone OS application marketplaces and how Windows expanded rapidly with the open architecture IBM PC compatibles. Altogether, definition and key success factors of operating systems were laid out for the interviewees in order to manifest ideas on what an operating system could potentially offer for a building and what would it require.

As an opening question, interviewees were asked about their general view on the current situation and future trends of building automation and other building ICT, reflecting it towards the literature reviewed in Chapter 3. Follow-up questions were composed from literature review implications; more closely, question themes were openness and interoperability of systems, platforms, user-centredness, disruption and killer applications. Openness and interoperability were widely discussed in chapters 3 and 4, which were conceived as critical themes for the interview study. As for the theme of user-centredness, user-friendliness was discovered as a key success factor in commercialising operating systems through GUI-based OS and user-centred development has been noted as a paradigm of digitalisation, thus being an important theme to discuss in the interviews. Theme of disruption was reflected to the occasional revolutions of computing history, especially to the sudden success of IBM PC compatibles and iPhone depicted in Chapter 2. Also, the concept of disruption coming from outside of the traditional industry was discussed with the interviewees which is a common theme in industry development discussion. The concept of killer application depicted in Chapter 2 was used as a baseline to gather application and service examples for an operating system and to discuss if a killer application could be developed for a building operating system. Two themes, namely Finland's potential and the expected realisation of a BOS were subsequently added as themes during the interviews and did not directly stem from the literature framework.

As depicted above, theoretical framework acquired from the literature review was used to compile the interview study question framework. Furthermore, theoretical framework is analysed against the answers acquired from the interviewees in order to compare the theoretical and empirical results of the study. In practice, thematic analysis was carried out with question themes based on the theoretical framework. Following chapter goes over the question themes, examining their patterns through a thematic analysis and reflecting them towards the literature.

A total of eight building digitalisation experts were interviewed during the interview study, listed in Table 1. Each interview was planned to last one hour, in practice this alternated from 40 minutes to 1½ hours. Interviews were made in the beginning of November 2017, between two weeks. The time for conducting interviews was short because of coercive restrictions in the thesis schedule, thus regrettably few people could be interviewed. A desirable saturation of interviews was only achieved from the scholar research and development side; the interview study would have desired more interviewees especially from the government and industry actors of built environment, including industry developers from companies and associations, private real estate owners, legislation makers and public sector developers.

Interviewees were chosen mainly from the KIRA-digi project community. All the interviewees have been a part of the KIRA-digi community as a maker or participant of various experimental projects in KIRA-digi and/or they have participated in society events held by KIRA-digi. When choosing the interviewees, certain aspects were considered, including their ability to look the industry from a holistic perspective and how wide they know the building automation, smart building and building digitalisation fields.

KIRA-digi is a government key project which aims to boost the digitalisation of real estate and construction sector in Finland. The project has a total of €8 million government funding between 2016 and 2018, administrated by the Ministry of Environment. KIRA-digi encompasses both the public and private sector of the industry, bringing together private companies, associations, ministries, government agencies and municipalities to develop the whole industry. Organisations involved in KIRA-digi include Ministry of Environment, Ministry of Agriculture and Forestry, Ministry of Finance, Association of Finnish Local and Regional Authorities, Building Information Foundation, Confederation of Finnish Construction Industries RT, RAKLI and Finnish Association of Building Services Industries. Major part of KIRA-digi are the experimental project funding rounds which enable a maximum of 40 % government funding for fast and concrete experiments. KIRA-digi also includes data management harmonisation, public sector development and legislation projects. Vision of KIRA-digi is to create an open and interoperable information management ecosystem for the built environment.

All of the interviews were recorded with permission. Interviewees were first asked about their background, after which this thesis was explained briefly and plainly to give a basic understanding of its content and purpose. Rest of the interview composed of question themes based on the research questions and literature review, acquiring insights of current building automation and IoT markets, openness of systems and data, and user-centredness. Furthermore, future-based questions were asked about the BOS platform, including its prospects, ownership, disruptiveness, potential services and applications, Finland's opportunities in creating it, and will the platform be created and when.

Interviewee	Position	Project or Department	Organisation
Tomi Teikko	Director	Intelligent Building	Tieto
Rita Lavikka	Postdoctoral Researcher	Digital Disruption of Industry	Aalto University
Jaakko Ketomäki	Professor of Practice	Smart Buildings and Services	Aalto University
Heikki Ihasalo	Professor of Practice	Smart Buildings and Services	Aalto University
Teemu Lehtinen	Chief Digital Officer	KIRA-digi	Building Information Foundation RTS
Esa Halmetoja	Senior Expert	Digitalisation of Facility Management	Senate Properties
Tero Järvinen	Technology Director	Innovation and Development	Granlund
Marko Tiitinen	Consulting Director	Building Industry	CGI

Table 1. Study Interviewees.

6 Results

Results of the interview study are laid out through a thematic analysis. Several patterns were observed in the interview data which are divided under their respective subchapters. These patterns mainly follow the question themes described in the last chapter which in turn are principally based on the theoretical framework of this thesis. Open data subchapter does not belong to the predetermined topic themes, as it was subsequently brought into the discussion.

Last subchapter reflects the results of the interview study against the literature review, inspecting congruencies and possible deviations between them, and provides a summary of results from the entire thesis. Discussion and suggestions are laid out in the following chapter.

6.1 Interoperability: Open or Closed System

Currently, building automation systems from different vendors are generally seen as closed. Different systems are often advertised as open standard compatible, whereupon they can be used with ZigBee, BACnet and/or other common open standards. However, the systems often have own private technology used on top of the open standard infrastructure, and these add-ons are usually made with proprietary solutions. Thus, many BA systems use open standards, but on top of them they use proprietary add-ons and protocols that hinder integrations with other systems. One interviewee mentioned, that they always need to distinctly demand that the BA system they order is open. Demanding an open system costs more, but the interviewee said that it is worth the extra cost, because it helps integration with other information systems.

Virtually all interviewees demanded for more open systems in the future. Issues mentioned in the last paragraph were described by interviewees who have had personal experience with using or ordering building automation systems; and they also demanded for open systems the most. As for interviewees who did not have specific experience of BA systems, they also demanded for more open systems based on their general knowledge of the industry. Benefits of open systems included improved integration and interoperability between different BA systems. BA experts especially demanded that systems should be more interoperable, which could be accomplished with more open systems being able to communicate better with one another. Moreover, open systems would help software developers to make use of the systems, making system management and data input & output easier. Generally, opening systems were seen as a possibility for more efficient data communication between BA systems and also between other information systems. APIs were seen as a solution to improve system communication.

Closed systems were also seen as a problem, where similar or equal data was collected by separate building automation or IoT systems. Having a separate system with security operations can be argued, but otherwise collecting same data multiple times is a waste of resources. There should be a solution to connect all overlapping systems, thus building data would not be collected concurrently in siloed systems. Furthermore, when data would be interchangeable between systems, it could be analysed and utilised more efficiently. One interviewee told an interesting metaphor where he compared current closed BA systems to grocery stores in the late Soviet Union. In the Soviet Union, meat, milk and vegetables were all bought from separate stores, which resembles the separateness of current BA systems.

As open systems, open ecosystems were seen as the future. One interviewee believed that traditional closed and siloed ecosystems will perish in the future. Open ecosystems, including open BA systems will prosper and gain business advantages over traditional closed ecosystems. However, interviewee also believed that ecosystems will never be completely open, that there will always be some business core kept closed. For example, Apple has an open ecosystem in a sense with its iOS, but entering the ecosystem requires that developers and users abide to the rigorous guidelines that the platform has. Nevertheless, iOS platform is virtually open for any software developer, yet following the strict guidelines is necessary.

6.2 Open Data

Discussion about open data with interviewees was varied. Mostly, open data was seen as a good thing. Open data can bring better and preference-based services and it helps collecting big data which can be used for machine learning and analysis.

Although open data was principally seen as a benefit, some of the interviewees brought up issues with openness of personal information. These discussions often referenced to *GDPR*, a EU regulation about individual data protection, and *MyData*, a set principles on how to manage, utilise and release personal information. Interviewees described that personal information used openly should not be able to be coupled with an individual, i.e., personal information should be adequately anonymised. GDPR and MyData proposes certain norms and guidelines on how the rights of personal information belong to individuals and how their information should be protected more efficiently.

Concerning personal information, individuals should have rights to define what information about them can be collected and utilised. Two of the interviewees were especially concerned about individuals' rights when the data is used for business purposes. Currently, IT giants like Google and Apple for example have leverage on this issue with smartphones. They can essentially dictate the terms of what information they can collect from their smartphone users and how they can utilise the information within their business. If users do not accept the terms, many of the essential services and applications will be blocked of use. Also, one interviewee depicted current personal data usage as a Wild West, where personal information can be used to all kinds of purposes and in various places without the individual knowing.

An interviewee brought up an interesting example of personal data usage for business purposes, where a large corporation uses a large amount of personal information to train its artificial intelligence. This large corporation also uses Finnish personal information for AI training, but in what extent does Finnish individuals know that their information is used for this purpose, and do they receive any benefits for allowing a business use their personal information.

In the government sector, opening data is a major challenge. Senate Properties which owns most of the government buildings in Finland does not have the right to open any data concerning their customers without a written permission. Some basic information can be shared openly, for example visitor rates and indoor climate data. However, for example the amount of personnel working in the tax office at a certain time cannot be released publicly. Also, army building data is naturally classified and cannot be shared openly. Interestingly, the openness legislation in Finland principally announces all government activities as open, but this openness is restricted from many angles. Overall, government building data is openly shared whenever it is not classified or restricted, it does not threaten national security and it does not conflict with restrictions specified by personal data legislation.

Considering open data in buildings, one interviewee talked about how real estate owners should be convinced to open their building data. They have access to most of the non-personal data from buildings, which could be opened rather easily. In addition to real estate owners, government initiatives and regulative encouragements should be put into practice to open data as much as possible.

Overall, open data was seen as an advantage in creating better services for buildings, their users, owners and maintainers. Open data makes open ecosystems possible, where data can be shared easily and without restrictions between businesses and organisations that want to utilise building data. Furthermore, combining easily accessible data brings opportunities for more and better services, applications and solutions. As for the need of restricting data, opening all data that can be securely opened was seen as a good solution. Also, anonymising personal information was required for open data purposes; coupling openly available data to an individual should not be possible or allowed. One interviewee mentioned that all data that you can sense, observe and investigate in your environment without special permissions or accesses should be open.

6.3 Who Owns the Platform

Ownership of a potential building operating system platform was discussed with all the interviewees. In the discussion, private platforms such as Android, iOS and Amazon were brought up and the question was if a building OS could be developed as a non-profit and open platform. For example, could building OS be a platform such as Linux or other open source projects, where a wide and global developer community builds the platform non-profit together according to open source licenses.

All interviewees saw potential in an independent, open and non-profit platform. This arguably correlates with how the interviewees also sought for more open systems addressed above. Some of the interviewees said that their organisations did not have the resources or incentive to start creating a building OS platform which would connect all the automation and IoT in a building. Moreover, a non-privatised platform was seen as an opportunity where platform owner's interests would not conflict with the business interests of application or service providers using the platform.

A couple of the interviewees brought up Tilaajavastuu's KIRA Platform of Trust as an example of a non-profit platform for buildings. The platform is in development, and its mission is to connect building information with all actors of the real estate and construction sector. It could also provide a platform for transferring data from a building OS. One interviewee mentioned, that such a non-private platform has not been attempted to build before. Some government-led projects have been in Singapore and China, and also in Finland there is a Location Information Platform (fin. *Paikkatietoalusta*) being developed by the Ministry of Agriculture and Forestry. One interviewee mentioned, that a neutral platform could possibly be created with blockchain technology.

One interviewee saw that the core infrastructure of a platform could be owned by a neutral and non-profit party. On top of the core infrastructure would be the service layer, where competition between service and application providers would take place. Yet in the end, consumer or user does not care who owns the platform, as long as (s)he can utilise the services and applications that (s)he wants.

One issue which was discussed in a neutral platform was, if it could lure a sufficient number of users. If there is no business incentive, is there any incentive to gather a critical mass of users and grow the platform nationally and globally? Another interviewee, who offers applications and services for buildings, saw neutral platform as a better option, because then there would be no conflict between business interests.

As was mentioned earlier, building automation and IoT markets are dispersed and heterogeneous. Some of the interviewees were pondering who would tackle the entirety and connect all the fragmented automation and IoT technology into a single platform. One interviewee was contemplating about who would own the building OS in a single building; building owner, resident, municipality, or someone else? Also, who will pay for its installation and maintenance?

6.4 Disruption from the Outside

Over half of the interviewees believed that built environment will be disrupted by someone outside of the traditional industry. These discussions addressed the building industry in general, and also it was discussed if a building operating system will be developed by a vendor outside of the existing industry of built environment and building automation.

Most of the interviewees believed that the current actors in the building and automation industry will not be able to disrupt the current markets with digital technology. Generally, building industry was described as a traditional and conservative industry where productivity has not grown in decades. Typically, it has been difficult for new actors to enter the industry. Also, building industry invests little in research and development compared to other industries. One interviewee argued that the building industry does not see its evolvement necessary because it makes same profit in any case, whether they invest in development or not. Furthermore, a few interviewees believed that the industry essentially needs a disruption from outside, until they realise that they need to change their operating and business models for the emerging digitalised world.

Interviewees who believed that the disruption will come from the outside depicted large platform enterprises such as Google, Amazon, IBM, Samsung and Apple to be the perpetrators of the presumed upcoming disruption. One interviewee said that Google and Apple have already started disrupting the smart home market with Apple HomeKit and Google Home. Earlier, traditional building automation firms like Honeywell were building their own products in the smart home market, but recently global IT-based conglomerates have seized the markets. One interviewee believed that the disruption will happen very soon, only in a couple of years. Especially if the global conglomerates believe in the business potential of building industry digitalisation, they will enter the markets sooner or later. Two of the interviewees were concerned how the Finnish RECS companies will be able to compete in their own market if a global company comes and takes over the industry. Will a global company dictate how the industry operates in the future if they conquer the market with a highly successful platform?

One interviewee believed that actors from outside of the industry will not be able to bring disruption. He argued that the industry is too conservative and there are too many factors that outsiders need to take into account when entering the industry. However, interviewee hoped that someone would come from the outside and *crack the game*. If a disruptor would do the same business more efficiently and with substantially lower costs, it would force traditional actors to change their operations likewise.

One interviewee believed that the disruption will need a collaboration of both the old industry and an actor from outside. He believed, that the industry alone cannot disrupt itself, but it needs an aide from the outside to succeed in disruption. Also, actors from the outside have tried to disrupt the industry alone already, yet they have not succeeded in it. Thus, disruption requires actors from inside the industry who are ready for a change, but also actors from the outside who can aid with technology and services for example. Additionally, another interviewee mentioned that the current built environment industry needs to accept the supposed upcoming disruption and believed that the current industry cannot accomplish the revolution alone.

6.5 Killer App: BOS Services and Applications

Several examples of potential building operating system services and applications were suggested by the interviewees. When asking about examples for potential services and applications, concept of killer app was explained to the interviewees. Majority of interviewees listed potential use cases for homes, but some focused more on offices, shopping centres, warehouses and public buildings. Overall, a wide scope of different applications, services and use cases were collected from the interviewees.

Most of the examples gathered from the interviewees were focused in residential buildings and homes. Energy management, including smart thermostats were popular examples. With smart energy management, residents and owners can accomplish energy and money savings, for example if you lower your home’s temperature at daytime when it is unoccupied. Other examples for home included indoor climate management, smart locks and chore easement. Chore easement examples included automatic refrigerator management, where the refrigerator tracks its grocery consumption and automatically orders more, for example when the milk has run out. A building application could also track when a household needs cleaning and order a cleaner (robot or a human). These could also be applied to other chores such as cooking, washing laundry and washing dishes. Also, sauna monitoring and management could also be done with an adequate building application, which tracks if the sauna is warm, or if it is overheated and there is a risk of fire.

Dampness and indoor climate monitoring was seen as a useful application for all types of buildings, including residential buildings. Currently, mould and dampness in public buildings, especially in schools have been a major debate in politics and media. With dampness and structural sensors, buildings could track if there is dampness or mould in the structures, insulation structures or the roof are leaking, air pressure is at an appropriate level or a bath room is not drying properly.

As for office buildings, many examples were also introduced. Interviewee from Tieto told about their own application for office buildings already in deployment, which monitors and manages office spaces. With the application, you can find a free workstation, find your colleagues, find an available meeting room, select workspace based on air quality, know noise levels, share your voice and ideas, create service requests and report problems, and visualise activities. Other examples for office included access control and parking place discovery. Smart access control would automatically recognise if a person can enter the building, opening the door if the person is authenticated to enter.

Shared office spaces were also discussed with a few interviewees. Today and in the future, working culture is changing and more people will do work remotely from the office headquarters. Some remote offices have already been piloted, where people can use their nearby local remote offices to work. With a widely-used office usage monitoring application, these remote offices could be monitored and you could check and reserve a workspace whenever you want to go to a local remote office.

Some examples of services and applications for warehouses, shopping centres, hospitals and retail stores were also shared. Warehouses need to be accessed more easily and you do not have to rent a fixed size room; you would pay only for the size of the items you are storing and the storage item management would be handled by robots.

Shopping centres are evolving ever more into amusement centres where people come to shop, eat, drink coffee, work, see a movie, and have a pleasant time. In shopping centres, people could use building-based applications to reserve spots in restaurants, cafes and movie theatres, find services and goods, et cetera. Smart and functional indoor navigation was one application discussed with a few interviewees, which was seen as a key application especially in large buildings. Also, one interviewee mentioned how important Wi-Fi availability is today, and how building application could reveal free Wi-Fi spots available inside a shopping centre for example.

In hospitals, people would not need to wait for their appointment close to the doctor's office, but all the patients could wait in cafeterias or entertainment rooms until they are informed through a building application when the doctor is ready to take them in. Thus, traditional waiting spaces would become redundant in hospitals and patients would be summoned to a doctor or a nurse remotely.

One interviewee believed that working, living and mobility will become one contiguous entity in the future. Each of these will be offered *as a Service*, where you can decide your service level for each element. If you want to spend less money on living, you can decide a lower service level for living and live in a smaller apartment farther from the city centre. Furthermore, if you demand faster and extensive travelling in the city and abroad, you can have a higher service level on mobility, which can include many fast transportation services such as taxi, airplane flights and a private car.

One interviewee described an interesting example on how a lamp maintenance could be carried out with a smart building operating system. When a light bulb burns out, the system notices this autonomously and starts a bidding competition on nearby maintenance firms, trying to find the cheapest and most suitable firm for the light bulb change. System automatically gives out information about what kind of lamp blew out so the maintenance knows which kind of lamp they need to order and replace. This would save a lot of time and money, when the maintenance person does not have to inspect the lamp personally, but the building system would lay out all the essential information about the maintenance procedure and also perform a bidding competition of the task with nearby maintenance firms.

Two of the interviewees emphasised how important user interface is for creating applications and services for buildings. With an adequate UI, users can investigate if lights are on or if there is a colleague present. Through a remote UI, buildings can be controlled remotely, for example if a whole floor is overheating, you can increase air ventilation or configure thermostats remotely. Furthermore, digital twin concept was brought up into discussion. Digital twin could work as an interactive user interface that depicts the building in a digital form, showing all the digital data available from a building, including temperature, space utilisation, people tracking, indoor climate et cetera. Digital twin could also be bidirectional, through which users and building managers could control building conditions. Via UI, building can be monitored and piloted.

Platform aspect of services and applications were also discussed with a few interviewees. There are many applications and services available already for buildings, but they are not universal in a way that they could be easily duplicated to other buildings, i.e., they are tailor-made for a specific building only. A building operating system might be able to answer this problem, enhancing interoperability of building applications and services amongst buildings. Thus, an application developed for a specific building operating system would in principle work with all buildings with the same or compatible building operating system. There are many factors to be considered with compatibility, knowing that buildings are widely heterogeneous and they comprise of various technology of varying age. However, building-independent technologies have already been introduced in the smart home industry, including Amazon Alexa which you can bring to any home with an electric outlet. Also, a computer vision based application has been developed in Finland to monitor water consumption, making it easily expandable to any building regardless of what automation systems or technical building services the building already has.

6.6 User-centredness

All the interviewees stood for user-centredness. Similarly, virtually all interviewees criticised how user-unfriendly current automation systems and building services are in general. Companies that were interviewed supported user-centred development and took end users into account when designing and creating products and services for customers. Overall, user-centredness was believed to make better services and products for the end users, making buildings more comfortable.

Current building automation and IoT systems were accused of being made only for buildings and engineers. They are complicated to use, they require expertise to be operated and the user interface seldom is visually pleasant. End users are not taken into account when designing these systems, thus user-centredness is missing. Also, buildings usually offer only the minimum according to standards and protocols, for example buildings temperatures are typically set to 21 °C, although building users might desire warmer or colder temperatures. In these situations, end users rarely have a proper interface to change indoor climate, perhaps only by opening a door or a window or trying to adjust the thermostat. One interviewee mentioned, that 38 % of building users are satisfied in indoor air conditions. End users should have possibility to give feedback on indoor climate and other building conditions and to control them according to their preferences.

On the other hand, many of the interviewees saw that the user should not be required to control building systems, but they would autonomously adapt to users' needs and preferences. At first, user might need to give feedback on what kind of environment he or she prefers, and in due time the system would learn and automatically offer comfortable indoor climate, temperature, services et cetera. This also applies to access control, where the visitor would not need to manually unlock the door, but rather the building would automatically open the door. One interviewee emphasised how end users merely want buildings to function properly, so that they do not encounter any problems when interacting with the building. Buildings would not need to be full of gadgets and cool settings, but that they would work properly and unobtrusively.

One interviewee thought that different users should have different user interfaces for the building. For an automation engineer, current systems are sufficient and they can operate them adequately. However, for a common man using current systems can be rather difficult and they require simpler user interfaces. As for real estate expert, he or she might know something about the building automation, thus being able to use a more complicated UI than the common man, but requiring a simpler UI than the automation engineer. Thus, separate UIs should be available to separate users depending on their needs and the level of expertise.

The needs of end users were a topic of discussion with a few interviewees. More closely, how does end users know what kind of systems or user interfaces they look for? How can end users demand for better building systems, services and applications if they are not aware of anything better? One interviewee mentioned that building certificates which take end users into consideration could drive the issue. Also, if a killer app would emerge for buildings which would substantially ease their life or lower their costs of living, they would start insisting for it.

One interviewee talked more philosophically about why buildings are built and what are their purpose. Essentially, buildings are made to provide a pleasant environment to act, work, sleep and eat without being at the mercy of mother nature. Thus, buildings offer easements and comfort to people’s lives. Furthermore, user-centredness is driven by the demands of people.

6.7 Finland’s Potential in Creating BOS

Recently, Finland has been keenly interested in platform economy, wanting to create a successful and global platform for Finland’s and Europe’s benefits and to compete with the number one platform possessor, the United States (Viitanen et al. 2017; Ailisto et al. 2016). Regarding these interests, all the interviewees were asked about Finland’s potential to create a building operating system platform, which would connect automation and IoT inside a building. Furthermore, the platform would require accumulating a critical mass of buildings, which would mass into a network of connected smart buildings, building up a smart city.

All of the interviewees believed that Finland could succeed in creating a BOS platform. Some were more enthusiastic about Finland’s opportunities, others were more sceptical but did not reject the idea. Especially, Finland’s engineering know-how was noted, often referencing to the partial collapse of Nokia, and how skilful engineers leaving Nokia have started their own start-ups. Other strengths of Finland that were mentioned included success stories of built environment oriented software companies such as Tekla and Solibri; talent in building information modelling and general ability to build integrations.

A couple of weaknesses were also mentioned. Primarily, these concentrated on the lack of funding and commercial skills. Funding is rather scarce in Finland, and trying to compete with China or the US for example can be difficult. Funding in Finland can be a fraction compared to large projects in the US or China. Although recently, plenty of funding have been allocated to platform economy, industrial internet and IoT especially by the Finnish technology and innovation funding agency Tekes which has improved the funding situation in Finland. One interviewee said that it is not the question of know-how; what Finland needs is major investments, good designing and comprehensive planning. The competitor can be a large global multi-billion-dollar conglomerate such as Apple, Google or Amazon.

General marketing skills of Finland were also criticised. Some of the interviewees believed that Finland could make products with same or even better quality than many other countries but in the end, Finland would lose in marketing. Finnish people would need to believe more in their products, marketing them more aggressively and stop underestimating their potential. Finns should have more faith in their prospect.

One interviewee mentioned Linux and contemplated if a BOS platform could be built similarly, accumulating a large and open developer community globally. Thus, BOS would become a global open source project, essentially being a neutral platform.

One interviewee especially believed that Finland needs support from other countries also. The platform cannot be built only by Finnish, and it must be done internationally with cooperation of all the biggest actors in the industry. Perhaps an international consortium driving BOS could create the neutral platform. Nonetheless, Finland can be a trendsetter and lead the way in the process.

6.8 When Will It Happen?

Some discussion was also had on when BOS will be created commercially and how will it happen. One interviewee believed that a prototype of BOS will come in merely a year. Another interviewee mentioned that such systems have already been built in the US, mentioning Lucid BuildingOS and Tridium Niagara. A third interviewee talked about the possibility of the system being built on top of Linux, and how it would gradually expand through experiments and pilots to neighbourhoods, cities, countries and finally worldwide, taking years or even decades to reach a critical mass. A fourth interviewee mentioned that the operating system will come in at least three years, mentioning that there are commercial prototypes already tested in smart home industry especially.

Two of the interviewees specifically talked about how the creation of BOS needs concrete actions. This idea has been discussed for a while already and its benefits can already be depicted, but now what is needed is experimentation and people engagement. If end users would become intrigued of BOS, their influence could substantially accelerate its development and diffusion. Also, government incentives and possible obligations would accelerate the process.

Link between killer apps and the platform was discussed with one interviewee. Perhaps an actor who will create the killer app will also build a platform on top of it. This has already happened with some companies which have created popular applications or services, later which a platform has been built on top of these.

6.9 Summary and Reflections to Literature

In the literature review, evolution of computers, operating systems, telecommunication, mobile computing devices and the internet was reviewed. Additionally, overview and evolvement of standards in telecommunications and electronic authentication was explained. Finally, overview of building automation systems and internet of things was covered.

Interoperability and integration of systems have been critical features in the evolution of ICT. In history, computers were hardwired and they did not have software nor operating systems. Each computer was a unique system, only being able to operate specific tasks without a possibility of reprogramming. Later, computers became general-purpose machines, ergo they enabled software and they were reprogrammable to various computing tasks. Especially after the Second World War, computers started to become more common, evolving from massive mainframes into minicomputers, personal computers and more recently into smartphones.

Similarly, various enterprise systems including enterprise resource planning systems have strived for interoperability and integration within an enterprise. ERPs and more widely enterprise systems have tried to enable information flow through the whole enterprise, integrating different functional areas of business for easier and more streamlined enterprise management. Modern ERPs replaced the old legacy enterprise systems in the turn of the millennium, providing a single backbone to all core areas of business, thus enabling better integrability between previously siloed information systems. ERP systems of today have notoriously long implementation times and their deployment can cost millions of dollars. Several future trends are also affecting ERP markets, including service-oriented architecture, cloud-based solutions and subscription-based pricing. Reflecting ERPs to building operating system,

Various standards and protocols have been introduced in the past for ICT systems and machines to enable their interoperability between different segments. These standards have evolved in two main ways: by conquering the market and being accepted publicly (*de facto*), or by governmental and legislative endorsement (*de jure*). A famous example of de facto standard is the IBM PC, consisting mainly of Intel microprocessor, Microsoft operating system and various other hardware. Unintentionally, IBM PC became an open architecture which was copied by various manufacturers, generating a myriad of inexpensive IBM PC compatibles. IBM PC quickly seized the market in the 1980s and 1990s, becoming a de facto standard of PC machines. As for de jure standards, major telecommunication standards such as GSM, UMTS and LTE have originated from de jure standardisation. More closely, GSM started as an initiative of Nordic countries and the Netherlands to build an Intereuropean telecommunication standard. Later, other European countries joined the consortium, and eventually European Telecommunications Standards Institute was established specially to continue GSM's development and expansion as a governmental standardisation scheme. In the end, GSM became both de facto and de jure standard worldwide. GSM's successors UMTS and LTE have continued similar legacy of de jure based standards.

Standards and protocols within building automation have become more open recently. Traditionally, building automation standards have been proprietary, only encompassing systems and devices of sole manufacturer. Today, BACnet, KNX, LonWorks and Zigbee are widely used open standards in building automation. Moreover, IoT systems in smart buildings uses several open telecommunication standards, including LTE, Wi-Fi, Bluetooth, NFC and IP.

Reflecting to literature, building automation systems are not as open as could be presumed. Based on the interviews, BAS use open standards such as BACnet but on top of the open standard there are typically proprietary protocols used which hinder interoperability and integration. This conclusion correlates with the literature review. Closed system dilemma also exists in the computer industry; closed systems were commonplace before IBM PC compatibles, although open systems have become favoured in the past three decades, including Linux and Android. Yet, Apple's devices and operating systems have mostly retained their closed nature. Nevertheless, ICT systems have overall become more open, although the level of openness varies. Internet browsers have also followed the open approach, including Google Chrome and Mozilla Firefox. After IBM PC, hardware architecture has become mainly open with PCs and smartphones, consisting of components from various hardware manufacturers.

In the past, platforms in the ICT industry have been privately owned. These include most of the operating systems, namely Android, iOS, macOS and Windows. These platforms are owned privately, but their level of openness varies. Android and iOS are open platforms in a sense because of their application marketplaces, enabling virtually any software developer to create their own applications to an Android or iOS device. However, software developers need to pay a fee to use the official marketplaces and there are guidelines for what kind of software is allowed. Android allows third-party application marketplaces, whereas iOS only allows the official Apple App Store. Windows for PCs work similarly, serving as a platform for various software of various developers. Windows does not have an exclusive application store similarly to Android. The only major purely open operating system is Linux, which is open source, it is developed by a large open community and it is not owned by a single corporation.

Discussing platform ownership in the interviews, many of them saw benefits in a neutral, non-private building operating system platform. Especially, there would be no business conflicts between the platform owner and the application or service provider. Therefore, core of the platform should be competition neutral, only serving as an infrastructure to facilitate efficient competition on top of it. Reportedly, neutral platforms have begun development in Finland, including Suomen Tilaajavastuu's KIRA Platform of Trust and Ministry of Agriculture and Forestry's Location Information Platform. It remains to be seen if a neutral platform will succeed in the global platform economy.

As for disruption from the outside, ICT industry has been a melting pot in this matter. Overall, ICT industry is still a young industry with many young companies. This is mostly the fact, since most of the major ICT conglomerates today did not exist before the Second World War. Nevertheless, ICT industry has had several disruptions during its brief existence, namely IBM PC, internet and smartphone revolution. Both revolutions actuated many start-ups which have later become multibillion companies, including Google, Amazon, Microsoft, Compaq, Intel, Facebook and Snap.

Computer industry has emerged several killer applications during its existence, including spreadsheet application for Apple II, GUI operating systems for PC and general internet browser for smartphones. Killer applications have had a clear impact on the success of computing devices that have incorporated such applications. As for discussing killer applications with the interviewees, they described several examples of applications and services that could be provided through the system. Some believed that a killer application would eventually revolutionise the markets, others believed that a single application or a service could not trigger a sudden revolution and that the transition would happen gradually. Potential applications and services were assumed to be user friendly with pleasant user interfaces that would provide ease and comfort to living and building operation. Interoperability was also included in the discussion, so that the applications and services could be used in any building having a same or compatible operating system. To create sincerely successful killer applications, such network effects would be required.

User-centredness has been a major accelerator of computer consumerism. Especially after the emerge of GUI and other enhanced user interfaces, computers have become ever more mainstream and ubiquitous. Furthermore, user experiences have become more prioritised in order to gain attraction and competitive edge. Considering building automation and smart buildings, user-centredness is still lacking, as was discovered both in the literature review and interview study. Current building automation systems are designed for engineers and buildings, not for end users. Smart home development has already brought some consumer-friendly products, which already has allegedly revolutionised the home automation industry. Overall, user-centredness has started changing the business and design models of building automation and smart building industry, urging for better UIs and UXs.

In addition to easier and more intuitive use of building automation and IoT, autonomous systems should also be taken into account in user-centredness. Building systems should be able to adapt automatically to users' needs and should detect if a room is too cold or warm for example. Furthermore, building owners and end users should be enlightened about the possibilities of better smart building systems, so that they could demand and insist for better serviceability, usability and also openness from the systems.

Interviewees widely believed that Finland could be the pioneer of building operating systems. Finland has talent especially in the engineering side in both telecommunications and built environment technology with companies such as Nokia, Kone, Solibri and Tekla. Finland is also believed to lack some essential features, notably marketing and funding. Finnish were thought to be too humble, lacking the courage or edge to market and promote their products more flagrantly. Furthermore, a small country like Finland does not have the same investment potential as USA for example, not being able to have grandiose funding in development projects. It was also explicitly mentioned in the interviewees that Finland cannot succeed alone, but it needs to collaborate extensively with others.

Considering past successes of Finland from the literature review, Finland was a forerunner in the Intereuropean telecommunication standardisation scheme. Beginning with Nordic NMT analogue cellular phone system Nordic countries enabled inter-Nordic telecommunication with open standards. NMT's open standards enabled more competition between device manufacturers, accelerating development and pushing the prices down. Later, NMT's successful legacy in the 1G era contributed to the development of 2G, bringing forth the de jure Intereuropean telecommunication standard GSM, which later expanded globally as the universal standard. Massive success of GSM aided its supporters amongst device manufacturing and service providing, including Nokia and Ericsson. Evidently, Nokia continued to be the forerunner in mobile telecommunication industry and later became the largest mobile phone manufacturer in the world until its gradual collapse since the end of the 2000s. Nokia no longer manufactures mobile devices, although the company still contributes extensively in the telecommunication technology, driving 4G and 5G technologies of the future.

7 Discussion and Suggestions

The purpose of this thesis is to provide holistic requirements for a building operating system. Thesis was carried out through a literature review and an interview study. Literature review built up the theoretical framework for the study and the interview study provided empirical results to complement and revise the theoretical framework. In addition to the main research question, four auxiliary research questions are laid out to aid the main research question and steer the research. Research questions are the following:

I. **What are the holistic requirements for a building operating system?**

A. What is an operating system, what are their purpose and how have they evolved?
B. What ICT systems are currently used in buildings?
C. What kind of development does the contemporary building ICT require in order to create a building operating system?
D. How can interoperability be achieved with a building operating system?

This chapter will go over the research questions, providing answers for them and discussing the thesis in its entirety. First, auxiliary questions starting from A are reviewed and finally the main research question is answered.

Auxiliary question A is mainly tested through the literature review of computing and operating system evolution. In the literature review, the definition of an operating system was understood in a holistic context and its purpose and evolution was described in depth. Question A aids the main research question by giving context to operating systems and providing implications on what an operating system would stand for in a building.

Operating system is essentially a system software which enables applications to interact with the computer's hardware. Operating systems provide convenient abstraction of hardware for software developers to easily exploit components of a computer through application programming interfaces. Thus, an operating system essentially operates the computer system, helping it to allocate resources of hardware and software, permitting user applications to use hardware. In a broader sense, the purpose of an operating system is to manage system devices and offer their resources to applications.

Operating systems have had a rich development over the past century. Along with the evolution of computing devices, where devices have shrunk from room-sized mainframes into handheld smartphones and computers have become reprogrammable general-machines, operating systems have also developed substantially. Originally, computers did not use operating systems since the applications were such rudimentary that programmers handled hardware resources by themselves in the program code. Later, first operating systems started to manage application execution in batches of multiple programming jobs. Later, multiprogramming was introduced which enabled parallel use of hardware resources, for example one job used the processor while another job used the input and output devices. In the 1970s, UNIX was introduced which was the first portable operating system, programmed in high-level language C. UNIX uses a simple consistent interface to peripheral devices, it enables multiprocessing and multiple users and it abstracts the machine architecture from the user. UNIX has been praised by its simplicity and consistency.

Major wave of operating systems began with the commercialisation of computers. Initially, command-line interfaces were the only option, but soon during the 1980s graphical user interfaces began revolutionising the markets. Apple was a major forerunner in GUI-based operating systems, aiming for user-friendliness. In the 1980s, IBM PC and its clones began dominating the PC markets which led to the success of Microsoft which was the main provider for IBM PC compatibles with their operating system DOS and various office applications. Soon, pursuing to compete with the superior GUI-based OS of Apple, Microsoft began developing its own GUI-enabled operating system Windows. At the same time, IBM was developing their GUI-based OS/2 which ultimately failed to compete against Windows on the IBM PC compatibles. Ever more user-friendly operating systems were created in the onset of computer commercialisation in order to captivate the consumers and win over the markets.

Ultimately, Windows took over the operating system market of PCs and has continued to dominate the markets until this day. In the 1990s, Linux was introduced as the first major free and open-software operating system. Furthermore, Linux is not owned by any single entity or company but it is developed by a large open developer community with thousands of developers. Linux has not succeeded to capture the PC markets, although it is widely used in servers and supercomputers, and is highly popular among computer scientists and hobbyists.

Operating systems have also been a crucial part of the development of smartphones. Mobile phone operating systems began to shape in the 2000s. Before, operating systems of mobile devices were simple hard-coded systems which were not portable to other devices. First major operating system, Symbian was launched in 2002 for the Nokia 7650 and it began to dominate the markets until 2008. Symbian suffered from serious deficiencies, mainly in application development which arguably led to its fall after iOS and Android was released. Overall, Symbian was difficult for developing applications for its cumbersome and inadequate software development kits.

In 2007, iPhone was released which essentially brought the concept of smartphone into mainstream, revolutionising mobile phones into pocket computers with calendar, e-mail, games and the internet. iPhone included a large touchscreen for video viewing and a universal web browser for browsing the entirety of the Web. The first iPhone was highly appraised for its user-friendliness and product design and it amassed a desirable number of users. In 2008, iPhone allowed third-party software on iPhone and launched the App Store for iPhone 3G which began the diffusion of third-party software on smartphones. App Store became a major platform for software developers and it quickly amassed thousands of applications and millions of downloads.

iOS also had two major competitors, Symbian and the recently released Android. Symbian continued to struggle with its SDK deficiencies and did not manage to build an app centric ecosystem as iOS and Android, thus losing the markets and gradually "burning" away. Android became a highly successful platform and currently dominates the mobile operating system markets. Compared to iOS, Android is an open platform developed by Google which uses a modified Linux kernel and is usable virtually in any mobile device, whereas iOS is a closed platform only usable in Apple devices. Nevertheless, both have succeeded to build up extensive application marketplaces with vibrant developer communities on their platforms, thus a key success factor for both have been the ability to mobilise software developers with serviceable SDKs and APIs.

Auxiliary questions B and C are answered with the aid of both literature review and interview study. In the literature review, different building ICT systems and their future trends were reviewed in Chapter 3 which were complemented with the interviewee discussions. Questions B and C aid the main research question by providing an overview of contemporary and prospective building ICT systems, including building automation systems, IoT systems and building operating system prototypes.

Building automation has been and still is the cornerstone of building ICT. Building automation systems control and monitor mechanical and electrical equipment inside a building, including HVAC, security, fire, power, water supply and elevator systems. Building automation’s general purpose is to increase user comfort at a reduced operation cost. Generally, building automation systems enable the utilisation of various sensors and actuators in a building in order to manage air ventilation, heating or lighting for example.

Building automation industry has continued to be a fairly closed ecosystem consisting of a few major vendors and manufacturers. Recently, open standards and protocols have emerged to enhance interoperability and integration of systems, including BACnet, LonWorks, KNX, Modbus, DALI and several web services, and they are utilised in a variety of BA systems available on the market. However, vendors and manufacturers often have proprietary add-ons on top of the open system, hindering interoperability and integration with other systems.

Furthermore, IoT and smart building products are already disrupting the building automation industry. Traditional wired and stationary solutions used in building automation seem to be reaching their end of the road, and perhaps soon they will be replaced with wireless and plug-and-play solutions of IoT-enabled devices. Furthermore, it remains to be seen if the whole building services engineering industry will be disrupted by modular plug-and-play solutions. Nevertheless, IoT and smart building industry is still attempting to resolve issues of compatibility between heterogenous devices. Other challenges for IoT include availability, reliability, mobility, performance, management, scalability and security & privacy (Al-Fuqaha et al. 2015).

Smart home industry is already disrupting the traditional home automation industry, as can be observed in market outlook and interviews. Google Home, Amazon Echo, Apple HomeKit and other branded smart home products have begun sweeping the smart home market from traditional vendors. Perhaps the whole building automation industry will be disrupted in a similar manner by global ICT conglomerates with user friendly and consumer-oriented solutions; or by a future unicorn start-up.

Auxiliary question D is answered mainly by the literature review of Chapter 4, although the topic of interoperability has been discussed throughout the thesis in both literature review and interview study. Question D provides answers to how interoperability could be achieved with a building operating system, pursuing to open the currently siloed systems in building automation.

Telecommunication standards have undergone an intriguing evolution towards interoperability, requiring heterogeneous devices to communicate wirelessly and in harmony. Telecommunication technology has evolved from quasi-vertical standardisation and integral problem solving into open standardisation and modular problem solving (Funk 2009). Traditionally, regulated monopolies determined standards and specifications in a local scale, attempting to conserve integrity of telecommunications systems. Furthermore, single administration singlehandedly developed standards and implemented them vertically in their country or other domain. These procedures began to change in the 1970s, when telecommunication monopolies started dismantling in the US and the markets were opened for various device manufacturers and service providers, pushing the prices down and alluring more competition. Similar development occurred in the Nordic countries with NMT.

In the past four decades, universal and globally acclaimed standards have been developed in the telecommunications industry. Especially GSM and its successors UMTS and LTE have become widely used standards globally. GSM initially started as a de jure standard between European countries but it later expanded as the de facto universal standard worldwide. UMTS and LTE have had a more global perspective from the beginning, but similar traits of governmental and organisational support have been exploited to stimulate their diffusion.

Standardisation towards interoperability has also been carried out in digital authentication. In Finland, the TUPAS de facto standard has been mainly used for identity authentication, utilising Finnish bank customers' electronic user identifiers such as a username and a PIN code, and a transaction authentication number. This has been developed through a collaboration of Finnish banks and it is supported by all the major Finnish banks. Another noteworthy identification method is *Mobiilivarmenne* which is supported by all the major telephone service providers, enabling strong electronic identification via a mobile phone. Mobiilivarmenne has achieved a similar collaboration of providers as the TUPAS standard, although Mobiilivarmenne has not diffused as well, possibly because of its service costs for users and low awareness.

Main research question of this thesis is to determine the holistic requirements for a building operating system. With the aid of the auxiliary questions, several success key factors of operating development and deficiencies of current building information systems have been mapped out. Additionally, literature review and especially interview study has helped to grasp the future trends of building ICT, giving a vision of what a building operating system requires. Next, a framework of requirements for a building operating system are laid out.

Depicted in the operating system review, operating systems provide *integration and interoperability* between hardware and software of a computer system. Essentially, the purpose of an operating system is to integrate hardware and software, ergo to connect processor, motherboard, video card, memory and peripherals for the utilisation of user applications through user application interfaces. Furthermore, enterprise resource planning systems strive for interoperability between functional areas of an enterprise. This basic property of interoperability is also required from a building operating system, connecting various technology inside a building. As was depicted in the literature review and interview study, various building automation systems are loosely integrated, thus a platform such as a building operating system is required to connect them together.

In order to create a BOS platform, *collaboration* is required between built environment industry and outside actors. The industry probably cannot evolve alone and it needs the appropriate technology and integration experts to connect all the heterogenous building automation and IoT under one operating system. Based on the literature review in Chapter 4, telecommunications and electronic authentication have required genuine collaboration between device manufacturers, service providers and governments in order to create universal standards. Similar type of collaboration is also required for a building operating system in order to expand it worldwide. Also, when considering the separateness of current building ICT systems, global open standardisation development and governmental endorsement similarly to telecommunications should be encouraged to achieve interoperability.

Building operating system requires an adequate level of *openness*. Primarily, open standards should be used similarly as in telecommunications to enable universal expansion of the system. As was discovered in the literature review and interview study, current building automation systems exploit open standards, but on top of the open infrastructure they often utilise proprietary and closed ad hoc solutions which hinder interoperability and hamper access to the building systems. Open systems such as Linux or Android provide easier integration with other systems and enable a wider and more democratic developer community. Overall, open systems can create a more vibrant ecosystem of device manufacturers, service providers, software developers and end users.

In addition to open systems, open data should be favoured as much as possible, as supported by the interview study discussions. Enabling efficient and diverse data flow of buildings would contribute to more competition and better quality, and more optimised services and applications to users. Furthermore, open data can introduce novel business and use case models that have not been envisioned yet. However, personal information should be appropriately anonymised if used as open data, and legislation such as GDPR in EU should be followed closely. Protection and rights of personal information must be taken seriously.

Platform ownership is another matter which affects the openness of a system. Previously, platforms have been mainly owned and administrated by private companies, demanding certain guidelines and fees for the use of the platform. Exceptions exist, mainly Linux which is open source and developed by a large and open developer community. Neutral and non-profit platforms have not yet been successful, but some effort for creating them have been carried out particularly in Finland. All things considered, a universal and global platform has an extensive influence on the whole market and its owner has a monopoly-like status on the markets. Platform companies can dictate the guidelines on how and to what purposes their platform can be used and they can request an arbitrary fee as the middleman. Market forces restrain platform owners to act excessively authoritarian but nevertheless, they have plenty of leeway in their platform operating models. Platform companies have already been addressed by governments and political unions which have demanded certain obligations from them to ensure competitiveness. Namely, Microsoft in the past have had extensive antitrust cases with the US and EU, because of Windows' anti-competitive practices and monopoly-like position in the PC operating system market. Modern platform companies such as Google, Amazon, Uber, Airbnb and others might face similar challenges with government intervention in the future.

A neutral platform administered by a consortium or a governmental union might be an option in the future. The core or infrastructure of a platform would be managed by a consortium which would provide certain guidelines and fees in order to maintain and regulate the platform. Another option to implement the building operating platform could be a distributed governance system with blockchain technology, which would enable more even value sharing and free markets (Mattila & Seppälä 2017). On top of the infrastructure, different service and application providers would rival each other, stimulating effective competition. Potentially, building operating system could be such a platform, although current trends and views suggest that a private company, possibly a global ICT company will create the BOS platform.

User-centred solutions should be encouraged when creating a building operating system. BOS should incorporate user-friendly and delightful user interfaces and experiences for various kind of end users, including building owners, apartment owners, residents, servicemen, visitors and service providers. Depicted in the literature review, user-friendly interfaces have widely contributed to the commercial success of operating systems. Especially, GUI-based operating systems revolutionised the operating system concept in the 1980s and the iPhone introduced wide touchscreen and swiping capabilities to enhance the smartphone experience. Interview study discovered that the current building ICT systems do not offer very user-friendly solutions, yet all of the interviewees demanded for more user-centred development.

Considering building operating system as a platform, it requires wide *expansion* to unleash its full platform potential. Platform aspect of a building operating system was widely discussed in the interviews, pondering how the system should achieve critical mass by expanding to neighbourhoods, districts, cities, regions, countries and ultimately worldwide. Obviously, in order to seize the markets, wide expansion is required which can also be seen in the evolution of operating systems and computing devices. Also, the global diffusion of telecommunications standards has been another example of wide expansion depicted in the literature review.

Building operating system requires *portability*, ergo applications and services developed for the system should be utilisable in any BOS-enabled building with similar capabilities. As was discovered in the interviews, current building applications are tailor-made for a single building and it requires plenty of manual work and tuning to duplicate an application into another building. As was discovered in operating system literature, operating systems for computing devices are typically portable to various devices of a similar type, and furthermore applications made for an operating system work with any device with the same or interoperable operating system. This similar portability is also required in buildings in order to enable eased portability of applications and services in a global scale.

Functional *developer tools* are crucial for an operating system. Building operating system should enable easy-to-use *SDKs* and *APIs* for software developers in order to attract a wide developer community around the operating system, thus generating more, better and diverse applications for building users. Convenient and functional SDKs and APIs have been the cornerstone for the success of Android and iOS which have amassed massive and active application marketplaces, bringing substantial value for the whole platform. Furthermore, building operating system should enable abstraction of various building devices, so that the application and service developers could easily tap into the building systems, similarly how operating systems provide abstraction in computing devices between hardware and software.

In conclusion, holistic requirements explored above for the building operating system are summarised into the following list and visually into Figure 3:

- **Integration and interoperability**. BOS needs to connect various technology inside a building, including building automation and IoT.
- **Collaboration**. In addition to technology, BOS connects various parties that need to collaborate with one another.
- **Openness**. BOS needs to use open standards and enable open data. In the best case, the platform would be developed and administrated by an open community.
- **User-centredness**. BOS requires user-centred solutions to make it appealing for the end users. Buildings are made to ease and comfort our lives, thus end users have to be taken into account when developing BOS.
- **Expansion**. BOS needs to spread globally in order to achieve a critical mass and unleash its full potential.
- **Portability**. Each building with similar capabilities should have the same BOS features, thus being able to use same services and applications globally.
- **Developer tools**. BOS requires convenient APIs, abstractions and SDKs for the needs of software and service developers.
- **Developer community**. A wide developer community is required to expand the platform and enable a wide range of services and applications for the building.

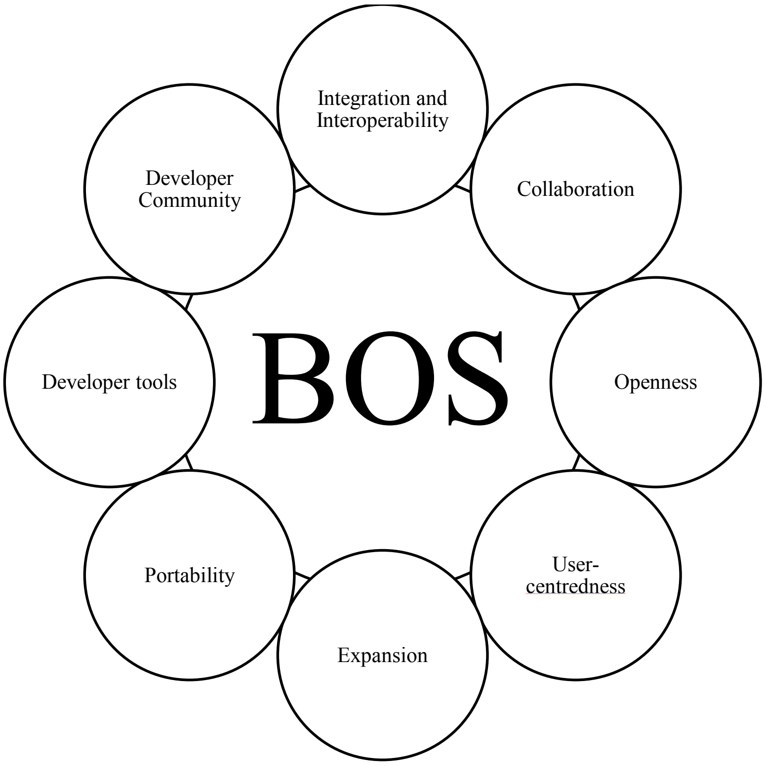

Figure 3. Holistic Requirements of A Building Operating System.

8 Conclusion

The purpose of this thesis was to determine the holistic requirements for a building operating system, ergo, to answer the main research question. For the purpose, a literature review was carried out to gather a theoretical framework for the study, which was further complemented with an empirical interview study. Literature review explored the evolution of various operating systems alongside the general evolution of computing devices. In addition, modern building information systems were reviewed, including building automation systems and IoT; and telecommunications and digital identity authentication were investigated through their evolution and standardisation towards interoperability. In the interview study, Finnish building digitalisation experts were interviewed about present and future state of building ICT. Lastly, gathered theoretical framework was reflected against the interview data and a framework of requirements for the main research questions was laid out with the aid of the auxiliary research questions.

Literature review consisted of studies from different research fields, aiming to provide a holistic view of the topics in question. More deeper examination of smart and intelligent building literature could have been carried out, although these topics have been addressed along the whole thesis. Furthermore, internet of things has been reviewed which entails essential parts of the smart building field. The extent of the operating system evolution literature was rather wide compared to other sections of the literature review.

Unfortunately, the number of interviews carried out in the interview study was fairly low, only eight in total. This resulted from coercive restrictions of the thesis schedule which only allowed a minimal amount of interviews to be conducted. To attain more support for the results, a more extensive interview study would be recommended. A desirable saturation of interviews was only achieved from the scholar research and development side; the interview study would have desired more interviewees especially from the government and industry actors of built environment, including industry developers from companies and associations, private real estate owners, legislation makers and public sector developers.

During the making of the thesis, many interviewees and industry experts have expressed how topical the thesis is. Currently, real estate and construction sector in Finland is quickly pushing forward in digitalisation, and many of the issues revolve around interoperability, integration and data flow management. Furthermore, current building systems are fragmented and siloed which further emphasises the necessity of a building operating platform. In recent years, commercial prototypes of building operating systems have been introduced as described in the thesis, stressing the topical nature of BOS.

At the moment, thesis subject matter is gaining attraction in research. Säynäjoki et al. (2017) discuss about the extraction of data from buildings to create more value. They suggest the Internet of Buildings framework, depicting the data flow between buildings, users and complementors through a central platform. Moreover, study emphasises how building data could be commercialised and distributed widely between different companies and industries. Also, research concerning the topic is being made by Esa Halmetoja, who is making a doctoral dissertation about the digitalisation of facility management. In his upcoming article, the data flow of buildings is discussed and an architectural framework is laid out for it. Furthermore, in Halmetoja’s research, digital twin is suggested as the interface of a digital building. Halmetoja also works for the Senate Properties in Finland, creating smart facility management systems for state-owned buildings. A recent article of Halmetoja (2016) discusses about the use of building information models in facility management.

Future research on the topic should concentrate on the integration and interoperability of existing building systems. Shared open standards should be implemented by the industry to gain similar interoperability as the telecommunications industry have achieved with GSM for instance. Furthermore, research on creating a core infrastructure of a neutral platform would be beneficial.

More importantly, the subject matter requires concrete actions in order to fulfil the vision of a building operating system. In addition to detailed and technical research, the topic requires concrete prototypes and experiments. More integrability should be tested and sought for between building systems, demanding actions from building owners, building users and system vendors. International standardisation and naming convention activities should be carried out in order to develop universal standards utilised widely in practice. Platform builders are also needed, whether the platform will become privately owned or neutral. Altogether, end users including all of us should be in the centre of development.

Bibliography

Ailisto, H., Collin, J., Juhanko, J., Mäntylä, M., Ruutu, S., Seppälä, T., Halén, M., Hiekkanen, K., Hyytinen, K., Kiuru, E., Korhonen, H., Kääriäinen, J., Parviainen, P. and Talvitie, J. (2016) *Onko Suomi jäämässä alustatalouden junasta?* Valtioneuvosto. Available at: http://www.otalib.fi/cgi-bin/thw/trip/?${BASE}=vttjure&${HTML}=wwwrecorden&${OOHTML}=wwwrecorden&${TRIPSHOW}=form=wwwabstracten&${FREETEXT}=R%3D75948 (Accessed: Nov 22, 2017).

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015) 'Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications', *IEEE Communications Surveys & Tutorials,* 17(4), pp. 2347-2376. doi: 10.1109/COMST.2015.2444095.

Alhojailan, M. I. (2012) Thematic analysis: A critical review of its process and evaluation. West East Journal of Social Sciences, 1(1), pp. 39-47.

Alsharif, M. H., & Nordin, R. (2017). *Evolution towards fifth generation (5G) wireless networks: Current trends and challenges in the deployment of millimetre wave, massive MIMO, and small cells*. Telecommunication Systems, *64*(4), pp. 617-637.

Aronson, J. (1995) A pragmatic view of thematic analysis. *The qualitative report*, *2*(1), pp. 1-3.

Atkinson, W. (2017) *The Future Of Building Automation Systems   * Available at: http://www.ecmag.com/section/systems/future-building-%E2%80%A8automation-systems(Accessed: Oct 25, 2017).

Bach, M.J. (1986) *The design of the UNIX operating system.* Englewood Cliffs, NJ: Prentice-Hall.

Barton, P. (2001) *Enterprise Resource Planning: Factors Affecting Success and Failure.* Available at: https://www.umsl.edu/~sauterv/analysis/488_f01_papers/barton.htm (Accessed: July 12, 2017).

Blind, K. and Gauch, S. (2008) 'Trends in ICT standards: The relationship between European standardisation bodies and standards consortia', *Telecommunications Policy,* 32(7), pp. 503-513. doi: 10.1016/j.telpol.2008.05.004.

Bolten, J.B. (2003) *E-Authentication Guidance for Federal Agencies.* Available at: https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/US_OMB/O031216B.pdf (Accessed: Oct 30, 2017).

Bond, B., Genovese, Y., Miklovic, D., Wood, N., Zrimsek, B. and Rayner, N. (2000) *ERP is dead–Long live ERP II.* Available at: http://www.sunlike.com.cn/internet/onlineerp/images/Long%20live%20ERPII%20By%20Gartner%20Group.pdf (Accessed: Oct 27, 2017).

Borgia, E. (2014) 'The Internet of Things vision: Key features, applications and open issues', *Computer Communications*. doi: 10.1016/j.comcom.2014.09.008.

Botta, A., de Donato, W., Persico, V. and Pescapé, A. (2016) 'Integration of Cloud computing and Internet of Things: A survey', *Future Generation Computer Systems,* 56, pp. 684-700. doi: 10.1016/j.future.2015.09.021.

Braun, V. and Clarke, V. (2006) Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3 (2). pp. 77-101.

Brinkmann, S. (2013) *Qualitative interviewing*. Oxford university press.

Brodkin, J. (2015) *Linux has 2,000 new developers and gets 10,000 patches for each version.* Available at: https://arstechnica.com/information-technology/2015/02/linux-has-2000-new-developers-and-gets-10000-patches-for-each-version/ (Accessed: October 13, 2017).

Burr, W.E., Dodson, D.F., Newton, E.M., Perlner, R.A., Polk, W.T., Gupta, S. and Nabbus, E.A. (2013) *Electronic authentication guideline.* Gaithersburg, Md: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology.

Changali, S., Mohammad, A. and van Nieuwland, M. (2015) *The construction productivity imperative.* Available at: https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/the-construction-productivity-imperative (Accessed: Nov 15, 2017).

Cohen-Almagor, R. (2011) 'Internet History', *International Journal of Technoethics (IJT),* 2(2), pp. 45-64. doi: 10.4018/jte.2011040104.

Columbus, L. (2017) *Internet Of Things Market To Reach $267B By 2020.* Available at: https://www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#4d96bf0609bd (Accessed: Oct 25, 2017).

Columbus, L. (2014) *Gartner's ERP Market Share Update Shows The Future Of Cloud ERP Is Now.* Available at: https://www.forbes.com/sites/louiscolumbus/2014/05/12/gartners-erp-market-share-update-shows-the-future-of-cloud-erp-is-now/#484f8be71fae (Accessed: Oct 27, 2017).

Conteh, N.Y. and Akhtar, M.J. (2015) 'Implementation Challenges of an Enterprise System and Its Advantages over Legacy Systems', *International Journal on Computer Science and Engineering,* 7(11), pp. 120-128.

Cooper, S. (2014) *Whatever happened to Netscape?* Available at: https://www.engadget.com/2014/05/10/history-of-netscape/ (Accessed: Oct 18, 2017).

Curran, J., Fenton, N. and Freedman, D. (2012) *Misunderstanding the Internet.* 1. publ. edn. London [u.a.]: Routledge.

David, P.A. and Steinmueller, W.E. (1994) 'Economics of compatibility standards and competition in telecommunication networks', *Information Economics and Policy,* 6(3), pp. 217-241. doi: 10.1016/0167-6245(94)90003-5.

Dawson-Haggerty, S., Krioukov, A., Taneja, J., Karandikar, S., Fierro, G., Kitaev, N. and Culler, D.E. (2013) *BOSS: Building Operating System Services. NSDI,* Vol. 13, pp. 443-458.

Dediu, H. (2017) *The Rise and Fall of Personal Computing.* Available at: http://www.asymco.com/2012/01/17/the-rise-and-fall-of-personal-computing/ (Accessed: Nov 27, 2017).

Deitel, H.M., Deitel, P.J. and Choffnes, D.R. (2004) *Operating Systems.* 3rd edn. New Jersey: Pearson Education.

DiCicco-Bloom, B., & Crabtree, B. F. (2006) The qualitative research interview. *Medical education*, *40*(4), pp. 314-321.

Dietrich, D., Bruckner, D., Zucker, G. and Palensky, P. (2010) 'Communication and Computation in Buildings: A Short Introduction and Overview', *IEEE Transactions on Industrial Electronics,* 57(11), pp. 3577-3584. doi: 10.1109/TIE.2010.2046570.

DistroWatch (2017) *DistroWatch Page Hit Ranking.* Available at: https://distrowatch.com/dwres.php?resource=popularity (Accessed: October 13, 2017).

Dixon, C., Mahajan, R., Agarwal, S., Brush, A.J., Lee, B., Saroiu, S. and Bahl, P. (2012) *An Operating System for the Home.*

Domingues, P., Carreira, P., Vieira, R. and Kastner, W. (2016) 'Building automation systems: Concepts and technology review', *Computer Standards & Interfaces,* 45, pp. 1-12. doi: 10.1016/j.csi.2015.11.005.

EconStats (2017) *Equity Index Data:NASDAQ Composite Index.* Available at: http://www.econstats.com/eqty/eq_d_mi_7.htm (Accessed: Oct 18, 2017).

Edwards, R., & Holland, J. (2013) *What is qualitative interviewing?* A&C Black.

Emery, M. (2013) 'Open or Closed Systems? Bridging the Gap', *systems. connecting matter, life, culture and technology,* 1(2), pp. 3-19.

European Commission (2004) *Commission Decision of 24 May 2004 relating to a proceeding pursuant to Article 82 of the EC Treaty and Article 54 of the EEA Agreement against Microsoft Corporation (Case COMP/C-3/37.792 — Microsoft)* European Union.

European Research Cluster on the Internet of Things (2017) *Internet of Things.* Available at: http://www.internet-of-things-research.eu/about_iot.htm (Accessed: Oct 25, 2017).

Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC (SI year and number). Available at: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG (Accessed: Nov 23, 2017).

Farley, T. (2005) 'Mobile telephone history', *Telektronikk,* 101(3/4), pp. 22-34.

FiCom (2014) *FiCom's (The Finnish Federation for Telecommunications and Teleinformatics) Application Guideline for ETSI's MSS Standards: V2.2.* Available at: http://mobiilivarmenne.fi/wp-content/uploads/2017/05/MSS_FiCom_Implementation_guideline.pdf (Accessed: Oct 30, 2017).

Fierro, G. and Culler, D.E. (2015) *XBOS: An Extensible Building Operating System.* Available at: http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-197.html (Accessed: Aug 17, 2017).

Flick, U. (2014) *An introduction to qualitative research*. Sage.

Forbes (2017) *The World's Biggest Public Companies.* Available at: https://www.forbes.com/global2000 (Accessed: Jul 19, 2017).

Fried, I. (2003) *EU closes in on Microsoft penalty.* Available at: https://www.cnet.com/news/eu-closes-in-on-microsoft-penalty/ (Accessed: Oct 18, 2017).

Friedrich, R., Le Merle, M., Gröne, F. and Koster, A. (2011) *Measuring industry digitization - Leaders and laggards in the digital economy.* Strategy&. Available at: https://www.strategyand.pwc.com/media/file/Strategyand-Measuring-Industry-Digitization-Leaders-Laggards-Digital-Economy.pdf (Accessed: Nov 22, 2017).

Fuentelsaz, L., Maícas, J. P., & Polo, Y. (2008) *The evolution of mobile communications in Europe: The transition from the second to the third generation*. Telecommunications Policy, *32*(6), pp. 436-449.

Funk, J. L. (2009) *The co-evolution of technology and methods of standard setting: the case of the mobile phone industry*. Journal of Evolutionary Economics, *19*(1), 73.

Gandhi, P., Khanna, S. and Ramaswamy, S. (2016) *Which Industries Are the Most Digital (and Why)?* Available at: https://hbr.org/2016/04/a-chart-that-shows-which-industries-are-the-most-digital-and-why (Accessed: Jul 12, 2017).

Gartner (2016) *Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage.* Available at: http://www.gartner.com/newsroom/id/3412017 (Accessed: Jul 19, 2017).

Gates, B. (1995) *The Internet Tidal Wave.* Available at: https://www.justice.gov/sites/default/files/atr/legacy/2006/03/03/20.pdf (Accessed: Nov 27, 2017).

Geissbauer, R., Vedso, J. and Schrauf, S. (2016) *Industry 4.0: Building the digital enterprise.* Available at: https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf (Accessed: Jul 19, 2017).

Gibbs, S. (2016) *Mobile web browsing overtakes desktop for the first time.* Available at: https://www.theguardian.com/technology/2016/nov/02/mobile-web-browsing-desktop-smartphones-tablets (Accessed: Oct 18, 2017).

GlobeNewswire (2017) *Global Smart Building Market Size Will Reach USD 36.0 Billion by 2020.* Available at: https://globenewswire.com/news-release/2017/05/25/996108/0/en/Global-Smart-Building-Market-Size-Will-Reach-USD-36-0-Billion-by-2020.html (Accessed: Jul 19, 2017).

Golafshani, N. (2003). *Understanding reliability and validity in qualitative research*. The qualitative report, *8*(4), pp. 597-606.

Goodger, B. (2006) 'Where Did Firefox Come From?', https://web.archive.org/web/20110623034401/. Available at: (Accessed: Oct 18, 2017).

Grand View Research (2017) *Industrial IoT Market Size Worth $933.62 Billion by 2025 / CAGR: 27.8%: Grand View Research, Inc.* Available at: http://www.grandviewresearch.com/press-release/global-industrial-internet-of-things-iiot-market (Accessed: Oct 25, 2017).

Gromov, G. (2002) *The First Decade of the Internet History: Brief Stats Story.* Available at: http://history-of-internet.com/#8 (Accessed: Oct 17, 2017).

Gruber, H. and Verboven, F. (2001) 'The evolution of markets under entry and standards regulation — the case of global mobile telecommunications', *International Journal of Industrial Organization,* 19(7), pp. 1189-1212. doi: 10.1016/S0167-7187(01)00069-8.

Hackett, R. (2016) *How Critical Real Estate Is to the Global Economy -- In One Chart.* Available at: http://fortune.com/2016/01/26/rea-estate-global-economy/ (Accessed: Nov 18, 2017).

Hall, S.P. and Anderson, E. (2009) 'Operating Systems for Mobile Computing', *Journal for Computing Sciences in Colleges,* 25(2), pp. 64-71.

Halmetoja, E. (2016) *Tietomallit ylläpidossa.* Available at: https://www.senaatti-areena.fi/filebank/6099-Tietomallit_yllapidossa.pdf (Accessed: Nov 22, 2017).

Heger, D.A. (2012) *Mobile Devices - An Introduction to the Android Operating Environment.* Available at: https://www.cmg.org/wp-content/uploads/2011/04/m_78_3.pdf (Accessed: Nov 27, 2017).

Design, Architecture, and Performance Implications. Available at: https://www.cmg.org/wp-content/uploads/2011/04/m_78_3.pdf (Accessed: Jul 12, 2017).

Heikkilä, J., Korhonen, P. and Paavola, M. (1999) *Matkapuhelinten sukupolvet.* Available at: https://www.netlab.tkk.fi/opetus/s38118/s99/htyo/47/index.shtml (Accessed: Oct 26, 2017).

Hennessy, J.L. and Patterson, D.A. (2014) *Computer Organization and Design.* 5th edn.Elsevier Science.

Internet Live Stats (2017) *Internet Users.* Available at: http://www.internetlivestats.com/internet-users/ (Accessed: Jul 19, 2017).

Internet World Stats (2017) *World Internet Usage And Population Statistics.* Available at: http://www.internetworldstats.com/stats.htm (Accessed: Jul 19, 2017).

Jacobs, F.R. and Weston, F.C. (2007) 'Enterprise resource planning (ERP) - A brief history', *Journal of operations management,* 25(2), pp. 357-363.

Jones, B. (2015) *Rags to Riches to Rags: The Rise and Fall of Internet Explorer.* Available at: https://www.digitaltrends.com/computing/the-rise-and-fall-of-internet-explorer/ (Accessed: Oct 18, 2017).

Jungner, M. (2015) *Otetaan digiloikka! Suomi digikehityksen kärkeen*, Helsinki: Elinkeinoelämän keskusliitto EK.

Kanter, J. (2012) *In European Court, a Small Victory for Microsoft.* Available at: http://www.nytimes.com/2012/06/28/business/global/in-european-court-a-small-victory-for-microsoft.html?_r=1 (Accessed: Oct 18, 2017).

Kastner, W., Neugschwandtner, G., Soucek, S. and Newmann, H.M. (2005) 'Communication Systems for Building Automation and Control', *Proceedings of the IEEE,* 93(6), pp. 1178-1203. doi: 10.1109/JPROC.2005.849726.

Kerttula, E. (2015) 'A novel federated strong mobile signature service—The Finnish case', *Journal of Network and Computer Applications,* 56, pp. 101-114. doi: 10.1016/j.jnca.2015.06.007.

Kiinteistö- ja rakentamisfoorumi (2011) *NYT/2025 Rakennettu ympäristömme.* Helsinki: Kiinteistö- ja rakentamisfoorumi. Available at: http://www.asuntotieto.com/asuntotietokeskus/esittely/KIRA_Rakennettu_ymp_netti.pdf (Accessed: Nov 27, 2017).

Kleinrock, L. (2010) 'An early history of the internet [History of Communications]', *IEEE Communications Magazine,* 48(8), pp. 26. doi: 10.1109/MCOM.2010.5534584.

KPMG Switzerland (2016) *SnapShot! on Digitalization - Real Estate Europe 2016.* Available at: https://assets.kpmg.com/content/dam/kpmg/ch/pdf/european-snaphot-on-digitalization-en.pdf (Accessed: Jul 19, 2017).

Krishnamurthy, S. (2009) 'CASE: Mozilla vs. Godzilla — The Launch of the Mozilla Firefox Browser', *Journal of Interactive Marketing,* 23(3), pp. 259-271. doi: 10.1016/j.intmar.2009.04.008.

Lasi, H., Fettke, P., Kemper, H. G., Feld, T., & Hoffmann, M. (2014) *Industry 4.0.* Business & Information Systems Engineering, 6(4), pp. 239-242.

Leiponen, A. E. (2008) *Competing through cooperation: The organization of standard setting in wireless telecommunications*. Management Science, 54(11), pp. 1904-1919.

Li Da Xu (2011) 'Enterprise Systems: State-of-the-Art and Future Trends', *IEEE Transactions on Industrial Informatics,* 7(4), pp. 630-640. doi: 10.1109/TII.2011.2167156.

Li, S., Xu, L.D. and Zhao, S. (2015) 'The internet of things: a survey', *Information Systems Frontiers,* 17(2), pp. 243. doi: 10.1007/s10796-014-9492-7.

Lohilahti, O. (2017) *Rakennusalalla työn tuottavuus ei ole kasvanut 40 vuodessa – onko allianssista tai leanista apua?* Available at: https://www.rakennuslehti.fi/2017/09/rakennusalalla-tyon-tuottavuus-ei-ole-kasvanut-40-vuodessa-onko-allianssista-tai-leanista-apua/ (Accessed: Nov 15, 2017).

Luo, B. (2013) *Ronald Wayne: On Co-founding Apple and Working With Steve Jobs.* Available at: https://nextshark.com/ronald-wayne-interview (Accessed: Jul 28, 2017).

Maeda, T., Amar, A. D., & Gibson, A. (2006) *Impact of wireless telecommunications standards and regulation on the evolution of wireless technologies and services over Internet protocol*. Telecommunications Policy, *30*(10), pp. 587-604.

Magenta Advisory (2017) *Suomen digimenestyjät 2017.* Magenta Advisory. Available at: http://www.magentaadvisory.com/fi/2017/01/24/digimenestyjat2017/ (Accessed: July 12, 2017).

Markus, M.L. and Tanis, C. (2000) 'The Enterprise System Experience — From Adoption to Success', in Zmud, R.W. (ed.) *Framing the domains of IT research: Framing the future through the past* Ohio: Pinnaflex Educational Resources, pp. 173-207.

Mason, J. (2002) *Qualitative researching*. Sage.

Mattila, J. and Seppälä, T. (2017) *Distributed Governance in Multi-Sided Platforms.* Washington D.C., United States: Industry Studies Association Conference 2017. Available at: https://www.industrystudiesconference.org/conference/papers/download/86 (Accessed: Nov 24, 2017).

McGibney, A., Rea, S. and Ploennigs, J. (2016) *Open BMS - IoT driven architecture for the internet of buildings* Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE. IEEE, 2016.

Mobiilivarmenne (2017) *Mobiilivarmenne.* Available at: https://mobiilivarmenne.fi/ (Accessed: Oct 30, 2017).

Mohd Nawi, M.N., Baluch, N. and Bahauddin, A.Y. (2014) 'Impact of Fragmentation Issue in Construction Industry: An Overview', *MATEC Web of Conferences,* 15, pp. 1009. doi: 10.1051/matecconf/20141501009.

Morgner F., Bastian P., Fischlin M. (2016) Securing Transactions with the eIDAS Protocols. In: Foresti S., Lopez J. (eds) Information Security Theory and Practice. WISTP 2016. Lecture Notes in Computer Science, vol 9895. Springer, Cham.

Moumina, A. (2001) *History of Operating Systems*. Available at: https://inf.ufes.br/~zegonc/material/Sistemas_Operacionais/history_of_operating_system_Moumina.pdf (Accessed: Nov 22, 2017).

NetMarketShare (2017) *Desktop Operating System Market Share.* Available at: https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0&qpsp=2015&qpnp=3&qptimeframe=Y (Accessed: Jul 29, 2017).

Niskanen, M. (2016) *Miljoonilla euroilla kehitetty mobiilivarmenne ei kiinnosta – Arvio: Käyttäjiä 200 000 * Available at: https://yle.fi/uutiset/3-9183014 (Accessed: Oct 30, 2017).

Ocock, T. (2010) *Guest post: Symbian OS – one of the most successful failures in tech history.* Available at: https://techcrunch.com/2010/11/08/guest-post-symbian-os-one-of-the-most-successful-failures-in-tech-history-2/ (Accessed: Oct 26, 2017).

Oesterreich, T. D., & Teuteberg, F. (2016) Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry. *Computers in Industry*, *83*, 121-139.

Open Group (2017) *Internet of Things.* Available at: http://opengroup.org/iot/top/index.htm (Accessed: Oct 25, 2017).

O'Regan, G. (2016) *Introduction to the History of Computing.* Cham: Springer International Publishing.

O'Reilly, T. (2007) 'What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software', *Communications & Strategies,* 65, pp. 17-37.

Pakarinen, S. (2017) *Rakennusteollisuus suhdannekatsaus lokakuu 2017.* Available at: https://www.rakennusteollisuus.fi/globalassets/suhdanteet-ja-tilastot/suhdannekatsaukset/2017/syyskuu-2017/rt-syksyn-2017-suhdanne.pdf (Accessed: Nov 18, 2017).

Panetto, H. and Cecil, J. (2013) 'Information systems for enterprise integration, interoperability and networking: theory and applications', *Enterprise Information Systems,* 7(1), pp. 1-6. doi: 10.1080/17517575.2012.684802.

Panetto, H., Zdravkovic, M., Jardim-Goncalves, R., Romero, D., Cecil, J. and Mezgár, I. (2016) 'New perspectives for the future interoperable enterprise systems', *Computers in Industry,* 79, pp. 47-63. doi: 10.1016/j.compind.2015.08.001.

Panorama Consulting (2017) *Clash of the Titans - An Independent Comparison of SAP, Oracle, Microsoft Dynamics and Infor.* Available at: https://www.panorama-consulting.com/wp-content/uploads/2017/10/White-Paper-Clash-of-the-Titans-2017.pdf (Accessed: Oct 27, 2017).

Parviala, A. (2017) *Pankki niisti joka verkkotunnistautumisesta jopa 50 senttiä – tänään hinta laskee, mutta otetaanko sentit sinulta?* Available at: https://yle.fi/uutiset/3-9587886(Accessed: Oct 30, 2017).

Piikkilä, V., Sahlsten, T., Salminen, J., Alhava, O., Järvinen, T., Stigzelius, J., Heinlahti, J., Sainio, S., Lähteenmäki, J., Kemppainen, J., Säteri, J., Niemi, J., Koivula, P., Kenttä, J., Pekkola, V., Mether, E., Ollenberg, J. and Lukkarinen, P. (2016) *Digitalisaatio rakennushankkeissa.* Espoo: Avoin automaatio ry. Available at: http://www.avoinautomaatio.fi/doc/julkaisuja/Digitalisaatio_rakennushankkeissa.pdf (Accessed: July 12, 2017).

Puhto, J., Snellman, S., Gussander, J., Kärkkäinen, H. and Pekkanen, J. (2016) *Digiselvitys 2016:Digitaalisuuden nykytila ja kehityssuunnat kiinteistö- ja rakennusalalla.* Tampereen teknillinen yliopisto. Rakennustekniikan laitos.

Pulkkis, G. and Yazdani, F. (2013) 'Mobile certificate based network services', *Issues in Informing Science & Information Technology,* 10, pp. 413-429.

Rakennusteollisuus (2017) *Rakennusalan työmarkkinat, perustietoa.* Available at: https://www.rakennusteollisuus.fi/Tietoa-alasta/Tyoelama/Tietoja-tyovoimasta-rakennusalalla/(Accessed: Nov 18, 2017).

Rashid, M., Hossain, L. and Patrick, J.D. (2002) *The Evolution of ERP Systems: A Historical Perspective.*

Research Nester (2017) *Internet of Things (IoT) Market : Global Demand, Growth Analysis & Opportunity Outlook 2023 * Available at: https://www.researchnester.com/reports/internet-of-things-iot-market-global-demand-growth-analysis-opportunity-outlook-2023/216 (Accessed: Oct 25, 2017).

Reuters Staff (2008) *Qualcomm halts UMB project, sees no major job cuts.* Available at: http://www.reuters.com/article/qualcomm-umb/qualcomm-halts-umb-project-sees-no-major-job-cuts-idUSN1335969420081113 (Accessed: Oct 29, 2017).

Robin, G. (1994) 'The European perspective for telecommunication standards', *IEEE Communications Magazine,* 32(1), pp. 40-44. doi: 10.1109/35.249772.

ROTI (2017) *Rakennetun omaisuuden tila 2017.* Available at: http://roti.fi/wp-content/uploads/2015/12/ROTI-2017-raportti.pdf (Accessed: Jul 12, 2017).

Salo, E. (2017) *Talotekniikan avoimet tiedonsiirtorajapinnat.* Master's Thesis. Aalto University. Available at: http://urn.fi/URN:NBN:fi:aalto-201709046854 (Accessed: Nov 27, 2017).

Salus, P.H. (1994) *A quarter century of UNIX.* United States.

Sarwar, M., & Soomro, T. R. (2013). Impact of smartphone's on society. *European journal of scientific research*, *98*(2), 216-226.

Säynäjoki, A., Pulkka, L., Säynäjoki, E. and Junnila, S. (2017) 'Data Commercialisation: Extracting Value from Smart Buildings', *Buildings,* 7(4), pp. 104. doi: 10.3390/buildings7040104.

Schneider Electric (2015) *Guide to Open Protocols In Building Automation.* Available at: http://blog.schneider-electric.com/wp-content/uploads/2015/11/SE-Protocols-Guide_A4_v21.pdf (Accessed: Oct 23, 2017).

Schroers, J. and Cuijpers, C. (2014) *eIDAS as guideline for the development of a pan European eID framework in FutureID.* Available at: https://lirias.kuleuven.be/bitstream/123456789/470230/2/OID+2014+paper+Jessica+Colette+v_final1.pdf (Accessed: Nov 22, 2017).

Scott, M. (2014) *Better together: why construction needs collaboration to work efficiently.* Available at: https://www.theguardian.com/sustainable-business/collaboration-construction-buildings (Accessed: Nov 18, 2017).

Sedera, D., Lokuge, S., Grover, V., Sarker, S. and Sarker, S. (2016) 'Innovating with enterprise systems and digital platforms: A contingent resource-based theory view', *Information & Management,* 53(3), pp. 366-379. doi: 10.1016/j.im.2016.01.001.

Serrano, N., Hernantes, J. and Gallardo, G. (2014) 'Service-Oriented Architecture and Legacy Systems', *IEEE Software,* 31(5), pp. 15-19. doi: 10.1109/MS.2014.125.

Shan, C.P. (2011) *Long antitrust saga ends for Microsoft.* Available at: https://www.seattletimes.com/business/microsoft/long-antitrust-saga-ends-for-microsoft/ (Accessed: Oct 18, 2017).

Shaul, L. and Tauber, D. (2013) 'Critical success factors in enterprise resource planning systems: Review of the last decade', *ACM Computing Surveys (CSUR),* 45(4), pp. 1-39. doi: 10.1145/2501654.2501669.

Shin, D., Kim, H. and Hwang, J. (2015) 'Standardization revisited: A critical literature review on standards and innovation', *Computer Standards & Interfaces,* 38, pp. 152-157. doi: 10.1016/j.csi.2014.09.002.

Silberschatz, A., Galvin, P.B. and Gagne, G. (2013) *Operating system concepts.* 9th edn.Wiley.

Snellman, S. (2016) *Digitaalisen liiketoiminnan nykytila rakennus- ja kiinteistöalalla.* Master's Thesis. Tampere University of Technology. Available at: http://URN.fi/URN:NBN:fi:tty-201604203861(Accessed: Jul 19, 2017).

Sorensen, C., De Reuver, M. and Basole, R.C. (2015) 'Mobile platforms and ecosystems', *Journal of Information Technology,* 30(3), pp. 195-197. doi: 10.1057/jit.2015.22.

Spyrelli, C. (2002) *Electronic Signatures: A Transatlantic Bridge? An EU and US Legal Approach Towards Electronic Authentication.* Available at: https://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002_2/spyrelli/ (Accessed: Oct 20, 2017).

Stallings, W. (2010) *Computer organization and architecture.* 8. ed., int. ed. edn. Boston, Mass. [u.a.]: Pearson.

StatCounter Global Stats (2017a) *Browser Market Share Worldwide.* Available at: http://gs.statcounter.com/browser-market-share#monthly-200901-201709 (Accessed: Oct 18, 2017).

StatCounter Global Stats (2017b) *Desktop Operating System Market Share Worldwide.* Available at: http://gs.statcounter.com/os-market-share/desktop/worldwide#monthly-200901-201706 (Accessed: Jul 29, 2017).

Statista (2017a) *Global GDP (gross domestic product) at current prices from 2010 to 2022 (in billion U.S. dollars).* Available at: https://www.statista.com/statistics/268750/global-gross-domestic-product-gdp/ (Accessed: Nov 18. 2017).

Statista (2017b) *Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2017.* Available at: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/ (Accessed: Oct 26, 2017).

Statista (2017c) *Operating Systems - Statistics and Facts.* Available at: https://www.statista.com/topics/1003/operating-systems/ (Accessed: Jul 29, 2017).

Statistics Finland (2016) *Kansantalouden tuottavuuskehitys 1976-2015.* Available at: http://www.stat.fi/til/ttut/2015/ttut_2015_2016-11-30_tie_001_fi.html (Accessed: Nov 15, 2017).

Stephenson, N. (1999) *In The Beginning was the Command Line.* Available at: http://cristal.inria.fr/~weis/info/commandline.html (Accessed: Jul 12, 2017).

Swenson, R.C. (2017) *What is a Commodore Computer? A Look at the Incredible History and Legacy of the Commodore Home Computers.* Available at: http://www.oldsoftware.com/history.html (Accessed: Jul 29, 2017).

Tanenbaum, A.S. (2009) *Modern operating systems.* 3rd edn. New Jersey: Pearson Education International.

Tanenbaum, A.S. (2006) *Structured computer organization.* 5. ed. edn. Upper Saddle River, NJ: Pearson Prentice-Hall.

The Economist (2017) *The construction industry's productivity problem.* Available at: https://www.economist.com/news/leaders/21726693-and-how-governments-can-catalyse-change-construction-industrys-productivity-problem (Accessed: Nov 15, 2017).

Tilson, D., Sorensen, C. and Lyytinen, K.J. (2012) *Change and control paradoxes in mobile infrastructure innovation: the Android and iOS mobile operating systems cases. System Science (HICSS), 2012 45th Hawaii International Conference,* pp. 1324-1333. IEEE 2012.

Tilson, D., Sorensen, C. and Lyytinen, K.J. (2011) *The paradoxes of change and control in digital infrastructures: the mobile operating systems case.* Mobile Business (ICMB), 2011 Tenth International Conference, pp. 26-35. IEEE 2011.

Turner & Townsend (2017) *International construction market survey 2017.* Available at: http://www.turnerandtownsend.com/media/2412/international-construction-market-survey-2017-final.pdf (Accessed: Nov 18, 2017).

Umble, M.M., Umble, E.J. and Haft, R.R. (2003) 'Enterprise resource planning: Implementation procedures and critical success factors', *European Journal of Operational Research,* 146(2), pp. 241-257. doi: 10.1016/S0377-2217(02)00547-7.

United Nations (2014) *World Urbanization Prospects.* United Nations. Available at: https://esa.un.org/unpd/wup/publications/files/wup2014-highlights.pdf (Accessed: Nov 18, 2017).

Väestörekisterikeskus (2016) *Väestörekisterikeskuksen ja Poliisihallituksen tiedote: Henkilökorttien hakumahdollisuudet laajenevat ja henkilökortin käyttöön tulee muutoksia.* Available at: http://vrk.fi/artikkeli/-/asset_publisher/vaestorekisterikeskuksen-ja-poliisihallituksen-tiedote-henkilokorttien-hakumahdollisuudet-laajenevat-ja-henkilokortin-kayttoon-tulee-muutoksia (Accessed: Oct 30, 2017).

Viitanen, J., Paajanen, R., Loikkanen, V. and Koivistoinen, A. (2017) *Digitaalisen alustatalouden tiekartasto.* Tekes. Available at: https://www.tekes.fi/globalassets/global/ohjelmat-ja-palvelut/kampanjat/alustatalous/alustatalouden_tiekartasto_web_x.pdf (Accessed: Nov 22, 2017).

W3Counter (2017) *Browser and Platform Market Share.* Available at: https://www.w3counter.com/globalstats.php (Accessed: Oct 18, 2017).

Watt, P. and McGeever, C. (1985) 'Macintosh vs. IBM PC at One Year'. *InfoWorld, January 15 1985.* pp. 15-16.

Weber, A.S. (2000) *Nineteenth century science.* Peterborough: Broadview Press.

Weng, T., Nwokafor, A. and Agarwal, Y. (2013) *BuildingDepot 2.0: An integrated management system for building analysis and control.* Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings. ACM.

West, J. and Mace, M. (2010) 'Browsing as the killer app: Explaining the rapid success of Apple's iPhone', *Telecommunications Policy,* 34(5), pp. 270-286. doi: 10.1016/j.telpol.2009.12.002.

West, J. and Mace, M. (2007) *Entering a Mature Industry Through Innovation: Apple's iPhone Strategy.* DRUID Summer Conference.

Yli-Villamo, H. and Petäjäniemi, P. (2013) 'Allianssimalli', in Koskenvesa, E., Heloma, T. and Laine, S. (eds.) *Rakentajain kalenteri 2013* Rakennustieto Oy, pp. 57-66.