Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Tuomas Lauri Wilhelm Juopperi

Real-time Space Occupancy Monitoring:

A Wireless Sensor Network Implementation

Master's Thesis
Espoo, Feb 15, 2019

Supervisor: Heikki Ihasalo, Professor of Practice, Aalto University

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author:	Tuomas Lauri Wilhelm Juopperi		
Title: Real-time Space Occupancy Monitoring: A Wireless Sensor Network Implementation			
Date:	Feb 15, 2019	**Pages:**	71
Supervisor:	Professor of Practice, Heikki Ihasalo		

This paper discusses solutions and technologies for automatically assessing the state of space reservation in a building, e.g., a university or an office building. Improving space utilization can benefit an organization or a company in multiple aspects. These aspects include enhanced user experience and improved workflow, and minimized losses from maintaining unused space. Additionally, the space utilization information can be utilized lower electricity usage, by optimizing HVAC (Heating, ventilation, and air conditioning) and lighting systems.

The goal of this study is to asses and compare different methods of gaining the real-time reservation status of a space. Both hardware and software factors are taken into account. The methods are first studied in the form of a literature review. This part of the study includes comparisons between relevant technologis. The most suitable solution is also inspected in practice.

Furthermore, as part of the study a real-time reservation status monitoring system was developed utilizing Philips hardware and recommendable software technologies. This experimental phase of the study also includes an overview of testing and installing such a system. A system built by an outside supplier was also installed for reference. Both systems employ PIR (Passive Infrared) sensors. The sensors are not designed for people counting purposes, which was not a core focus of this study.

Gathering space utilization data can have further, far stretching benefits. Therefore, this paper also discusses different use cases for the occupancy data. The value of such data can be significant, due to property costs taking up a large portion of companies' expenses. Effectively utilizing this data can therefore prove to be remarkably advantageous.

Keywords:	Wireless Sensor Network, PIR sensors, IoT, Computer Science, Presence sensors, Space occupancy monitoring
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan tutkinto-ohjelma

DIPLOMITYON TIIVISTELMA

Tekija:	Tuomas Lauri Wilhelm Juopperi		
Tyon nimi: Reaaliaikainen tilojen käytön seuranta: langaton sensoriverkko implementaatio			
Paivays:	15. helmikuuta 2019	**Sivumaara:**	71
Valvoja:	Professor of Practice, Heikki Ihasalo		

Tämä tutkimuspaperi käsittelee ratkaisuja ja teknologioita automaattiseen tilojen varausasteen määrittämiseen julkisissa rakennuksissa, kuten yliopistoissa tai toimistoissa. Tilojen käytön tehostaminen voi hyödyttää organisaatiota tai yritystä monilla tavoin. Näihin tapoihin kuuluvat muun muassa parantunut tilojen käyttökokemus ja työn sujuvuus, sekä minimoidut käyttämättömien tilojen ylläpitokustannukset. Lisäksi tilojen käyttöasteinformaatiota voidaan hyödyntää sähkönkulutuksen alentamiseksi, optimoimalla LVI- ja valaistusjärjestelmiä.

Tämän tutkimuksen tavoite on arvioida ja vertailla eri tapoja hankkia tieto tilojen reaaliaikaisesta varaustilanteesta. Sekä laitteisto, että ohjelmisto tekijät on otettu huomioon. Tapoja tutkitaan aluksi kirjallisuuskatsauksella. Tässä osassa tutkimusta vertaillaan eri teknologioita. Sopivimman ratkaisun toimintaa tutkittiin myös käytännössä.

Lisäksi osana tutkimusta kehitettiin reaaliaikaisen varaustilanteen seurantajärjestelmä hyödyntäen Philips Hue laitteita ja suositeltavia ohjelmistoteknologioita. Tämä kokeellinen tutkimusvaihe sisältää myös yleiskuvan vastaavien järjestelmien testaamisesta ja asentamisesta. Myös kolmannen osapuolen toimittama järjestelmä asennettiin tiloihin vertailukohteeksi. Molemmat järjestelmät käyttävät PIR (Passive Infrared) sensoreita. Sensorit eivät ole tarkoitettu henkilölaskentaa varten, koska se ei ollut tämän tutkimuksen ydintavoite.

Tilojen käyttödatan keräämisellä voi lisäksi olla muita, kauaskantoisia hyötyjä. Siksi tutkimuksessa käsitellään myös erilaisia tapoja hyötykäyttää sitä. Datan arvo voi olla merkittävä tilakustannusten ollessa merkittävä osa yritysten kuluja. Datan tehokas käyttö voi osoittautua huomattavan suotuisaksi.

Asiasanat:	Langaton sensoriverkko, PIR sensorit, IoT, Tietotekniikka, Läsnäolotunnistimet, tilankäytön seuranta
Kieli:	Englanti

Acknowledgements

Firstly, I wish to thank my supervisor and instructor Professor Heikki Ihasalo and the rest of our research group: Mika Maaspuro and Petri Jantunen. They have eagerly shared with me their knowledge and supported me through out this process. Professor Ihasalo has allowed me the freedom to work as I saw fit, while still giving me help and guidance whenever needed. Much of my gratitude is also owed to the numerous representatives and experts who helped me along the way. Additionally, I would like to thank the ACRE (Aalto University Campus & Real Estate) department and its' representatives for funding this project and giving me the opportunity to be a part of developing this great University.

I would also like to express a sincere thank you to Aalto University as a whole, and the Finnish education system. They have allowed me and countless others to invest fully in the pursuit of knowledge and higher understanding. The advancement of science and technology is unquestionably built upon the work of others before us. Let us strive to maintain this throughout our own lives.

Lastly, a massive thank you to my family and friends, who have given me more support than they perhaps even know. I hope that this thesis can give something to the people, and help make all our lives a tiny bit easier.

Thank You!

Espoo, Feb 15, 2019

Tuomas Lauri Wilhelm Juopperi

Abbreviations and Acronyms

IoT	Internet of Things
PIR	Passive Infrared
HVAC	Heating, Ventilation, and Air Conditioning
API	Aplication Programming Interface
UI	User Interface
WSN	Wireless Sensor Network
WMN	Wireless Mesh Network
WM	Wirepas Mesh
LoRa	Long Range
BLE	Bluetooth Low Energy
LPWAN	Low Power Wide Area Network
TSDB	Time Series Database
HTTP	Hypertext Transfer Protocol
JWT	JSON Web Token

Contents

Chapter 1

Introduction

Inefficient space utilization can cause a multitude of problems. If not taken in to account, it can at the least lead to unnecessary operating costs. On the one hand maintaining unused workspaces alone causes ineffectual expenses, on the other hand a workspace deficit or inefficient usage of workspaces can hinder workflow and lead to further problems. In addition to declines in productivity, these problems can include employee dissatisfaction, inefficient energy usage, and security problems. [37]

Space utilization optimization can be enhanced by giving users more accurate information about the current occupancy situation within the workplace. This can improve work flow in practice by making it easier and faster to find a free meeting room. This occupancy data can be gathered with a network of sensors. The data can also be used to determine requirements for additional workspaces.

Space reservation systems relying solely on user input often lead to over estimated booking lengths. For example, if a group has an appointment that takes approximately 30 minutes, they are likely to book a space for an hour. If the meeting is over in 30 minutes, the space utilization is 50 % at maximum. Additionally, the utilization percentage may be even lower, considering the size of the group compared to the size of the reserved room. This leads to the users paying for services they did not use, and the space provider not being able to support as many customers as possible.

Some workspaces may allow multiple groups. If the space reservation system is binary, i.e., a space is either free or occupied, it limits the possibilities of space utilization. Therefore, it can also be beneficial to asses the *level* of occupancy.

Scalability is one of the most important factors when reviewing technology options. Dealing with the scalability problem requires new and innovative technologies. The issue includes matters, such as ease of installation, protocol extensibility, and computational power of the hardware used. One example of a solution to the growth of the internet in general is the development of IPv6. As the IPv4 address space of around 4 billion (4 294 967 296) 32bit addresses was proving to be far too scarce, IPv6 was built to give the internet room for growth. With 128bits used, IPv6 allows 340 undecillion, i.e., $3.4 * 10^{38}$ unique addresses to be used. This should be enough for now. Additionally, the vast address space means that each end device and node on the internet can be provided a unique IP address [70]. Fortunately, The growing IoT field has a variety of turnkey solutions for space occupancy monitoring. The available modern systems are often designed for wide scale corporate use. The suppliers can deliver low-cost hardware with scalable, low-energy network protocols. [33] [55]

The wide spread effects the internet has had on our lives are undeniable. Most people in developed countries use multiple devices that access information utilizing this extensive network, on a daily basis. IoT is the next step in the digital age. IoT, as a whole, encompasses the vast spectrum of equipment, home appliances, sensors, and industrial machines, which are, or will soon be, connected to the internet aswell. The overall number of connected devices is expected to grow to 50 billion by 2020 [70] [25]. By then, the amount of IoT devices will be approximately 36 billion [62]. A large portion of the devices will likely be sensors of varying types. Therefore, studying the underlying technologies and specific use cases is beneficial.

1.1 Research question

The aim of this thesis and the combined experiment project is to enhance work flow by further optimizing space utilization. We aim to find an answer to the following question: What are the most suitable hardware and software technologies for gathering real-time occupancy data within ad-hoc workspaces, and how the gathered data can be analyzed. The factors taken in to account are cost efficiency, privacy, reliability, low maintenance, and accuracy.

Additionally, the goal is to give the reader an overview of the factors associated with implementing and developing a space occupancy monitoring WSN (Wireless Sensor Network) system.

1.2 Research methods

The problem cases mentioned previously can be mitigated by implementing a system for assessing the real-time reservation status, and the level of occupancy within a public space. Such as system can give users feedback according to the real-time situation within the area.

First, a literature review was carried out in order to determine the relevant hardware and software technologies. The methods were selected by researching previously studied, similar implementations. In addition to the literature research, companies within the field were contacted to discover the most common technologies. After taking into account relevant factors, the most reasonable technology choice was selected. Subsequently, an experiment implementation was also developed using the methods deemed most suitable. The in-house built implementation was also compared to a turnkey solution provided by an outside supplier. [27] [18] [53]

The focus of this paper is on detecting presence in general and minimizing unused space. People counting is not a core focus, but is discussed concisely. The implementation is not designed for people counting purposes.

1.3 Structure of the Thesis

This thesis is structured as follows: Chapters 2 and 3 discusses the most relevant methods of space occupancy monitoring. Relevant hardware, software, and networking technologies are discussed here generally. Factors such as accuracy, ease of installation, and scalability are taken into account.

Chapter 4.1 discusses the environment in which the experiments were conducted. The variety of workspaces used in the experiment, and their characteristic features are depicted here. Additionally, other use cases for space utilization monitoring systems are addressed in this chapter.

Chapter 4 discusses the methods used. I.e., the selected technologies are discussed more in depth, with their respective drawbacks and benefits.

Chapter 5 describes the final implementation. Both the hardware and software components are presented in detail. The stages of implementing such systems and matters that should be taken into account when planning a similar project are discussed here.

Chapter 6 includes data analysis and general evaluation of the implementation. Possible benefits are discussed in more detail here.

Chapter 6.3 is designated for further discussion on the subject and the thesis. Possibilities provided by WSN systems of this type are explored concisely.

Chapter 7 comprises a conclusion of the study. It aims to give an overview of the themes discussed and assess the usefulness of a real-time space utilization monitoring system.

Chapter 2

Networking technologies and software

This chapter discusses the software aspects of a occupancy monitoring system. Firstly different topology designs relevant to WSNs are depicted. The other addressed elements are networking protocols and software employing varying types of topology designs. This chapter aims to give insight into the underlying functionalities of such a system. The technologies are selected on the basis of relevance to the subject. The software selection for backend and frontend development is extensive, and largely up to preference. However, this chapter aims to give resources and perspective for developers.

The selection of available network technologies and protocols for IoT solutions, such as sensor networks, is growing rapidly. This growth comes from a necessity. In 2015 the amount of devices connected to the internet was 25 billion. It is expected that in 2020 there could be around 6.5 internet-connected devices per person. Figure 2.1 shows the explosive increase in internet-connected devices.

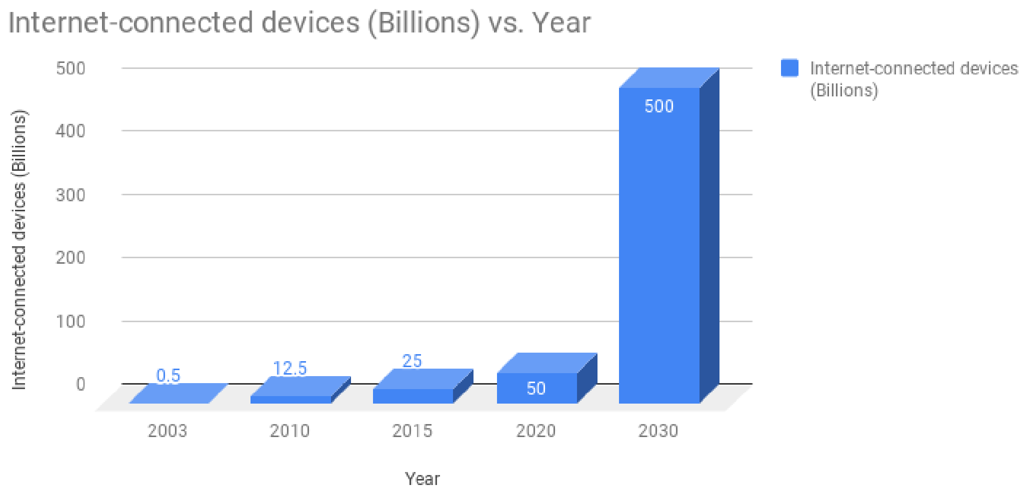

Figure 2.1: The rising number of internet-connected devices (adapted from [25] and [10]).

The growth is largely due to the expansion of the IoT. Devices are often no longer bound to a specific user. Billions of devices with specific functionalities become parts of larger networks, such as sensor networks. This is in addition to the fact that common home appliances are quickly becoming internet nodes. [25]

The choice between network technologies and protocols is an important one. This choice can largely affect both the reliability and power consumption of a system. Power consumption of wireless sensor nodes has been effectively reduced as technology has advanced. There are products on the market, which offer battery lives of up to 10 years. In some cases power consumption may be considered low enough to allow the use of energy harvesting techniques, as in the case of EnOcean devices [2] [46].

Wireless sensor nodes most commonly utilise either the IEEE 802.15.4 based ZigBee or Bluetooth LE for data transfers. The data rates of WSNs are generally minuscule. In addition to lowered power consumption, these devices have become smaller and easier to install. New multipurpose sensor nodes are a topic of wide spread interest, and finding the most suitable solutions for energy management and functionality is essential. [6]

Wireless Sensor Networks are commonly built on technologies using either a star, mesh or tree topology. The difference between the three is that the latter two typically have multiple routes between nodes. Generally, the remote station, i.e. a base station or a bridge is used to gather the data from the nodes. The nodes themselves contain the necessary sensors, power sources, micro processors, additional memory, analog to digital converters and transceivers. In case of a PIR sensor node, the sensing data is converted to a digital format, stored in the external memory, and analyzed by the microprocessor. The data is then sent to the base station using the transceiver. The transceiver can also receive commands and configurations from the base station. [57]

2.1 Star topology networks

A star topology is arguably the simplest topology type in wireless networking. A star topology network consists of a base station and nodes. The nodes communicate with the base station with no direct inter-node interaction. Wireless Sensor Network implementations that employ a star topology have been shown to be more robust, and have higher throughput, atleast in an outdoor environment. However, a star topology is commonly limited by range. However, some LPWAN (low-power wide-area network) networking protocols function in frequencies, which allow long range communication between devices and base stations.

An example of a star topology is displayed in figure 2.2 [47].

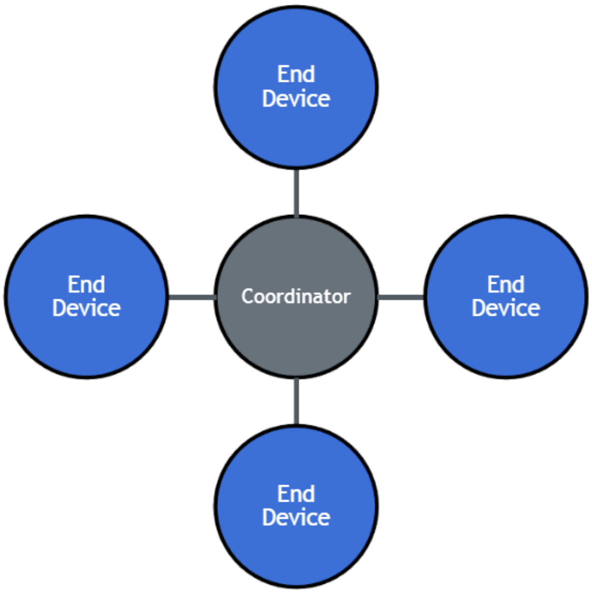

Figure 2.2: Star topology [47]

2.2 Tree topology networks

A tree topology is similar to a star topology in that each node has one route to the base station. However a tree topology supports limited inter-node communication. A tree topology network is an expanded version of a star topology network, i.e., the network can be spread wider by adding further nodes to the branches. An example of a network built using a tree topology is displayed in figure 2.3.

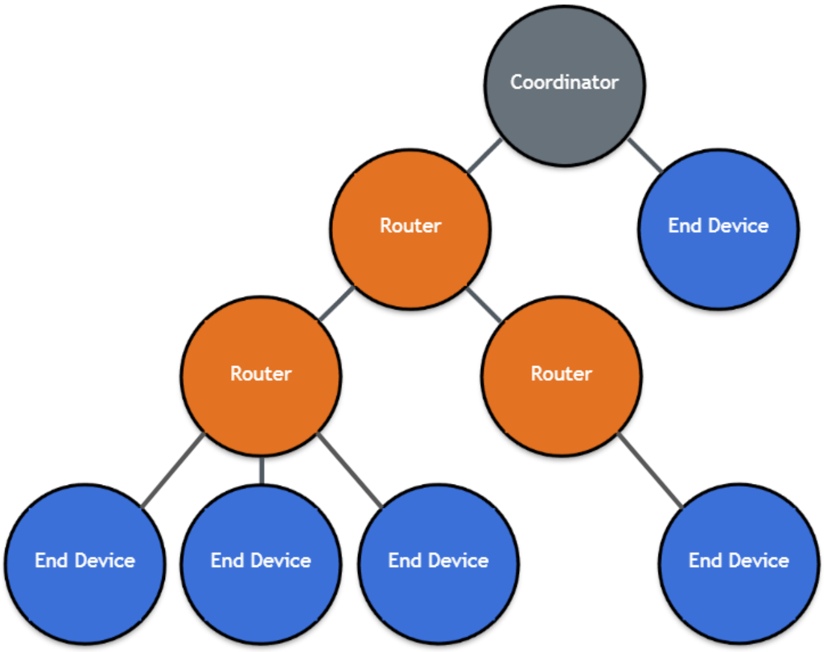

Figure 2.3: Tree topology [47]

2.3 Wireless mesh networks

A mesh network is a common multihop network topology choice for Wireless Sensor Networks. Distinctive features of a mesh network include a self routing network with direct inter-node communication. Mesh networks offer reliability by providing multiple routes between sensors. And can Additionally, they can be cheaper to implement, since they generally require fewer base stations to operate. However, multihop networks have been generally shown to have lower throughput, and higher packet loss and delays in an outdoor environment [57] [47].

Wirepas Mesh is mesh networking protocol. It is a good example of mesh networking technology designed to be used in WSN's. It is developed by a Finnish company going by the same name. Wirepas Mesh is a self correcting, bi-directional wireless mesh network (WMN). The goal of the Wirepas project was to create a de-centralized radio communications protocol that could be used on practically any device, independet of the radio chipset type or frequency used. According to the companys tech-report [65], the expectations have been met.

The mesh network is targeted towards IoT applications, with reliability and scalability as core values. According to the technical overview, there is no upper limit for the number of devices. The protocol has reportedly been used to create a network of 700 000 smart meters in a single mesh network, and has been proven to allow over a thousand radio transceivers per cubic meter to work simultaneously. Sparse installations are also possible. [65]

Implementations using WM protocol have achieved high energy-efficiency. Sensors, and data routing devices can run for years on a single battery, while using WM. An IPv6 router with BLE radios reportedly uses only 20Î¼A of power. Since the devices form the network, no further infrastructure is needed. A case study reported a public building using a single router for a network of 50 sensors, within a public building, in this case, a shopping center. [49] [65]

Wirepas core principles:

De-centralized

All devices are routers

Multi-gateway support

Collision free

Multi-channel

Hardware and frequency independence

In a mesh network, the nodes function as routers. They can independently optimized TX powers and frequency channels, aswell as select the most reliable neighbours. Sensor data is routed to a gateway, of which there can be many for load balancing reasons. The radio spectrum is devided in the time and frequency domains, to ensure no collisions. The system uses synchronized operations that remove communication overhead and an efficiently shared frequency band, which lead to increased capacity and the possibility for high device density. However, perhaps the most interesting factor in WM is the hardware independency. It enables service providers and hardware manufacturers to optimize for best performance across the design process.

The hardware used in the system developed for the experiment phase of this study uses a ZigBee based mesh network. Generally, the nodes communicate with the base station directly but an interjacent node can repeat signals between the base station and nodes otherwise out of reach.

Figure 2.4 shows an example of a mesh topology network.

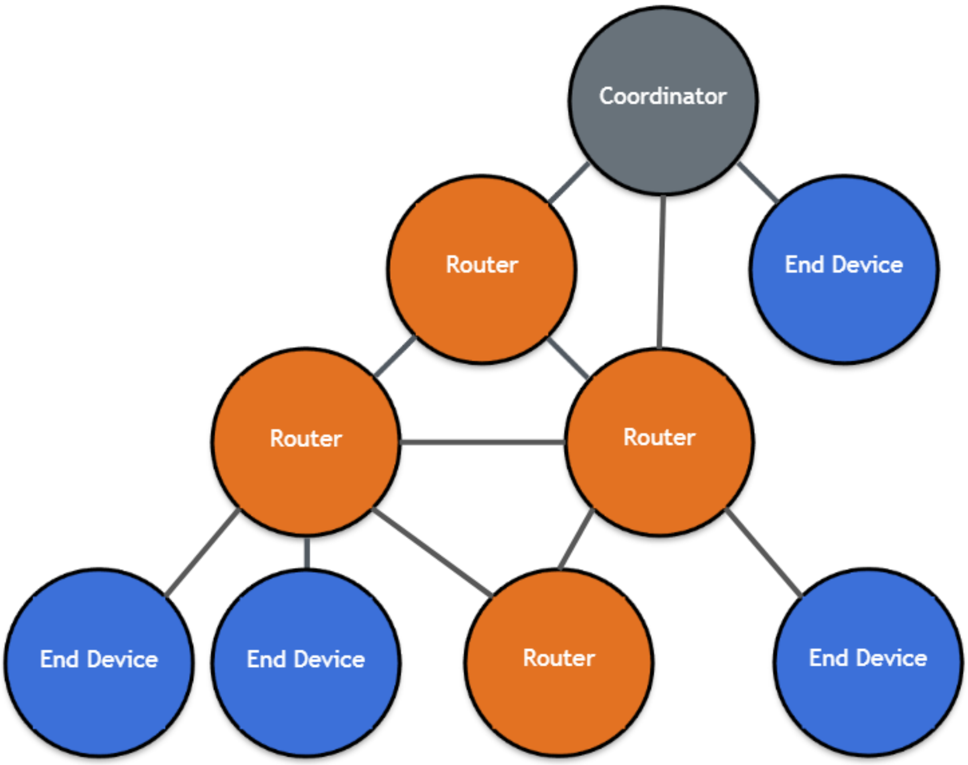

Figure 2.4: Mesh topology [47]

2.4 Bluetooth Low Energy

The original Bluetooth can be considered the standard for short range communication between end devices. It was developed together by Ericsson, Nokia and Intel in 1998. It was standardized under IEEE 802.15.1 in 2002. It is currently supported by most smart phones and laptops and it has become the technology of choice for almost all wireless peripherals, such as portable audio equipment. [63] [52]

In WSN systems, Bluetooth LE is commonly the protocol utilized between an end device and a gateway or base station. The internet connected gateway or base station gathers the data from the sensor nodes via Bluetooth LE. The gathered data is then accessed and analyzed by administrators over, a LAN connection using IP routing. [52]

Bluetooth Low Energy (LE) has previously been proposed as the missing link for wireless sensor networks. This relatively new protocol, that is also known as a "Smart" protocol, is a part of Bluetooth 4.0. It is a very viable network technology for the growing IoT. BLE offers a bandwidth of circa 200 kbps, which is sufficient for most common sensor network applications. [63]

Bluetooth LE utilizes the 2.4GHz Bluetooth radio band. The new low energy technology adaptation is compatible with devices that support Bluetooth 4.0. Most smart phones developed after 2011 are built with this support. Unlike the 79 channels used by regular Bluetooth, LE uses only 40 channels. With a transmit output of 10mW, it was designed to allow running a wireless sensor for at least a year on a single, approximately 200mAHr coin cell. Further analysis of the power consumption of Bluetooth LE implementations with various modulation schemes can be found in [52]. [63]

Bluetooth LE is a competing technology to ZigBee. Bluetooth is a widely used protocol and due to this it may become the standard for wireless sensor networks in the future.

2.5 ZigBee

In the case of WSN's, ZigBee is similar to Bluetooth LE in its functionality; sensors nodes communicate with a base station that is then connected to the internet or a Local Area Network. Although ZigBee is not a standard commonly supported by every-day devices such as laptops and smart phones, it is very well suited for wireless IoT devices. Therefore it is commonly utilized in WSN systems and similar implementations [20] [69] [26] [13] [67].

ZigBee is based on the IEEE 802.15.4 standard. Its advantages include low cost hardware, high reliability and low complexity. However, zigbee is restricted by its communication capacity and, in the case of WSNs, it is also limited by the computation power of the nodes. The latter is naturally not a fault of ZigBee itself, but it does introduce problems. The aforementioned restrictions make using conventional security measures difficult. Methods, such as public key cryptography are not well suited for ZigBee WSN systems.

The security limitations of ZigBee, and WSNs in general are a real problem. In the medical field, for example, private patient data gathered by sensors may be in jeopardy. [48] and [50] discuss security threats of ZigBee and platforms that use it as their main network protocol.

2.6 5G

5G is the new generation of cellular networks. The standardization of its application support requirements is on-going at the Association Project Third Generation (3GPP). Device-to-device or Machine-to-machine communication is one of the applications that will likely have a substantial impact on WSN systems. [32]

5G will likely allow easier implementation of city wide networks of IoT systems affecting water supply and public safety systems. Studies of simulated WSNs with early versions of 5G networks and different routing protocols have already been done in [30]. [32]

Fully IP-based networking using 5G would simplify development of WSN systems significantly.

2.7 EnOcean

EnOcean is a German networking technology that has a growing userbase in Europe and the rest of the world. With innovative energy harvesting methods as its core objectives, EnOcean stands out as a significant competitor to technologies such as ZigBee. The harvesting methods include kinetic and ambient light energy harvesting. EnOcean is described as an ultra-low-power wireless communication technology. During the preliminary review phase EnOcean was one of the top choices. However, finding a supplier proved yet problematic. [29]

Developers of EnOcean had to make specific design choices to achieve the inherent ultra low energy usage. Firstly, the messages are formated to be extremely short. Other choices deemed necessary, but reasonable include not implementing message acknowledgements or access regulation methods such as Carrier Sense Multiple Access (CSMA). The two aforementioned factors can lead to increased message collisions and transmission errors. [29]

The drawbacks are combated by sending each message multiple times. The collision rate was determined to be 11% for 20 devices, each sending messages with one second intervals. Due to the method of repeating messages, a success probability of 99% was calculated, which can be expected to be enough for non critical systems. [29] [46]

2.8 LPWAN

LPWAN stands for low-power wide-area network. SigFox, and other similar technologies, such as LoRa, INGENU and TELENSA, offer a wide area coverage for Wireless Sensor Networks and other similar, low energy systems. The data is sent straight to a remote base station and server. This means that the client only needs to locally install sensor nodes. This lowers installation overhead and simplifies the installation process significantly. However, they are limited by relatively high delays and a low number of messages. A SigFox node, for example, is limited to 140 messages per day. In the use case of space utilization monitoring however, the delays are insignificant. [41]

LPWAN devices commonly send data in ultra-narrow, sub-GHz carrier band to a remote base station. The data can then be pulled from the cloud and accessed by administrators. [41]

2.9 Software

A WSN systems software generally encompasses the firmware for the hardware, a database solution for the data gathered, software for visualizing the data, and software for controlling the hardware, i.e., a controller. The firmwares of systems used in this study were proprietary and were not modified during the study. The controller generally includes a backend and a user interace (UI). A status page displaying the real-time reservation status of rooms is a significant part of the UI. Additionally, a database for storing the sensor systems configurations and other system specific information is a relevant part of the controller backend. The implementation discussed in chapter 5 incldues further descriptions of the selected software components.

The database choice for the sensor data is an important one. For time related data, such as space usage data, a time series database (TSDB) is the most obvious choice. Time series databases are optimized for handling time series data, i.e., data points indexed by time. [66] describes time series databases as follows: "Time series database is a set of time series, each of which is an order list of real values measured at equal intervals.". In the case of this study, the time series data consists of presence measurements. The systems periodically stores the presence state received from each sensor. This state describes either that presence was recently detected (1) or presence was not recently detected (0). [24]

There are currently many TSDB alternatives for developers, such as InfluxDB, Druid, TimescaleDB, Cube Prometheus and Graphite. InfluxDB has the most active community, and is the most popular MIT lisenced open source TSDB available, according to the current amount of stars and forks on GitHub. [24]

Telegraf and MQTT are good choices of brokers for the data. A broker, in this context, describes the tool that handles gathering the data from the sensor system to the InfluxDB, or other similar database. Telegraf is developed by influxdata, and is likely the best match with InfluxDB. Grafana and Chronograf are both compatible with InfluxDB. Chronograf is developed by influxdata, however, the more comprehensive functionalities of Grafana may make it a more suitable alternative in some cases. [24] [3] [23] [17] [22]

The development of the controller software included choices of technology for both the frontend and backend. JavaScript (JS) stands as the most used programming language according to many GitHub metrics [16]. These can be considered to represent the demand and availability of information concerning the language. Therefore, technologies that utilize or are built on JavaScript are safe and reasonable alternatives. For backend development the popular Node.JS is a sensible choice, although there are a wide array of options available. For the configuration data, authentication system and other system relevant information, a non relational database, such as the open source MongoDB is a reasonable solution. Alternatively, another non relational database, such as Cassandra or a relational database, such as MariaDB could have been used. [36]

When developing a control panel frontend for administrators, simplicity and functionality is key. Technology choices are therefore mostly up to preference. However, due to the popularity of JS, the React library was used to improve the interactive UI. React allows efficient UI updates according to state changes, which can be usefull in many cases. [16] [40] [14] [15]

2.10 Conclusion

The most widely used and optimal networking technologies were determined to be ZigBee and LBE. They present adequate data transfer rates and are optimized for low energy communications. ZigBee and Bluetooth are both continuously developed and comprehensively studied. They have a significant representation in WSN systems overall. They are reliable and handle transfer errors well. LBE could have easily been chosen as the transfer protocol instead of ZigBee, if suitable hardware were as easily available.

Additionally, the emergence of 5G presents promising implications. LPWAN technologies were also considered as prominent choices due to the ease of installation of LPWAN sensor systems. The justifications behind these choices are discussed in detail in chapter 5.

There is a multitude of software solutions available for developing systems around existing hardware. The most notable choices were the selection of the data broker telegraf and the TSDB InfluxDB. Both function as intended and are configurable for specific cases. InfluxDB has a relatively large userbase and good documentation. Software aspects and best practices evolve rapidly and it is advisable to do further research before beginning a similar project in the future.

Chapter 3

Relevant hardware solutions

This chapter discusses hardware methods of detecting people and movement in an environment. In the scope of this study, the detection is done in order to collect occupancy data for workspaces and convey the real-time reservation status to users. The detection can be executed using a multitude of technologies, such as PIR sensors, depth sensors or standard cameras utilizing machine vision techniques. The technologies are selected based on their prevalence in the field. [34]

The most prominent methods continuously observe the space itself. Observing people flow through passageways can be implemented with relatively high accuracy [35]. However, a single miscalculation leads to the system having false information of the actual reservation status within the space in question. This means that further sensors would be required to oversee and correct the reservation infromation. This study aims to present a solution for assessing real-time reservation status of a workspace, and giving reliable information on whether a room or workspace currently has occupants. Therefore, counting people flow was deemed an unsuitable method.

Some implementations use different sensor types in unison to achieve higher accuracy. This is a promising and logical approach to the problem. Studies have been done on sensor fusion and some hardware providers already have solutions that implement forms of sensor fusion. The battle for the best hardware solutions is on-going. During the study, multiple companies within the field were contacted. The selection of hardware solutions is wide and many companies also provide software for data visualization and analysis, in addition to the hardware. [9] [68] [12] [31] [4]

3.1 Passive infrared sensors

A PIR sensor passively observes the space in front of it. It is activated when an object that has a different temperature than the environment is moving in its' field of vision. PIR sensors function by detecting miniscule voltage changes caused in the wiring by this by movement. Contrary to many other directional sensing systems PIR sensors do not need daylight or an additional signal emitting from the target. [8]

PIR sensors also have very low energy consumption. This is due to them only being activated when movement is detected. This advantage has been utilized to minimize electricity and memory space usage of a security system [8]. Most of the PIR sensors available have battery lives measured in years. This leads to low maintenance overhead caused by the hardware.

Generally, PIR sensors are described to have a range of 10 meters and a field of vision of 110°, which is enough to cover most office rooms. However, their accuracy diminishes at high ranges. [1]

During the review phase of the study, it was concluded that PIR sensors are the standard for presence monitoring within the industry. PIR sensors are cost efficient and sufficiently accurate. The cost of sensors considered during the review phase were under 50 euros. PIR sensors have been used to implement power saving solutions and intrusion detection systems [18] [53]. PIR sensors have also been used in fusion with other sensor types, including ultrasonic sensors [4]. Using multiple PIR sensors can also give approximations for people count and location within an area. [27]

3.2 Signal detection

Detecting signals, such as those emitted by Wi-Fi or Bluetooth devices is a functional way of detecting presence. Carry-on bluetooth tags are one implementation method. Mobile phones with Bluetooth turned on can also be used for this purpose. However, these solutions bring variables that are user specific and not always controllable or they require additional hardware.

Utilizing bluetooth beaconing for indoor user localization has been studied, and is a relevant method in some use cases [11]. Using Wi-Fi signal strength for assessing the location of devices within a space has been studied aswell. In [58], researchers were able to achieve accuracies of over 85% using Wi-Fi signals, with no additional hardware. In the same study it was stated that similar methods applied to Bluetooth signals achieved accuracies of over 96%. These results indicate that accuracy is not a concern.

3.3 Depth sensors and radar systems

A depth sensor measures the distance between itself and objects in its field of vision. I.e., a signal is sent and the propagation times of the reflecting signal are analyzed. Even though PIR still seems to be the hardware standard among suppliers within the industry, depth sensors have also been studied as tools for people detection. Using multiple overhead depth sensors and depth sensors alongside cameras have been studied with promising results [64] [12].

Radar systems function in similar way. A signal (ultrasound or electromagnetic radiation) is sent and the reflection times are analyzed to gain information about objects around the sensor. Technologies include pulse-Doppler radars, which inspects the targets velocity. In [9], researchers used RGB-Depth (RGB-D) sensors alongside radar systems to detect fall accidents in elderly people. The transmit powers of common radar systems are low enough to not be dangerous to humans. The transmit powers are comparable to those of Wi-Fi routers. This, however, also means that the power consumption is commonly too high for low maintenance battery-powered systems. [9]

Both the systems present no privacy concerns. For RGB-D sensors, if the sensors depth data is used exclusively, the system is completely anonymous. Depth and radar sensing are also not affected by changes in lighting. [9]

3.4 Cameras

An RGB camera is essentially a standard camera. RGB cameras are arguably the most cost efficient way of accurately detecting movement and counting people. A single camera can be used to count people with high accuracy [5]. RGB cameras have even been used to detect the heart rate of multiple people in their field of vision [38].

RGB camera systems may require tuning due to changes in light levels in the observed area. Although perhaps the most accurate way to detect movement if well adjusted, they can present privacy concerns. This was also the reason why cameras were not chosen as the solution for the implementation developed during this study. Installing cameras to the work spaces was deemed unreasonable.

However, privacy conscious solutions for human detection have been developed. One solution uses low resolution video, as in [34]. In [7], video feed was analyzed by splitting crowds into segments. A Gaussian process was then used to estimate the people count in a segment. In [45], researchers used Microsoft Kinect hardware and attachable 3D printed "privacy optics" to blur out the image, while retaining the ability to do motion tracking with the hardware. With the aforementioned method, anonymity is achieved before any data is stored. Other privacy conscious solutions have also been studied. [56]

However, cameras may not always be suitable. Even if the underlying technology guarantees anonymity, it can be seen as questionable to install cameras in some environments.

3.5 Sound sensors

Detecting occupancy levels with sound alone is problematic, due to false positives caused by bypassers. This was especially true for this project, since in the experiment environment it was ruled that the doors are to be left open if the space is not occupied. Additionally, a person working alone in a workspace may not cause sound for long periods of time. Overall, the method was considered improbable to work alone. [19]

Background noises can be enough to cause false positives in presence detection. Additionally, a person working alone may sit quietly enough to not allow using a higher volume threshold for determining occupancy. Even small movements may not provide enough sound to determine presence accurately. Thus, using other sensors is likely generally a better option even when movements are infrequent. However, using sound to detect context-dependent events and classifying sounds has been studied in [19] and [54].

Sound sensors can still be usefull. The cost of a microphone is in the range of a few euros. Although using sound alone can be less than effective without high quality context analysis, using sound in addition to e.g., a PIR sensor can provide additional accuracy. The Yanzi motion+ sensors, for example, are equiped with a sound pressure sensor and a PIR sensor [68]. Using sound together with PIR has also been studied in [44]. [28]

3.6 Conclusion

For the purposes of this study, PIR was selected as the hardware method. It can be considered the most suitable hardware method for purely assessing the state of presence within a space. The accuracy of these sensors was deemend satisfactory. Although small movements are not always detected, the reservation status of a room can be assessed on a satisfactory level using a longer observing window. Even a single sensor provided adequate accuracy. In order to improve the accuracy and reliability of the sensor system, installing multiple sensors per room should be considered. Using sensor fusion is also a reasonable next step. PIR sensors are widely available and require little maintenance, due to low energy consumption. The justifications behind this choice are discussed in detail in chapter 5.

Cameras are likely the most accurate method, but they unfortunately bring forth security and privacy issues. Due to these issues, they were deemed unsuitable for this project. Further examining the use of RGB cameras considering privacy would be beneficial. Other methods had their own considerable drawbacks. Radar systems were deemed impractical due to energy consumption. Counting people flow was deemed impractical for the purpose of detecting presence. Sound sensors were deemed to be error prone when used for presence detection, however utilizing them in unison with other sensors could provide additional accuracy.

Chapter 4

Environment and methods

This chapter covers the experiment environment and methods selected for the final implementation and the reasons for choosing them. Both the hardware and software choices are covered here.

A low energy, wireless sensor network (WSN) of presence sensors was used to gather occupancy data. The selected sensing method was PIR, due to privacy and accuracy factors. WSN systems are at the core of the growing Internet of Things (IoT). One sensor was installed in each ad-hoc workspace. [62]

4.1 Experiment environment

This section describes the experiment environment and factors related to different space types. The experiments were done within the Aalto University School of Science. The subjet rooms were ad-hoc rooms of varying sizes. Additionally, other environments and workspaces that may be relevant in similar cases are discussed concisely.

4.1.1 Experiment rooms

The implemented solution was put to test in five ad-hoc work spaces at Aalto University. An ad-hoc room is a workspace that can not be reserved in advance. A space is reserved simply by walking in to an empty room. The rooms included three smaller rooms with seating for only a few people, with facilities comprising of a sofa and two stools. Other rooms inlcuded more typical facilities: a large table in the center, with 6-8 chairs around it.

Figure 4.1 depicts the floor plans for each room. Their sizing is close to the relative sizes between the rooms. Additionally, the positioning and direction of sensors in each room is depicted by a black rectangle and outward stretching circular lines presenting the direction of the sensor's vision.

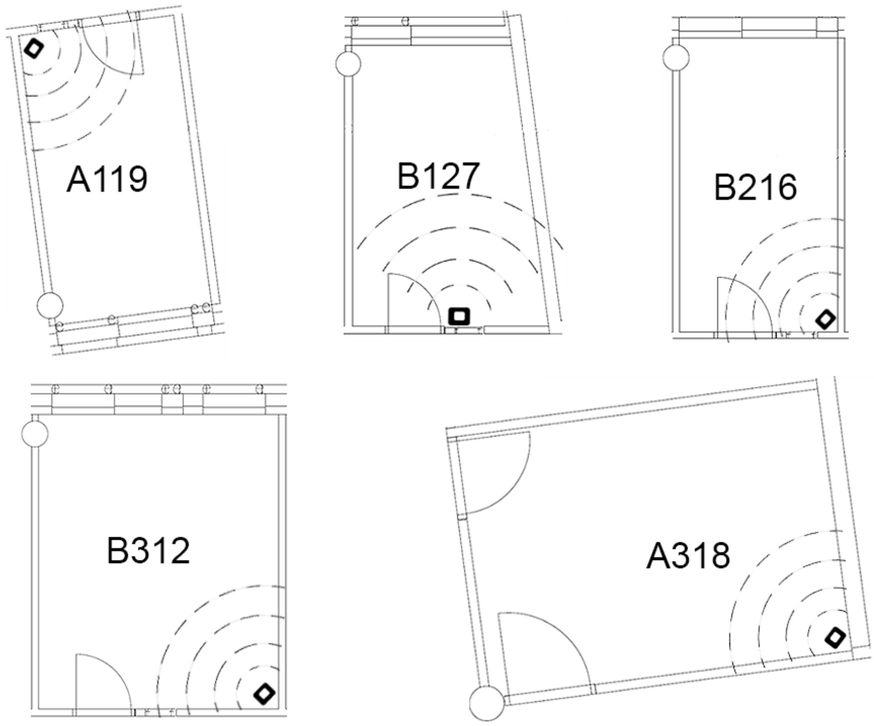

Figure 4.1: This figure shows the floorplan of each room and the positioning of sensors.

The smaller rooms have a disadvantage in that they commonly have fewer occupants. The room is thus expected to have less movement within, as implied by the results discussed in 6. However, since the room is smaller, the people inside are more likely to be closer to the sensors. This leads to a higher probability of movements being detected by the PIR sensors.

4.1.2 Other relevant environments

Standard pre-reservable offices and meeting rooms receive lesser practical benefits from space occupancy monitoring. Since these spaces are reserved beforehand and the reservation status is determined by those reservations, it is not as beneficial to the users to know the real-time circumstances within the work space. However, meeting lengths can not be perfectly predicted and overbooking, i.e. reserving a longer time frame than likely necessary, is a real problem. The effects of overbooking include diminished efficiency in space utilization. Inquiring the reserver whether they would like to end their reservation prematurely, according to the real-time status of the room, could be a way to mitigate this problem. Such an inquiry could relatively easily be implemented in to existing reservation software, utilizing the data gathered by a real-time space occupancy monitoring WSN system.

Mobile work stations are similar in practice to the ad-hoc rooms. They are intended to be used in a "first come first served" manner, i.e. a workstation is reserved by sitting down. Mobile work stations can be described as larger tables with multiple computers available for any employee. These areas often have constant traffic and people moving around. Multiple service providers in the industry present solutions for these kinds of workstations during the review phase. These solutions commonly included PIR sensor attached under the table at each work station.

4.2 Selected hardware methods

During the technology review phase discussed in chapter 3, multiple different hardware technologies for assessing space occupancy were considered. These methods include RGB cameras, sound sensors, doorway laser sensors, PIR sensors, signal detecting methods, and radar systems.

Ultimately, a network of Passive Infrared Sensors (PIR) were selected for assessing the level of occupancy in the experiment environments. PIR sensors are energy efficient and provide sufficiently accurate information. They are also the most common solution for companies working on space occupancy monitoring, as established during the review phase of this study. Two technology providers were selected for comparison. One sensor from each provider was used per room. Philips HUE system was selected as the hardware for the inhouse built solution. The Philips' HUE family of products have an extensive userbase and widescale availability. Additionally, they are relatively cheap and easy to install and provide an open API for developers. Reference hardware and software were provided by third parties for the turnkey solution. [43] [42]

In addition to the Hue sensors, Hue bulbs were installed outside the rooms in question. The bulbs were programmed to change their color according to the current reservation status of the room. Figures 4.2, 4.3, and 4.4 display the Philips Hue sensors, bridges and bulbs used in the implementation, respectively. Figure 4.5 displays the reference systems sensors. The final hardware implementation is described in detail in section 5.1.

Figure 4.2: Philips Hue, Motion sensor (046677473389) [42]

Figure 4.3: Philips Hue, Bridge (046677458478) [42]

Figure 4.4: Philips Hue, E27 White and color ambiance bulb (8718696592984) [42]

Figure 4.5: Sensors used in the reference system.

The Philips Hue family is not problem free, however. The most crucial hindrance of the Hue system are its security flaws. The only password dependent security measure shielding the system from abuse is a password protected WiFi network. In addition to this, each bridge will have a list of users with a 40 character username that must be used as an URL parameter when sending commands to the bridge and further to the sensors. However, these usernames can be generated if the IP of the system is known.

A Philips HUE system may also be compromised by adding a malicious lamp within reach of the network, giving an attacker control of the lights, and possibly making the attached bridge a part of a bot network, as shown in [51]. The bridges in this implementation are therefore connected to a separate Local Area Network (LAN) and not connected to the internet. This will prevent the largest threats of such attacks and is advicable. However, malicious lamps may still be able to reach other HUE systems with their ZigBee messages. Results from simulations done in [51] argue that 15,000 randomly located lamps in an area the size of Paris could be enough for one malicious lamp to infect them all.

Using Yanzi hardware was a part of the objectives, but problems with vendors in addition to the limited time frame lead to them being left out of the experiment, unfortunately. Yanzi uses pure IPv6 communications with security as one of their main cornerstones of development. For future work, it is advisable to lean towards Yanzi over Philips HUE in its current state. [50] [51] [44] [68]

4.3 Selected software methods

The software was developed utilizing the open API of the Philips HUE system. The Philips Hue system is built with developers in mind, and there is a rather extensive user base. Due to this, a considerable amount of information is available online. Hue hardware employs ZigBee networking, which is a very suitable technology for a WSN implementation. [43]

The selected software solutions, most relevant libraries and tools were chosen according to factors stated in section 2.

The developed software system consists of the following parts:

Controller UI: React, HTML

Controller backend: Node.js, MongoDB

Data broker: Telegraf

Database for sensor data: InfluxDB

Data visualization: Grafana

The software implementation as a whole and its functionalities are described in detail in section 5.2. Additionally, some usage examples are depicted there. The development process is also described concisely.

Chapter 5

Implementation

This chapter explains the final implementation that was used for the data collection. Both the hardware and software implementations are discussed in detail. The data acquired is reviewed and analyzed in chapter 6.

5.1 Hardware

This section describes the process of implementing the hardware components used in the experiment part of this study. Hardware specific features are discussed further and in depth.

The in-house implementations hardware consisted of a Philips Hue Bridge, five Philips Hue Motion Sensors, and five Philips Hue Bulbs. The comparative, turnkey system used ready to install sensor hardware. Figure 5.1 shows the bulbs installed outside the rooms. The bulbs in question require an E27 socket. [42]

Figure 5.1: Bulbs installed outside the rooms which were selected as the experiment environment.

When selecting the colors it is advisable to consider the distinctiveness of the colors. A person with deuteranopia, i.e. red-green color blindness, for example, may have problems discerning the colors. For the red and green colors, we used hue values of 65280 and 25500, respectively. A hue variable can be passed to the bulbs in a HTTP message to adjust the color. It was determined that the green, i.e. the value 25500 was easily distinguishable, but the red, value 65280 was not optimal. Making it a brighter shade of red was adviced by a person with deuteranopia. Information on selecting color schemes with attention to this factor can be found in [60], [60] and [21]

The sensor hardware used was small and compact. The reference system's presence sensors are comparable to a typical roof attachable fire alarm, with a circumference of approximately 8 centimeters. The Philips Hue presence sensors measure at 55 mm vertically and and horizontally, and have a height of 20 mm [42]. The Philips Hue bridge measures at approximately 9 cm vertically and horizontally, and has a height of approximately 2,5 cm [42]. All of the devices are light but sturdy and easy to handle.

The Philips Hue bridge has two connections: a power input and a network cable connection for an RJ-45 cable. The included power adapter has the following specifications: 100-240V AC / 50-60Hz, output voltage: 5 V DC 600mA, standby power: 0.1 W max. The device itself has a maximum power consumption of 250mA. A 1 meter long ethernet cable is also included in the package.

Installing the reference hardware supplier's sensors is extremely straight forward. The sensors can be attached with velcro, or with screws straight to the wall. The company also supplies the client with mounts, that make installing the hardware in room corners easier. In the experiment implementation the sensors were installed in the corner of each room, at a height of approximately 1.5 - 2 meters.

As the reference system's sensors are directly connected to external base stations using LPWAN networking technology, no additional user installed bridges or base stations are needed. However, in some cases signal propagation may be prevented by strong structures. In such a case an additional repeater can be added. The repeater strengthens the signal between the sensors and the external base stations.

The Philips sensors were attached similarly to the reference hardware ones. The attachment was done as close to the comparative hardware as possible, to minimize biased results. The Hue sensors are distributed with a wall mount of their own. The wall mount is a simple round stub, with one screw hole and a magnet. The sensors are then attached to the magnet, and can be easily removed if needed. The extremely easy removal of the sensor from their mount presents an increased risk of theft in public spaces, however. [42]

A philips Hue Bridge was installed in a server room, as close to the center of the sensors as possible. The distance between the bridge and the furthest sensor was under 30m. In cases where the range would not be enough, an additional Hue bulb can act as a repeater when attached between the problem light and the bridge. It was recognized, that a stairway with thick concrete walls was enough to prevent communication between a sensor or a light and the bridge. However, the signal was able to penetrate thinner inner walls and floors without problems. Planning the positioning of the bridge accordingly is thus reasonable, in order to avoid additional hardware costs. [42]

Adding new sensors to the system requires the user to pull a plastic cover from the sensor which results in the battery being plugged in. The sensor can then be detected by pressing the "Scan" button in the control panel, and after a while using either the "All sensors" or the "New sensors" button (shown in figure 5.3) to find the attached sensors. The procedures of adding a new Hue bulb to the system are equivalent, except instead of plugging in a battery, the bulb must be screwed in to a E27 socket. Additionally, the user is required to use the buttons refering to lights. Further software related functionalities are described in section 5.2.

5.2 Software

This section describes the software implementation developed for the system. The software can be used to control the Philips Hue Hardware and for visualizing the gathered data. Figure 5.2 displays the system and its software components. The code can be accessed at [61].

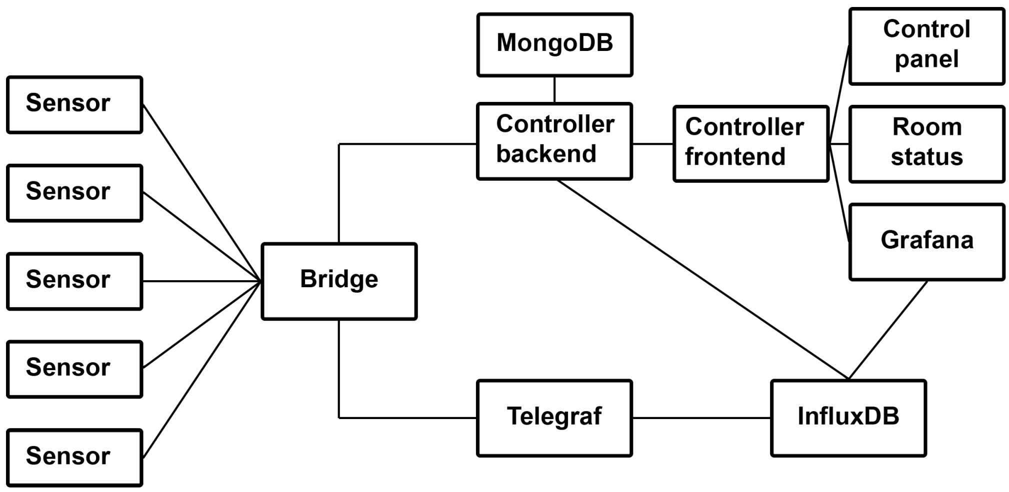

Figure 5.2: Concept map of the system with all its software components and their relations.

It is rational to approach understanding the system by describing what happens to the sensor data. We will do this by looking at figure 5.2. On the left side of figure 5.2 we have the sensors. They gather presence data, which is in boolean form, i.e., true or false. A true state for one sensor implies, that the sensor has detected movement, i.e., there has been a moving heat signature in front of the sensor within the last few seconds. If no movement is detected in approximately 5-15 seconds, the state of the sensors' presence value is returned to false, i.e., no presence is detected. The bridge gathers this data via ZigBee messages from all the sensors. The state of all the sensors associated with the bridge can then be queried from the bridge.

The bridge is connected to a LAN and can send and receive HTTP messages. Interactions with the sensors happen invariably via the bridge. There are two system components that send and receive messages from the bridge. Firstly, Telegraf [23] acts as the agent that periodically collect the presence state of each sensor via the bridge, and reports, i.e., stores the data to an InfluxDB instance. A five second interval was used in this implementation. Secondly, the controller manages the settings and configurations of the sensor system. The configuration files of each bridge and associated sensors and bulbs are stored in the MongoDB as backups. The controller's frontend, i.e., the control panel, is the user interface (UI) for managing these configurations. The frontend consists of three parts. A control panel for managing preferences and configurations, a status page for displaying current occupancy status of each room, and a data visualization dashboard built using Grafana [17]. The control panel is displayed in figure 5.3. Figure 5.4 displays the status page. Some examples of how occupancy data can be displayed using Grafana can be seen in figure 5.5.

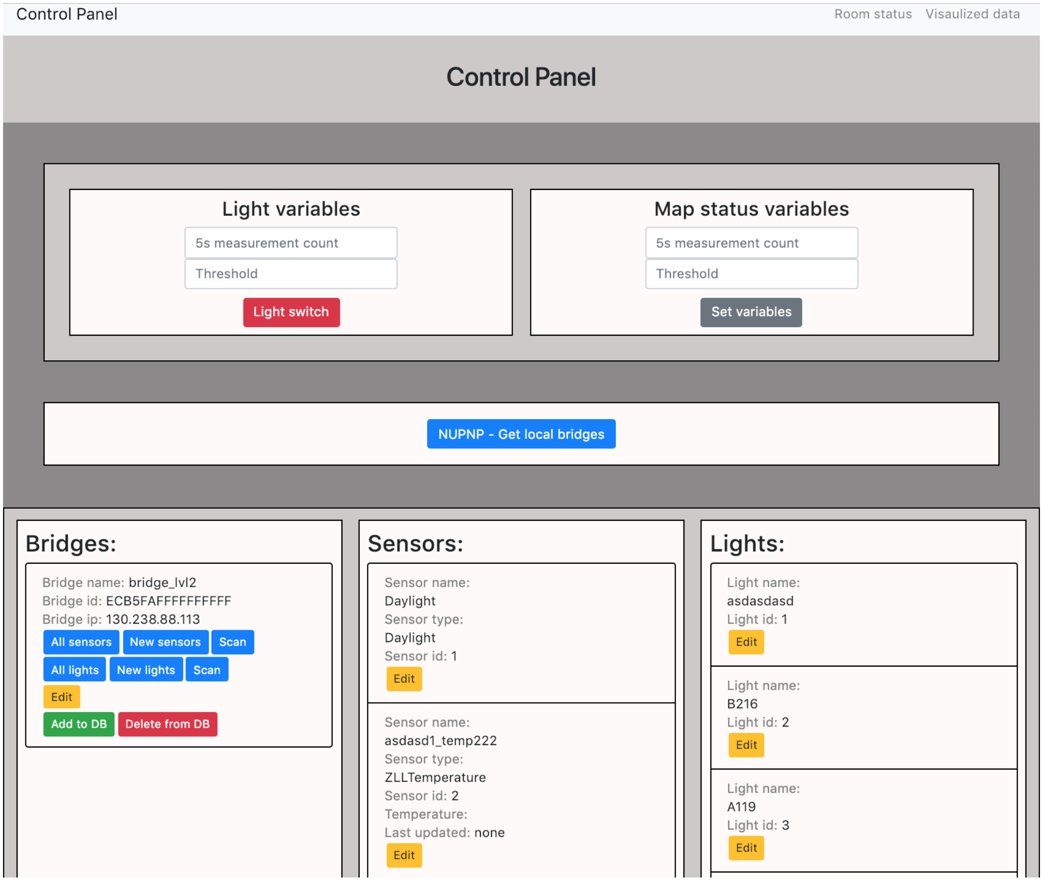

Figure 5.3: The control panel is part of the UI. It is used for managing the configurations of the system.

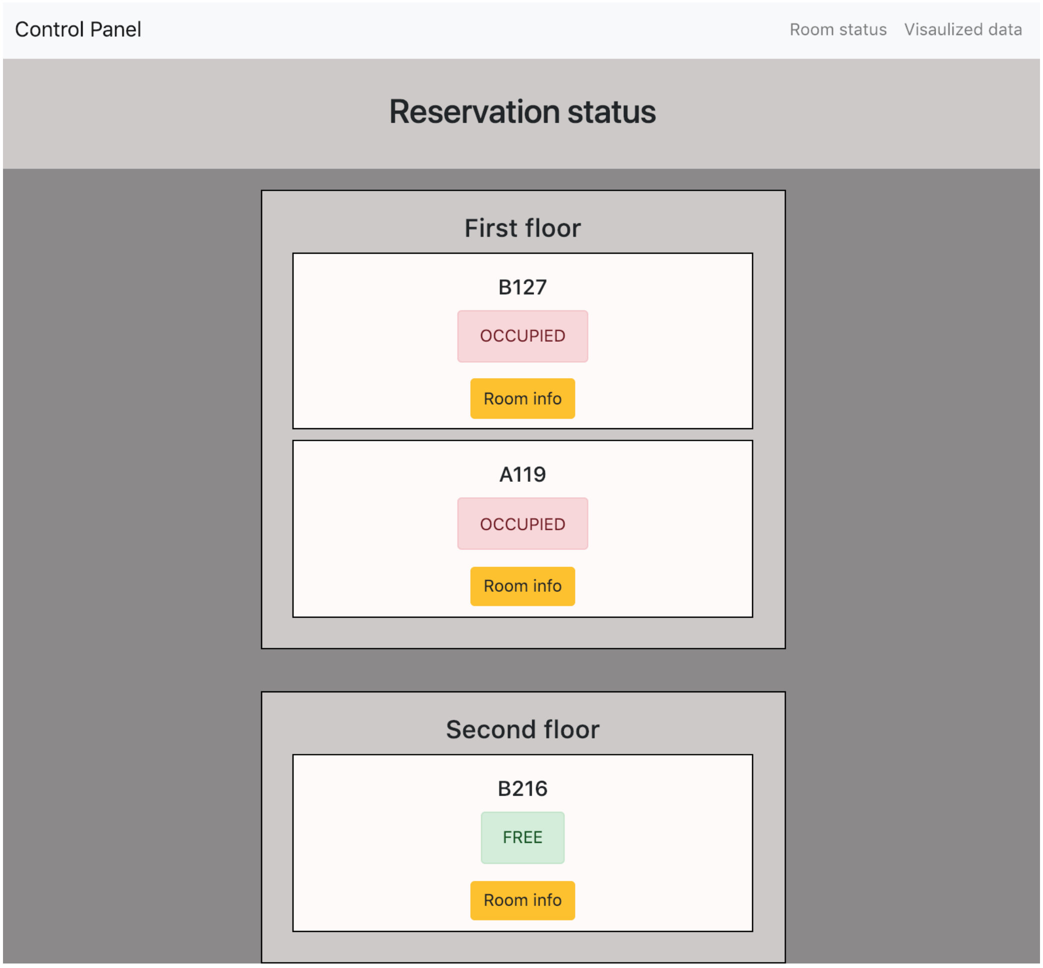

Figure 5.4: The status page displays the current reservation status of each room.

Figure 5.5: Visualizing presence data with Grafana.

The networking configurations between the different hardware components (For the hardware used, this means communications between a Hue Bridge and Hue sensors) are proprietary and were not adjusted in any way. The technologies used in the experiment phase of this study implement different networking methods for their inter-device networking. The Philips Hue system builds a short range, ZigBee based network, where all the sensors communicate with a locally installed bridge, which has a wired LAN connection. The Hue WSN nodes can work as extenders, thus creating a mesh network, with nodes essentially forwarding messages from the base station to other nodes. The reference sensors from the outside supplier use a Low Power Wide Area Network technology (LPWAN), that is considered a star topology network. [41]

5.2.1 Control panel

The control panel (figure 5.3) is a part of the developed front end. It was built using React and HTML. The control panel is essentially an UI for managing the Hue hardware's configurations and the variables that control the lights and status page. The former includes configurations, such as the name of a bridge, sensor or light, and the sensitivity of a sensor. The latter variables include a "5s measurement count", i.e. the windows size. If the "5s measurement count" is set to, 60, the window size will be 5 minutes. There are separate variables for the light management and status page, although they will not commonly differ. Additionally, a room scheme was created, that included information on the facilities and capacity of a given room.

The system uses the name of a sensor and light to connect them. I.e., the name of a light bulb must match the relevant sensor's name for the system to adjust the color of the right bulb. During the experiment, the name or code of a room, i.e. "B216" was used as the name for the sensor and light installed in the room.

5.2.2 Status page

The room status page (figure 5.4) (also referred to as "map status page") is built using the same technologies as the control panel UI. The status page's functionalities are straightforward. When on the page, the backend is periodically queried for the status of each room, which is determined by data retrieved from the InfluxDB instance. The behaviour, i.e. the window size and sensitivity of status changes is based on the "Map status variables" shown in figure 5.3 and discussed in the previous section, section 5.2.1. Creating an API for the current reservation status was one of the main objectives. The status information can then be displayed in other reservation systems, such as mobile applications, as needed.

A user visiting the status page has the ability to request further information about the room, by pressing the "Room info" button, shown in 5.4. This information currently comprises the rooms seating and facility information. Additional information about rooms could be added.

5.2.3 Grafana, InfluxDB and Telegraf

Grafana has direct support for InfluxDB and is a great tool for visualizing and analyzing large data sets [17] [24]. InfluxDB can be added as a data source, and the contents of the database can then be visualized, according to user created queries. A dashboard can be saved with desired query results visualized with a variety of graph types. In the software implementation, grafana was used as is. I.e., the grafana dashboard was displayed on its own default page and it was running on a different port to the rest of the frontend. A grafana dashboard can also be displayed in an iframe on a website. This can be considered more userfriendly, and a cleaner solution.

Grafana queries the InfluxDB for current and historical presence data. InfluxDB can be queried directly from the dashboard, and drawing different graphs is made simple. InfluxDB is queried in a similar fashion to many sql databases. I.e., the query is built in the form showin in listing 5.1. It is naturally essential to handle backups for the gathered data when going into production. [24]

Listing 5.1: Basic query format for InfluxDB

```
SELECT <field_key>[,<field_key>,<tag_key>]
FROM <measurement_name>[,<measurement_name>]
```

Some further examples of grafana and InfluxDB usage are depicted in listings 5.2 and 5.3. Listing 5.2 depicts an example of an hourly average queried by first grouping in 5 minute segment maximum values and then by one hour segments. Listing 5.3 is an example of a continuous query.

Listing 5.2: Hourly average of "state_presence" values grouped by 5 minutes

```
SELECT mean("max")
FROM (
        SELECT max("state_presence")
        FROM "http"
        WHERE $timeFilter AND ("name" = 'B216')
        GROUP BY time(5m))
WHERE $timeFilter
GROUP BY time(1h)
```

Listing 5.3: An example of a continuous query. This is used to group larger data segments into smaller portions. This query groups 5 second booleans called "state_presence" into 5 minute segments with the maximum of said key's values. Continuous queries create a dataset of their own in to the database, that can later be queried.

```
CREATE CONTINUOUS QUERY presence_5min
ON test
BEGIN
        SELECT max("state_presence") AS "presence_5min"
        INTO presence_5min
        FROM http GROUP BY time(5m), "name"
END
```

Below are some listings (5.4 and 5.5) of queries used in the Grafana dashboard.

Listing 5.4: Selecting the average reservation status from the previous hour with a 5 minute window, for a room named B216.

```
SELECT mean("max")
FROM (
        SELECT max("state_presence")
        FROM "http"
        WHERE ("name" = 'B216') AND (time > now() - 1h)
        GROUP BY time(5m)
         fill(null)
)
```

Listing 5.5: Daily averages for the hours between 8-15, after 2018-12-12.

```
SELECT first("mean")
FROM (
        SELECT mean("presence_1h_mean")
        FROM "presence_1h_mean"
        WHERE time > '2018-12-12T08:59:00Z' GROUP BY time(6h,9h))
WHERE time > '2018-12-12T08:59:00Z'
GROUP BY time(1d,7h), "name"
```

Telegraf was used as the data broker between the Hue bridge and InfluxDB [23]. The broker periodically requested the status of each sensor via HTTP requests. A simplified example of the telegraf config file that was used can be found in appendix A. The state_presence variable was not posted to InfluxDB without using the converter processor to convert it into an integer. This caused further complications, due to which attempts to query for all the sensors via one URL were unsuccessful. Currently the telegraf configuration file uses direct URLs for each sensor, depicted by its' ID. For further work, improving this would slightly simplify the configuration process, when adding sensors. Additionally, the telegraf configuration could be implemented as a funtionality in the control panel.

5.2.4 Backend

The backend was developed with Node.js. It includes functions for both communicating with the Hue Bridge via HTTP requests and querying data from the InfluxDB instance. The queried data can then be displayed in the frontend, or forwarded to other external systems.

Huejay npm package's discover() function was used to implement the NUPNP search for finding bridges in the LAN [59]. There are other node npm packages available aswell, but they were not used in the project [43]. The npm package called influx is used for connecting to the InfluxDB instance from the backend [39]. An example of setting up the InfluxDB connection for this specific case can be found in listing 5.6.

Listing 5.6: Example of setting up an InfluxDB instance in the Node.js back-end using the influx npm package

```
// Set up an InfluxDB instance
const influx = new Influx.InfluxDB({
  host: '<INFLUXDB_IP_ADDRESS>:<INFLUXDB_PORT>',
  database: '<DBNAME>',
  schema: [
    {
      measurement: 'http',
      fields: {
        'state_presence': Influx.FieldType.INTEGER
      },
      tags: [
        'name'
      ]
    }
  ]
})

// Check that the InfluxDB database '<DBNAME>' exists.
// If it doesn't, create one by the same name.
influx.getDatabaseNames()
  .then(names => {
    if (!names.includes('<DBNAME>')) {
      return influx.createDatabase('<DBNAME>');
    }
  }).catch(err => {
    console.error('Error creating influxDB!')
  })
```

The following, listing 5.7, is an example query to InfluxDB using Node.js, and the influx instance created previously in listing 5.6.

Listing 5.7: Example of a InfluxDB query using the influx npm package in Node.js

```
influx.query(
  `
  SELECT "state_presence","name","url" FROM http
  GROUP BY "name"
  ORDER BY time DESC
  LIMIT ${measurementCount}
  `
)
.then(result => {
  console.log(result);
}
```

The backend uses a JSON Web Token (JWT) based authentication scheme for access control in the control panel. The room status page, however, requires no authentication and it is designed to be available to all users. The API end points relevant to the control panel require authentication aswell.

The API includes functions that handle data queries from InfluxDB (as seen in listing 5.7), aswell as status requests and configuration file requests to the Hue bridge. The configuration files can be stored in a MongoDB instance, if desireable. Additionally, the MongoDB instance includes an implementation for setting up rooms. The rooms can be managed with their own functions, and are designed to include information about the room facilities. The API also includes functions that change the configurations for the sensors, light bulbs or bridges themselves. The aforementioned functions can be called via the Control Panel frontend by authenticated users. The control panel is discussed in detail in section 5.2.1.

The backend includes models for users, bridges and rooms. The user model includes a username and a password only, and is designed for administratorion purposes. The bridge model includes the bridges username used for calls to the bridge's API. It also includes the name, id and ip of the bridge. Additionally, the sensors and lights associated with the bridge, and their configs can be stored under a bridge item. The room model includes the name of the room, which should match the name of the sensor and possible light bulb associated with it. Additionally, a room has information about the number of seats and the types of facilities available.

The backend is built together with the React frontend and they can be run simultaneously [14] [15]. Requests from the frontend, i.e. the client are proxied to the backend, which runs on a separate port. Therefore, allowing API connections from external sources requires opening the backend port, i.e. 3001. The frontend runs on port 3000 by default.

Chapter 6

Results and evaluation

This chapter includes analysis of the gathered data and evaluation of the implemented systems. The raw data discussed consists of presence data gathered by the sensors and aggregated into the InfluxDB instance with telegraf. The system evaluation includes overall assessment of the system developed during this study and comparisons to the turnkey reference system.

The system was evaluated by determining if the reservation status assessments made by the sensors and displayed by the system were correct. This was done by human monitoring. I.e., a person walked periodically past each room, inspected whether they were actually occupied or not and noted the status of the monitoring system for the given room. Additionally, the amount of people in the room was inspected through windows between the corridor and the room.

6.1 Data analysis

We gathered data on the functionality of the real-time space occupancy monitoring system for approximately a month. The raw data is in boolean form. It is aggregated in to the database with five second intervals. Each datapoint is associated with one sensor, and is either a one or a zero. A one represents a state where the sensor has detected movement within the last few seconds (approximately 5-15s). A zero represents a state where the sensor has not detected any movement within a similar time period.

This data was then used to assess the real-time space occupancy status. The status was determined by looking at the max presence value in a time frame of a given length. If movement was detected within this time frame, the space was determined to be in use. Additionally, the data was used to calculate historical utilization percentages. For this, we calculated the

System	Correct	False	Free (f)	Occupied (f)
IS	89%	11%	6%	5%
RS	80%	20%	15%	5%

Table 6.1: The accuracy of the systems. It displays the percentages of correct and false assessments for each system.

System	Correct	False	Free (f)	Occupied (f)
IS	89%	11%	6%	5%
RS	85%	15%	8%	7%

Table 6.2: Accuracies of the systems. Double lense removed from the RS.

portion of five minute windows when rooms were reserved within a desired time frame. For the historical data analysis in this study, a time frome of working hours between 9-15, monday to friday, was used.

We will first look at the accuracy of the systems. The reference system had an additional lense, which was removed half way in, to study the effects it had on accuracy. The reference system will be refered to as RS and the in-house developed system will be refered to as IS, for short.

Table 6.1 and table 6.2 depicts some accuracy numerics for the systems. This former displays the RS values before removing the secondary lense, and the latter after removing it. The tables display the portion of correct and false assessments by each system. Additionally, the percentage of false free (Free (f)) and occupied (Occupied (f)) states are depicted.

After removing the the lense, it can be concluded that the accuracy of the RS did improve. However, the in-house developed system using Philips hardware did perform better, overall.

The portions of falsely assessed states displayed in tables 6.1 and 6.2 are calculated from all the measurements. I.e., there was a 6% chance that the in-house built system falsely claimed a room was free (ratio between false free assessments and all noted assessments). The probabilities of a free assessment being false, however, were 9% and 21% for the IS and RS, respectively. After removing the double lensing from the RS, the ratio between false free assessments and all free assessments dropped to 13%, which is quite notable. The effect on false occupied assessments was a slight increase, but not outside the error margin.

We will now discuss the historical data displayed by each system. Although the RS was set to use a 5 minute window for the real-time status of

System	Mon	Tue	Wed	Thu	Fri
IS	42%	34%	41%	40%	32%
RS	63%	57%	56%	64%	52%

Table 6.3: Space utilization per weekday (hours 9-15) for each system.

each room, it used a one hour window to determine utilization percentages for the historical data. I.e., a room was determined to be occupied for a full hour, if movement was detected. This lead to inflated utilization readings. The percentages can be seen in table 6.3. The values are from data gathered over three weeks, and the results are separated into weekdays. Data gathered over longer periods would be desireable, however, the inflated values proposed by the RS are evident. The mentioned factors together with the accuracy metrics displayed in tables 6.1 and 6.2 suggest that metrics from the IS are more dependable.

What comes to the utilization of spaces during each weekday, no reliable conclusions should be drawn from the data. The spaces are utilized relatively evenly throughout the week, with perhaps a slight dip on fridays. Trustworthy analysis of this would require longer time frames of data gathering.

In addition to the accuracies and utilization metric analysis, another hypothesis was addressed: could the data gathered be used for people counting? I.e., could the amount of "presence = 1" states within a time frame be used to predict the amount of people within a work space. To determine this, the amount of people within a reserved room was noted, and the ratio between "presence = 1" measurements and all measurements within the time frame was considered. Only the data gathered by the IS was applicable to this part of the study. The results can be seen in table 6.4.

The second line displays the portion of "presence=1" measurements, i.e., five second frames where presence was detected. E.g., if a room had only one person in it, 20% of the five second measurements had detected movement, and the rest showed no presence. It can be seen, that this ratio does seem to be correlating with the amount of people in the room. However, with a group size of four, a slight decline in the value can be observed. Additionally, the standard deviation and variance listed on rows three and four, respectively, are rather high. It was noted that some groups presented significantly less movement than others, even with larger group sizes.

The data presented in table 6.4 indicates some correlation between the amount of presence=1 measurements and people in a room. This data is highly dependent on room size and interior design factors, such as furnishing. Therefore, these numbers should not be taken as universal facts. It can be

num. of people	1	2	3	4
presence=1	0.20	0.37	0.51	0.34
stdev	0.19	0.21	0.32	0.22
var	0.04	0.05	0.10	0.05

Table 6.4: Portion of "presence=1" measurements according to the amount of people in a room. This depicts the ratio between 5 second windows with "presence=1" and 5 second windows with "presence=0 OR 1"

stated with reasonable doubt, however, that the measured ratios are generally affected by the number of people within a monitored space.

The five minute window for determining the space reservation status for rooms was deemed reasonable. It is short enough to not cause excessive false reservations after a room becomes free, and long enough to not easily determine a space falsely free. Figures 6.1, 6.2, and 6.3 display graphs created using Grafana [17]. Figure 6.1 shows comparison between a three minute window and five minute window. Additionally, a 60 point moving average of the 5 second measurements is displayed in the top graph. A 0 state describes a free assessment and a >0 state describes an occupied assessment. The window size is essentially the same as the with the 5 minute window. The occupant only left the room for a longer period between 18:30 and 18:40. It can be observed, that the three minute window was prone to falsely assessing the room to be free, periodically.

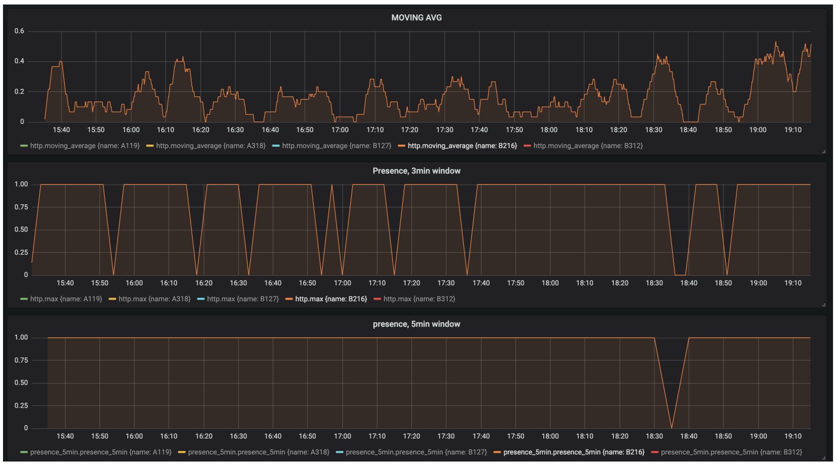

Figure 6.1: Graphs displaying the reservation status (moving average, three and five minute windows) of a room with one person, over approximately 3.5 hours. [17]).

Figure 6.2 compares the one minute and five minute windows. It can be observed that even the five minute window occasionally fails to correctly assess the room status. A 10 point moving average of 30 second windows is also included.

Figure 6.2: Graphs displaying the reservation status (moving average, one and five minute windows) of a room with 2-4 people moving in an out during the time frame. [17]).

In figure 6.3 we can observe that even two people can in some cases be enough to allow a three minute window to correctly assess the reservation status. A 10 point moving average of 30 second windows is also included.

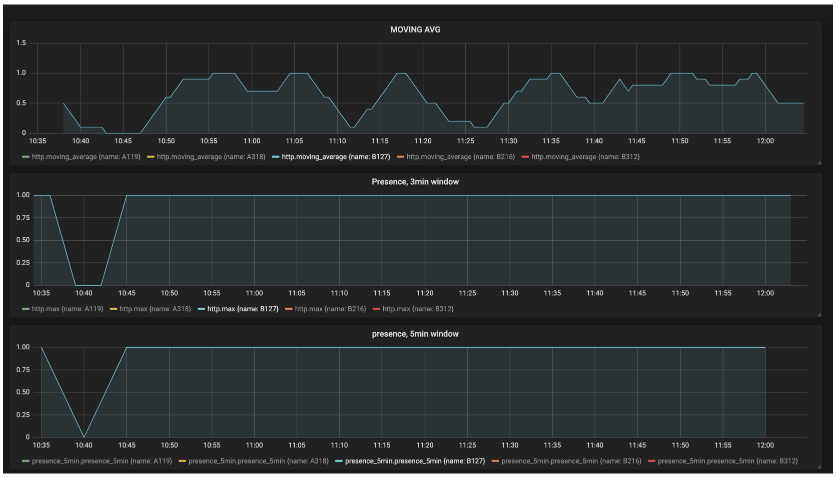

Figure 6.3: Graphs displaying the reservation status (moving average, three and five minute windows) of a room with two people during the time frame after 10:45. [17]).

6.2 Comparison

According to the data analysis in the previous section, the in-house built system performed better. However, installing the RS required less effort. Naturally, the IS demanded design and development hours in addition to requiring locally installed base station hardware. In general, a turnkey solution is likely the most reasonable choice for most companies.

Additionally, the accuracy metrics do not differ enough to dismiss the RS. It does give users better knowledge of the real-time reservation status of public rooms. The historical data of both systems can be used to give insight on additional space requirements for corresponding personel. The variables could also be tweaked by the supplier. The original values were used as is, to display the differences between perceived space utilization. However, the higher overall accuracy of the IS does suggest that the historical data gathered by it should be more precise aswell.

Real-time space occupancy monitoring systems, and the data gathered by them are relevant to companies working in many industries. For those not working within the IT field, ease of installation is likely a main goal. It may not always be reasonable to invest considerably in the implementation such systems. Thankfully, many suppliers within the industry have relatively dependable systems, with ease of installation as a principle. However, the IS developed during this study indicates that such systems can have space to improve, both in hardware and software.

6.3 Discussion

Space occupancy monitoring, and space utilization optimization are a widely discussed and studied field. There are a multitude of ways for gathering the data required for such a system. Each method has its' advantages and drawbacks. The validity of one method over another is dependent on the use case. Thus, when developing such a system, or planning for implementation, it is up to the corresponding parties to make the relevant choices. However, PIR sensing seems to be the standard within the field, as of yet. The overall availability of technologies is excellent. Many of the technologies are continuously developed. For the growing IoT, the most influential technology advancement will likely be the wide spread expansion of 5G.

The objective of this study was to make decision making easier in the future, by providing information on both the software and hardware components and factors relevant to implementing such systems. It can be presumed, that the accuracies of the IS and the RS are enough to expect enhancements in work flow, to some extent.

Having real-time space occupancy information has a wide spectrum of advantages. The gathered data can also be used to analyze the requirements for further space acquisitions. Space maintenance and lease expenses play a large role in overall expenses for companies. Therefore, the technologies discussed in this paper will likely be developed further. The accuracy of present hardware solutions will improve and network technologies will be polished to be more efficient. Novel methods for achieving even better results are likely to come.

The implementation developed in-house is discussed in section 5. It was deemed to be camparable to other industry solutions. Adding multiple sensors to each room would likely achieve higher accuracy, with minimal time and money investments. The UI could be developed further. The control panel could be reviced and include more functionalities, such as controlling the colors of the reservation lights.

The thesis combines information relevant to real-time space occupancy monitoring and WSNs in general. The software developed is under IBM Public Lisence, and it is free to use and further development is encouraged.

Chapter 7

Conclusions

The importance of space occupancy monitoring comes from multiple factors. These include overall employee satisfaction through streamlined workflow, in addition to tangible benefits, such as lowered energy and space lease expenditure.

This study was intended to answer the following questions: what are the most suitable hardware and software technologies for gathering real-time occupancy data within ad-hoc workspaces, and how the gathered data can be analyzed. Additionally, the goal was to recount factors relevant to developing an occupancy monitoring WSN system.

To answer the first question, a review of hardware and software technologies relevant to real-time space occupancy monitoring and WSN systems was conducted. Multiple methods of people and movement detection were discussed. Many of them were deemed unsuitable due to matters of privacy and accuracy, or maintenance overhead caused by high power consumption.

The review was conducted mainly by researching existing literature, but information was also attained from companies working on space utilization monitoring systems. The most common hardware method for detecting occupancy was found to be PIR sensors. In addition to their high availability, PIR sensors are also a relatively cheap solution and require very little power.

Although the selection was largely limited to PIR sensors, there were systems employing other methods available aswell. These included camera systems and depth sensors.

The most common networking technologies and protocols for WSN systems were determined to be ZigBee and Bluetooth Low Energy. They are very suitable for the purpose because of their energy efficiency, sufficient range and bandwidth, as well as availability. Most commercial WSN systems employ one or the other. In regards to WSNs and the IoT field, the most promising networking technology currently in development is 5G. Additionally, EnOcean, which employs energy harvesting methods, such as solar panels and Wirepas Mesh were considered potentially substantial technologies to consider in the future.

The data gathered by the monitoring systems was analyzed with accuracy as the main baseline. The analysis includes comparison between the reference system and the system developed during this study. It was determined, that the in-house developed system was able to achieve higher accuracy. The factors that may affect the accuracy differences between the systems include both the software, which comprises of the networking protocols, firmware and data processing methods, and hardware, which includes the accuracy of the sensor inside, the lense, and the overall design of the device. In the implementation developed during this study, a single PIR sensor was able to correctly assess the occupancy status of a room with 89% accuracy. The accuracy of the reference system was 85%. Adding multiple sensors in each room was not studied, however, it would likely increase the accuracies further.

The possibility of approximating the number of people in a room was also studied. A correlation was found between the amount of movements detected and the amount of people in a given room. However, variance in the data was high. The values, discussed in chapter 6, are likely highly dependent on differences in the experiment environment. The historical data can be valuable to landlords, when assessing the requirements for future expansions.

The results implicate that a system employing only a single PIR sensor per room can provide relatively accurate real-time information of the occupancy status. This information, if relayed efficiently to users, can be expected to improve work flow. The data gathered can also be used in HVAC and lighting systems to minimize energy consumption.

The implementation process was multifaceted. The whole project was concluded in three months. This included the preliminary review, development, and installation. Thereafter, data was gathered for approximately one month. The in-house developed implementation utilizes Philips Hue hardware. The open API of the Hue family of products was a deciding factor in the selection of the hardware provider. However, Hue systems have proven security flaws stemming from firmware design and ZigBee related matters. One sensor was installed in each experiment room. Philips Hue bulbs installed outside the rooms depicted the reservation status, i.e. whether movement was detected inside. The third party reference system was extremely easy to install due to it employing LPWAN networking with no on-site base stations.

The development phase for the in-house system took approximately two months. Although the product is functional, further development would likely improve user firendliness considerably. Additional functionalities, could also be implemented. These include the ability to adjust the color configuration of reservation bulbs, configuring thresholds and window lengths separately for each space, and allowing multiple sensors per space. The most notable necessary software choices were associated with managing the sensor data. InfluxDB and Telegraf were used to store and aggregate the data, respectively. It is advisable to research the current best practices and preferred technologies before beginning a similar project. The Philips hardware itself was easy to install. The most significant matter to consider was finding a central position for the bridge or base station.

Overall, it was determined that developing software for existing occupancy monitoring hardware with open APIs is feasible and relatively straight forward. Additonally, such a system can, as of yet, provide competitive results when compared to turnkey services provided by suppliers in the industry. However, the development process for a high grade system requires substantial research and work hours.

With the growth of the Internet of Things and WSN systems becoming more common, studying the underlying software and hardware technologies is important. This study aims to help the development of better solutions for monitoring space occupancy, both for real-time utility and historical data gathering.

For future work, studying the effects of increasing the number of sensors per room would be beneficial. Additionally, further empirically comparing different sensing methods in comparison to and in fusion with PIR sensors would be logical. Additionally, studying methods of clearly conveying the reservation status to users in varying environments would likely enhance practical implementations.

Bibliography

[1] Akbas, S., Efe, M. A., and Ozdemir, S. Performance evaluation of pir sensor deployment in critical area surveillance networks. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems* (May 2014), pp. 327–332.

[2] Arcari, F. D., Costa, C., Pereira, C. E., Netto, J. C., Torres, G., Souza, M., and MÃ¼ller, I. Development of a wirelesshart - enocean adapter for industrial applications. In *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)* (Nov 2017), pp. 181–186.

[3] Arlen Nipper of Cirrus, Andy Stanford-Clark. Mqtt. `http://mqtt.org/`, 1999. Accessed: 2019-01-09.

[4] Bai, Y., Li, Z., and Xie, Z. Use of multi-frequency ultrasonic sensors with pir sensors to enhance the sensing probability of an embedded surveillance system. In *2010 10th International Symposium on Communications and Information Technologies* (Oct 2010), pp. 170–175.

[5] Cao, J., Sun, L., Odoom, M. G., Luan, F., and Song, X. Counting people by using a single camera without calibration. In *2016 Chinese Control and Decision Conference (CCDC)* (May 2016), pp. 2048–2051.

[6] Castillo-Secilla, J. M., Aranda, P. C., Outeirino, F. J. B., and Olivares, J. Experimental procedure for the characterization and optimization of the power consumption and reliability in zigbee mesh networks. In *2010 Third International Conference on Advances in Mesh Networks* (July 2010), pp. 13–16.

[7] Chan, A. B., Liang, Z.-S. J., and Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), pp. 1–7.

[8] Chandra Nepal, G., Biswa, R., Maya Adhikari, D., Chodon, P., Gyeltshen, S., and , C. Passive infrared (pir) sensor based security system. *International Journal of Electrical, Electronics and Computer Systems 14* (06 2013), 772–778.

[9] Cippitelli, E., Fioranelli, F., Gambi, E., and Spinsante, S. Radar and rgb-depth sensors for fall detection: A review. *IEEE Sensors Journal 17*, 12 (June 2017), 3585–3604.

[10] Cisco. Internet of Things At a Glance. Tech. rep., Cisco Systems, 2016. [Online; accessed 16.10.2018].

[11] Coskun, H., Albayrak, Y., and Helhel, S. Indoor location service with beacons on embedded linux. In *2015 23nd Signal Processing and Communications Applications Conference (SIU)* (May 2015), pp. 2370–2372.

[12] Dan, B., Kim, Y., Suryanto, Jung, J., and Ko, S. Robust people counting system based on sensor fusion. *IEEE Transactions on Consumer Electronics 58*, 3 (August 2012), 1013–1021.

[13] Elarabi, T., Deep, V., and Rai, C. K. Design and simulation of state-of-art zigbee transmitter for iot wireless devices. In *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (Dec 2015), pp. 297–300.

[14] Facebook. react. `https://github.com/facebook/react`, 2011. Accessed: 2019-01-09.

[15] Facebook. reactjs.org. `https://reactjs.org/`, 2011. Accessed: 2019-01-09.

[16] GitHub. Github octoverse 2018, top languages over time. `https://octoverse.github.com/projects`, 2018. Accessed: 2019-01-09.

[17] grafana. Grafana. `https://github.com/grafana/grafana`. Accessed: 2019-01-09.

[18] Harikrishnan, R., and Sivagami, P. Intelligent power saving system using pir sensors. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)* (April 2017), vol. 2, pp. 573–577.

[19] Heittola, T., Mesaros, A., Eronen, A., and Virtanen, T. Context-dependent sound event detection. *EURASIP Journal on Audio, Speech, and Music Processing 2013*, 1 (Jan 2013), 1.

[20] Hirakata, Y., Nakamura, A., Ohno, K., and Itami, M. Navigation system using zigbee wireless sensor network for parking. In *2012 12th International Conference on ITS Telecommunications* (Nov 2012), pp. 605–609.

[21] Huang, J., Tseng, Y., Wu, S., and Wang, S. Information preserving color transformation for protanopia and deuteranopia. *IEEE Signal Processing Letters 14*, 10 (Oct 2007), 711–714.

[22] influxdata. Chronograf. `https://github.com/influxdata/chronograf`. Accessed: 2019-01-09.

[23] influxdata. Telegraf. `https://github.com/influxdata/telegraf`. Accessed: 2019-01-09.

[24] influxdata. Influxdb. `https://github.com/influxdata/influxdb/`, 2013. Accessed: 2019-01-09.

[25] Javed, F., Afzal, M. K., Sharif, M., and Kim, B. Internet of things (iot) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys and Tutorials 20*, 3 (2018), 2062–2100.

[26] Kim, M., Jang, I., Lim, S., and Kang, T. Implementation of zigbee-vlc system to support light control network configuration. In *2016 18th International Conference on Advanced Communication Technology (ICACT)* (Jan 2016), pp. 153–156.

[27] Lai, K., Ku, B., and Wen, C. Using cooperative pir sensing for human indoor localization. In *2018 27th Wireless and Optical Communication Conference (WOCC)* (April 2018), pp. 1–5.

[28] Lau, B. P. L., Wijerathne, N., Ng, B. K. K., and Yuen, C. Sensor fusion for public space utilization monitoring in a smart city. *IEEE Internet of Things Journal 5*, 2 (2018), 473–481.

[29] Li, X., Chen, G., Zhao, B., and Liang, X. A kind of intelligent lighting control system using the enocean network. In *2014 International Conference on Computer, Information and Telecommunication Systems (CITS)* (July 2014), pp. 1–5.

[30] M, V. W., Iskandar, Hendrawan, and Arifianto, M. S. Wireless sensor network on 5g network. In *2018 4th International Conference on Wireless and Telematics (ICWT)* (July 2018), pp. 1–5.

[31] Maguire, B., and Desai, S. Dempster-shafer fusion for personnel detection: Application of dempster-shafer theory with ultrasonic micro-doppler and pir sensors. In *2012 15th International Conference on Information Fusion* (July 2012), pp. 2209–2214.

[32] MartÃnez, I. S. H., Salcedo, I. P. O. J., and Daza, I. B. S. R. Iot application of wsn on 5g infrastructure. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)* (May 2017), pp. 1–6.

[33] Mekki, K., Bajic, E., Chaxel, F., and Meyer, F. Overview of cellular LPWAN technologies for iot deployment: Sigfox, lorawan, and nb-iot. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018, Athens, Greece, March 19-23, 2018* (2018), IEEE, pp. 197–202.

[34] Miyazaki, N., Tsuji, K., Zheng, M., Nakashima, M., Matsuda, Y., and Segawa, E. Privacy-conscious human detection using low-resolution video. In *3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015, Kuala Lumpur, Malaysia, November 3-6, 2015* (2015), IEEE, pp. 326–330.

[35] Mohammadmoradi, H., Munir, S., Gnawali, O., and Shelton, C. Measuring people-flow through doorways using easy-to-install IR array sensors. In *13th International Conference on Distributed Computing in Sensor Systems, DCOSS 2017, Ottawa, ON, Canada, June 5-7, 2017* (2017), IEEE, pp. 35–43.

[36] MongoDB Inc. Mongodb. `https://www.mongodb.com/`, 2007. Accessed: 2019-01-09.

[37] Munir, S., Arora, R. S., Hesling, C., Li, J., Francis, J., Shelton, C., Martin, C., Rowe, A., and Berges, M. Real-time fine grained occupancy estimation using depth sensors on arm embedded platforms. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (April 2017), pp. 295–306.

[38] Nisar, H., Khan, M. B., Yi, W. T., Voon, Y. V., and Khang, T. S. Contactless heart rate monitor for multiple persons in a video. In

2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) (May 2016), pp. 1–2.

[39] NODE-INFLUX. An influxdb client for node.js. `https://github.com/node-influx/node-influx`. Accessed: 2019-01-09.

[40] NODE.JS FOUNDATION. reactjs.org. `https://nodejs.org/en/`, 2009. Accessed: 2019-01-09.

[41] OSMAN, N. I., AND ABBAS, E. B. Simulation and modelling of lora and sigfox low power wide area network technologies. In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (Aug 2018), pp. 1–5.

[42] PHILIPS. Meet hue homepage. `https://www2.meethue.com/`. Accessed: 2019-01-09.

[43] PHILIPS. Philips hue developer program. `https://developers.meethue.com/`, 2018. Accessed: 2019-01-09.

[44] PINOL, O. P., RAZA, S., ERIKSSON, J., AND VOIGT, T. Bsd-based elliptic curve cryptography for the open internet of things. In *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)* (July 2015), pp. 1–5.

[45] PITTALUGA, F., AND KOPPAL, S. J. Pre-capture privacy for small vision sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 11 (Nov 2017), 2215–2226.

[46] PLOENNIGS, J., RYSSEL, U., AND KABITZSCH, K. Performance analysis of the enocean wireless sensor network protocol. In *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)* (Sep. 2010), pp. 1–9.

[47] PRAMONO, S., PUTRI, A. O., WARSITO, E., AND BASUKI, S. B. Comparative analysis of star topology and multihop topology outdoor propagation based on quality of service (qos) of wireless sensor network(wsn). In *2017 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)* (Oct 2017), pp. 152–157.

[48] RAZOUK, W. Zigbee security within the framework of iot. In *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications* (Nov 2014), pp. 265–265.

[49] Research, A. Case study: Lindstroem group a reinvented workflow optimization solution enabled by thingsee iot platform and wirepas mesh, 2018. [Online; accessed 17.10.2018].

[50] Ronen, E., and Shamir, A. Extended functionality attacks on iot devices: The case of smart lights. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)* (March 2016), pp. 3–12.

[51] Ronen, E., Shamir, A., Weingarten, A., and O'Flynn, C. Iot goes nuclear: Creating a zigbee chain reaction. In *2017 IEEE Symposium on Security and Privacy (SP)* (May 2017), pp. 195–212.

[52] Sachan, D., Goswami, M., and Misra, P. K. Analysis of modulation schemes for bluetooth-le module for internet-of-things (iot) applications. In *2018 IEEE International Conference on Consumer Electronics (ICCE)* (Jan 2018), pp. 1–4.

[53] Sahoo, K. C., and Pati, U. C. Iot based intrusion detection system using pir sensor. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)* (May 2017), pp. 1641–1645.

[54] Salamon, J., and Bello, J. P. Unsupervised feature learning for urban sound classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (2015), IEEE, pp. 171–175.

[55] Samuel, S. S. I. A review of connectivity challenges in iot-smart home. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)* (March 2016), pp. 1–4.

[56] Senst, T., Eiselein, V., Badii, A., Einig, M., Keller, I., and Sikora, T. A decentralized privacy-sensitive video surveillance framework. In *2013 18th International Conference on Digital Signal Processing (DSP)* (July 2013), pp. 1–6.

[57] Sharma, H., and Sharma, S. A review of sensor networks: Technologies and applications. In *2014 Recent Advances in Engineering and Computational Sciences (RAECS)* (March 2014), pp. 1–4.

[58] Singh, V., Aggarwal, G., and Ujwal, B. V. S. Ensemble based real-time indoor localization using stray wifi signal. In *2018 IEEE International Conference on Consumer Electronics (ICCE)* (Jan 2018), pp. 1–5.

[59] SQMK. huejay npm package. `https://github.com/sqmk/huejay`. Accessed: 2019-01-09.

[60] TANAKA, G., SUETAKE, N., AND UCHINO, E. Visibility improvement of undiscriminatable colors by modification of yellow-blue components for protanopia and deuteranopia. In *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (Jan 2009), pp. 562–565.

[61] TUOMAS JUOPPERI. Philipsensors code. This repository includes the code that was developed during this study.

[62] VASCONCELOS, F., FIGUEIREDO, L., DE ALMEIDA, A., AND FERREIRA, J. C. SMART sensor network: With bluetooth low energy and CAN-BUS. In *2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Bari, Italy, September 18-20, 2017* (2017), IEEE, pp. 217–223.

[63] WANT, R., SCHILIT, B., AND LASKOWSKI, D. Bluetooth le finds its niche. *IEEE Pervasive Computing 12*, 4 (Oct 2013), 12–16.

[64] WETZEL, J., ZEITVOGEL, S., LAUBENHEIMER, A., AND HEIZMANN, M. Towards global people detection and tracking using multiple depth sensors. In *2018 International Symposium on Electronics and Telecommunications (ISETC)* (Nov 2018), pp. 1–4.

[65] WIREPAS. White paper: Wirepas mesh overview. Tech. Rep. WP-PD-123, Wirepas Ltd., Visiokatu 4, 33720 Tampere, 2018. [Online; accessed 17.10.2018].

[66] XIAO, H., FENG, X.-F., AND HU, Y.-F. A new segmented time warping distance for data mining in time series database. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)* (Aug 2004), vol. 2, pp. 1277–1281 vol.2.

[67] XIAOMAN, L., AND XIA, L. Design of a zigbee wireless sensor network node for aquaculture monitoring. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)* (Oct 2016), pp. 2179–2182.

[68] YANZI. Yanzi homepage. `https://www.yanzi.se/`. Accessed: 2019-01-09.

[69] ZHANG, M., AND HU, Q. A hybrid network smart home based on zigbee and smart plugs. In *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)* (Nov 2017), pp. 389–392.

[70] ZIEGLER, S. Considerations on ipv6 scalability for the internet of things - towards an intergalactic internet. In *Global Internet of Things Summit, GIoTS 2017, Geneva, Switzerland, June 6-9, 2017* (2017), IEEE, pp. 1–4.

Appendix A

Telegraf configuration file

```
# Telegraf agent configuration
[agent]
#Data collection interval for all inputs
interval = "5s"

# Convert values to another metric value type
[[processors.converter]]
## Tags to convert
##
## The table key determines the target type,
## and the array of key-values
## select the keys to convert.  The array
## may contain globs.
##   <target-type> = [<tag-key>...]
[processors.converter.tags]
string = []
integer = []
unsigned = []
boolean = []
float = []

## Fields to convert
##
## The table key determines the target type,
## and the array of key-values
## select the keys to convert.  The array
## may contain globs.
```

```
##   <target-type> = [<field-key>...]
[processors.converter.fields]
tag = []
string = []
integer = ["state_presence"]
unsigned = []
boolean = []
float = []

# OUTPUTS
[outputs]
[outputs.influxdb]
# The full HTTP endpoint URL for
# your InfluxDB instance
url = "<INFLUX_DB_URL>:<INFLUX_DB_PORT>"
# The target database for metrics.
# This database must already exist
database = "YOUR_DATABASE" # required.


# Read formatted metrics from one or more
# HTTP endpoints
[[inputs.http]]
## One or more URLs from which to read
## formatted metrics
# fill with direct address to each presence sensor.
# Now reading data from sensors with ids 6 and 9.
urls = [
"http://<BRIDGE_IP_ADDRESS>/api/<USERNAME>/sensors/6",
"http://<BRIDGE_IP_ADDRESS>/api/<USERNAME>/sensors/9"
]

## HTTP method
method = "GET"

## Amount of time allowed to complete the HTTP request
timeout = "5s"

## Data format to consume.
## Each data format has its own unique set of
## configuration options, read
```

```
## more about them here:
## https://github.com/influxdata/telegraf/
## blob/master/docs/DATA_FORMATS_INPUT.md
data_format = "json"

tag_keys = [
"name",
"uniqueid",
"productname"
]

json_string_fields = ["state_presence"]
```