universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„ILLAC: An Indoor Localization Framework for Location-aware Access Control in Fog-Computing based IoT Environments“

verfasst von / submitted by

Kaspar Lebloch BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2019 / Vienna, 2019

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 935
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Medieninformatik
Betreut von / Supervisor:	Univ. Prof. Dipl.-Math. Dr. Peter Reichl, Privatdoz.

Contents

Abstract

Access control, as one of the major pillars of IT security, has seen numerous proposed improvements and novel concepts over the last years. The dire need for systems to be secure before handling highly sensitive data oftentimes requires tedious configuration by skilled technicians, especially in pervasive and IoT contexts. This has led to the development of access control models like Attribute Based Access Control, Capability Based Access Control, and other even context-aware access control models allowing fine-grained access right modelling and possibly supporting automation of access right management. This thesis covers conception, implementation, and testing of an indoor localization-based extension to an existing access control system for Fog-Computing based IoT environments. The design and implementation of an event-based room-level indoor localization system based on Bluetooth Low Energy, as well as the application and design of localization based access policies is discussed. Furthermore, a proof of operability and a performance evaluation for both the localization system and the location-based access policy evaluation were performed and the results support the technical argument towards the application of location-aware access control in future IoT environments.

1 Introduction

Achieving a sufficient level of security has always proven to be a major challenge for designers of IT- and especially Internet of Things (IoT) systems. The oftentimes constrained nature of IoT devices, be it in processing power, memory, or even power supply for battery powered devices, prohibits the application of heavyweight security protocols and mechanisms in such environments. Since system security is generally dependent on skillful work on configurations, an automatically self-configuring access control system could potentially support more secure configurations as compared to a system set up by a non-professional user. Moreover, the granularity of access right policies in current access control systems is often insufficient given the heterogeneous and distributed nature of the IoT. Recent approaches to the problem of access control include relying on attributes or capabilities of the user to determine whether access to a resource should be granted or denied. Another novel approach towards rethinking access control concepts was the introduction of contextual variables into the decision making process. The context of an interaction may make implications towards the authenticity of a request or access to resources may be desired to only be granted depending on location, hence location-aware access control has been introduced as a context-dependent method of access control.

1.1 Scenarios for location-aware access control

In order to motivate the idea of location-aware access control, some exemplary scenarios shall be presented:

1. Location-aware access control in a smart home
2. Location-aware access control in a business environment
3. Burglary deterrent and impediment

These scenarios serve the purpose of illustrating the security improvement and simplification, which can be achieved through the application of location-aware access control in IoT environments.

1.1.1 Location-Aware Access Control in a Smart Home

The physical security of a smart home and its appliances can be significantly improved by implementing a layer of access control on said appliances. For example: The stove in a smart home is "smart", it is connected to the homes network. Interactions with the stove are communicated to the smart home hub, and access control policies decide whether access should be granted or not. Through location-aware access control, it is possible to define policies allowing the stove to only be turned on, if a specific person (e.g. an adult) is in the kitchen, preventing toddlers and pets from unintentionally activating the stove without supervision. Moreover, a location-aware access control system can be set up to alert the owners of interaction with the resource (e.g. the stove) being target of an unsupervised access and directly ask them for permission.

1.1.2 Location-Aware Access Control in a Business Environment

The distribution of physical keys to authorized persons already represents access control in a non-digital space. This authorization process can be translated to the digital space using location-aware access control: The position of an employee having access to specific floors is being tracked throughout their workplace building. The employee, being a tech support technician, has access to the server room floor located in the basement of the building. Instead of using their physical key in the keyhole presented in the elevator, the access control system detects their presence in the there, and automatically approves their request to go to the restricted floors. Again, if an unauthorized person were to select a restricted floor, a supervisor can be alerted to the request and give their approval to temporarily authorize the person's access.

1.1.3 Burglary Deterrent and Impediment

The Philips Hue [46] platform already offers burglary deterrent functionality, as pre-programmable light on-/off cycles can be put in place before going on vacation. A location-aware access control system can enhance such features in case of a real break-in. Policies can be defined to put the smart building or smart home into an alarm state upon detection of local resource or detection of movement access while no known user is on the premises. The alarm state may turn on all the lights inside the perimeter and disable the ability to turn

them off unless authorized. The system may also activate acoustic alarms from any smart speakers and alert the police or neighbours of the break-in. Since the system is aware of the users location, it automatically detects the absence of the users and can arm the alarm mechanism automatically each time no registered user is present at the supervised object.

1.2 Research Questions

The main scientific contribution of this thesis is the elaboration on the following research questions:

1.2.1 RQ1: Reliable, Fast, and Accurate Indoor Localization With Affordable Hardware

- How fast can an indoor localization system based on affordable consumer grade hardware detect the presence/absence of the user?
- How fast can it detect changes in user position?
- How often and how long does it assign a false user position?

1.2.2 RQ2: Location-Aware Access Control and Access Control Automation

- How can the automation of access control mechanisms be supported by location data?
- What data models are needed, how can access policies be modeled for achieving that goal?
- What are the implications this would have on IoT access control systems in terms of intelligence and automation, what are possible downsides?

1.3 Motivation

Many of the security problems present in current IoT environments stem from the inability of the users to correctly represent their access control needs in the system. In my opinion, this may be due to different reasons, such as an access control model that is too coarse-grained and doesn't allow its users to set up access control the way they want to. Instead users are forced to lower or even disable the defences of their system in order to set up access for all parties as desired. Another reason for this inability may soberly be presented by the users' own incompetence, in which case the best solution would be to take initial access right management out of their hands and try to automate it in a smart way. Of course, such cutbacks into the users' sovereignty over their own devices have to be backed by tried and proven security gains. Still, I firmly believe, the insights gained from the research on automation of access control may serve as a starting

point for the setup of default access rules to further increase the security of the users, their devices, and their privacy.

1.4 Contributions

This thesis is based the design of a location-aware access control component and its evaluation in a real-life scenario. The gained insights provide information for research towards the development of automated access rights management in IoT environments through context-awareness. Down the line, these insights will also benefit the users of IoT systems as their devices will be more secure by default and the hassle of setting up access control will be reduced.

1.5 Structure of this Thesis

The remaining thesis is structured as follows: First, in Section 2, an overview of the state of the art in indoor localization techniques and mechanisms, as well as an overview of existing context-aware access control concepts and tangential technologies is presented. In Section 3, the chosen approach and its implementation is described in detail, beginning with the already existing access control framework followed by the developed event-based Localization Engine and the application of resulting localization data in access policies on IoT resources. A thorough analysis of performance and function of both the localization, as well as the access policy evaluation engine is presented in Section 4. Closing with Section 5, conclusions towards the application of the developed framework in real-world scenarios are drawn and compliance with the EU General Data Protection Regulation (GDPR), as well as ethical arguments for and against the application of location tracking technologies in Fog-Computing based IoT networks are discussed.

---

2 Related Work

2.1 Indoor Localization Techniques

There are many different ways of acquiring approximate user position readily available today, but given the major differences in technologies and their respective performance, a unique standard hasn't been developed yet [49]. There are numerous ways of determining indoor user position using the following wireless technologies:

- WiFi [3, 7, 27]
- Bluetooth Low Energy [15, 54]
- Light [50, 45]
- RFID [6, 24]
- Inertial Sensors [28]
- GSM Signals [34]
- Computer Vision [25]
- Ultrasound [51, 38]

The following subsubsections will address the approaches proposed and their applications using each technology, as well as their benefits and drawbacks with respect to the use case location-aware access control.

2.1.1 Localization Mechanisms Using Wireless Technologies

Using wireless technologies, exact positional information can be extracted using a triangulation or a tri-/multilateration method or in the case of fingerprinting techniques by performing similarity checks with fingerprints collected during the setup of the system. Another popular, albeit not as accurate method is proximity detection, where an object gets assigned to a zone according to the node it appears to be closest to.

Triangulation. For triangulation, the angles of attack are extracted from a signal using antenna arrays or steerable antennas. With the known distances between multiple different static beacons, the exact position of the target can be calculated. This method is especially susceptible to interferences generated by phenomena of reflection from differently shaped surfaces, so called "multipath" phenomena [6].

Multilateration. Multilateration relies on correctly determining the distance of the localizable object to at least three (for 2D-localization) or four (for 3D localization) static nodes. These range measurements can be done by analyzing Received Signal Strength (RSS), Time Of Arrival (TOA), Time Difference Of

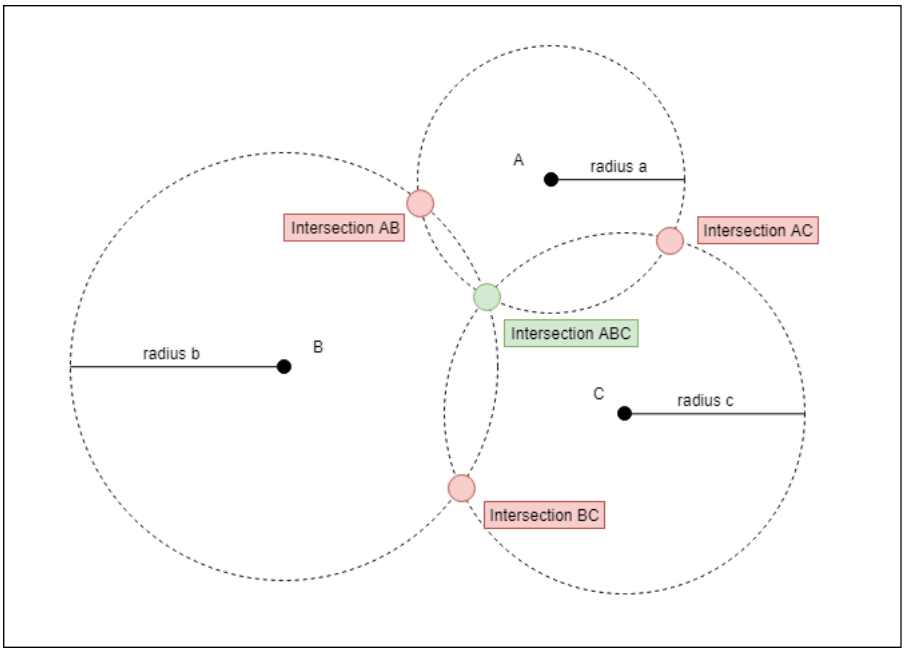

Figure 1: The principle of trilateration to localize an object in 2D, based on illustration in [6]

Arrival (TDOA), or Received Signal Phase (RSP) [6]. Figure 1 depicts the process of trilateration in two dimensions. For each node, the object may only be on a circle with radius r equal to the distance calculated. Since two circles mostly intersect in two points, a third distance measurement is needed to determine the exact location. An example for three dimensions could be understood analogous with spheres instead of circles.

Fingerprinting. Localization using fingerprinting relies on the property of Radio Frequency (RF) environments being stable over time, but unstable concerning space. Signal strengths from static stations vary strongly when the client device is moving around since the received signal strength drops when the client moves away from a station, but not when it is still for a long period of time. This property allows for the collection of signal strength data for specific locations, so called "fingerprints", as depicted exemplary in Figure 2. Once collected, the data associated with a specific position can be used to determine the position from any new fingerprint.

Proximity Detection. Trivial methods of proximity detection rely on deploying sending or receiving nodes assigned to rooms or zones. Received signal strength can be used as an indicator of proximity to such a node, and thus help

```
"fingerprint":{
    "timestamp" : 1557573510 ,
    "rssis"       :{
            "WIFI-abcdefg" : -99,
            "WIFI-hijklmn" : -20,
            "WIFI-opqrstu" : -211,
            "WIFI-vwxyzab" : -150
    }
}
```

Figure 2: Example RSS fingerprint in JSON

to assign the position of the user to a zone or room.

2.1.2 WiFi

WiFi Fingerprinting. The "RADAR in-building User Location and Tracking System" [3] proposed by Bahl and Padmanabhan intends to collect received signal strength data from a mobile object (beacon) at multiple receiving stations within a building. The collected data is then manually associated with a position on the floor map during the setup phase. During the operational phase, the system collects those signal strength indicators and attempts to find the nearest neighbours in the classified data. Once those are determined, a guess in the center of the space spanned up by the neighbours is used to determine the final location. In the same paper, the authors also propose a radio propagation model, which helps generating signal strength data akin to the data collected during setup, with the approximate precalculated spread of the radio signal. This further improved their tracking results and can reduce the time required for the setup of the system. The authors claim the roles of collecting and broadcasting entity can be reversed as well without negative impact on performance. It is even recommended to do so once a large number of clients is expected to be using the system [3].

There have been many improvements on such techniques using "fingerprints" of received signal strength values for indoor localization, such as the EZ algorithm proposed by Chintalapudi et al., which does not require pre-deployment effort. Users are regularly sending RSS reports to a localization server and occasionally, if available in close proximity to windows, also GPS signals with absolute positions. The system then uses a genetic algorithm to determine the location from constraints modelled by the EZ algorithm [7].

A similar approach at building location information for usage in RSS fingerprint based localization systems, is Zee [40], a crowdsourcing based method. In addition to RSS fingerprints, the client devices also collect data from inertial sensors as described in Section 2.1.6 and [28] which helps assigning positions to fingerprints without user interaction. Zee only needs a map of the perimeter

to combine with the inertial sensor data to collect data for fingerprint based localization [40].

Another recent approach is the application of convolutional neural networks on fingerprinting data alone. The approach presented by Mittal et al. [29] transforms WiFi signatures into images, which in turn are analyzed by a framework of convolutional neural networks. This approach rewarded them with an average localization error smaller than two meters [29].

Localization using 802.11ad. Among the more recent approaches at determining user position through WiFi technology, the exploitation of information gained from directional radio as is used in the IEEE 802.11ad standard[23] was introduced by Bielsa et al.. The authors succeeded in modding the firmware of off-the-shelf 60GHz access points to extract Received Signal Strength (RSS) and Signal-to-Noise Ratio (SNR) measurements for each antenna in the array. Using Fourier transformations and particle filters, the authors claim to achieve sub-meter accuracy of localization in 70% of the cases[4].

2.1.3 Bluetooth Low Energy(BLE)

BLE proximity detection. Han et al. tested a system with distributed anchors in different rooms, which determine whether a person left or entered a room depending on thresholds applied to the RSS [19]. This method is easy and simple to set up and operate, as well as affordable.

BLE fingerprinting. The main benefit of BLE fingerprinting over WiFi fingerprinting is its susceptibility to an effect called "fast fading", the rapid drop of signal strength over distance. This effect leads to locally strongly varying Received Signal Strength (RSS) values for deployed BLE Beacons, which in turn is beneficial for localization precision [14]. In order for BLE fingerprint based localization to localize moving users in a sufficiently precise way the authors recommend the deployment of 1 beacon per 30 square meters to reach accuracies of $<$4.8 meters [15].

More recent improvements on algorithms determining user location using BLE fingerprinting technique have increased the accuracy in dense deployment scenarios of 1 beacon per 9 meters to about $<$2.56 meters 90% of the time [54].

2.1.4 Light

There are several approaches at localization using visible or invisible parts of the light spectrum to help localize users. Earlier approaches such as infrared light emitting worn badges regularly broadcasting identifiers through pulse width modulated light bursts. These approaches were developed by Want et al. [50] to be used instead of pagers, to help route calls directly to the current position of the user, instead of waiting for the user to call back. The benefits of this technique compared to using pagers to contact the user is the determination where the user is, especially given the fact that a message sent to a pager may

not always be received due to bad reception. To introduce such a system based on infrared light bursts, the perimeter, where localization is being deployed has to be outfitted with corresponding receivers.

Recently, there have also been approaches using passive retroreflectors for visible spectrum light for real time localization [45]. The deployment of the proposed localization system only requires affixing multiple photodiodes on unmodified light sources to sense the retroreflected signal. This technique uses trilateration to determine an objects position and can achieve centimeter-level accuracy and single digit angular error for the orientation of a device.

2.1.5 RFID

Radio Frequency IDentification (RFID) can be used to track and localize assets and users. Since RFID readers with greater range are often large in size or expensive, trivial approaches rely on gates, whose passage assigns the position of an asset or user to a specific zone [6].

LANDMARC. More sophisticated approaches like LANDMARC [31] use active RFID tags as beacons with absolute positions and estimate the position of object tags through signal strength differences to those reference tags. Applying this technique results in a cheaper system, given fewer expensive readers have to be acquired, is more easily adaptable to environmental dynamics through free deployment of tags, as well as the production of more accurate and reliable location information [31].The methodology for determining an objects position in LANDMARC relies on a k-nearest-neighbours algorithm. Improvements to the LANDMARC system as proposed by Guang-Yao et al. include the reduction of candidates for neighbouring tags, which is claimed to reduce the latency for locating objects [24].

2.1.6 Inertial Sensors

Li et al. designed algorithms for step detection, stride length estimation, and heading direction determination using smart phone inertia sensors [28]. Based on these procedures the authors developed an end-to-end system and claim to be able to reach a mean accuracy 1.5m to 2m dependent on the position of the smart phone on the body (hand held or pocket).

2.1.7 GSM Signals

Based on the ubiquity and rare occurrences of inference on GSM band based radio frequency coverage, Otsason et al. [34] developed a solution for indoor positioning using a GSM-signal based fingerprinting technique. The authors claim to reach a median accuracy of 5 meters in large multi-floor buildings.

2.1.8 Computer Vision

The problem of Simultaneous Localization And Mapping (SLAM) is an old and well researched matter [10]. The vSLAM [25] (visual Simultaneous Localization And Mapping) algorithm developed by Karlsson et al. proposed tackling the problem with computer vision technology in combination with odometry that is the application of inertial sensors to determine movement speed and direction, which is presented in Section 2.1.6.

2.1.9 Ultrasound

Most approaches using UltraSound (US) technology for localization actually rely on a combination of US and RF signals. The Cricket [38] localization system is based on so called "crickets", small ultrasonic and RF emitting beacons built from components, which are available for less than $10. The Crickets simultaneously transmit data on RF and US channels, which then in turn can be picked up by clients in close proximity. Using the time difference of arrival of the signals, the client can estimate its own distance to the beacon, since RF signals travel at the speed of light, whereas US signals travel at the speed of sound. The combination of RF and US signals also provides the benefit of easy RF beacon selection, since the Cricket system assumes no proximity, if the RF signal is received, while there is no corresponding US signal available. Since US signals are more affected by obstacles (such as walls), they will only be received in close proximity. The "Cricket Location-Support system" has been developed with privacy in mind, and is explicitly a system allowing users to locate themselves without tracking them.

An alternative system based on ultrasonic signal transmission has been proposed by Ward et al. [51]. The "Location Technique for the Active Office" proposes widespread deployment of ultrasonic receivers within the supervised perimeter and relies on remotely activating the tracked mobile devices through RF messages, which in turn broadcast their US signal from a special dome shaped transmitter for reliable trilateration.

2.2 Data Models for Indoor Localization

The proposed application of user position in access control systems requires a model with representations relevant to said systems. In order to provide location data to an access control system, a required granularity of user position has to be established. In accordance with this, Niu et al. considered a common data model for indoor location-aware services [32], which is supposed to represent locations of objects independent of the source of their location data in a unified way. While this model incorporates high precision and powerful representations of objects within an indoor space, it relies on accurate object positions represented in Cartesian coordinates relative to a root point of a building. Acquiring such detailed position data usually requires expensive tracking hardware or, in case of lower accuracy tracking methods, fails to provide a measure of error or approximation for the determined location.

Tóth et al. proposed a data model for hybrid indoor positioning systems [49], which helps combine multiple sources of location information into a single object containing measurements from available sensors, a representation of a position, as well as a zone this position resides in. While the data model is highly expressive and can represent various different information about a measurement, it fails to semantically divide between measurement and resulting position. This representation assumes that either the system consuming the data is interested in the exact measurements, or the system creating the data is already aware of the position, in which case an unnecessary overhead (the measurement data) would be transmitted.

Samsungs "Smart Things" [47], a popular smart home solution models presence but not positions. Only the state of a presence sensor may be available, but the API itself does not allow for requesting a derived position of the user.

2.3 Fog Computing and Access Control Models

The distributed nature of the IoT and the spatial relevance and availability of the data produced by sensor and actuator networks motivates the concept of moving processing and storage capabilities from the Cloud and distributing them geographically along the edge of the network. This distribution and decentralization can easily go hand in hand with the application of novel access control models, which are more suited to IoT contexts, where access to a resource may insufficiently be described by Discretionary or Mandatory Access Control models based on identities or roles and even be desired as dependent on the context of the local system.

2.3.1 Fog Computing

Fog computing has been defined as an extension to the Cloud Computing paradigm by Bonomi et al. [5] in 2012, in order to improve scalabily and overall responsiveness for data intesive applications, such as IoT systems. The main characteristics given to the Fog computing principles by the authors are (i) edge location, (ii) geographical distribution, (iii) Large-scale sensor network support, (iv) Support for very large number of nodes, (v) Support for mobility, (vi) Real-time interactions, (vii) Predominance of wireless access, (viii) Heterogeneity, (ix) Interoperability and federation, and (x) Support for on-line analytic and interplay with the Cloud [5]. Given these characteristics, Fog-Computing based IoT networks, and local access control can benefit from the reduced latency introduced by geographical distribution of data processing devices, called Fog nodes, which highly advocates the application of Fog-Computing principles. In addition, the distributed nature allows for offline-availability of resources and Fog-based access control mechanisms given a severed connection to the internet. Another major benefit of the Fog-Computing principle is achieved through the distribution of the users' data, thus greatly enhancing privacy and bringing

control over data back into their hands.

2.3.2 Access Control Models for IoT environments

The heterogeneous nature of the IoT and its partaking devices and the vast number of connected devices greatly reduces the applicability of traditional access control models such as Identity Based Access Control (IBAC) and incentivizes the application of novel concepts as described below.

Role Based Access Control (RBAC). RBAC has been scientifically documented by Ravi S. Sandhu in 1998 [44]. It allows for access rights to a resource to be assigned to certain roles within the system in such a way that each user assuming the specified role is allowed access to the resource.

Attribute Based Access Control (ABAC). ABAC was mentioned by Yuan and Tong [52] in 2005. It is introduces as an access control model for service oriented architectures that provides more fine-grained modelling of access rights than existing models. This is achieved through the design of the model which allows access rights to be based on subject, object, and environment attributes.

Capability Based Access Control (CapBAC). Based on the work of Dennis and van Horn [8] describing semantics for multiprogrammed computations, where computations are granted access to memory words depending on a list of capabilities, access control models for other applications have been developed [35]. Capabilities are oftentimes represented as a token allowing the user access to a resource. While generally highly applicable in heterogeneous systems, the application in IoT environments introduces the issues of capability propagation and revocation [35].

Directions for access control in IoT environments. In the outstanding work by Ouaddah et al. [35] a highly detailed overview over challenges for existing access control models during application for IoT contexts is given. The author's OM-AM model depicting Objectives, Models, Architectures and Mechanisms allowed them to classify different access control models according to the requirements introduced by the IoT.

Context-Awareness in Access Control. The term "interaction context" is easily understood by the reader, but consensus about the definition is hardly reached [1]. The most accepted definition is "any information, that can be used to characterize the situation of an entity" [9, 1]. One significant context variable is the user's location.
According to Ouaddah et al. [35], all mentioned models allow context awareness to some degree, whereas RBAC and CapBAC provide limited support for context-awareness. Roles fail to depict context variables altogether and the phenomenon of role explosion for complex systems as could be based on context-

aware dynamic role assignment as described in the model proposed by Zhou et al. [53] makes the system less manageable. The problem limiting CapBAC's context-awareness is the regular change in context variables and the subsequent propagation or revocation of single capabilities as the whole system has to be aware of each change, which introduces significant complexity. ABAC based models are context-aware per definition, since attributes of the environment are seen as context variables and ABAC allows for the creation of access policies based on those environmental attributes.

2.4 Resulting Implications For the Design of ILLAC

In order to develop the Indoor Localization framework for Location-aware Access Control, in short ILLAC, the following considerations were taken based on the research on related work. Selecting a localization framework with the aim of improving access control mechanisms through the provisioning of localization data about the user depends on many factors. For the development of a localization system that may even be deployed in smart homes the following requirements have to be fulfilled:

- (i) reliability
- (ii) response time
- (iii) cost efficiency
- (iv) ease of use
- (v) adequate precision

Most researched systems fulfill (i) and (ii), as well as (v) for our purposes. Since the use cases require the system to be affordable and easy to set up and operate, these will be the major factors impacting that decision. Table 1 reflects our classification of technologies according to the remaining requirements. This has led to the decision of making ILLAC's first iteration's Localization Engine BLE proximity detection based.

Since the granularity of a data model for location-aware access control has to depend on the accuracy of the information provided to it and there is little margin for error as access to critical resources may unjustifiably be denied, the data model has to be robust enough to compensate for tracking errors. The data models described in most cases provide positions in relative Cartesian coordinates, which may, depending on the error introduced by the tracking technology, be sub-optimal for evaluation in a policy evaluation engine. This led to the decision that most researched data models proved themselves to be inapplicable for the desired purpose but provided a reasonable starting point for the development of ILLAC's own location data model.

technology	cost efficiency	ease of use
BLE fingerprinting	+	-
BLE proximity detection	+	+
WIFI fingerprinting	+	-
802.11ad localization	-	-
Light-based localization	-	-
RFID localization	-	+
Inertial sensors	+	-
GSM signals	+	-
Computer vision	+	-
Ultrasound	+	-

Table 1: Comparison of technologies according to requirements

Building ILLAC upon Fog-Computing principles provides enhanced privacy and latency, since the data is being kept and processed in Fog nodes in close proximity to the user, ideally even in the user's own network. Hence, the user is not necessarily dependent on an active internet connection or a third party for data processing or storage.

ABAC is a highly suitable access control model for a context-aware, and in ILLAC's special case, a location-aware access control engine. Due to its granularity and flexibility, additional attributes such as user position are easily introduced and evaluated. Hence ILLAC will be built on an ABAC-based access control engine in order to provide enhanced security for the users of the system.

3 Solution Design

The following section provides details concerning the design and implementation for achieving location-aware access control with ILLAC. First, the technologies supporting ILLAC are introduced. Second, in subsection 3.1 an overview of the architecture of the system is given, followed by details about the developed Localization Engine and its components in subsection 3.2. Next, the Event Coordinator service responsible for coordinating context events in the system will be introduced in Section 3.3, before closing the design chapter with the presentation of the location-aware access control mechanisms built on aforementioned components in Section 3.4.

3.1 Architectural Overview

3.1.1 Used Technologies and Protocols

Python. Python is an object-oriented script programming language [39]. Due to its platform independence and active community providing libraries as well as the rich environment of standard and system libraries for many different protocols, it is extraordinarily well suited for rapid prototyping and deployment of small projects in different environments.

Spring Boot Framework. The Spring Boot Framework is a Java framework aimed at the easy and quick deployment of production-grade Java applications [37]. It embeds deployment environments for web applications like Apache Tomcat by default, and requires little configuration. Additionally it provides the benefit of easy integration of select starter dependencies into projects. It is used to develop and run all independent services In the CosyLab IoT security framework (see subsection 3.1.2) and provides the APIs for their consumers.

MongoDB. MongoDB is a document based database storing data in JSON-like format [30]. It provides Ad hoc querying as well as indexing and aggregation. MongoDB databases can easily be distributed and operated as clusters. Document based databases as opposed to relational databases do not have a semantic link between objects in the database by default. Such links have to be implemented by the user in the application's business logic.

RabbitMQ. One popular message broker for AMQP is RabbitMQ[36], which is used for inter process communication for some of the components around ILLAC. It allows for easy user and queue management through a web interface and is easily integrated into Spring Boot and efficient to use through annotations.

MQTT. Message Queuing Telemetry Transport (MQTT) is a lightweight messaging protocol enabling devices to communicate in constrained (e.g. IoT-) environments [12]. It follows the widely known publish/subscribe pattern and allows connected producers to publish messages to topics on a message broker.

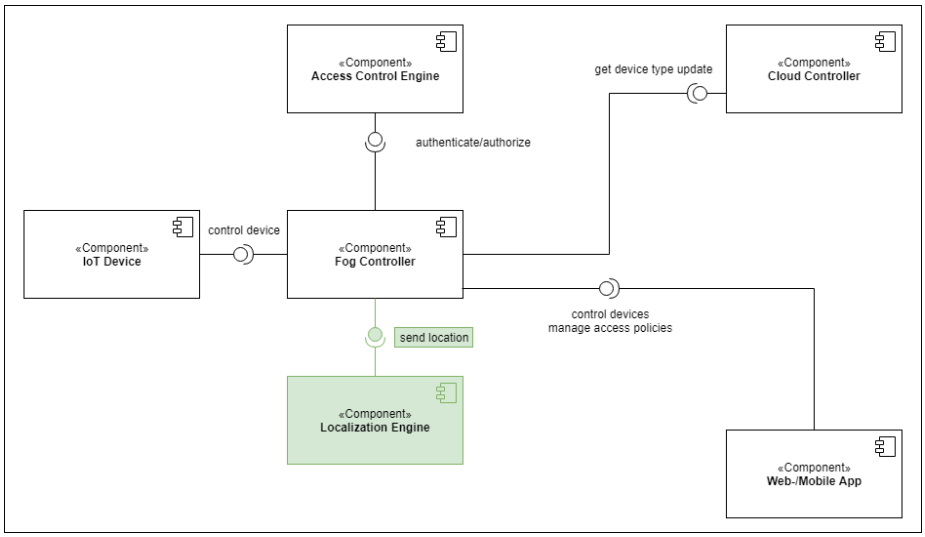

Figure 3: Overview of the architecture of CosyLab IoT security framework

A consumer can subscribe to any number of topics and receive the messages once published. As with many message queuing protocols, especially the "wildcard" subscriptions using `/superTopic/*` to subscribe to all subtopics of a super-topic. This feature proves itself to be useful when the exact names of the topics are unknown at the time of development due to generation during runtime.

AMQP. Advanced Message Queuing Protocol (AMQP) is a versatile communication protocol for the exchange of business messages between applications as well as organizations[33]. AMQP provides queue based messaging, as well as publish/subscribe patterns and remote procedure calls. For the purposes of ILLAC, queue based messaging and publish/subscribe pattern based communication was chosen and set up.

3.1.2 CosyLab IoT Security Framework

ILLAC has been designed and developed as part of an ongoing project at Cooperative Systems Research Group (COSY) at University of Vienna. To give broader understanding about the context and preconditions presented to the design of ILLAC, parts of the IoT Security framework currently under development as part of the COSY Lab will be described in the following paragraphs.

The CosyLab IoT security framework is a prototypical Fog-Computing based access control framework for application in smart home or smart building scenarios. A coarse overview of the components of the system can be seen in Figure 3. It currently consists of four major components and the projected minor

component developed for this thesis:

1. Cloud Controller
2. Fog Controller
3. Access Control Engine
4. Web-/Mobile App
5. *Localization Engine*

The following paragraphs provide an outline of the responsibilities of these components in the framework, as well as their interdependencies and interactions.

1. Cloud Controller. Since the CosyLab IoT security framework follows the basic principles introduced by the Fog-Computing concept, the Cloud Controller plays a minor role during the runtime of the system. Its main purpose is to serve as a registry for connected Fog nodes and supported device types, as well as their functionalities (e.g. "turn on" for smart outlets, "set color" for smart light bulbs). This allows for easy setup of new Fog nodes and hassle-free updates of device functionalities within the systems, as those are retrieved periodically from the Cloud component. While the Cloud-based component provides valuable services, it is not required for the operation of an instance of a Fog node, which consists of the following components.

2. Fog Controller. The Fog Controller can be seen as the core component of the system. It provides an interface for controlling the registered smart devices through a REST API. All requests towards changing or reading the state of a smart device have to be authorized against the Access Control Engine. Once authorized, the Fog Controller component reformats the request according to the specifications of the smart device and forwards it to the device. Furthermore, the Fog Controller provides an API that is passed through to the Access Control Engine for creating, updating and deleting access policies. Another essential element of the Fog Controller is the so-called *Event Coordinator (see Section 3.3.* It listens on message queues for context events happening in the system and redistributes received events to the appropriate components.

3. Access Control Engine. The Access Control Engine is currently, as of writing of this thesis, an ABAC-based policy evaluation engine. It provides a registry of user accounts and policies. These policies can be created for so-called "policy anchors", the specific resource to which access is to be controlled. This can be a single functionality of a device, (all functionalities of) a device, or all devices of a certain type. Currently, the engine only supports evaluation of policies based on the "functionality of device" and "device" anchors. Upon an incoming authorization request from the Fog Controller, the Access Control

Component	Queue/Topic	Activity
Localization Engine	localizationEngine.location	send
Localization Engine	localizationEngine.zones	send
Fog Controller	localizationEngine.zones	receive
Event Coordinator	localizationEngine.location	receive
Event Coordinator	eventCoordinator.localization	publish
Access Control Engine	eventCoordinator.localization	subscribe

Table 2: RabbitMQ Queues and Topics for Message Transmission (Exchanges are selected according to sending component)

Engine retrieves the relevant policies from its database and evaluates the request based on the attributes of the user and the system's context against the deposited policies for the requested resource. It then returns the result of this evaluation to the Fog Controller to either proceed with the requested operation, or inform the user of the failed authorization.

4. Web-/Mobile App. In order to provide an easy-to-use interface for interaction with policy, user, and device management, as well as for interaction with smart devices, an AngularJS [2] -based web- and a native Android[16] app are provided by the CosyLab IoT security framework. These apps communicate with the Fog Controller component through the provided HTTP REST API. Migration to AMQP as major communication protocol for the front end apps is currently being investigated. Since the web app and the Android app provide almost identical features, no differentiation between the two has been made here.

3.1.3 Communication Model

The framework is using Advanced Message Queue Protocol (AMQP) as protocol for communication between the described components. Each of the major components is assigned an exchange it declares its message queues or topics on. The Localization Engine sends its messages to a message queue, which has one consumer: the Event Coordinator component. The Event Coordinator component uses a publish/subscribe style topic exchange to send its messages to the subscribed consumers, namely the Access Control Engine and future components currently under development. The relevant topics and queues as well as their exchanges are laid out in Table 2.

3.2 Localization Engine

The development one of the Localization Engine was the main technical contributions to the project covered by this thesis. It was inspired by the scientific work of Han et al. [19], as well as the Happy Bubbles open source presence detection project[20], and developed towards to the needs of location-aware access

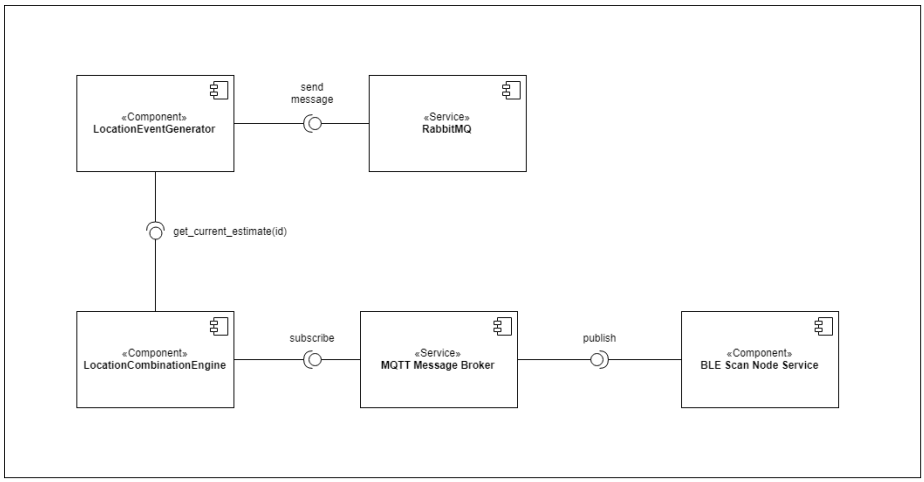

Figure 4: Architecture Diagram of the Localization Engine

control. It consists of three components working together to achieve room scale localization:

1. BLE-Scan Node Service
2. Location Combination Engine
3. Location Event Generator

These components and their inter-dependencies, as well as the message queue services used for communication between them are illustrated in Figure 4.

3.2.1 BLE-Scan Node Service

Each zone within the perimeter where localization is provided has to be outfitted with a proximity detection node. Upon first start, a version 4 universally unique identifier (UUID4) [26] is generated by the script running on the node and saved to a file for loading during future application (re)starts. During operation, the systems Bluetooth module is used to continually and periodically scan for nearby Eddystone unique identifier (uid) BLE beacons [17] broadcasting instance identifiers and namespaces. Using the bluepy.btle [22] and PAHO [11] MQTT libraries, recorded RSS values for each received beacon get published to the queue corresponding to its node identifier on an MQTT broker running on the device responsible for the evaluation of the signal strengths and subsequent determination of the beacons' positions. Figure 5 depicts the payload generated by the node and sent to the MQTT topic associated with the nodes identifier.

```
{
  "timestamp": 1234567890,
  "location": "2a3706d0-6f24-4469-ad50-1842b66dd3bd",
  "devices": [
    {
      "mac": "11:22:33:44:55:66",
      "rssi": "-60.2",
      "type": "eddystone",
      "namespace": "c051e10ca7e000000000",
      "instance": "000000000001"
    },
    {
      "mac": "66:55:44:33:22:11",
      "rssi": "-199.0",
      "type": "eddystone",
      "namespace": "c051e10ca7e000000000",
      "instance": "000000000002"
    }
  ]
}
```


Figure 5: Examplary payload published by a proximity detection node

3.2.2 Location Combination Engine

The location combination engine was developed as the Python script `BlEval.py`. Its structure allows for it to be imported into any other script and its member functions to be called. After initializing, the location combination engine connects to the MQTT broker, subscribes to all subtopics of `bleLoc` (`/bleLoc/*`), and waits for messages. Upon arrival of new RSS data from a node (as depicted in Figure 5), new node identifiers are saved if the zone was yet unknown. Subsequently the devices with the namespace identifier assigned to the systems environment are selected, and the instance identifiers and associated RSS values are extracted from the payload. Thereafter, the RSS values are saved in a two-dimensional dictionary according to the zone they were received, and the instance identifier they belong to. An example of this internal state and structure can be found in Table 3. The location combination engine provides a member function for retrieving the current position of an instance ID to its importing script. Upon call of the function, the zone of the device with the provided instance ID is determined through the comparison of the RSS values for each node for the instance ID. The node with strongest (highest) RSS value is assumed to be the closest to the device, a measure of confidence derived from the difference with regards to the next highest RSS is returned along with the assumed node identifier. Through the usage of another function provided by the location combination engine, the calling script can request an array of all zone identifier keys known to the engine.

instance ID / node ID	"2a37..."	"f6d3..."	"7762..."	"2e45..."
"000000000001"	-99.0	-20.5	-60.2	-255
"000000000002"	-255	-255	-255	-54.7
"000000000003"	-25.2	-55.0	-255	-166.8

Table 3: An internal state of the location combination engine

3.2.3 Localization Event Generator

The Localization Event Generator component relies on the Location Combination Engine. A record of the current position of each instance identifier in the system is kept and saved periodically by the event generator, to be reloaded upon component failure. Through regular calls of the zone estimation function of the combination engine, location changes are detected. Events are generated either upon such detected location changes, or after a predefined time of an instance ID remaining in proximity of the node. The generated events are subsequently sent to a message queue *"localizationEngine"* provided by RabbitMQ for further processing in the Event Coordinator component of the Fog Controller.

3.3 Event Coordinator

The purpose of the Event Coordinator component is the collection and redistribution of context events such as location, behaviour, user interaction or device state changes. Such events are generated by different engines in order to support context-aware access control. Currently, context events published by the Localization Event Generator component are received by the Event Coordinator through an assigned message queue, and subsequently published as slightly reformatted event objects with resolved zone objects (see Section 3.4.1) to a RabbitMQ topic exchange for potential consumption by other components in the framework, such as the Access Control Engine.

3.4 Location-Aware Access Control

3.4.1 Data Models for Location-Aware Access Control

In order to provide a representation of location and localization data, specific data models had to be developed. For representing the locations and easy policy creation, a zone object was modeled. For updates on user position, an event based model was selected.

Zones. The zone object, which is the highest necessary and achievable level of precision identified for the application in access rules, provides nullable string values of its location parameters (building, floor, room, and name). It also carries its specific node identifier as described in Section 3.2, since mapping of node identifiers to zone objects happens within the Fog Controller component. In accordance with the database model in the CosyLab IoT Security Framework,the zones were intentionally designed as not relational. This is also beneficial due to the fact that the policy evaluation engine, as an interchangeable component, is supposed to be entirely agnostic and independent of the structure of the locations. The proposed logical hierarchy within the zone object still allows for the creation of rules based on the presence in rooms, floors, or even just buildings. Figure 6 illustrates an exemplary zone object used throughout the framework. Newly detected zones have to be mapped manually through the web-/mobile app. It is recommended to start the node daemon on the node devices in sequence and map the zones alongside as to avoid confusion upon the appearance of multiple unmapped zone identifiers at the same time.

Events. In order to keep all components informed about the changes of context variables necessary for their operation, an event-driven information exchange model was chosen as to eliminate the need for database synchronicity or additional communication which would have come at the price of significant transmission overhead. Since the aim of the access control engine under development at COSY is the introduction of context-aware fine-grained access control in IoT environments, the decision to design an abstract context event definition was made. This ensures that every event carries a timestamp and the type identifier of its type (e.g. `TYPE_LOCALIZATION`, `TYPE_BEHAVIOUR`, ...) .

```
{
  "nodeId": "8aa8be8a-f59b-403b-a1c7-51289f82a6c9",
  "zone": "Center",
  "room": "Office 1.09",
  "floor": "1",
  "building": "S6"
}
```

Figure 6: JSON representation of a zone object

```
{
  "timestamp": 1561628471435,
  "type": "localization",
  "proxyId": "000000000001",
  "confidence": 0.43,
  "action": "entered",
  "zone": {
    "nodeId": "8aa8be8a-f59b-403b-a1c7-51289f82a6c9",
    "zone": "Center",
    "Room": "Office 1.09",
    "Floor": "1",
    "Building": "S6"
  }
}
```

Figure 7: JSON representation of a localization event

Localization events extend that base by a zone object as depicted in Figure 6, an action [`ACTION_ENTERED`(user entered a zone), `ACTION_LEFT`(user left a zone), `ACTION_REMAINED`(user remained in a zone for the a predefined interval)], a localization proxy identity equivalent to the instance ID from the Eddystone BLE beacon, and the measure of confidence generated by the Location Combination Engine (see Figure 7).

3.4.2 Location-Based Access Policies

Access policies in the Access Control Engine consist of one or multiple access rules. Access rules can be of numeric, String-based or Boolean types and, developed as part of this thesis, location based. There exist composite access rules which can consist of any number of rules of any type connected with the logical operators `AND` and `OR`. Numeric, String, and Boolean rules allow for the com-

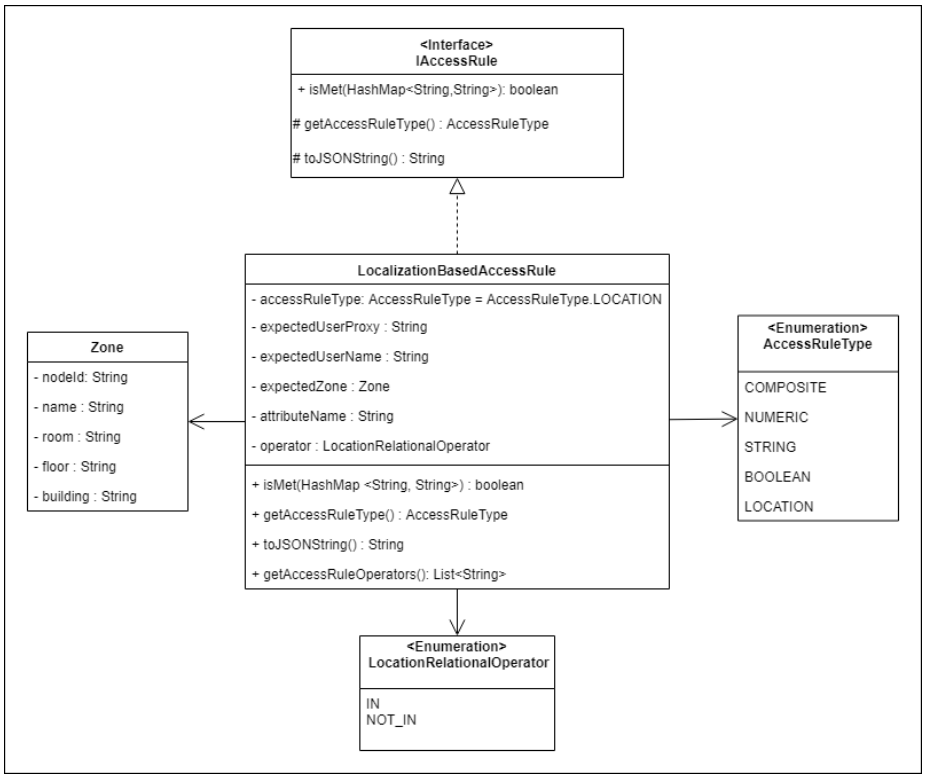

Figure 8: UML class diagram illustrating the location based access rule class and its dependencies

parison of any user attribute of the defined type to the attributes stated in the access rule. The LocationAccessRule type requires an expected zone object and a location specific operator [`IN` (user is currently in the specified zone), `NOT_IN` (user is currently outside of the specified zone)]. A class diagram illustrating the structure of localization based access rules can be found in Figure 8.

The fields of the zone object of the location access rule are nullable, which allows for the creation of policies not only based on presence within a zone, but also for access policies based on rooms, floors, or even buildings. While evaluating whether the rule is fulfilled by the users and the environments attributes, the access rule follows the logical hierarchy of the zone. If no zone name or identifier is specified rooms are compared, if no zone and no room are specified, only the floor and building are checked. Further steps up the hierarchy are treated analogously. Combined with the power of the recursive composite access rule, complex fine-grained access policies for resources can be created. Figure 9 presents an access policy where access to the resource is only allowed, if the user "Kaspar" with userProxy "000000000001" is in "Office 1.09" and the requesting user is over 15 years of age.

```
{
  "accessRuleType": "COMPOSITE",
  "accessRules": [
    {
      "accessRuleType": "NUMERIC",
      "expectedValue": 15,
      "attributeName": "Age",
      "operator": "GREATER_THAN"
    },
    {
      "accessRuleType": "LOCATION",
      "expectedUserProxy": "000000000001",
      "expectedUserName": "Kaspar",
      "expectedZone": {
        "room": "1.09 Office",
        "floor": "1",
        "building": "S6"
      },
      "attributeName": "Location_Event",
      "operator": "IN"
    }
  ],
  "operator": "AND"
}
```


Figure 9: Composite access rule with location component

4 Evaluation

4.1 Test Deployment

The tests validating ILLAC were conducted at the offices of COSY. The components of the Localization Engine were deployed to different computers throughout the premises (see Figure 11).

Most localization node services ran on Raspberry Pi Zero W, and Raspberry Pi 3 Model B and B+. The Raspberry Pi 3 Model B provides a Quad Core 1.2GHz 64bit CPU and 1GB of RAM [41], while the B + model increases CPU clock speed to 1.4GHz at 1GB of RAM [42]. The Raspberry Pi Zero W supplies a 1GHz single-core CPU and 512MB of RAM [43]. All of the models come with Bluetooth Low Energy out of the box, which makes their application ideal for the desired purpose of scanning for BLE beacons, especially considering the price point of €35,- for the 3 Model B+ and €10,- for the zero W. The tracked devices were Android smart phones equipped with the Android app "Beacon Simulator" by Vincent Hirribarren [21]. Beacon Simulator allows the user to specify beacon data which is subsequently broadcast by the smartphone according to specifications.

Since ILLAC is currently only supporting Eddystone format beacons, the simulator was configured to broadcast an Eddystone beacon. Transmission power was set to "Low $\sim$ -75dBm" to reduce the possibility of false positives in presence detection and the broadcast frequency was set to "Low latency $\sim$ 10Hz" to ensure timely detection. The namespace of the test environment was defined as C051E10CA7E000000000 (Cosielocate) and instance identifiers were given out as incremental integer numbers starting at 000000000001.

For the test deployment scenario, three nodes were distributed in the offices of COSY. One was deployed on a Raspberry Pi Zero W, one on a Raspberry Pi Model B, and the last one was deployed on the central node running on a Raspberry Pi Model B+. The distribution of the nodes within the office and the positioning in the rooms can be seen in Figure 11. The tests were conducted during regular work times with about 3 to 8 employees being present and moving around throughout the office, regularly opening and closing doors to each of the equipped rooms. The tracked device was either placed on the desk, table, or kept in the pocket during each test.

4.2 Proof of ILLAC's Operability

In order to provide proof of operability one detailed scenario for the Localization Engine and another one for location-aware access control in smart home contexts was defined.

4.2.1 Localization Engine

The correct operation of the Localization Engine is validated through a scenario involving all tracked zones and movements between them. Room changes have

Figure 10: Raspberry Pi 3 Model B and Raspberry Pi Zero W used during the evaluation

to be detected correctly and as fast as possible.

Validation Scenario:

1. The user starts with the tracked device in hand in room 1.10.
   Expected outcome: No change in position, last event was user entering room 1.10.

2. The user puts the tracked device into their pocket.
   Expected outcome: No change.

3. The user leaves room 1.10 and enters the hallway.
   Expected outcome: No immediate change.

4. The user slowly walks across the hallway and enters room 1.05.
   Expected outcome: A "leave"-event for room 1.10 and an "enter"-event for room 1.05 may already be generated upon entering.

5. The user remains in room 1.05 for two minutes, sitting down at the desk.
   Expected outcome: A "leave"-event for room 1.10 and an "enter"-event for room 1.05 has already been generated, or is generated early during the stay.

6. The user stands up and leaves room 1.05.
   Expected outcome: No change.

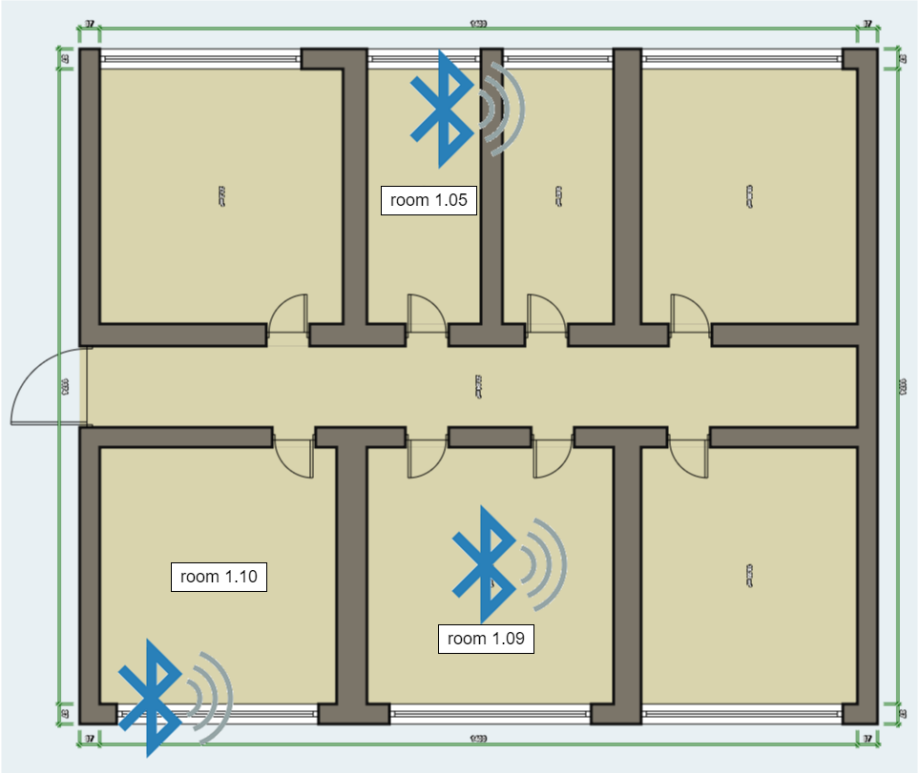

Figure 11: Deployment scenario: Localization nodes in offices of COSY. Each Bluetooth logo represents a single tracking node.

7. The user swiftly walks across the hallway and enters room 1.09.
   Expected outcome: A "leave"-event for room 1.05 and an "enter"-event for room 1.09 is generated.

8. The user removes the tracked device from the pocket.
   Expected outcome: Unless already generated, a "leave"-event for room 1.05 and an "enter"-event for room 1.09 is generated.

9. The user takes a seat in the armchair.
   Expected outcome: Unless already generated, a "leave"-event for room 1.05 and an "enter"-event for room 1.09 is generated.

10. The user places the tracked device on the table next to the armchair.
    Expected outcome: No change.

11. User and tracked device remain in position for another two minutes.
    Expected outcome: No change.

The deployment scenario chosen for conducting this test is described in Section 4.1.

Results. The test was executed and the system fired the events as expected albeit with present latency. Table 4 shows the timeline of the events as well as the time at which they occurred. The results are according to the expected outcomes and prove general operability of ILLAC's Localization Engine. A graphical representation of the route taken and relevant events as well as their timestamps can be seen in Figure 12.

4.2.2 Location-Aware Access Control in an Office Context

In order to prove operability of the location-aware access control component of ILLAC, a validation scenario was designed and executed. It consists of the definition of location based access policies on some of the resources available in the test environment. The test subject was ordered to interact according to the designed scenario, walking from room to room and accessing or reloading the device interaction page of the web app. Since policies were put in place for all features of the devices, once a device was shown in the list, the test subject was able to interact with all of its functionalities.

Validation Scenario:

1. The following policies are defined:

   - The user can access all features of the lamp in room 1.09 when present there.
   - The user can read the value of the temperature sensor in room 1.05 only when not present there.

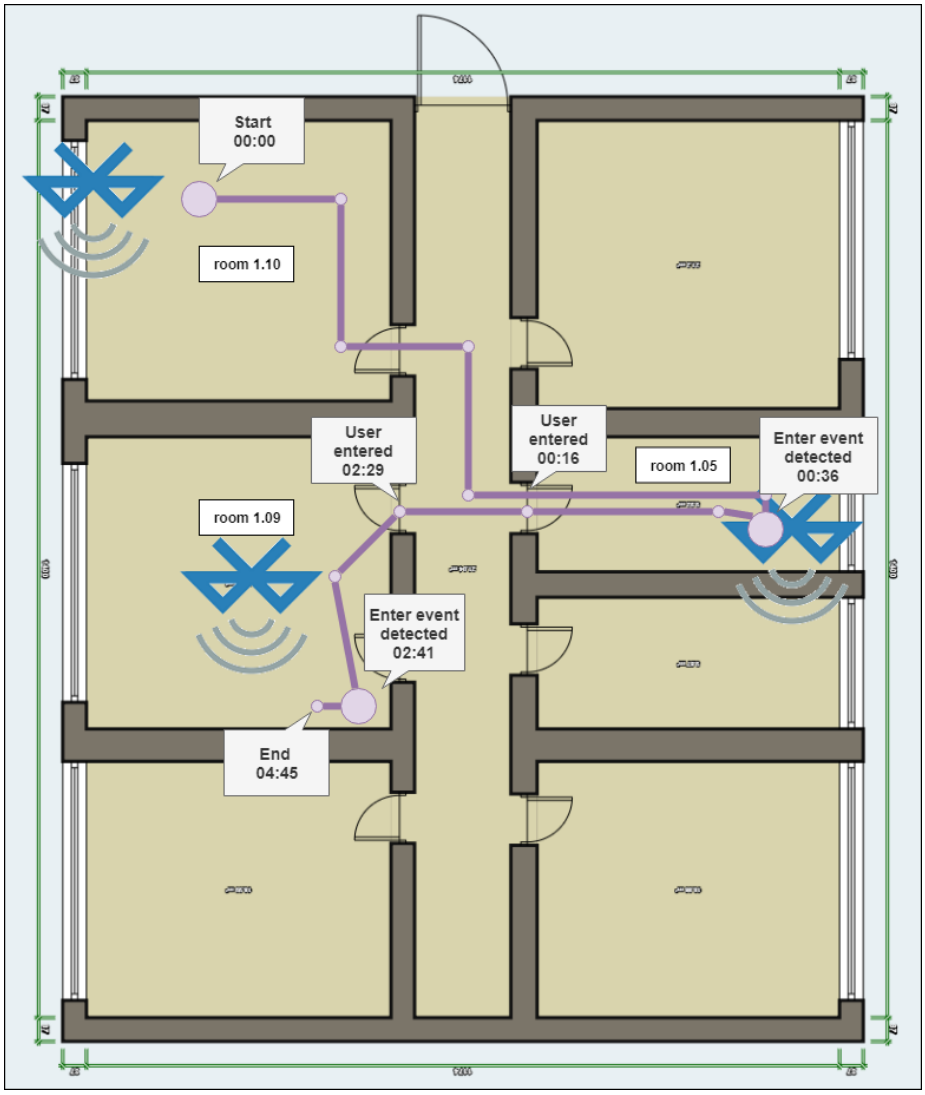

Figure 12: Selected events from Table 4 on the map of the deployment scenario

Time (mm:ss.SS)	Event
00:00.00	Start of scenario
00:06.42	Tracked device entered pocket of user
00:13.36	User left room 1.10
00:16.63	**User entered room 1.05**
00:23.10	User took a seat at the desk
00:36.14	Leave event for room 1.10 was generated
00:36.15	**Enter event for room 1.05 was generated**
02:24.24	User stood up
02:28.93	User left room 1.05
02:29.57	**User entered room 1.09**
02:36.17	User removed tracked device from pocket
02:41.23	User took a seat in the armchair
02:41.59	Leave event for room 1.05 was generated
02:41.60	**Enter event for room 1.09 was generated**
02:45.47	User placed tracked device on table
04:45.39	End of scenario

Table 4: Timeline of the proof of operability test for the Localization Engine

- The user can read the value of the light sensor in room 1.10 as long as in floor 1.

2. The user starts in the room 1.10.

3. The user refreshes the view on the sensors/devices tab of the web app.
   Expected outcome: Lamp interaction interface is not visible, sensor values of temperature and light sensor are displayed.

4. The user moves to room 1.09.

5. The user refreshes the view.
   Expected outcome: The lamp interface is visible and the lamp turns on upon toggle, sensor values of the light and temperature sensors are displayed.

6. The user attempts to toggle the lamp.
   Expected outcome: The lamp turns on.

7. The user moves to room 1.05.

8. The user refreshes the view.
   Expected outcome: The lamp interaction interface is not visible, the temperature sensor value is no longer displayed, the light sensor value is still displayed.

9. The user leaves the floor.

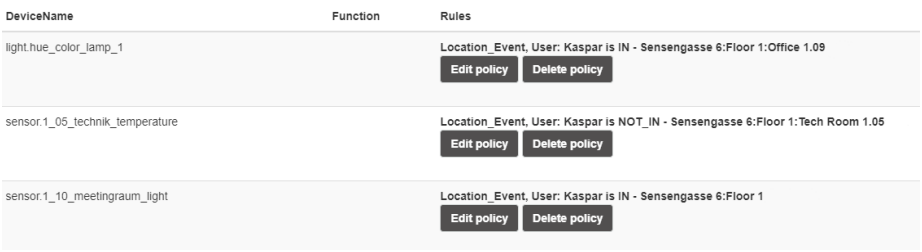

Figure 13: Policies for the proof of operability scenario

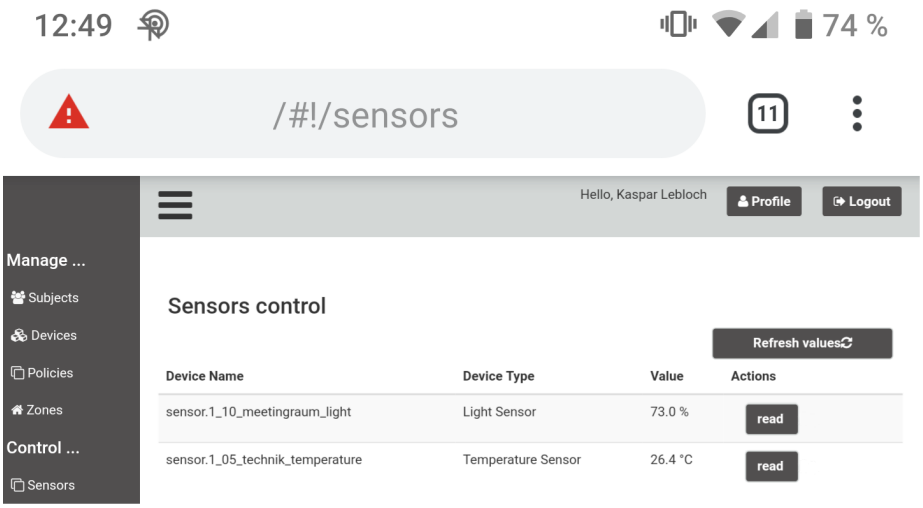

Figure 14: User interface when in room 1.10

10. The user refreshes the view.
    Expected outcome: The user can only read the value of the temperature sensor in room 1.05.

Execution and Results. First, the policies were set up according to the validation scenario. A screenshot of the policy page can be seen in Figure 13. The test was conducted by following the steps as defined in the validation scenario. A documentation of the interface states is provided through screenshots in Figures 14 to 17. The system worked as intended as long as it was given the necessary time for the system to detect the changes in localization (up to 30 seconds). For the performance evaluation of the system see Section 4.3.1). Access was granted or denied according to the access policies defined and the user position at the time of interaction.

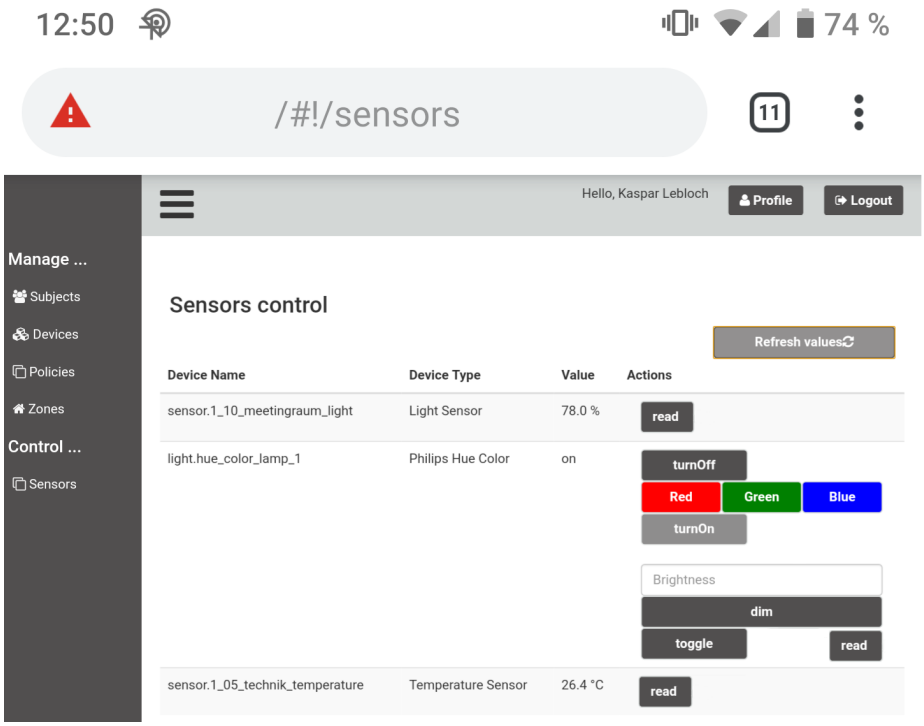

Figure 15: User interface when in room 1.09

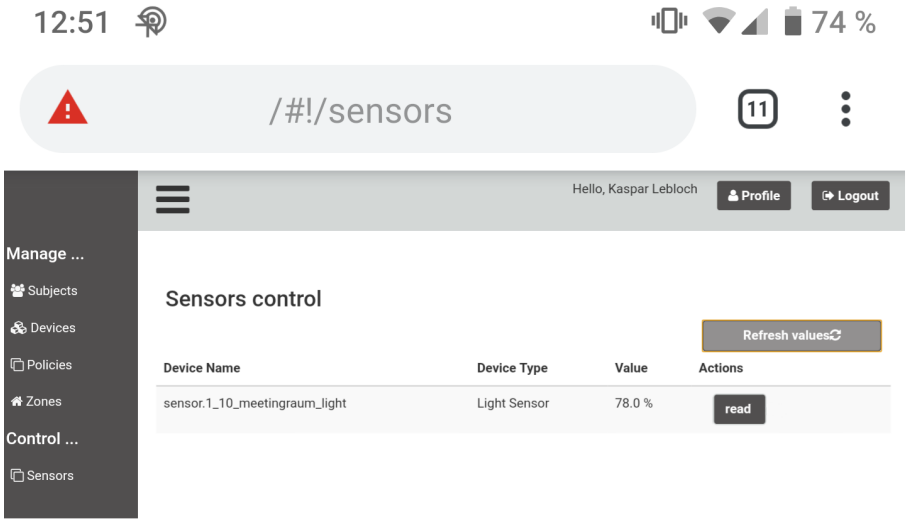

Figure 16: User interface when in room 1.05

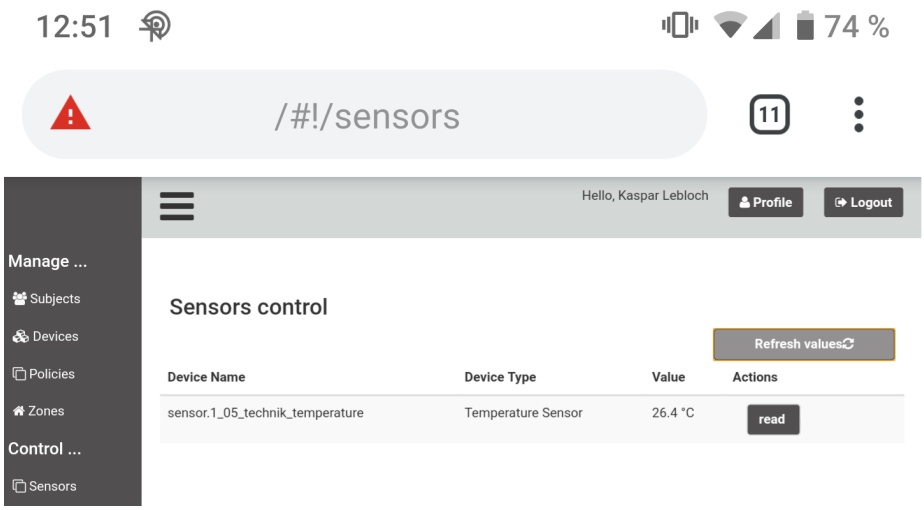

Figure 17: User interface after leaving the first floor

4.3 Performance Evaluation

4.3.1 Localization Engine

In order to determine the performance of the Localization Engine, four metrics were evaluated:

1. Latency and accuracy when entering a zone.
2. Latency and accuracy when leaving a zone.
3. Latency and accuracy when changing between two zones.
4. False positive events when staying in a zone.

Each latency test was conducted 50 times and the latency was measured manually using a browser based software stopwatch [18]. The results of the measurements are visualized in Figure 18. The node deployment was done according to the described deployment scenario (See Figure 11). False positive event measurement was done during thirty minute to one hour time frames and occurrences of unexpected events were documented for extraction of their number and cumulative duration of incorrect zone assignment.

Latency and Accuracy When Entering a Zone. In order to simulate the transition from an unknown zone to a known zone, the tracked Android phone was statically placed inside an office room. The beacon was turned on, and time measurement was started simultaneously. The mean detection latency of an "enter"-event was 9.54 seconds with a standard deviation of 5.53 seconds.

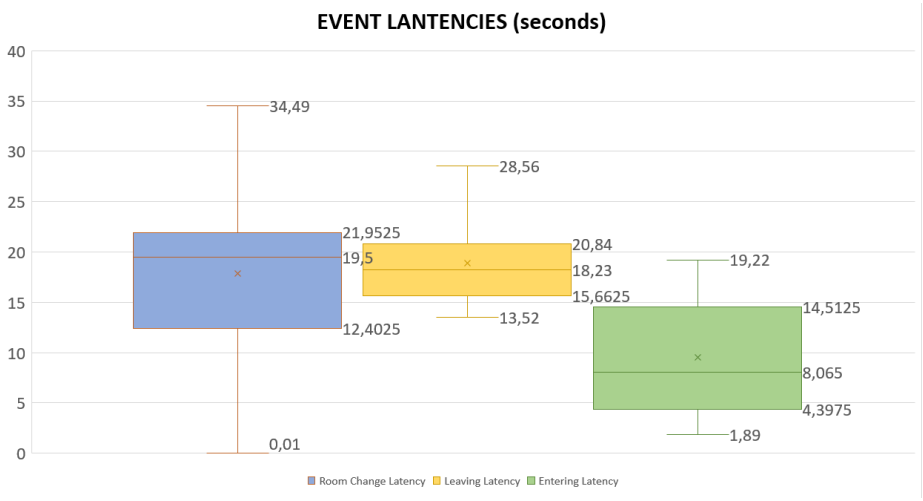

Figure 18: Box and whiskers plot of latency data of different scenarios

Half of the events were detected less than 8 seconds upon first signal transmission with a minimum of 1.89 seconds and a maximum of 19.22 seconds. This is to be attributed to the fact, that the periodical scan and publish of the node script may not always happen just after a person entered a room, and the periodicity of the estimation may add additional delay. During the tests, the Localization Engine was able to correctly assign the users position on the first try in 100% of the cases.

Latency and Accuracy When Leaving a Zone. The simulation of leaving a zone was achieved by turning off the beacon service on the tracked Android phone. Again, simultaneously time measurement was started. The mean detection latency of the "leave"-event was 18.91 seconds, with a standard deviation of 3.86 seconds. The median was 18.23 seconds, the longest detection took 28.56 seconds, and the shortest took 13.52 seconds. Detecting a "leave"-event always takes longer than any other event, as the tracked device has to leave the vicinity of all nodes to be recognized as no longer present. Given the configured publish intervals for the BLE Scan Nodes and the fact that they are not synchronized, detection delays of double the entering latency were to be expected. During the tests for generating "leave"-events, the engine was again able to correctly assign the users position to "left" on the first try in 100% of the cases.

Latency and Accuracy When Changing Between Two Zones. For the evaluation of the systems performance towards the detection of zone changes, a test subject was given a tracked device and ordered to walk between rooms equipped with tracking hardware. Upon entering the time measurement was started and upon detection of the correct room, it was stopped. The mean zone

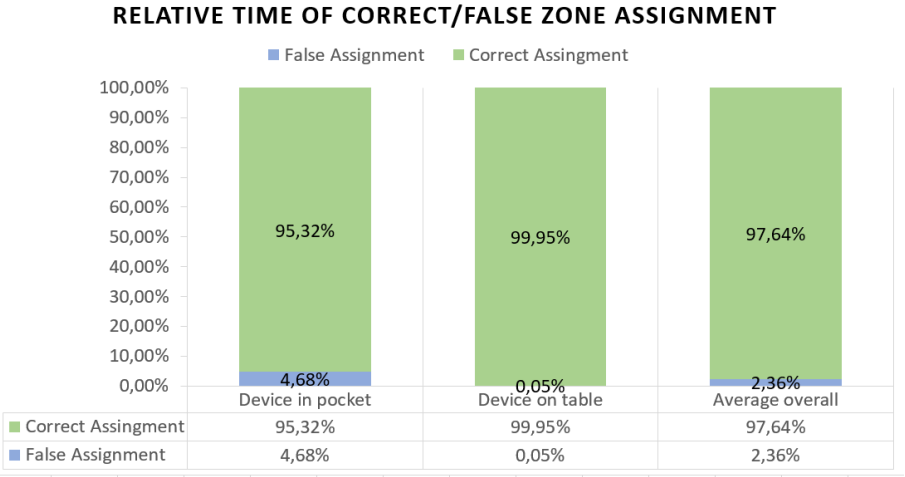

Figure 19: Relative time of correct and false room assignment for device in pocket and device on table scenarios as well as overall.

change detection (of the correct zone) was 17.84 seconds, with a standard deviation of 7.3 seconds. The median was 19.5 seconds, the maximum time taken to assign the user to the correct zone was 34.49 seconds and the minimum was 0.01 seconds. During most times when high detection times (>25 seconds) occurred, there was at least one false assignment preceding the assignment to the correct zone. In total, the engine was able to correctly assign the users position on the first try in 74% of the cases, but always assigned the correct zone after a maximum of the aforementioned 34.49 seconds.

False Positive Events When Staying in a Zone. The validation of the systems stability and resilience against false positive room change detection was tested by leaving the tracked device with the beacon service on in a room in the offices for 30 minutes to an hour, during different days of the week and at different times of day. The events fired by the engine for that device were extracted and false positive ("enter"- or "leave"-) events were counted. A total of 7 measurements were taken and there were 14 false positives detected, averaging out at 2.2 false positives per hour with false room assignment for about 2.7% of the total time. The results were significantly different for the "tracked device in pocket" versus the "tracked device on table" scenarios. The average percentage of false assignment duration for the "tracked device in pocket" scenario was 4.68% at 4.33 false positives per hour and 0.07% at 0.33 false positives per hour for the "tracked device on table" scenario. The physical distance between the two positions (pocket, table) was always less than 50 centimeters. Figure 19 visualizes the relative times of correct and false zone assignment for the pocket and table scenarios.

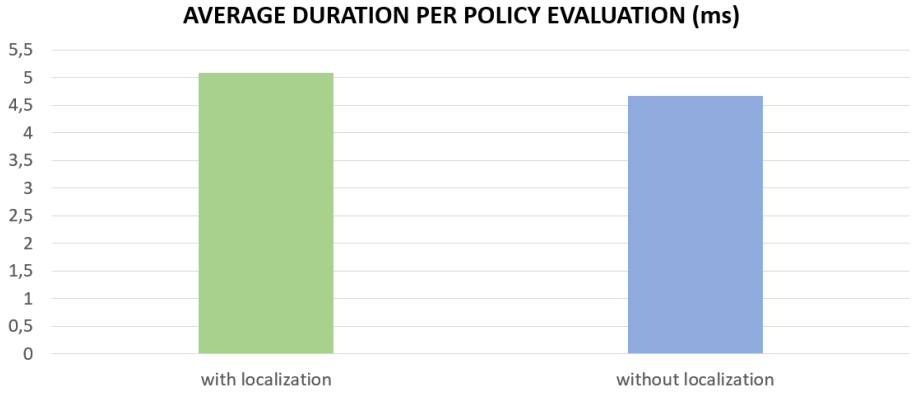

Figure 20: Comparison of average evaluation time per policy for location-based and location-agnostic policies

4.3.2 Policy Evaluation

In order to determine the performance impact of localization-based access policies on the existing system, policy evaluations were executed and timed in the access control engine. Simulation of a rich policy set was achieved by repetition of the validation procedure. Each test was conducted 50 times, for each the policies with and without localization-based rules. For each test, the validation was repeated for 10, 100, and 1000 iterations in order to detect whether the increase in evaluation time scales linearly. On average, each policy evaluation duration increased by 8.01% for localization-based access policies. The effect was slightly above average for the 1000 evaluations scenario with 9.2%, lowest for the 100 evaluations scenario with 7.34%, and 7.66% for the 10 evaluations scenario. Figure 20 illustrates the difference in average evaluation time for a policy with and without localization component, Figure 21 shows how location based access policy evaluation scales compared to location-agnostic policies when evaluating larger numbers of policies.

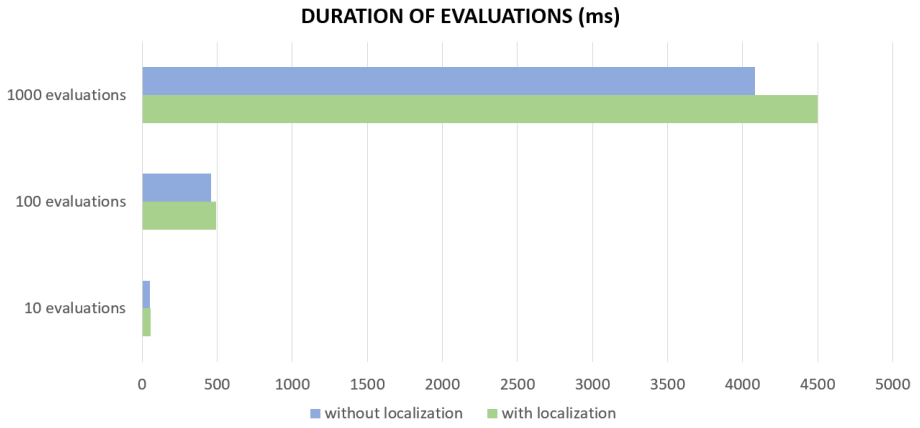

Figure 21: Comparison of 10, 100, and 1000 policy evaluations for location-based and location-agnostic policies

5 Conclusions and Future Work

This final section presents an elaboration on the research questions posed in Section 1.2 and further goes into detail about the author's ethical concerns regarding indoor tracking technology and concerns about the performance of ILLAC when applied in a smart home context.

5.1 Concerning the Posed Research Questions

5.1.1 RQ1: Affordable Localization Engine

The results presented in Section 4 show that reliability and accuracy were easily achieved in this first iteration of ILLAC running on a set of Raspberry Pi computers for the price of €10,- to €35,- plus SD cards and power supply.

The speed of detection. Since ILLAC's Localization Engine takes on average around 20 seconds to determine a change of user position, this question can only be answered subjectively according to the requirements of the consuming sytem. For location-aware access control in a context where users are stationary most of the time, this latency will suffice.

Reliability and accuracy. ILLAC's Localization Engine was able to detect every event correctly after enough time and produced few localization errors when users remained stationary. The component ran reliably for multiple weeks and picked up the user anytime tracking was enabled on the tracked device. Reliability and accuracy are seen as achieved.

5.1.2 RQ2: Location-Aware Access Control and Access Control Automation

The design of ILLAC's event based localization data model and location based access rules, described in Section 3, showed that access control can be supported by location data through a simple data model. The decided granularity for ILLAC was to have room scale localization for policy evaluation, which was achieved. The data model even allows for the application of a more precise localization engine to provide a higher level of granularity using the zones which can be defined as desired.

Introducing location as an additional parameter in access control models further supports the development of more secure default access policies and access control automation. This was proven to be viable in practice through the functional analysis and performance evaluation of the model's localization based access policies as well as their performance during the authorization procedure. Location-aware access policy evaluation did not introduce significant overhead in authorization times and policy management based on positions was easily executed. For the advancement of access control automation, insights gained from the analysis of movement profiles leading to automatic generation of location-based access policies is an option worth investigating.

Possible problems with the application of location tracking systems such as legal obligations introduced by handling movement data and other concerns regarding data security are further discussed in the following sections.

5.2 Performance and User-Acceptance

Given the fast pace of today's societal lifestyle, interactions with and orders given to devices are expected to be executed instantaneously. The thought of having to wait up to half a minute to be allowed to turn on the light in a room the user just entered is not exactly appealing given the user's understandable demands to a smart home. Pairing these requirements with ILLAC's current performance creates demand for a mitigation strategy towards either the improvement of ILLAC's detection latency or an increase in security high enough to sell the downsides. For some use cases such as the smart stove scenario described in Section 1.1.1, this might be the case, for others such as in Section 1.1.2, where the alternative is the turn of a key, it will certainly be not.

5.3 Privacy: Legal and Ethical Concerns

When it comes to location tracking, people are usually easily concerned, and rightfully so. Apart from the danger resulting from having an observing malicious entity knowing your position in real time, keeping a database of movements or even movement patterns of a person, can lead an observer to a series of conclusions about an observed persons lifestyle, health or, in an employer-employee context, assumed productivity. Especially given the fact that a lot of times the user setting up and controlling the system is already the person in power within

the deployment context, the application of ILLAC requires a lot of trust from the users not only towards the system being secure against intrusion, but also towards the system administrators benevolence. ILLAC is currently in a proof of concept phase and not sufficiently secured against threats (see Section 5.4), be it from inside or outside the system. Nonetheless, even a completely secure system leading up to the access control engine my never be enough to protect the users' privacy. With sufficient knowledge of the systems' access policies, an entity with access to any account on the system, could, by probing through periodical requests onto a resource and checking whether access has been granted or denied, build a movement profile of a user within the system. This leaves us with few reasons to adopt location-aware access control in many contexts, but there will be some, where the enhanced security through strict access control will outweigh the loss of privacy.

5.3.1 The Privacy Advantages of Fog-Computing

As described in Section 2.3.1, one of the main benefits of the Fog-Computing principle is that it brings the capabilities conventionally provided by Cloud services closer to the end user. This immediate proximity is ideally in the user's own network, under the user's own control. The decentralization and distribution of the user's data not only reduces the amount of transmissions traversing possibly insecure networks, but also decimates the potential damage in case of a breach. The user, as self-sovereign platform operator is a target of minimal value for an attacker in the first place, compared to a Cloud platform with potentially millions of users. Even if an attack occurs, only the data of a small number of users is leaked.

5.3.2 Operational Responsibility

Location data of the acquired granularity is highly sensitive. It is understandable that few users would want to leave this sort of data in the hands of a corporation, so the Fog-Computing approach offering self-sovereignty is attractive. Still, any tech-savvy user would want access to the source code to convince themselves that there are no back doors implemented. Even after having assured themselves of no malicious code and state-of-the-art system security, some users may never want to take the responsibility of keeping the system up to date and resilient against upcoming threats. Most may not even think about this problem to begin with. A decision about the operation of a system the likes of ILLAC has to be sufficiently informed and the user has to keep the attached responsibilities in mind.

5.3.3 Securing Data Against Abuse From the Inside

When it comes to the data generated by ILLAC abuse is possible in a manifold of ways. While ILLAC was designed to be an indoor localization framework for enabling location-aware access control, there is a plethora of applications for the information produced. In a smart home context, the data can be used to trigger automations based on presence in different rooms in the house, and reduce overall energy consumption. In an office context, it can be used to automate timekeeping, or Heating, Ventilation, and Air Conditioning (HVAC). This could save employees tedious hours of filling out bland time sheets, and saving employers fair amounts of money in energy costs. The problem arises when the system is used in a way, that was not intended or approved by all affected parties. Keeping track of an employees toilet habits, or their regular trips to the coffee machine or their long stay in the break room may let an employer draw conclusions not doing the employees performance justice.

5.3.4 Data Protection - GDPR Compliance

Since the 25th of May 2018 the General Data Protection Regulation (GDPR) is applicable in the European Union. Although the GDPR is just an EU regulation, its effects are already noticeable on a global scale. As of now, there are discussions about unifying privacy law in the United States of America and there has been an adequacy decision by the European Commission on Japan ensuring data flow based on strong protection guarantees [13].

Regarding the legal assessment of ILLAC in the light of data protection law, it makes sense to split the application domain into two possible scenarios: Home use and Workplace use.

Home use: In Article 2 of the GDPR, the material scope of the regulation is defined. Paragraph 2 lit. c states that "This Regulation does not apply to the processing of personal data: by a natural person in the course of a purely personal or household activity" [48]. Since ILLAC is supposed to be deployed as a Fog-Computing based component and operated by the users themselves without the support or interference of any third party, it can be argued, that the GDPR does not apply to this use case. This would also include the data generated about family members.

Workplace use: The processing of location data in connection with user identity by an employer is a scenario which falls into the scope of the GDPR. In order to justify this processing of personal data, Article 6 of the GDPR provides several possible justifications for the lawfulness of this processing, out of which the possibly relevant are presented [48]:

1. **Consent of the employee:**
   This could generally apply, but the relationship of dependence between

an employee and an employer may induce serious doubts of the voluntary nature of consent in such a scenario.

2. **Contractual obligations of the system operator:**
   This scenario is hardly arguable, since the contract in question has to be between the data subject and the processing entity. This could be a contract set up between employee and employer forcing the employee to provide access to personal location data for the benefit of the operation. Such a contract has to be assessed towards its lawfulness on an individual case basis. Again, the voluntary nature of the employees consent to signing such a contract may be seriously doubted.

3. **Legitimate interests:**
   Depending on the situation of deployment, an employer may have legitimate interests to secure the operation to a very high degree and introduce employee-location-based access control. Every time such legitimate interests are invoked, a weighing of interests has to be performed to decide whether the privacy needs of the employee outweigh the processing needs of the employer.

5.4 Future Work and Open Issues

While ILLAC has demonstrated to be viable proof of concept for location-aware access control in Fog-Computing based IoT environments, there is some potential for improvement. Mainly an improvement of performance of the Localization Engine, especially concerning detection latency as well as some privacy-enhancing features (having been declared out of scope for this thesis) remain on the list of future work for ILLAC to be a market-ready deployable framework.

5.4.1 Improving Localization Accuracy

Although the performance of ILLACs policy evaluation processing has proven to be more than satisfying for the application in smart home contexts and small office contexts, the precision and latency of the Localization Engine as well as the errors upon room change in the produced localization data leave room for improvement. The inevitable advent of 5G technology and upcoming WiFi standards may provide an interesting starting point for future research into indoor localization techniques, especially considering sub-meter accuracy and precise device orientation detection has already been reached using 802.11ad WiFi[23, 4]. Despite WiFi fingerprinting techniques showing promising results in localization latency and precision, the technology is in most cases dependent on rigorous setup procedures, which are oftentimes complex or error prone for the end user. In order to enable future improvements the Localization Engine was designed in a modular way and allows simple additions of new location determination components of any technology. By simply extending the provided

interface and exposing the corresponding functions to the location event generator, a new localization (combination) engine can be added as an additional source of information. The location event generator can then choose a location depending on the confidence provided by each engine as to support highest tracking reliability.
Since the major proportion of the false positive localization events were generated during the "tracked device in pocket" scenarios with the user sitting at a table and the tracked device being under the table, using devices that are usually carried above the waistline may improve overall accuracy of the Localization Engine. BLE beacons don't necessarily have to be emulated by smart phones and are oftentimes even sold as wristbands. Assuming the tracked devices don't move into a position where reflection phenomena are overly prevalent, the accuracy of the Localization Engine would increase.

5.4.2 Improving Localization Latency

Improvements to localization latency are expected to be achievable, since the localization solution presented by Han et al. [19] also based on BLE proximity detection. Using thresholding instead of ILLAC's direct comparisons of RSS values the authors were able to lower maximum detection latency to 10 seconds.

The main problem with latency in ILLAC's Localization Engine is the synchronous existence of presence-implying received signal strength values for multiple node identifiers. This situation may be mitigated by synchronizing the scans performed in the nodes in order to ensure that each measurement is a momentary snapshot of the systems state. Given all proximity values in the Localization Engine are refreshed at the same time, no old values recorded prior to user movement may introduce noise into the evaluation.

Another more trivial approach may be increasing the scanning and position determination frequency. While the system is currently running stable, increasing the frequency may bottleneck the system and delay its position determination function due to too many received messages. If a large number of nodes is deployed, this issue will scale and potentially cause a decrease in performance. The frequency of five seconds was chosen in order to ensure a stable operation of the localization component and allow for future scalability.

5.4.3 Privacy Enhancing Features

In its current state as a proof of concept the Localization Engine relies on static broadcast identifiers, which are easily impersonated since any user can detect and copy the Eddystone uid BLE beacon namespace and instance identifiers and rebroadcast them at any time. Moreover, the static broadcast allows for third parties to set up tracking hardware outside of or close to the smart envir-

onment and use the beacons for tracking even against the users will. In order to mitigate this form of abuse and the associated security risk Google provides the "Eddystone eid" BLE beacon standard [17], which generates the broadcast identifier from a predefined key and a time offset that is also broadcasted. To confirm the identity of an eid beacon a Cloud API is provided, reintroducing a dependency on internet connectivity and thus preventing pure offline operation. A proprietary implementation of such an algorithm is entirely possible, and will be goal of future iterations of ILLAC.

5.4.4 Mobile App Integration

In order to keep necessary setup effort minimal, integrating the BLE beacon functionality into the mobile app poses an attractive solution. This would require to integrate a component for live updates on Bluetooth characteristics on the smart phone. The main benefits of such an approach are easy and fully automatic synchronization of proxy identity indentifiers, and in case of the implementation of more privacy preserving beacon technology as mentioned in 5.4.3, automatic as well as secure key exchange. This drastically increases usability by reducing necessary user interaction during setup of the system.

5.4.5 Limitations of the Technology

Throughout the work on this thesis and in discussions with colleagues some issues that may create problems leading to unintended behaviour were brought up. One obvious problem is the used proxy identity device: Unless the BLE beacon device is permanently and inseparably affixed to the user's body, the proxy identity device can always be separated from the person it is assigned to. This can lead to abuse by unauthorized users, as they can simply seize the tracked device from an authorized user and move it to the required zone in order to gain access to a desired restricted resource. When it comes to remote authorization scenarios, as described in 1.1.2, the Fog node is again dependent on internet connectivity in order to notify the resource owner of the requested access and prompt the owner for authorization.

5.5 Summary

ILLAC has been a project developed over the course of a year providing a proof of concept for the envisioned solution for the automation of access control in smart home environments. Moreover, I do firmly believe that the contributions made to the research concerning security in IoT contexts with special focus on smart access rights management are of significant weight. While legal aspects may prevent the application of ILLAC in many business contexts where skilled technicians for setting up complex access control schemes are generally available

as a mitigation strategy, it will still be applicable in smart homes and can at least assure better physical and information security for home users.

Since the testing of ILLAC was done on generally affordable and easily available hardware and the achieved performance was sufficiently satisfying for a first iteration of a prototype, I have high confidence that indoor localization and functional location-aware access control can and will be available for everyone.

Future improvements made to the Localization Engine both in terms of latency and accuracy could make ILLAC even more viable for the application in the development of a new generation of smart home systems, given the potential for energy savings and comfort that can be achieved through a location-aware smart home.

Overall, working on the project leading up to this thesis has been a very interesting and insightful experience.

References

[1] Alegre, U., Augusto, J. C., and Clark, T. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software 117* (2016), 55–83.

[2] AngularJS. Angularjs web development framework. `https://angularjs.org/`. (Accessed: 2019-07-01).

[3] Bahl, P., Padmanabhan, V. N., Bahl, V., and Padmanabhan, V. Radar: An in-building rf-based user location and tracking system. Institute of Electrical and Electronics Engineers, Inc. ACM SIGMOBILE Test-of-Time Paper Award, 2016.

[4] Bielsal, G., Palacios, J., Loch, A., Steinmetzer, D., Casari, P., and Widmer, J. Indoor localization using commercial off-the-shelf 60 ghz access points. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (2018), IEEE, pp. 2384–2392.

[5] Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (New York, NY, USA, 2012), MCC '12, ACM, pp. 13–16.

[6] Bouet, M., and Dos Santos, A. L. Rfid tags: Positioning principles and localization techniques. In *2008 1st IFIP Wireless Days* (2008), Ieee, pp. 1–5.

[7] Chintalapudi, K., Padmanabha Iyer, A., and Padmanabhan, V. N. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 173–184.

[8] Dennis, J. B., and Van Horn, E. C. Programming semantics for multiprogrammed computations. *Communications of the ACM 9*, 3 (1966), 143–155.

[9] Dey, A. K. Understanding and using context. *Personal and ubiquitous computing 5*, 1 (2001), 4–7.

[10] Durrant-Whyte, H., and Bailey, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine 13*, 2 (2006), 99–110.

[11] Eclipse Foundation. Eclipse PAHO - MQTT and MQTT-SN software. `https://www.eclipse.org/paho/`. (Accessed: 2019-07-01).

[12] Edited by Banks, A., Briggs, E., Borgendale, K., and Gupta, R. MQTT Version 5.0, OASIS Standard. `https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html`. (Accessed: 2019-06-25).

[13] European Comission. Japan Adequacy Decision. http://europa.eu/rapid/press-release_IP-19-421_en.htm. (Accessed: 2019-07-01).

[14] Faragher, R., and Harle, R. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)* (2014), vol. 812.

[15] Faragher, R., and Harle, R. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications 33*, 11 (2015), 2418–2428.

[16] Google LLC. Android smart phone operating system. https://www.android.com/. (Accessed: 2019-07-01).

[17] Google LLC. Eddystone format. https://developers.google.com/beacons/eddystone. (Accessed: 2019-06-25).

[18] Google LLC. Google Stopwatch. https://www.google.com/search?q=stopwatch. (Accessed: 2019-07-01).

[19] Han, G., Klinker, G. J., Ostler, D., and Schneider, A. Testing a proximity-based location tracking system with bluetooth low energy tags for future use in the or. In *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)* (2015), IEEE, pp. 17–21.

[20] Happy Bubbles Technology. Happy bubbles presence detection. https://www.happybubbles.tech/. (Accessed: 2019-06-25).

[21] Hirribarren, V. Beacon simulator android app. https://play.google.com/store/apps/details?id=net.alea.beaconsimulator. (Accessed: 2019-06-30).

[22] Ian Harvey. bluepy - a Bluetooth LE interface for Python. https://ianharvey.github.io/bluepy-doc/l. (Accessed: 2019-07-01).

[23] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 3: Enhancements for very high throughput in the 60 ghz band. *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012)* (Dec 2012), 1–628.

[24] Jin, G.-y., Lu, X.-y., and Park, M.-S. An indoor localization mechanism using active rfid tag. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)* (2006), vol. 1, IEEE, pp. 4–pp.

[25] Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., and Munich, M. E. The vslam algorithm for robust localization and mapping. In *ICRA* (2005), pp. 24–29.

[26] Leach, P., Mealling, M., and Salz, R. A universally unique identifier (uuid) urn namespace. `https://tools.ietf.org/html/rfc4122`. (Accessed: 2019-06-25).

[27] Lee, S., and Husen, M. N. Recalibration-free indoor localization with wi-fi fingerprinting of invariant received signal strength. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), IEEE, pp. 4649–4655.

[28] Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., and Zhao, F. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing* (2012), ACM, pp. 421–430.

[29] Mittal, A., Tiku, S., and Pasricha, S. Adapting convolutional neural networks for indoor localization with smart mobile devices. In *Proceedings of the 2018 on Great Lakes Symposium on VLSI* (2018), ACM, pp. 117–122.

[30] MongoDB, Inc. What is MongoDB. `https://www.mongodb.com/what-is-mongodb`. (Accessed: 2019-07-01).

[31] Ni, L. M., Liu, Y., Lau, Y. C., and Patil, A. P. Landmarc: indoor location sensing using active rfid. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).* (2003), IEEE, pp. 407–415.

[32] Niu, L., Matsumoto, S., Saiki, S., and Nakamura, M. Considering common data model for indoor location-aware services. In *Proceedings of the 4th International Workshop on Location and the Web* (2014), ACM, pp. 25–32.

[33] Organization for the Advancement of Structured Information Standards. About AMPQ. `https://www.amqp.org/about/what`. (Accessed: 2019-06-25).

[34] Otsason, V., Varshavsky, A., LaMarca, A., and De Lara, E. Accurate gsm indoor localization. In *International conference on ubiquitous computing* (2005), Springer, pp. 141–158.

[35] Ouaddah, A., Mousannif, H., Elkalam, A. A., and Ouahman, A. A. Access control in the internet of things: Big challenges and new opportunities. *Computer Networks 112* (2017), 237–262.

[36] Pivotal Software. Rabbitmq. `https://www.rabbitmq.com/`. (Accessed: 2019-07-01).

[37] Pivotal Software. Spring boot. https://spring.io/projects/spring-boot. (Accessed: 2019-06-30).

[38] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (2000), ACM, pp. 32–43.

[39] Python Software Foundation. Python3 FAQ. https://docs.python.org/3/faq/general.html#what-is-python. (Accessed: 2019-06-25).

[40] Rai, A., Chintalapudi, K. K., Padmanabhan, V. N., and Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (2012), ACM, pp. 293–304.

[41] Raspberry Pi Foundation. Raspberry pi 3 model b specs. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. (Accessed: 2019-06-30).

[42] Raspberry Pi Foundation. Raspberry pi 3 model b+ specs. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. (Accessed: 2019-06-30).

[43] Raspberry Pi Foundation. Raspberry pi zero w specs. https://www.raspberrypi.org/products/raspberry-pi-zero-w/. (Accessed: 2019-06-30).

[44] Sandhu, R. S. Role-based access control11portions of this chapter have been published earlier in sandhu et al. (1996), sandhu (1996), sandhu and bhamidipati (1997), sandhu et al. (1997) and sandhu and feinstein (1994). vol. 46 of *Advances in Computers*. Elsevier, 1998, pp. 237 – 286.

[45] Shao, S., Khreishah, A., and Khalil, I. Retro: Retroreflector based visible light indoor localization for real-time tracking of iot devices. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (2018), IEEE, pp. 1025–1033.

[46] Signify Holding. The official site of philips hue. https://www2.meethue.com/en-usl. (Accessed: 2019-07-01).

[47] Smart Things. Smart things API documentation. https://docs.smartthings.com/en/latest/ref-docs/reference.html. (Accessed: 2019-07-01).

[48] The European Parliament and the Council of the European Union. General Data Protection Regulation. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679. (Accessed: 2019-07-01).

[49] Tóth, Z., Magnucz, P., Németh, R., and Tamás, J. Data model for hybrid indoor positioning systems. *Production Systems and Information Engineering 7*, 1 (2015), 67–80.

[50] Want, R., Hopper, A., Falcao, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems (TOIS) 10*, 1 (1992), 91–102.

[51] Ward, A., Jones, A., and Hopper, A. A new location technique for the active office. *IEEE Personal communications 4*, 5 (1997), 42–47.

[52] Yuan, E., and Tong, J. Attributed based access control (abac) for web services. In *IEEE International Conference on Web Services (ICWS'05)* (July 2005), p. 569.

[53] Zhou, Z., Wu, L., Hong, Z., Liang, Z., Jun, L., Sheng-Jun, X., Dengfeng, C., Peng, X., Peixin, Q., Xilong, Q., et al. Context-aware access control model for cloud computing. *International Journal of Grid and Distributed Computing 6*, 6 (2013), 1–12.

[54] Zhuang, Y., Yang, J., Li, Y., Qi, L., and El-Sheimy, N. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors 16*, 5 (2016), 596.

6 Appendix

6.1 Deutsche Zusammenfassung

Als eine der Säulen von IT-Sicherheit haben Zugriffskontrollmechanismen in den letzten Jahren eine Vielzahl an Verbesserungen und neuen Konzepten präsentiert bekommen. Nachdem besonders im Kontext von IoT oft höchstsensible Daten behandelt werden, sind die Anforderungen an die Sicherheit solcher Systeme außerordentlich hoch. Diese Umstände führten unter anderem zur Entwicklung von Zugriffskontrollmodellen wie Attribute Based Access Control, Capability Based Access Control, oder anderen Konzepten, die sogar Kontextinformationen berücksichtigen können und die Automatisierung von Zugrifskontrollmanagement ermöglichen. Diese Arbeit behandelt die Konzeption, Umsetzung, und Funktionsüberprüfung einer Lokalisationskomponente für ein Zugriffskontrollsystem, das für Fog-Computing-basierte IoT Umgebungen entwickelt wurde. Es wird das Design und die Umsetzung sowohl eines eventbasierten Bluetooth Low Energy - Lokalisierungssystems, als auch von positionsbasierten Zugriffsrichtlinien diskutiert. Weiters enthält diese Arbeit einen durch Experimente erbrachten Nachweis für die ordnungsgemäße Funktion des Lokalisierungssystems und der positionsbasierten Zugrifssrichtlinien, die ein technisches Argument für die Anwendung von positionsbasierter Zugriffskontrolle in IoT-Umgebungen unterstützen.