

UNIVERSITY OF BAMBERG

International Software Systems Science Degree Program in the
Faculty of Information Systems and Applied Computer Sciences

MASTER'S THESIS

Context-aware Anonymization of Mobility Data for Hospital BLE-based Asset Tracking

BY

Marcel Soler Lopez

Supervisors:
Prof. Dr. Daniela Nicklas,
Simon Steuer

Chair of Mobile Systems

Bamberg 2026

Dieses Werk ist als freie Onlineversion über das Forschungsinformationssystem (FIS; https://fis.uni-bamberg.de) der Universität Bamberg erreichbar.
Das Werk steht unter der CC-Lizenz CC BY.

Lizenzvertrag: Creative Commons Namensnennung 4.0
https://creativecommons.org/licenses/by/4.0/



URN: urn:nbn:de:bvb:473-irb-117599x
DOI: https://doi.org/10.20378/irb-117599

Abstract

The increasing use of digital technologies in healthcare settings raises significant concerns about the privacy of mobility data. This thesis investigates how location privacy can be preserved when tracking medical assets –specifically VAC pumps– using Bluetooth Low Energy (BLE) technology in hospitals. Conducted as part of the Explanym project, which aims to develop explainable anonymization techniques for mobility data, the work addresses the research question: *How can privacy be preserved for hospital mobility data, taking into account contextual factors?* The methodology combines requirements engineering and privacy threat modeling using LINDDUN GO. Based on this input, the open source application PrivProxy was developed. PrivProxy applies Location Privacy-Preserving Mechanism (LPPM) based on the contextual state of a tag and includes a front-end interface to explain the applied mechanisms to users. Designed with scalability in mind, PrivProxy supports rule-based, explainable privacy protection. While full integration with the hospital's tracking infrastructure is pending, initial testing demonstrates the system's potential to enhance privacy in healthcare asset tracking.

Acknowledgements

I would like to express my gratitude to Prof. Dr. Daniela Niklas, whose insightful input and feedback greatly improved the quality of this thesis. I am also grateful to my supervisor, Simon Steuer, for the joint development of the exiting topic and his guidance throughout the thesis.

Special thanks go to the Explanym project team for their active participation in the workshop and for the valuable discussions during the project meetings, which greatly supported and enriched this work. I would also like to thank Kim Wuyts for providing major parts of the training material, which was instrumental in the privacy threat modeling phase.

I am incredibly grateful to my friends and partner for their support, encouragement, and patience –especially in moments when motivation was low. Their feedback, emotional support, and belief in me made a real difference. Finally, a heartfelt thank you to my dog, Fritzi, whose daily joy, playfulness, and presence motivated me to start each day with energy and a smile.

Contents

Glossary xv

Acronyms xviii

CHAPTER 1 **Introduction** 1

1.1 Problem Statement 2
1.2 Structure of Work 2
1.3 Methodology 3
1.4 Related Work 3

CHAPTER 2 **Theoretical Background** 7

2.1 Explanym Research Project 7
2.2 Anonymization 8
2.3 Anonymization in Mobility 16
2.4 Asset Tracking in Hospitals 19
2.5 Context-Aware 20

CHAPTER 3 **Asset Tracking System and Use Case: VAC Pumps** 23

3.1 System Architecture and Stakeholders 23
3.2 Components and Tracking Technology 25
3.3 Hardware 27
3.4 Use Case VAC pumps 28

CHAPTER 4 **Privacy Threat Modeling using LINDDUN GO** 33

4.1 Threat Modeling 33
4.2 Objective and Scope 34
4.3 LINDDUN Overview 34
4.4 System Representation 38
4.5 Execution and Results 40
4.6 Security Considerations 43

CHAPTER 5 **Requirements** 45

5.1 Constraints 45
5.2 Functional Requirement 45
5.3 Non-Functional Requirements 48

CHAPTER 6 **Implementation of Priv Proxy** 51

6.1 Implementation Approach 51
6.2 Design and Architecture of Priv Proxy 52
6.3 Project Structure and Technologies 54
6.4 Rule Logic and State Management 57
6.5 Web Interface and User Interaction 63
6.6 Applying the Implementation: VAC Pump Tracking Scenario 66
6.7 Challenges During Implementation 68

CHAPTER 7 **Evaluation 71**

7.1 Evaluation Setup 71
7.2 Results 73
7.3 Addressing Requirements and Threats 77
7.4 Discussion 79

CHAPTER 8 **Conclusion 81**

8.1 Limitations 81
8.2 Future Work 82

References 85

Disclosure 89

List of Figures

2.1 Process of federated learning. Adapted from Gecer and Garbinato [GG24]. 18

3.1 Overview of the Asset Tracking System. 23
3.2 Stakeholder Map for VAC Pump Tracking System. Adapted from Robertson and Robertson [RR12]. 24
3.3 Web Interface of HPM® 25
3.4 Web interface of Track. 26
3.5 iBeacon Payload format for Safectory Beacons. Adapted from [KS14]. 28
3.6 Tracking Hardware. 28
3.7 VAC Pumps. 29
3.8 UML State Diagram. Source: Müller and Halbritter [MH24]. 30
3.9 Test Setup Floor 9 for Field Test 1. 31

4.1 Threat tree for the identifying threat type. Source: DistriNet [Dis24]. 36
4.2 Comparison of LINDDUN GO and Elevation of Privilege with representative cards. 38
4.3 The DFD used for the LINDDUN Go workshop. Adapted from Shostack [Sho14]. 39
4.4 Game board for LINDDUN GO. 40
4.5 Ubertooth One Bluetooth sniffer. 44

5.1 Asset tracking system with anonymization for incoming data. 46
5.2 Asset tracking system with anonymization for outgoing data. 47

6.1 Event Processing Network for PrivProxy. 52
6.2 Directory structure of PrivProxy. 55
6.3 Landing Page (Tag Overview Page). 64
6.4 Tag overview page after entering the tag ID. 64
6.7 Rule administration page. 64
6.5 Admin section for editing tag attributes. 65
6.6 Access to the rules and state management page is restricted to Admin users only. 65
6.8 State administration page. 65
6.9 State Diagram for VAC pumps use case. 67

6.10 Mapping of Müller's states [MH24] (left) to the corresponding Priv Proxy states (right). 69

7.1 Message volume comparison between Track and PrivProxy. 72
7.2 VAC Usecase with PPMs. 73
7.3 VAC Usecase without PPMs. 74
7.4 Bed Usecase with PPMs. 74
7.5 Bed Usecase without PPMs. 75

List of Tables

2.1 Original Table of Patients. 10
2.2 3-Anonymous Version of Table 2.1. 10
2.3 3-Diverse table. 12

4.1 Mapping table for the element-based LINDDUN approach. Source: Sion et al. [Sio+18]. 36
4.2 Mapping table for the interaction-based LINDDUN approach. The highlighted (*colored and emphasized*) elements are where the privacy threat is associated. Source: Sion et al. [Sio+18] 37

Listings

6.1 Pseudocode for Allow Filter Rule. 59
6.2 Pseudocode for Deny Filter Rule. 59
6.3 Pseudocode for the filter_tag function. 59
6.4 Pseudocode for the Track Filter Rule. 60
6.5 Pseudocode for Heartbeat Rule. 61
6.6 Pseudocode for Track By State Allow Rule. 62
6.7 Pseudocode for Track By State Deny Rule. 62
6.8 Pseudocode for Geofence Threshold Rule. 62

Glossary

DDoS Distributed Denial of Service; an attack in which one or multiple systems overwhelm a server, service, or network with traffic to disrupt or shut down access for legitimate users. 43

geofence A virtual boundary defined by geographic coordinates, often used in location-based services. Geofences can trigger actions or alerts when a device enters or exits the designated area. 22, 26, 52, 53, 57, 58, 59, 62, 66, 68, 71, 72, 73, 74, 77

LINDDUN A privacy-focused threat modelling framework, where LINDDUN is a mnemonic for the privacy threat categories: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non-compliance. iii, v, vii, ix, xi, 2, 3, 15, 33, 34, 35, 36, 37, 38, 39, 40, 42, 44, 45, 79, 80, 81, 82

STRIDE A security-focused threat modelling framework, where STRIDE is a mnemonic for the security threat categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. 34, 35, 36, 38, 44, 82

trilateration A method of determining the position of a point based on the known positions of three or more reference points and measuring the distance to those points. 26

Acronyms

AID Anonymous ID. 5

API Application Programming Interface. 54, 56, 68

BLE Bluetooth Low Energy. iii, 1, 2, 3, 5, 19, 21, 26, 27, 29, 43, 44

DBA Database Administrator. 39, 40

DFD Data Flow Diagram. ix, 33, 34, 36, 37, 38, 39

DoS Denial of Service. 44

EPA Event Processing Agent. 52, 77

EPN Event Processing Network. ix, 52

GAN Generative Adversarial Network. 5

GDPR General Data Protection Regulation. 1, 14, 36, 37

HIS Hospital Information System. 26, 68

HTK Hygiene Technologie Kompetenzzentrum. 7, 40

IPS Indoor Positioning System. 19

LBS Location-Based Service. 15, 16, 17

LPPM Location Privacy-Preserving Mechanism. iii, 16, 17, 18, 51, 75, 79

MIA Membership Inference Attack. 15

MOBI Mobile Systems. 7, 40

PET Privacy-Enhancing Technology. 34, 37

POI Point of Interest. 16

PoPETs Proceedings on Privacy Enhancing Technologies. 3

PPM Privacy-Preserving Mechanism. x, 4, 5, 8, 15, 16, 18, 19, 24, 34, 41, 43, 46, 47, 51, 52, 57, 58, 63, 66, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83

PSI Privacy and Security in Information Systems Group. 7, 40, 43

PSY Personality Psychology and Psychological Assessment. 7

ROF Reduction in Observable Features. 76, 79, 83

RSSI Received Signal Strength Indicator. 21

RTLS Real-Time Locating System. 19

SDG Synthetic Data Generation. 14, 15

SIG Special Interest Group. 26, 27

SQL Structured Query Language. 55

UML Unified Modeling Language. 39

VAC Vacuum-Assisted Closure. iii, vii, viii, ix, x, 1, 2, 3, 7, 8, 20, 23, 24, 26, 28, 29, 30, 31, 42, 51, 63, 66, 67, 71, 72, 73, 74, 75, 76, 78, 81, 83

1 Introduction

As digital technologies increasingly impact people's daily lives, the collection and processing of mobility and healthcare data raises profound privacy concerns. At the 38th Chaos Communication Congress (38C3)[1] in 2024, the two most-attended presentations each highlighted significant privacy concerns in both areas.

The first, *Wir wissen wo dein Auto steht Volksdaten von Volkswagen* [2], revealed how Volkswagen collects and accesses vehicle, movement, and diagnostic data. In particular, it was found that movement data from 800,000 electric vehicles, along with owner contact information, was found to be exposed online. The presentation illustrated how mobility data can reveal highly sensitive information, including where people live, where they work, and even their sexual or religious affiliations. Previous research has shown that drivers can be re-identified using seemingly innocuous telemetry –such as brake pedal usage– if reference data is available [Ene+16]. The presenters also reflected on the challenge of presenting such data responsibly –visualizations included– in order to raise awareness without compromising individual privacy. The second most popular event concerned the healthcare sector. In *"Konnte bisher noch nie gehackt werden": Die elektronische Patientenakte kommt – jetzt für alle!*[3] presented vulnerabilities in the electronic patient file system that demonstrated how attackers could gain access to sensitive patient data, weeks before its release.

Under Article 9 of the General Data Protection Regulation (GDPR), health data, as well as information disclosing religious beliefs, political opinions, or sexual orientation, is classified as special category data requiring enhanced protection. As mentioned before, mobility data –although not explicitly listed– can similarly reveal such sensitive information and therefore merits equivalent safeguards under data protection regulations.

This thesis brings the two worlds together by examining mobility data in a healthcare context. It focuses on an asset tracking system based on Bluetooth Low Energy (BLE) that is used to monitor the movement of Vacuum-Assisted Closure (VAC) pumps - medical devices that are connected to patients. The movement patterns of the tracked VAC pumps can inadvertently reveal sensitive information: for example, a pump located in the oncology department could imply a cancer diagnosis, while frequent device interactions could be used to infer employee activities.

1: The Chaos Communication Congress is an annual multi-day convention organized by the Chaos Computer Club (CCC).

2: English translation by the author: We Know Where Your Car Is People's Data from Volkswagen. Link to the source: https://media.ccc.de/v/38c3-wir-wissen-wo-dein-auto-steht-volksdaten-von-volkswagen (last accessed 28.04.2025).

3: English translation by the author: "Could Never Be Hacked Before: The Electronic Patient File Is Coming – Now for Everyone!". Link to the source: https://media.ccc.de/v/38c3-konnte-bisher-noch-nie-gehackt-werden-die-elektronische-patientenakte-kommt-jetzt-fr-alle (last accessed 28.04.2025).

1.1 Problem Statement

The privacy implications of BLE asset tracking systems are becoming increasingly relevant as Bluetooth technology becomes more widespread. In 2024, global annual shipments of Bluetooth devices reached 4.9 billion, with continued growth projected [SIG25]. BLE-based asset tracking is already being adopted across various industries to enhance operational efficiency. However, when introduced in sensitive environments such as healthcare, additional privacy concerns arise, due to the sensitivity of the handled data [Baz+21].

This thesis is part of the Explanym research project, which develops explainable anonymization techniques for mobility data in both urban and healthcare settings. In the healthcare domain, the focus lies on the development and prototypical implementation of an anonymization component for tracking VAC pumps. The goal is to enable effective use of mobility data while preserving privacy.

These considerations lead to the following research question guiding this thesis:

> How can privacy be preserved for hospital mobility data from VAC pumps, taking into account contextual factors?

In addition to answering this question, the thesis addresses key privacy challenges in mobility data research. Mokbel et al. [Mok+24] have emphasized privacy as a central concern, highlighting open issues such as the lack of well-defined threat models, the need to balance privacy and utility, and the importance of explainability and public understanding of privacy risks. Each of these challenges is addressed in this thesis and explored in the context of hospital asset tracking.

1.2 Structure of Work

The thesis is organized as follows: Section 1.3 outlines the methodology used. Section 1.4 gives an overview of related work. Chapter 2 covers the theoretical background, including concepts of anonymization, anonymization of mobility data, asset tracking systems, and context. Chapter 3 describes the asset tracking system and the VAC pump use case that underlie the anonymization application developed. This chapter also sets the stage for the privacy threat modeling workshop using LINDDUN GO discussed in chapter 4. The results of the workshop contribute to the system requirements outlined in chapter 5. Chapter 6 details the implementation of PrivProxy, the anonymization software developed, while chapter 7 presents the evaluation of the results. Finally, chapter 8 concludes the thesis and discusses possible avenues for future work.

1.3 Methodology

To develop a privacy-preserving application for indoor asset tracking using BLE, a combination of literature research, system analysis, privacy threat modeling, and iterative implementation was conducted.

First, a literature search was conducted to identify foundational work on privacy in indoor localization using BLE. The databases IEEE Xplore[4], ACM Digital Library[5], Springer Link[6], and ScienceDirect[7]. The following search parameters were used:

- *Title*: Privacy OR Anonymous*.
- *All fields*: Indoor AND (BLE OR Bluetooth Low Energy).
- *Publication years*: 2015–2025.

4: https://ieeexplore.ieee.org/ (last accessed 13.03.2025).

5: https://dl.acm.org/ (last accessed 13.03.2025).

6: https://link.springer.com/ (last accessed 13.03.2025).

7: https://www.sciencedirect.com/ (last accessed 13.03.2025).

In addition, the Proceedings on Privacy Enhancing Technologies (PoPETs) were reviewed. Due to the relevance of the journal, a full manual review of the tables of contents from 2015-2025 was performed instead of a keyword-based search. Papers addressing privacy in location-based systems were selected, including work on outdoor localization whose strategies may be partially applicable to indoor environments. This exploratory approach uncovered relevant research that might have been missed by a broader catalog search.

Information about the asset tracking system and the VAC pump use case was gathered through bi-weekly briefings, related work by fellow students (see section 1.4), and documentation available on the project's internal GitLab instance.

This information provided the necessary background for conducting the LINDDUN GO workshop. Literature for the LINDDUN approach was primarily obtained from the official website[8] and supplemented with scientific papers from IEEE Xplore[1], with particular emphasis on publications authored by the original developers of the approach.

8: https://linddun.org/ (last accessed 18.03.2025).

The LINDDUN workshop resulted in a list of identified privacy threats. Based on these threats, as well as the requirements from project documentation and Explanym meetings, system requirements were developed (see chapter 5).

The Implementation followed an iterative approach, with regular progress discussions during the Explanym bi-weekly meetings. Feedback from these sessions was used to continuously adapt and refine the system.

1.4 Related Work

Several papers have been published in the context of the Explanym project. Ackermann et al. [Ack+23] discuss key challenges related to the collection, management, and anonymization of mobility data in the context of smart cities. Their work outlines various real-world data sources, highlighting differences in structure, granularity, and potential privacy

risks. In addition, they offer a concise overview of existing anonymization techniques for mobility data, drawing primarily on the survey by Jin et al. [Jin+23]. Finally, they reflect on the results and limitations of previous anonymization projects, highlighting recurring obstacles and lessons learned for future initiatives.

Waclawek et al. [Wac+24] examine how different types of explanations influence the understanding and acceptance of crowd-counting technologies. They distinguish between functional explanations, which emphasize the purpose of the project, and mechanistic explanations, which describe the inner workings of the system. Their work combines exploratory interviews, a field study, and an online survey. Findings indicate that functional explanations promote project understanding more effectively than mechanistic explanations. In addition, the study reveals a positive correlation between trust in the project and its acceptance.

Dietz et al. [Die+24] conduct a literature review with a user-centric focus to explore the practical barriers to adoption of AI and ML-based intrusion detection solutions. They examine challenges related to data adequacy, reproducibility, explainability, practicality, usability, and privacy.

Müller and Halbritter [MH24] wrote a master's thesis entitled *Identifying Data-Induced Risks in Hospital Asset Tracking* as part of the Explanym project. While Müller conducted the majority of the research, she collaborated with Halbritter to analyze the data-related requirements in asset tracking. This collaboration included expert interviews, activity- and state diagram modeling, and the creation of a data dictionary. Some of the results of this collaboration are included in this thesis. Based on the data dictionary and supported by literary research, Müller identified risks that are often overlooked in IT risk assessments and developed photo stories to effectively communicate these risks to stakeholders.

The following related work was conducted independently of the Explanym project but offers valuable complementary perspectives. Cunha et al. [Cun+24] developed the privacy toolkit Privkit[9], which addresses the challenges posed by the lack of standardized privacy definitions, configuration processes, and evaluation methods for PPMs. Privkit streamlines the process of configuring and evaluating PPM by providing a modular, extensible system. It currently supports location and face data types and includes several pre-integrated PPMs. In addition, the toolkit allows PPMs to be evaluated against a range of attack scenarios and metrics, enabling comprehensive privacy and utility assessments.

9: https://github.com/privkit/privkit (last accessed 27.04.2025)

Primault et al. [Pri+19] provide a comprehensive overview of location privacy, distinguishing between online and offline use cases. They categorize PPMs into six groups based on the core algorithms employed. They also review the evaluation metrics for these mechanisms, focusing on three dimensions: privacy, utility, and performance, with an emphasis on the privacy-utility trade-off.

Zuo et al. [Zuo+21] conducted a literature mapping study on data anonymization in digital health. Their findings suggest that while anonymization

of health data is theoretically feasible, further research on practical implementations is needed to achieve a better balance between privacy and utility.

Zidek et al. [ZTH18] present the Bellrock proximity framework, which uses BLE to repurpose static personal devices, such as laptops and smartphones, as proximity beacons. The framework allows devices to act as beacons without revealing the owner's location, using access control mechanisms to protect privacy. The Bellrock Proximity Framework can extend existing beacon infrastructures or create new coverage where none exists. The implementation relies on pseudonymization, where devices are assigned an Anonymous ID (AID) that changes whenever the device becomes stationary. These AIDs can be unmasked on demand by a cloud service with different access levels to provide additional usage potential.

Müller's master's thesis [Mül20] builds on the Bellrock proximity framework by implementing the Crowd Beacons system, which allows mobile devices to act as dynamic BLE beacons. The approach extends existing static beacon infrastructures, even in smaller locations, by introducing anonymous dynamic beacons based on frequently changing AIDs, preserving user privacy while improving indoor localization coverage.

The literature on PPMs for location data is extensive, ranging from custom anonymization algorithms for indoor localization [Kon+15] to protocol-based approaches [Jär+19]. Additionally, machine learning techniques have been utilized, such as the application of Generative Adversarial Network (GAN) for privacy-preserving taxi demand prediction [Oze+23]. In the context of indoor location-based services, reinforcement learning has been used to balance privacy and service quality, allowing users to define privacy profiles that determine the trade-off between privacy risk and service reward [FKS16]. While explicable anonymization remains an underexplored area, user-centric models have been proposed, such as the "feeling-based" location privacy approach by Xu et al. [XC09], in which users specify a comfortable region instead of a precise location. Other related approaches and theoretical foundations are discussed in chapter 2.

Although this work focuses primarily on anonymizing data generated by asset tracking systems, privacy and security considerations in such systems remain essential for real-world implementations. Sartayeva et al. [SC23] conducted a systematic review of the security and privacy of indoor positioning methods. They categorize positioning techniques into non-collaborative, collaborative, and hybrid approaches. For each category, they analyze the associated security and privacy challenges, providing a comprehensive overview of the topic. A more detailed discussion of privacy and security considerations relevant to the asset tracking system used in this thesis is provided in chapter 4.

2 Theoretical Background

This chapter provides the theoretical foundation for this thesis. It begins with an introduction to the Explanym research project. It then provides an overview of anonymization, including key strategies and common attack types for both record-level and aggregated data. It subsequently analyzes the distinct challenges and characteristics associated with anonymization within the mobile domain, while introducing relevant anonymization categories relevant to this context. The focus then shifts to asset tracking systems, with an emphasis on healthcare and indoor environments, as the system under study operates in this area. Finally, the concepts of context and context awareness are explored in detail.

2.1 Explanym Research Project

This work is part of the Explanym research project, which investigates practical anonymization techniques for mobility data with a focus on explainability and acceptability. The project is carried out by interdisciplinary teams combining academic research and industry expertise.

Explanym focuses on two major use cases: one in urban mobility, such as intermodal transportation, and another in clinical environments, such as medical device tracking. This thesis addresses the latter use case. In this work and unless otherwise stated, the Explanym project refers explicitly to the clinical use case and its designated/associated project group.

The clinical project group includes the university chairs Mobile Systems (MOBI), Privacy and Security in Information Systems Group (PSI), and Personality Psychology and Psychological Assessment (PSY), as well as industry partners Safectory GmbH[1], which provides the tracking infrastructure and equipment, and Clinaris GmbH[2], which develops the hospital asset tracking application. Additional clinical coordination is provided by Hygiene Technologie Kompetenzzentrum (HTK) GmbH[3]. More information about the system components of the industry partners can be found in chapter 3.

The primary focus of this thesis is the anonymization of data from VAC pump tracking at the Bamberg Clinic[4] (see section 3.4), one of two field tests conducted within the broader project. Although the second field test, involving person tracking in a geriatric rehabilitation facility, is not

1: https://safectory.com/ (last accessed 03.05.2025).

2: https://www.clinaris.com/ (last accessed 03.05.2025).

3: https://hygiene-tk.de/ (last accessed 03.05.2025).

4: https://www.sozialstiftung-bamberg.de/ (last accessed 03.05.2025).

covered in this thesis, the developed system has been designed with the adaptability to support both scenarios.

The project consists of two field tests. The first focuses on anonymizing data from VAC pump tracking at the Bamberg Clinic (see section 3.4), which forms the primary context for this thesis. While the second field test, concerning person tracking in a geriatric rehabilitation center, is outside the scope of this thesis, the developed system is designed with adaptability for both use cases in mind.

The project's progress was coordinated through bi-weekly meetings where team members discussed upcoming challenges, shared updates, and defined next steps. Notes from these meetings, along with detailed documentation of the project's progress, are stored and managed in an internal GitLab instance. In addition to these regular meetings, on-site visits to the hospital were organized to generate real-world test data and validate the functionality of the system components. In addition, the findings and results from these meetings and field tests were shared among the project teams in a dedicated workshop.

2.2 Anonymization

In systems that process personal or sensitive information, anonymization can play a key role in protecting the privacy of individuals. The GDPR, in recital 26, defines anonymized data as *"[...] information that does not relate to an identified or identifiable natural person [...]"* [EC16]. Anonymization therefore involves transforming data to ensure that it cannot be linked to any individual. Anonymization is an example of a Privacy-Preserving Mechanism (PPM), a broader class of techniques designed to preserve privacy.

The approach to data anonymization depends on several factors. One key factor is the structure of the data itself. A distinction can be made between structured, semi-structured, and unstructured data [CMV21; Gad+24]. Structured data is typically tabular, with each row representing an individual. Semi-structured data often consists of events or activities, such as GPS location traces, associated with timestamps and descriptive metadata. Unstructured data, on the other hand, includes formats such as free text, images, or video, and lacks a defined schema. This work focuses on anonymizing structured and semi-structured data, as the asset tracking system does not include unstructured data.

In addition, the trust and adversary models influence the choice of appropriate anonymization techniques. These models define the boundaries of the system and help ensure that the techniques selected are consistent with the privacy goals and system constraints.

The trust model specifies which components of the system are considered trustworthy. The most common approach is the centralized trust model, where a single trusted entity (the curator) has full access to the original data, performs the anonymization, and publishes the results. The goal is

to preserve statistical utility while preventing privacy attacks [Pri+19]. Alternative models include distributed approaches, where anonymization is shared among multiple components, and local models, where data is anonymized locally, at the point of creation.

The adversary model defines the assumed capabilities, knowledge, and goals of potential attackers. In the context of anonymization, adversaries are typically assumed to be honest but curious, such as data scientists who follow protocol but try to extract as much information as possible from the data [Gad+24; Pri+19]. They are assumed to possess and apply all relevant external knowledge, with the goal of inferring sensitive information about targets through privacy attacks.

Anonymization can be performed in different modes –offline, online, or hybrid (e. g., batch processing) [CMV21; Pri+19]. These modes differ in when and how anonymization is applied relative to data collection and processing.

Offline anonymization is applied to static datasets after data collection is complete. It is often used for releasing public data or for research, where the entire dataset is anonymized at once without incorporating new data.

Online anonymization processes data in real-time, using only current and historical information. Since future data is not accessible, methods must account for the incomplete dataset. Anonymization techniques used in online mode can be applied in offline mode, but not vice versa, since offline anonymization requires access to the entire dataset.

Hybrid anonymization processes data in batches, making it suitable when higher latency is acceptable. Like online anonymization, hybrid anonymization works without future data and therefore must account for incomplete data sets.

According to Gadotti et al. [Gad+24], anonymization results can be divided into two types. The first is record-level data, where individual records are anonymized but still retained. The second is aggregate data, where individual records are not retained. In aggregate data, information is retained in formats such as statistical summaries or machine learning models. Each category encompasses different types of attacks. Record-level anonymization techniques and their associated attack types are examined first, followed by aggregate data anonymization techniques and their associated privacy attack types.

One common attack on record-level data is the linkage attack, which seeks to re-identify individuals by connecting information across datasets [Gad+24]. This can range from simple exact matching attacks that use identical information such as zip code, gender, and date of birth, to more sophisticated so-called robust matching attacks that can handle noise or disturbances in the data. The most advanced form is the profiling attack, where datasets from non-overlapping time periods are linked, even when significant changes in the data have occurred.

To address linkage attacks, Sweeney [Swe02] introduced the concept of k-anonymity in response to an attack she conducted, demonstrating how individuals could be re-identified by linking publicly available datasets through quasi-identifiers such as zip code, gender, and date of birth. Quasi-identifiers are attributes that, when combined with external records, can uniquely identify individuals [Mac+06]. Two other attribute types are identifiers, like name, social security number, and address which allow direct identification and sensitive attributes which are to be protected.

k-anonymity prevents linkage attacks, by ensuring that each record in a dataset is indistinguishable from at least $k-1$ other records based on a set of quasi-identifiers (e. g., age, zip code, and gender). Records that share the same values for all quasi-identifiers form what is called an equivalence class. Within each equivalence class, individuals cannot be distinguished from each other based on the quasi-identifiers alone.

To achieve k-anonymity, techniques such as generalization and aggregation can be used. Generalization replaces specific values with broader categories, while aggregation groups records and replaces values with, for example, the average of the group.

Table 2.1 shows a record containing the quasi-identifiers –age and zip code– along with the sensitive attribute disease. The data allows a linkage attack: If zip code 13068 corresponds to a small village and an external registry lists the ages of its residents, a linkage attack could reveal the identity of patient ID 2. If that person is the only 34-year-old in the village, the person could be re-identified and the sensitive information –a cancer diagnosis– would be exposed.

In contrast, table 2.2 shows a 3-anonymity data set divided into three equivalence classes. In each class, each combination of quasi-identifiers appears in at least three records. Under the same conditions, it would no longer be possible to determine whether the 34-year-old individual was included in the dataset. If their inclusion were known, it would remain unclear whether this person had the flu or cancer.

	Age	Zip Code	Disease
1	29	13054	Flu
2	34	13068	Cancer
3	37	13053	Flu
4	45	14850	Cancer
5	50	14851	Cancer
6	52	14850	Cancer
7	61	15052	Flu
8	62	15050	Diabetes
9	64	15068	Flu

Table 2.1: Original Table of Patients.

Age	Zip Code	Disease
20–39	130**	Flu
20–39	130**	Cancer
20–39	130**	Flu
40–59	148**	Cancer
40–59	148**	Cancer
40–59	148**	Cancer
60–65	150**	Flu
60–65	150**	Diabetes
60–65	150**	Flu

Table 2.2: 3-Anonymous Version of Table 2.1.

A few years after the introduction of k-anonymity, two privacy attacks were identified that can compromise datasets even if they satisfy the k-anonymity property: the homogeneity attack and the background knowledge attack [Mac+06]. A homogeneity attack exploits the lack of diversity in sensitive attributes within an anonymized group. For example, in the previously shown 3-anonymized dataset (table 2.2), all individuals aged 40-59 with zip code 148*** have the same diagnosis. If someone is known to be in this group, their diagnosis can be inferred due to the lack of diversity of the sensitive attribute.

Background knowledge attacks rely on external information. Consider a scenario where the clinic publishes the 3-anonymized dataset shown in table 2.2, and Mallory knows that her 62-year-old neighbor Alice is included. Observing that Alice hasn't had the flu and has stopped buying sweet pastries, Mallory concludes that Alice likely has diabetes, even though several diagnoses appear in the equivalence class.

To address the shortcomings of k-anonymity, Machanavajjhala et al. proposed an additional privacy measure called ℓ-diversity, which can be used alongside k-anonymity [Mac+06]. A dataset satisfies ℓ-diversity if every equivalence class contains at least ℓ well-represented values for the sensitive attribute. The term *well-represented* is not defined in a single, fixed way. Instead, Machanavajjhala et al. [Mac+06] suggest several approaches. One of them is ℓ-entropy, which measures the entropy of sensitive values within each equivalence class E:

$$\text{Entropy}(E) = -\sum_{s} p(E,s) \log p(E,s)$$

where S is the domain of the sensitive attribute, and $p(E,s)$ represents the fraction of records in E that have the sensitive value s. For a dataset to satisfy entropy ℓ-diversity, each equivalence class must satisfy $\text{Entropy}(E) \geq \log(\ell)$. However, since entropy ℓ-diversity may be too restrictive for some cases, a weaker variant is recursive (c, ℓ)-diversity, which limits how dominant the most frequent value can be within an equivalence class.

One year after the introduction of ℓ-diversity, Li et al. [LLV07] demonstrated two effective attacks against datasets anonymized using k-anonymity with ℓ-diversity: the skewness attack and the similarity attack.

A skewness attack exploits cases where a sensitive attribute –such as a virus diagnosis– is unevenly distributed in the population. If only 1% of people have the virus, an equivalence class with 5 virus cases and 5 non-virus cases still satisfies entropy 2-diversity.

However, knowing that someone is in this group gives a 50% chance that they have the virus –much higher than the population average. This illustrates that ℓ-diversity does not protect against inference when the overall distribution is highly skewed.

In contrast, a data set is vulnerable to a similarity attack if the sensitive values are technically different but semantically similar, allowing inference about individuals. In table 2.3, the sensitive attributes –disease and

income– meet the formal criteria for entropy 3-diversity. However, if a person is known to have a very low income, it can be inferred that the person is likely to have a stomach problem. Conversely, if someone has a very high income (e. g., the 80k outlier), it can be inferred that the person has been diagnosed with cancer.

Age	Zip Code	Disease	Income (€)
2*	130**	Gastritis	2k
2*	130**	Gastric Ulcer	3k
2*	130**	Stomach Cancer	4k
≥40	148**	Arrhythmia	8k
≥40	148**	Diabetes	10k
≥40	148**	Lung Cancer	80k
3*	150**	Pneumonia	7k
3*	150**	Bronchitis	8k
3*	150**	Flu	12k

Table 2.3: 3-Diverse table.

Although the table satisfies 3-anonymity and entropy 3-diversity, privacy is still compromised due to the semantic similarity and background knowledge that links quasi-identifiers to specific sensitive values.

To defend against skewness and similarity attacks, Li et al. introduced the privacy measure t-closeness [LLV07], which is defined as follows: An equivalence class satisfies t-closeness if the distribution of a sensitive attribute within the equivalence class differs from the overall distribution by no more than a threshold t, measured by a distance metric. A data set satisfies t-closeness if all equivalence classes satisfy this condition. Li et al. propose different distance measures for categorical attributes (e. g., disease) and numerical attributes (e. g., income).

In summary, the record-level data anonymization techniques presented provide incremental protection against re-identification. k-anonymity ensures that all quasi-identifiers in each record are indistinguishable from at least $k - 1$ other records. ℓ-diversity enhances this by ensuring that sensitive attributes have well-represented values in each group. Finally, t-closeness ensures that the distribution of sensitive attributes within a group is similar to the distribution of the entire dataset. Nevertheless, a minimality attack can still exploit weaknesses in how these properties are achieved.

The minimality attack targets anonymization algorithms based on the minimality principle, which applies the least amount of generalization or suppression necessary to satisfy privacy properties such as k-anonymity [Cor+10]. By assuming that the algorithm seeks to maximize data utility while making minimal transformations, an attacker can reverse-engineer the original values by analyzing the applied generalizations, exploiting both the potential transparency of the anonymization process and its efficiency-driven design. Cohen demonstrated this attack on a real-world dataset with a conservative k-anonymity configuration where each attribute serves as a quasi-identifier [Coh22].

The record-level data anonymization approaches have the advantage that their privacy guarantees, such as k-anonymity, can be directly evaluated on a given dataset. However, datasets often exhibit high dimensionality, characterized by a large number of attributes, which increases the probability of unique record combinations [Gad+24]. Consequently, significant data utility is often lost during anonymization. Furthermore, identifying quasi-identifiers is challenging, as even seemingly harmless attributes can enable re-identification when combined with other external datasets. Thus, anonymization remains highly context-dependent and is only effective as long as no external information is available –an assumption that doesn't hold in practice. Differential privacy addresses these limitations by providing protection that does not rely on assumptions about external knowledge.

The principle of differential privacy can be traced back to the earlier concept of Randomized Response, introduced in 1965 by Warner [War65]. It is a technique for gathering truthful responses to sensitive questions in surveys while preserving respondent privacy.

In the Randomized Response method, a respondent is asked a sensitive question, such as *Have you ever used illegal drugs?* Instead of answering directly, the respondent follows a randomization process. For example, the respondent may flip a coin. If the coin lands heads, the respondent answers truthfully. If the coin lands tails, the respondent answers a non-sensitive question with a known distribution. This randomization ensures that the researcher cannot definitively link any individual's response to the sensitive question, thereby protecting privacy.

This randomization introduces uncertainty, allowing individuals to respond truthfully without being directly exposed. While individual answers are obfuscated, statistical analysis of aggregated responses can still reveal accurate insights [DR14].

Differential privacy, introduced by Dwork et al. in 2006 [Dwo06], offers formal, quantifiable privacy guarantees by focusing on the behavior of the data access mechanism rather than modifying the dataset directly [Gad+24].

A randomized mechanism M satisfies ε-differential privacy if, for all neighboring datasets D and D' that differ in at most one element, and for all measurable subsets $S \subseteq \text{Range}(M)$, the following condition holds [DR14]:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S]$$

Here, $\varepsilon \geq 0$ is a non-negative parameter, often referred to as the privacy budget. ε quantifies the privacy loss: smaller values imply stronger privacy. The set S represents any subset of possible outputs of M [DR14].

Differential privacy ensures that the result of a query remains nearly identical whether or not an individual's data is included in the dataset, with the privacy guarantee being provable. This can be achieved by adding noise

to the data. One of the key advantages of differential privacy is its computability: the total privacy loss over multiple queries can be computed by summing the individual losses. This is in contrast to k-anonymity, where repeated data releases can gradually reveal more information [Gad+24]. With differential privacy, the cumulative privacy loss over each release can be monitored and controlled. In addition, the parameter ε enables precise adjustment and hence fine-grained control of the trade-off between privacy and utility

While differential privacy provides strong, provable privacy guarantees and has been widely studied in theory, its adoption in practice remains limited [Gad+24; Pri+19]. Similar to the randomized response approach, it requires large data sets to provide meaningful insights, which can be problematic, especially when dealing with minority groups or outliers [Gad+24; WBH19].

Managing the privacy budget is another practical concern. It is not always clear whether the budget should be allocated globally across all users or individually per user. When users collaborate, privacy guarantees can weaken and budget management becomes more complex [Pri+19].

Choosing the right value for ε is also critical. A low ε improves privacy but may render the data nearly useless, while a high ε improves utility at the expense of privacy. Finding this balance is often a non-trivial task.

Gadotti et al. argue that while differential privacy provides a solid foundation for privacy discussions, its strict application is not always necessary and may need to be adapted to enable broader practical use [Gad+24].

The position is consistent with the GDPR's interpretation of anonymization. Recital 26 states:

> *[...] To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly [...]* [EC16].

This aligns with the GDPR's focus on realistic, rather than hypothetical, re-identification risks.

A prominent practical application of differential privacy is Google's Gboard application, which combines differential privacy with federated learning to improve next-word prediction [ZY24]. Instead of sharing raw input data, only aggregated model updates are shared (for details on federated learning, see section 2.3). While the example of Google's Gboard illustrates the use of aggregated data in the context of differential privacy, the concept of aggregate data and its role in anonymization is further explained in the following paragraphs.

Aggregate data refers to information summarized across records, such as summary statistics, machine learning models, data query systems, or Synthetic Data Generation (SDG) [Gad+24]. While summary statistics are straightforward, the focus here is on the latter approaches. Data query

systems allow analysts to interact with data in either interactive or non-interactive settings. In interactive settings, analysts submit queries to the system that tailors the results to preserve privacy, but accesses the real data. In non-interactive settings, the data owner (curator) releases only pre-computed, aggregated results. An example of a non-interactive approach is providing access to a machine learning model: although the model is trained on real data, it no longer has access to the underlying data and only exposes aggregated information [Gad+24]. While machine learning models aggregate data, they do not inherently guarantee privacy. To address this, differential privacy is built into the training process. For simpler models such as logistic regression, noise can be added directly to the loss function [CM08], while for complex models such as deep neural networks, noise is typically injected into the gradients during training [Yu+21]. SDG creates artificial data that retains the properties of the original data, allowing standard analytical methods to be applied to the artificial data without modification [Gad+24]. However, the generated data is still based on the original data. Similiar to machine learning approaches, differential privacy techniques are often applied during the training process, as demonstrated in approaches such as differentially private Generative Adversarial Networks (GANs) by Xie et al. [Xie+18].

Although aggregate anonymization is generally more robust than record-level anonymization [Gad+24], it remains vulnerable to several types of attacks. In Membership Inference Attack (MIA), the adversary attempts to determine whether a particular individual was part of the dataset, which may in itself constitute a privacy violation, such as revealing affiliation with a political group [Gad+24]. In attribute inference attacks, an adversary attempts to discover sensitive attributes associated with individuals in the original dataset, such as medical conditions [Gad+24]. In reconstruction attacks, the adversary attempts to reconstruct parts or even the entire dataset from the exposed aggregate data [Gad+24].

In anonymization, the privacy-utility trade-off refers to balancing privacy with the utility of data. While privacy in context of anonymization is often narrowly interpreted as confidentiality [Gad+24], frameworks such as LINDDUN (see chapter 4) emphasize that privacy encompasses broader dimensions. PPMs aim to preserve privacy in data without making it useless. Utility can be evaluated through metrics such as distribution fidelity or model performance, but its relevance is highly context-dependent. Achieving the right balance between privacy and utility requires appropriate methods to assess both. However, measuring privacy is often an intricate challenge.

There is no universal standard for measuring privacy, as methodologies vary from domain to domain. Wagner and Eckhoff [WE18] reviewed over 80 privacy metrics and categorized them by domain, including communication systems, databases, location-based services (LBS), smart metering, social networks, and genomic data. These metrics include:

- *Uncertainty*: Higher uncertainty for the adversary implies greater privacy.

- *Information Gain*: Measures how much information the adversary obtains; more gain means more privacy loss.
- *Data Similarity*: Focuses on data patterns, such as the frequency of values; includes metrics like k-anonymity, ℓ-diversity, and t-closeness.
- *Indistinguishability*: Assesses whether the adversary can distinguish between different outputs of a privacy mechanism.
- *Adversary's Success Probability*: Indicates how likely the adversary is to succeed.
- *Error*: Measures the distance between the adversary's estimate and the true value.
- *Time*: Evaluates the time it takes for the adversary to succeed.
- *Accuracy/Precision*: Assesses how close and consistent the adversary's estimate is.

In the context of mobility data, anonymization presents unique challenges. The following section explores how these privacy metrics apply to mobility data and the techniques used to address privacy concerns in this domain.

2.3 Anonymization in Mobility

While the previous chapter introduced core anonymization concepts, this section focuses on those specific to mobility data. A key concern is the identification of Point of Interest (POI), a location where individuals spend time. Clustering algorithms can easily detect these Point of Interest (POI)s, and even a single sensitive location, such as a hospital or religious site, can reveal personal information [Pri+19]. In addition, co-location patterns can indicate social relationships [Pri+19]. Beyond spatial data, the temporal dimension also poses risks: users often follow consistent patterns of movement (e. g., daily commutes), which can allow re-identification even when individual locations are obscured.

The term Location-Based Service (LBS) refers to a service that provides information or functionality based on the location of a user or object. Examples include navigation applications, asset tracking systems, or weather services. The PPMs used in the context of LBS are referred to as Location Privacy-Preserving Mechanism (LPPM). LPPMs are a subset of PPMs that focus on protecting location data and must address additional challenges that are often unique to spatial and temporal information.

The architectures for implementing LPPMs can be categorized into trusted third party, non-trusted third party, peer-to-peer, and local architectures [Pri+19]. Each architecture defines a specific trust model that determines which components of the overall system are trusted and where the privacy mechanisms must be applied. Only the local architecture can operate both offline and online; all others require an online connection.

Each architecture starts with a sending device that wants to send data to the LBS while applying privacy protection mechanisms. In a trusted

third party architecture, a fully trusted external entity called a proxy performs the LPPM and has full access to the raw data. In non-trusted third party architectures, the data is protected before it reaches the third party servers, so there is no need to trust them. Peer-to-peer architectures rely on device cooperation to apply privacy protocols before data is sent. In local architectures, all privacy mechanisms are performed directly on the sending device, without the involvement of external entities.

According to Primault et al. [Pri+19], the LPPMs can be grouped into several conceptual categories, each with distinct strategies and trade-offs. One such strategy involves the use of mix zones, which are defined areas where users are not tracked. When a user exits a mix zone, they are assigned a new pseudonym, making it difficult to link their identity across zones. The effectiveness of this method relies on careful placement of these zones and a sufficiently large number of users to ensure anonymity.

Another common approach is generalization, where location accuracy is deliberately reduced–for example, by truncating decimal places in latitude and longitude. This can also include cloaking, where a location is only revealed when at least k users are present, in line with intuitive privacy models such as k-anonymity. Generalization is typically applied in offline scenarios and often follows deterministic logic. However, similar to mix zones, its effectiveness in real-time systems is limited unless a large user base is available.

To address this limitation in online use cases, some systems use dummy-based methods. These involve generating fake users or queries –called dummies– to mask the real user's activity. For example, a LBS might receive multiple concurrent queries, only one of which reflects the user's actual location. This technique can provide artificial k-anonymity, especially in scenarios where data might otherwise be discarded. However, it poses challenges in generating realistic dummy data. Most adversary models require knowledge of the presence of dummies, making their plausibility essential. Repeated queries can reveal inconsistencies in dummy behavior, allowing adversaries to infer real data. In addition, machine learning can compound this risk by detecting subtle patterns that distinguish real from dummy data.

Perturbation-based approaches take a different tack by perturbing the data before it is sent to a LBS. These techniques often rely on differential privacy to introduce noise in a controlled manner, providing strong privacy guarantees against re-identification.

Some methods are protocol-based, embedding privacy protection directly into the system design through cryptographic protocols. These approaches can provide high levels of privacy and often achieve a good balance between privacy and utility, especially when tailored to specific use cases. However, they tend to be computationally expensive and may not be practical in all scenarios.

Finally, rule-based approaches offer the most flexibility, allowing systems to dynamically select and combine different privacy-preserving techniques based on context. While this composability enables tailored

responses to different threats or use cases, it also introduces complexity: poorly designed rule combinations can unintentionally weaken overall privacy guarantees.

Although not explicitly categorized by Primault et al., federated learning has gained significant attention as an emerging privacy-preserving machine learning approach, particularly in the mobility domain [GG24]. The primary appeal of federated learning lies in its ability to address security and privacy concerns associated with machine learning [GG24]. Unlike traditional methods, federated learning eliminates the need to send sensitive data to a central server. Instead, computation occurs locally on devices, and only the resulting model parameters are shared, enhancing privacy.

Federated learning involves three key roles: the data generator, the learner, and the aggregator [GG24]. The data generator (e. g., BLE beacons) creates the data, while the learner performs machine learning tasks that require computational resources. The learners do not communicate with each other. The learner's output, the model parameters, are sent to the aggregator, which merges the parameters and sends updates back to the learners. Multiple aggregators can work together to build an aggregated model. An overview of this process is shown in Figure 2.1.

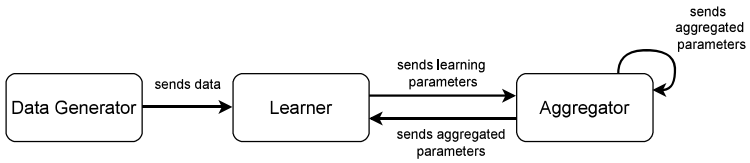


Figure 2.1: Process of federated learning. Adapted from Gecer and Garbinato [GG24].

Since model parameters and the resulting model remain vulnerable to attacks (see section 2.2), federated learning should be combined with additional PPMs, such as differential privacy, to preserve privacy [KL22].

LPPMs can be evaluated along three dimensions, as defined by Primault et al. [Pri+19]: performance, utility, and privacy. Performance metrics examine factors such as execution time, communication overhead, energy consumption, scalability, and fault tolerance. Utility metrics assess how well quality of service is preserved after a Location Privacy-Preserving Mechanism (LPPM) is applied and are highly dependent on the specific application context. Privacy metrics, as previously discussed in Section 2.2, quantify the strength of protection offered, based on the assumed adversary and trust models.

While these metrics offer valuable insights for comparing systems, there is currently no universally accepted standard for evaluating LPPMs, making objective comparisons across approaches difficult [Gad+24; Pri+19].

2.4 Asset Tracking in Hospitals

Asset tracking is critical in hospitals, where equipment, beds, and other assets must be located quickly to support clinical workflows, reduce costs, and improve patient care [Baz+21]. While GPS is effective for outdoor positioning, it performs poorly indoors due to signal attenuation and limited accuracy [Kim+15].

To overcome this, hospitals utilize Indoor Positioning System (IPS), often referred to as Real-Time Locating System (RTLS) when operating in real or near-real-time. These systems use a variety of sensing technologies. A review by Bazo et al. found RFID to be the most commonly used in hospitals (24 cases), followed by Wi-Fi (5), BLE (4), UWB (4), infrared (2), ZigBee (1), and ultrasound (1) [Baz+21].

Each technology has its advantages and drawbacks. RFID often requires manual scanning and may interfere with medical devices [Tog+08]. Wi-Fi can leverage existing infrastructure but consumes more power than other options [Baz+21]. Comparative studies show that Wi-Fi generally offers higher accuracy due to stable RSSI measurements. However, technologies like BLE and ZigBee are appealing alternatives when energy efficiency, low deployment costs, and latency are important [Van+16]. BLE 5.0, in particular, provides improved range and performance compared to earlier versions [Kar+19]. Infrared is accurate but requires line of sight, while ultrasound struggles to track multiple devices in the same room due to interference. UWB offers high accuracy but comes with higher costs compared to BLE or ZigBee.

Selecting the right tracking technology involves evaluating factors such as positioning accuracy, energy consumption, interference susceptibility, maintenance effort, privacy, security, and scalability [Baz+21]. In hospitals, these decisions are further complicated by clinical constraints, limited staff time for maintenance tasks, and the need to avoid interference with medical devices [Baz+21].

The system analyzed in this thesis is based on BLE. Although the choice of technology was not made within the scope of this work, BLE forms the basis of the existing implementation, without exploring alternative technologies in detail. BLE has also been studied in the context of patient monitoring [Kom+16], and recent research has shown that machine learning can significantly improve positioning accuracy [DAl+20; Peš+18].

Privacy concerns are especially relevant in hospital settings. BLE-based IPS can be augmented with PPMs. For instance, Shubina et al. propose a differential privacy approach using machine learning [Shu+23], while Ruiz et al. explore federated learning for BLE-based positioning systems [REE24]. Other technologies such as RFID [RBA17] and UWB [Mül+22] have also been studied for privacy-preserving applications.

The asset tracking system discussed in this thesis is described in detail in chapter 3.

2.5 Context-Aware

A widely accepted definition of context from Dey and Abowd is [Abo+99]:

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.*

Although there is no universal consensus –given the variety of definitions, categorizations, and interpretations of the term context– Dey and Abowd's definition has gained wide acceptance, largely due to its broad and inclusive nature [AAC16; Per+14].

Applying this definition to an asset tracking system in a hospital environment, users are hospital staff –such as nurses or logistics personnel –who interact with the system to locate and manage medical equipment. The application is the tracking system that facilitates this interaction.

Each asset, such as a VAC pump, is equipped with a tracker, making the asset itself an entity whose context (e. g., location, usage status, movement history) can be characterized. Although patients are not direct users of the system, they are entities relevant to the interaction and therefore part of the contextual landscape.

Context in this scenario may include, but is not limited to:

- The current room or ward where the VAC pump is located.
- The timestamp of its last movement.
- The assigned patient ID (if available).
- The responsible staff member.
- The status of the asset (e. g., in use, idle, in stock).

In addition, Dey and Abowd define four primary context categories essential for characterizing an entity's context: location (e. g., asset position), identity (e. g., patient or device UUID), activity (e. g., in use, idle, or being cleaned), and time (e. g., last movement timestamp) [Abo+99]. These correspond to the core situational questions: where, who, what, and when, and are typically captured by the system [Abo+99].

In contrast, secondary context is information that is not directly sensed or collected, but is derived from primary context elements through processing, aggregation, or interpretation [Abo+99]. For example, predicting the availability of hospital assets in an asset tracking system may involve combining several primary context elements –such as location (e. g., Bamberg Hospital) and time (e. g., 3:00 p.m.). These inputs allow the system to analyze real-time and historical data to infer asset availability. Another example is retrieving a patient's medical records using their ID: the ID itself is the primary context, while the retrieved records are the secondary context derived from it.

The concept of context-awareness is based on the use of contextual information. Dey and Abowd define context-awareness in systems as follows [Abo+99]:

> *A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*

In asset tracking, this means the system integrates contextual data –such as location, identity, activity, or time– to infer situational meaning and provide task-relevant responses. What is considered relevant depends on the specific application tasks.

Because context-awareness is task-dependent, its implementation varies across different environments. The following examples illustrate how contextual data is used to meet the unique demands of each system.

In mobile applications, contextual information can be used to detect proximity, monitor status, and respond to situational cues. Sykes et al. [SPN15] demonstrate this with several context-aware applications developed using iBeacon technology. One application uses RSSI of Bluetooth to determine whether a key is nearby, enabling real-time retrieval of items. Another monitors temperature during food transport to ensure cold chain compliance by detecting environmental deviations. For healthcare applications, they have developed two other applications. The first uses orientation sensors to monitor whether a wheelchair remains upright, triggering alerts when abnormal positioning is detected –potentially indicating that the person has fallen out. The second uses patient identification data to determine required equipment and procedural steps, guiding caregivers through clinical tasks to support safe and accurate execution.

In indoor positioning, contextual information can refine spatial estimates by integrating environmental and behavioral data. Peši'c et al. proposed BLEMAT (Bluetooth Low Energy Microlocation Asset Tracking), a system that improves indoor localization by analyzing contextual information such as Bluetooth signal strength, and historical data [Peš+18]. By using these inputs with the help of machine learning, BLEMAT significantly improves the accuracy of BLE-based tracking.

In privacy-preserving positioning, context can preserve utility while protecting sensitive data. Meftah et al. [MRC20] addressed the privacy risks associated with transmitting raw location data. Their IndoorHash method uses nearby WiFi infrastructure as contextual input to generate similarity-based location hashes. These hashes support location-based services without exposing precise coordinates, providing a privacy-preserving yet functional alternative to sending raw location data.

In commercial environments, context can be used to personalize services, but also to manage privacy preferences. Banerjee et al. [Ban+19] presented Hip-Car (Hierarchical Indoor Positioning for Context Aware Recommendation), a system designed for physical retail spaces that combines hierarchical indoor positioning with privacy controls. It uses contextual information, such as spatial relationships within a building, to

generate personalized, context-aware recommendations. At the same time, it allows location sharing to be restricted to broad zones (e.g., floor level) by applying privacy policies that tailor the type and granularity of information shared based on contextual relevance.

While Dey and Abowd provide a broad conceptual definition of context, Etzion and Niblett offer a more implementation-focused perspective that aligns well with event-driven systems [EN10] –such as the PrivProxy application developed in this thesis.

> *A context is a named specification of conditions that groups event instances so that they can be processed in a related way. It assigns each event instance to one or more context partitions. A context may have one or more context dimensions and can give rise to one or more context partitions.*

Here, context partitions refer to groupings of event instances based on shared conditions. These partitions are not necessarily exclusive; a single event may belong to multiple contexts simultaneously.

Etzion and Niblett describe the following context dimensions [EN10]:

- *Temporal context*, where events are grouped by time intervals –such as fixed periods (e.g., 24 hours), intervals between specific triggers, or time windows.
- *Spatial context*, where event grouping is based on location data, such as GPS coordinates or room identifiers, geofences, or proximity to mobile devices.
- *State-oriented context*, defined by the state of external entities. For example, events are included in a context only when a device is active or a patient is checked into a ward.
- *Segmentation-based context*, which organizes events based on attributes within the event data itself. For example, all events with the attribute `deviceOnPatient=True` can be grouped together.

These context dimensions are not mutually exclusive. They can be combined to define complex, real-world contexts –such as all events involving mobile assets on a specific floor during night shifts when cleaning status is idle.

Context-awareness enables systems to go beyond static responses and instead adapt to changing situations, user tasks, and environmental conditions. In hospital asset tracking, this means the potential for better workflows, safer patient care, and potentially enhanced privacy. The design and technology of the asset tracking system relevant to this thesis will be discussed in the following chapter.

3 Asset Tracking System and Use Case: VAC Pumps

This chapter introduces the asset tracking system used in Explanym and its role in tracking VAC pumps. It describes the hardware of the system and the technologies used. Based on the asset tracking system, the application for applying privacy-preserving mechanisms is developed and further explored in the following chapters.

3.1 System Architecture and Stakeholders

Figure 3.1 gives an overview of the entire tracking system, omitting some components and features of Clinaris and Safectory as they do not affect the workflow. The system consists of three subsystems: Clinaris, on the left, provides a web application for managing hospital assets. Safectory, in the middle, provides Clinaris with asset location data. The hospital, on the right, contains the tracking infrastructure, with beacons permanently installed in various areas and tags attached to the assets to be tracked.

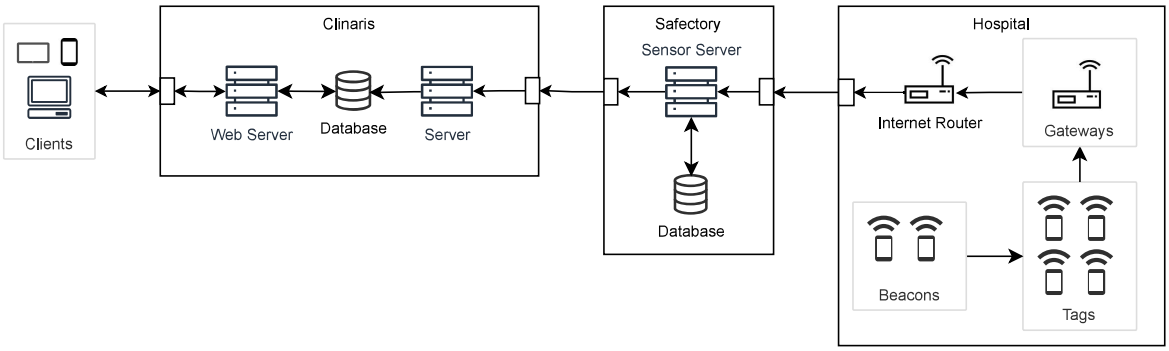


Figure 3.1: Overview of the Asset Tracking System.

.

The asset tracking system involves several stakeholders with different roles and interests. To provide a clear overview, a stakeholder map (figure 3.2) was created based on the approach of Robertson and Robertson [RR12]. The stakeholder map provides an overview of the key stakeholders in the asset tracking system. At the center is the system itself, surrounded by three areas: the operational workspace, which includes stakeholders who interact directly with the system; the surrounding businesses, which consist of stakeholders who benefit from the system without direct interaction; and the wider area, which includes those with influence or interest in the system.

Robertson and Robertson define specific roles for each domain [RR12]. In our use case, in the operational work area, the roles include operational support and maintenance (Clinaris or Safectory, depending on the subsystem), and operators/users (hospital staff).

The hospital and the German Federal Ministry of Education and Research are included in the business area because they both benefit from the system. The Ministry also acts as a sponsor, funding the Explanym project and benefiting from its research results. The University of Bamberg is also included in this area in its role as a consultant, supporting the development of a system that provides PPMs and contributing academic expertise.

The wider environment includes the hospital's data protection officer as a legal expert and adversaries as negative stakeholders. Negative stakeholders either seek to undermine the success of the system or pose a direct threat, such as hackers or fraudsters. A key focus of the Explanym project is the impact of the asset tracking system on special interest groups– patients, visitors, and hospital staff –who may be directly or indirectly affected by it.

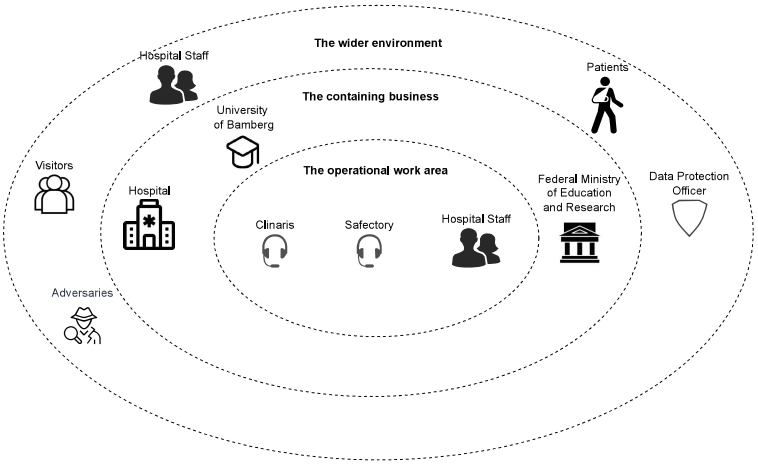


Figure 3.2: Stakeholder Map for VAC Pump Tracking System. Adapted from Robertson and Robertson [RR12].

3.2 Components and Tracking Technology

Before delving into the tracking technology, the individual system components shown in 3.1 will be examined in detail.

Clinaris provides the web-based HPM®[1] application, which is accessible through a browser. The application allows hospital staff to manage medical assets and monitor the reprocessing workflow, including documentation, maintenance and calibration scheduling.

1: https://www.clinaris.com/ (last accessed 12.04.2025).

Different user roles have specific accesses and views within the system. Technicians can track assets that need to be reprocessed, while the nursing department can request maintenance, cleaning, and monitor asset status. One of the key benefits of HPM® is location tracing, which allows users to see where each asset is located and its current status. Asset status is managed within Clinaris and is determined based on location data and actions recorded in HPM®. The web interface shown in 3.3, includes

- ① A status overview of all tracked assets listed.
- ② Locations of tracked assets.
- ③ A shortcut menu for a tracked asset.

Additional features were implemented for the initial field test, including a random patient number generator for testing purposes that replaces real patient numbers with arbitrary ones. Notifications were also added to alert users when the number of available assets falls below a critical threshold.

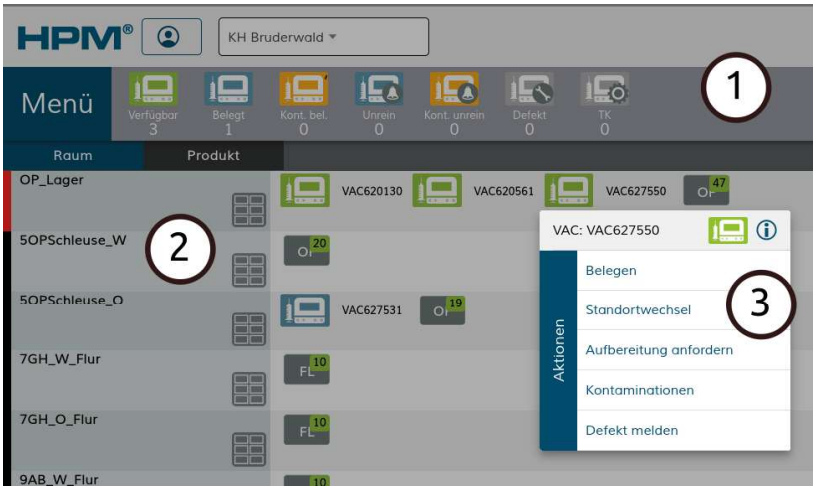


Figure 3.3: Web Interface of HPM®

While HPM® is specifically designed for hospital environments, Safectory's Track² platform supports general asset tracking. The web application allows users to query asset locations and visualize them on an OpenStreetMap[3]. The web interface shown in 3.4 includes:

2: https://safectory.com/assettracking (last accessed 12.04.2025).

3: https://www.openstreetmap.org/about (last accessed 03.04.2025)

- ① A list view of all tracked assets.
- ② Status information for a tracked asset.

- ③ An OpenStreetMap view with a saved floor plan.
- ④ A list of events that have occurred.

All devices, including beacons, tags, and BLE gateways, can be configured remotely. Settings include trigger sensitivity (motion detection threshold), heartbeat intervals, and whether the location is determined by a single beacon or multiple beacons using trilateration. In addition, Track has the ability to define geofences so that events such as an asset entering or leaving a defined zone can be detected.

Users can manage location data, create user groups and control access permissions through the web application. Most of these operations are also available through Safectory's API, which plays a key role in implementing the anonymization software in chapter 6.

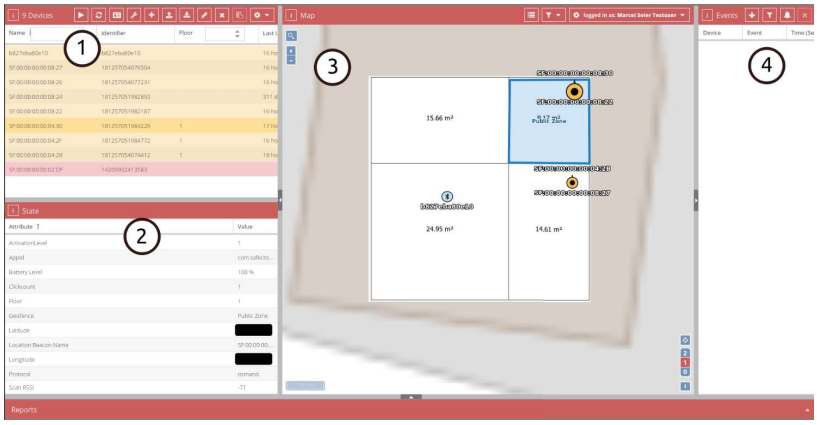


Figure 3.4: Web interface of Track.

In the initial field test of the Explanym project, the Track application and HPM® were provided by their respective vendors and were not hosted onsite by the hospital. The Explanym project team chose not to integrate the Hospital Information System (HIS) to ensure that only location data and no additional sensitive patient data was stored in the system, especially given the external hosting of Track and HPM®.

The tracking system relies on specific technologies and communication protocols. The associated hardware is described in section 3.3. As shown in the hospital section of figure 3.1, the tracking hardware consists of four components: beacons, tags, an Internet router, and BLE gateways. The Internet router forwards the data to the Track back-end. Communication between beacons, tags, and gateways is based on BLE.

Bluetooth consists of several wireless communication specifications that operate in the 2.4 GHz ISM band, the same as WiFi standards 802.11x, and are governed by the non-profit Bluetooth Special Interest Group (SIG). Bluetooth technology supports short-range communication –typically about 10 to 100 meters [Has+18]. In point-to-point communication, devices can communicate through ad hoc networks called piconets, each consisting of a master and one or more slave devices, with each piconet using its own timing clock and frequency hopping sequence to avoid interference with nearby piconets [Has+18].

Bluetooth is often divided into two types: Classic Bluetooth (BR/EDR), optimized for higher data throughput (e. g., audio streaming), and BLE, designed for low-power, battery-efficient applications. Although the protocols are largely independent, dual-mode chipsets allow both to run in parallel using a common antenna.

Over time, Bluetooth has evolved significantly, with continuous improvements in range, throughput, security, and functionality. The latest version, Bluetooth 6.0, was released by SIG in 2024.

Because of BLE's focus on energy efficiency, it serves as the foundation for many low-power applications –such as proximity-based services– and is central to protocols such as the iBeacon protocol, which is used as the communication protocol by the tracking hardware.

In the asset tracking system, beacons are stationary and continuously broadcast BLE advertising messages. Tags are mobile and can receive these broadcasts just like any other BLE device. Upon receiving a beacon's message, the tag sends its own broadcast containing information about the detected beacon. This message is then picked up by a BLE gateway and forwarded to the back-end of the Safectories application track, allowing the system to determine the location of the tag.

The iBeacon protocol, introduced by Apple in 2013, uses BLE to transmit a payload containing a UUID, major, minor, and measured power [KS14]. These values allow hierarchical location identification. For example, the UUID can be the same for all beacons within a hospital, while the major and minor values can identify the floor and ward, respectively. This structure allows for flexible and scalable deployment. The measured power is used to estimate the proximity between the beacon and the receiving device.

There is no communication back to the beacon; beacons only transmit. Devices receiving the signal can determine which beacon sent it and estimate the distance to it based on signal strength. The transmission rate of these signals affects both update frequency and battery life, making it an important configuration parameter in real-world deployments.

In Safectory's implementation, the standard iBeacon protocol is extended by an additional byte to transmit the battery level as a percentage. The first 25 bytes of the payload remain consistent with the original protocol (see figure 3.5).

The tags are configurable and also use BLE advertising messages to communicate. They support multiple modes, including the ability to report multiple beacon detections, send heartbeats, and events such as button presses using custom advertising messages.

3.3 Hardware

This section introduces the hardware used for Asset Tracking. Figure 3.6 illustrates the devices involved.

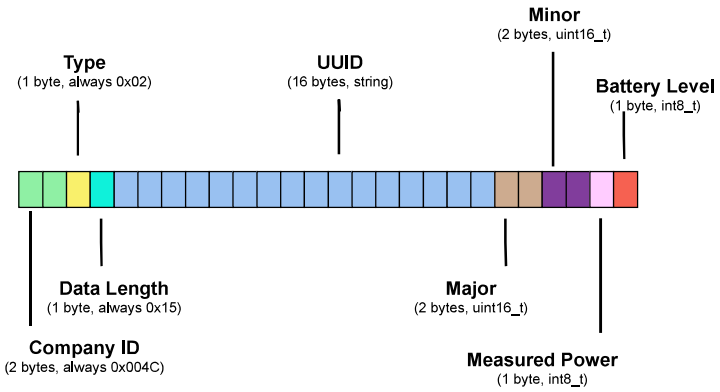


Figure 3.5: iBeacon Payload format for Safectory Beacons. Adapted from [KS14].



Lynmouth Tag / Beacon. Campanella Tag. BLE Gateway.

Figure 3.6: Tracking Hardware.

This project uses Safectory's Lynmouth[4] beacons and tags, based on Texas Instruments' ARM-based CC2640R2F chipset with 275kB of flash memory, 28kB of RAM, and support for Bluetooth up to version 5.1. The devices measure approximately 43 x 43 x 19mm and weigh 20g.

In addition, the system uses Safectory's Campanella[5] tags, which use the same chipset and RAM but offer a more compact form factor (32 × 8.5 mm) and reduced weight (11 g) at the expense of shorter battery life.

The final component is Safectory's BLE gateway[6], which uses a 1.2GHz quad-core CPU and 1GB of LPDDR2 SDRAM. It supports Bluetooth v4.1 EDR Class 1 and WLAN 802.11 b/g/n.

4: For the datasheet see: https://safectory.com/wp-content/uploads/2023/10/2023-10-16_safectory_Lynmouth_datasheet_EN.pdf (last accessed 17.09.2024).

5: For the datasheet see: https://safectory.com/wp-content/uploads/2023/10/2023-10-16_safectory_Campanella_datasheet_EN.pdf (last accessed 17.09.2024).

6: For the datasheet see: https://safectory.com/wp-content/uploads/2022/11/2022-10-12_safectory_Gateways_datasheet_EN.pdf (last accessed 17.09.2024).

3.4 Use Case VAC pumps

Vacuum-Assisted Closure (VAC) pump therapy, or negative pressure wound therapy, promotes wound healing by treating complex or poorly healing wounds. For this project, two types of VAC pumps were tracked: the big and small VAC pumps, as shown in figure 3.7.

A variety of assets from different categories were considered for the field test, including hospital beds, heavy-duty beds, wheelchairs, food containers, goods carts, laundry, service notebooks, pulse oximeters, chest



Small VAC Pump. Big VAC Pump.

Figure 3.7: VAC Pumps.

pumps, VAC pumps, sterilized surgical trays, pharmaceuticals, and personnel tracking (e. g., patients with restrictions, staff with special qualifications, or staff during infectious disease outbreaks). However, personnel tracking was excluded from the first field test, which was intended to serve as a pilot, and would be considered for the second test. Ultimately, tracking of VAC pumps or hospital beds remained, with VAC pumps being selected due to their simpler implementation, clearer lifecycle, and easier approval by the privacy officer and works council. All these different options demonstrate the wide range of possibilities for tracking in a clinical setting.

As part of the Explanym project, Müller and Halbritter created a state diagram based on expert interviews to map the states of a VAC pump [MH24]. The diagram 3.8 illustrates key stages in the pump's lifecycle, including storage, ordering, attachment to a patient, removal, cleaning, and return to storage. The cleaning process consists of wipe disinfection only, while other medical devices may require more complex decontamination procedures. The diagram also shows the corresponding status in Clinaris HPM® at each stage. It also serves as a template for the implementation of state based anonymization (see chapter 6).

For the first field test, a total of nine BLE gateways and 22 beacons were installed on three floors to track five VAC pumps.

- *Floor 7*: One BLE gateway and two beacons were installed at a single station to detect station entrances.
- *Floor 11*: Two stations were equipped with one BLE gateway and two beacons each.
- *Test Floor 9*: Six BLE gateways and 16 beacons were placed in a designated test area, allowing for room-precise tracking. However, no patients with tracked VAC pumps were treated there.

Figure 3.9 shows the setup for Test Floor 9. The goal was to create a controlled environment where realistic conditions could be simulated and

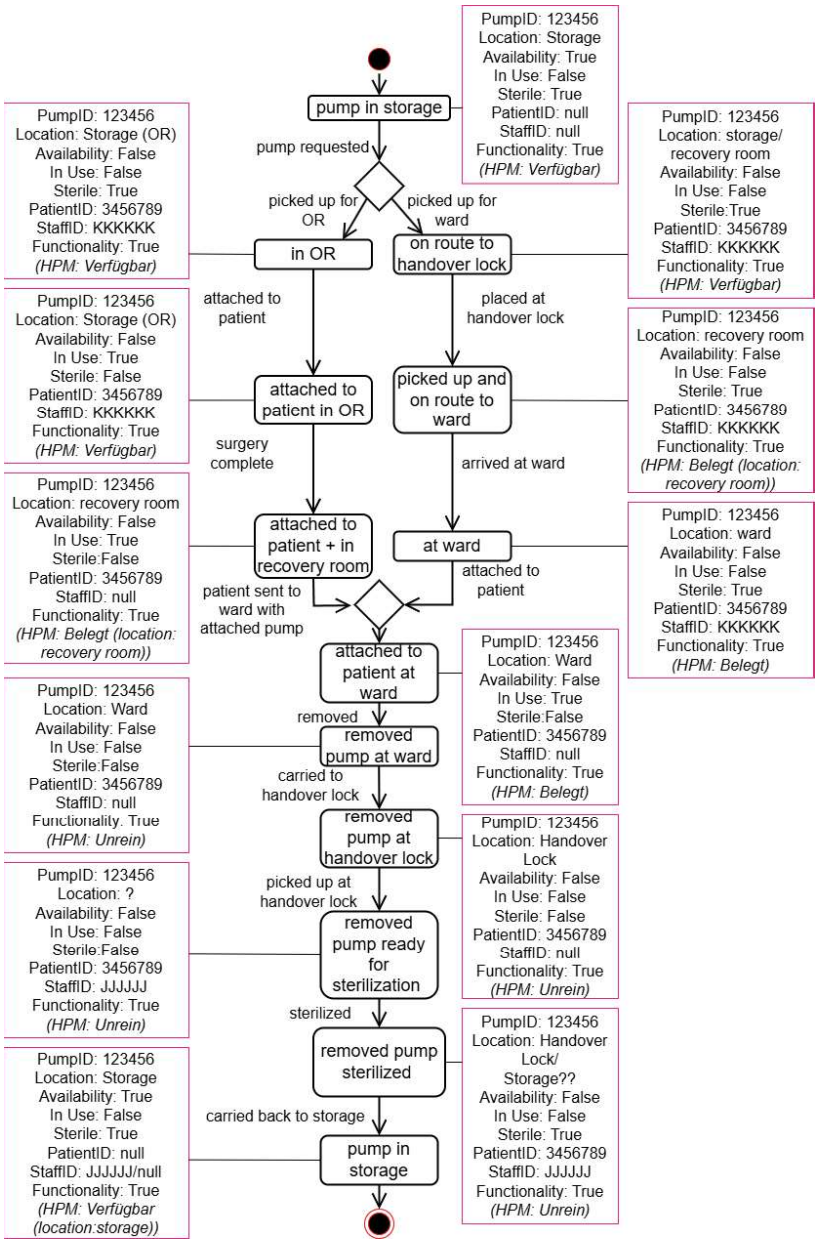


Figure 3.8: UML State Diagram. Source: Müller and Halbritter [MH24].

tested with precise tracking while avoiding patient tracking. In contrast, data on Floor 7 and Floor 11 was collected with minimal precision, primarily to generate data and evaluate the overall setup, with a strong emphasis on safeguarding the privacy of the tracked individuals.

Tracking VAC pumps provides benefits in monitoring availability and preventing loss or theft. However, the need for tracking is greater for other equipment, such as beds or wheelchairs, where additional benefits such as ensuring disinfection procedures and enabling traceability of infections could be achieved. For the Explanym project, tracking VAC pumps was well-suited for testing both the asset tracking system and the anonymization application.

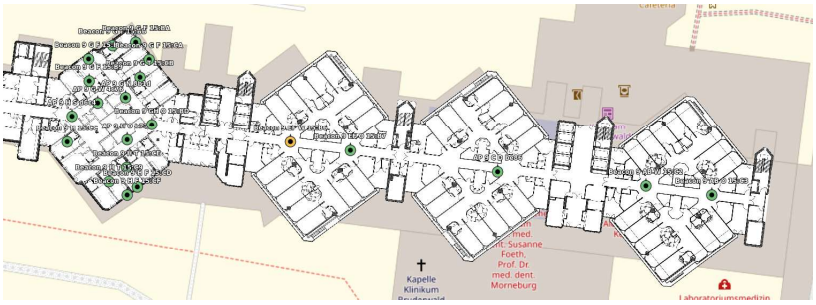


Figure 3.9: Test Setup Floor 9 for Field Test 1.

More details on VAC pump tracking can be found in the Implementation chapter (section 6.6), which explains how the PrivProxy application protects the generated data, and in the Evaluation chapter (section 7.1), where PrivProxy is tested in a clinical setting.

4 Privacy Threat Modeling using LINDDUN GO

This chapter introduces threat modeling and the LINDDUN Privacy Threat Modeling Framework, detailing its scope and objectives in the context of the Explanym project. In addition the chapter describes the LINDDUN GO method and a workshop conducted for the Explanym project using this approach. The chapter concludes with a list of privacy threats identified during the workshop.

4.1 Threat Modeling

According to the Threat Modeling Manifesto, *"Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics"* [Bra+23].

System representations, often presented in the form of a Data Flow Diagram (DFD), are essential in threat modeling. In the past, most threat modeling approaches focused primarily on security aspects [Den+11; Sho14]. Over time, privacy considerations and new privacy-focused approaches have emerged. Therefore, privacy-related issues have been increasingly incorporated into threat modeling approaches [Den+11; Sho14; WSJ20]. While security and privacy are often aligned, they can sometimes conflict.

Independent of its focus, threat modeling offers advantages that can benefit the overall system. Threat modeling addresses system problems that other techniques may miss [Sho14], contributes to the development of better overall products [Sho14], supports in the definition of requirements [Den+11; Sho14], and helps to detect bugs early in the development process [Sho14].

Although threat modeling can be applied at any stage of a system's lifecycle, it is most effective early on [Bra+23; Den+11; Sho14]. Early adoption benefits from greater design flexibility and minimizes the cost and effort of implementing changes compared to later stages of the system lifecycle.

For performing threat modeling, Shostack outlines a four-phase process [Sho14]:

1. *Model the system:* Create a representation of the system, such as a DFD, to support phase two.

2. *Find threats:* Analyze the system model to elicit threats.
3. *Address threats:* Use one of the following approaches to deal with identified threats:
   - Eliminate: Completely remove the threat (e. g., disable remote access).
   - Mitigate: Reduce the likelihood or impact of the threat (e. g., add encryption).
   - Transfer: Transfer the threat or risk to another entity (e. g., delegate authentication to a third party).
   - Accept: Accept the risk (e. g., accept risks associated with hardware tampering).
4. *Validate:* Validate the result for completeness and effectiveness.

In the context of this thesis, the four-phase process is applied as follows: The results of the *model the system* (1) phase are presented in section 4.4. The *find threats* (2) phase is outlined in section 4.3, with the identified threats discussed in section 4.5. The *address threats* (3) phase is covered in section 4.5, with its contributions forming part of chapter 5. Finally, the *validate* (4) phase is covered in chapter 7.

4.2 Objective and Scope

The objective of the LINDDUN GO workshop is to identify threats that can inform the creation of additional privacy-related requirements or the adaptation of existing requirements. By analyzing where the threats occur in the system, the workshop will provide insights that can help determine appropriate locations for implementing the PPMs and integrating relevant Privacy-Enhancing Technology (PET), thereby enhancing the overall privacy of the system.

The DFD in figure 4.3 defines the scope of the workshop. While the Clinaris system component is treated as an external entity, additional assumptions were made during the workshop to further refine the scope, as detailed in chapter 4.5.

4.3 LINDDUN Overview

LINDDUN is a privacy threat modeling framework inspired by the well-known security threat modeling method STRIDE [Den+11; Sho14]. STRIDE is a mnemonic where each letter represents a type of security threat. Deng et al. identified a gap in threat modeling because privacy considerations were not adequately addressed [Den+11]. To fill this gap, they developed LINDDUN, drawing inspiration from STRIDE and incorporating similar components to enable synergies.

Today, LINDDUN is used in both academic [AHR24; MK18] and industrial [ISO19; Nwe+22] contexts as a comprehensive privacy threat modeling method [WSJ20].

Similar to STRIDE, LINDDUN uses a mnemonic where each letter represents a different type of threat associated with a privacy property.

- ***L**inkability* arises when the privacy property of unlinkability is threatened, allowing data items to be linked [Den+11]. This can pose a personal threat if a data subject becomes identifiable, or a societal threat if linked data items are used to infer information about groups. For example, if a company analyzes consumer behavior data and links it to specific demographic information, it could target certain groups with discriminatory marketing tactics or pricing.
- ***I**dentifiability* arises when the privacy properties of anonymity and pseudonymity are threatened [Den+11]. It occurs when a data item can be used to distinguish a data subject from others [Sio+18]. For example, if a company conducting a survey asks respondents for their birthday, and the birthday is unique for some people, those people can be identified.
- ***N**on-repudiation* arises when the privacy property of plausible deniability is threatened [Den+11]. It occurs when a data subject performs an action that they cannot deny [Sio+18]. For example, in an online voting system, if it is possible to see who voted for whom, a voter could not deny their choice, which is a problem if repudiation is desired for privacy reasons.
- ***D**etectability* arises when the privacy properties of undetectability and unobservability are threatened [Den+11]. It occurs when an attacker can determine the existence of an item of interest, regardless of their ability to access it [Sio+18]. For example, during a login process, if the error message indicates whether the email is registered, it reveals that the owner of the email address is in the system.
- ***D**isclosure of information* arises when the privacy and security property of confidentiality is threatened [Den+11]. It occurs when an adversary gains access to data items related to data subjects that they should not have access to [Den+11]. For example, if a website uses HTTP instead of HTTPS, an eavesdropper can intercept and read the communication, potentially exposing sensitive user information such as login credentials or private messages.
- *Content **u**nawareness* arises when the privacy property of content awareness is threatened [Den+11]. It occurs when a data subject is unaware of how their personal data is processed, stored, or shared, and the purposes behind those activities [Sio+18]. For example, a mobile application may collect location data without informing users or explaining how it will be used.
- *Policy and consent **n**on-compliance* occurs when the privacy property of policy and consent compliance is threatened [Den+11]. This happens when personal information is processed, stored, or shared in violation of laws, regulations, or policies. For example, a company that provides

Element Type	L	I	N	D	D	U	N
Process	X	X	X	X	X	-	X
Data Store	X	X	X	X	X	-	X
External Entity	X	X	-	-	-	X	-
Data Flow	X	X	X	X	X	-	X

Table 4.1: Mapping table for the element-based LINDDUN approach. Source: Sion et al. [Sio+18].

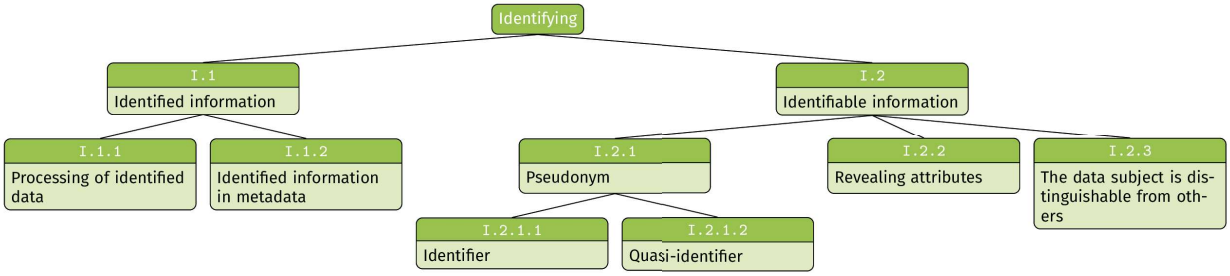


Figure 4.1: Threat tree for the identifying threat type. Source: DistriNet [Dis24].

services to people in the EU may violate GDPR by failing to honor a data subject's request to delete their information, or by continuing to send marketing emails despite unsubscribing.

The first five threat types fall into the category of hard privacy, while the last two fall into the category of soft privacy [Den+11]. Soft privacy prioritizes data security –ensuring that data is processed only for specific purposes and with consent, using measures such as policies, audits, and access controls [Den+11]. In contrast, hard privacy prioritizes data minimization – limiting data generation and trusting as few entities as possible [Den+11]. Each category has its own toolbox of techniques and strategies for addressing associated threats.

The original LINDDUN follows an element-based approach, similar to element-based STRIDE. Each DFD element is analyzed individually, and certain threat types are excluded based on the DFD element. The table 4.1 summarizes this mapping.

The following section outlines the six steps of the LINDDUN process as described in [Den+11]. First, a system representation is created as a DFD. In the second step, each DFD element is assigned a threat type using the 4.1 mapping table. Threats are identified using threat trees. An example of a threat tree is shown in figure 4.1. The current iteration of LINDDUN provides threat trees that include descriptions and examples to support the threat elicitation process [Dis24].

In the third step, identified threats are documented as misuse cases. Misuse cases depict harmful scenarios in which an adversary (e. g. a fraudster) intentionally exploits the system [RR12]. In the counterpart, use cases describe expected interactions between users and the system to achieve a desired outcome [RR12].

Source		Destination	L	I	N	D	D	U	N
Process	Flow	Process	X	X	X	X	X	-	X
Process		Data Store	X	X	X	X	X	-	X
Process		External Entity	X	X	X	X	X	-	X
Data Store		Process	X	X	X	X	X	-	X
External Entity		Process	X	X	X	X	X	X	-
Process	Flow	*Process*	X	X	X	-	-	-	X
Process		*Data Store*	X	X	X	-	-	-	X
Process		*External Entity*	X	X	X	-	-	-	X
Data Store		*Process*	X	X	X	-	X	-	X
External Entity		*Process*	X	X	X	-	-	X	X
Process	*Flow*	Process	X	X	X	X	X	-	-
Process		Data Store	X	X	X	X	X	-	-
Process		External Entity	X	X	X	X	X	-	-
Data Store		Process	X	X	X	X	X	-	-
External Entity		Process	X	X	X	X	X	-	-

Table 4.2: Mapping table for the interaction-based LINDDUN approach. The highlighted (*colored and emphasized*) elements are where the privacy threat is associated. Source: Sion et al. [Sio+18]

In the fourth step, the identified threats are evaluated and prioritized through a risk assessment. The LINDDUN approach does not mandate a specific method, as there are numerous risk assessment techniques, and the best choice depends on the context of the system.

In the fifth step, privacy requirements are derived from the misuse cases. In the sixth and final step, appropriate PETs are selected to meet these requirements and mitigate the identified privacy threats.

The entire process is supported by knowledge elements, such as the threat trees or the mapping table. The content of the knowledge elements is regularly updated [WSJ20]. A major update was made with the introduction of the GDPR. In particular, the content for the threat types content unawareness, and policy and consent non-compliance have been adapted to the GDPR content [WSJ20].

The LINDDUN approach has been criticized for its complexity [Sio+18; WSJ20] and the high level of threat modeling and privacy expertise required to apply it effectively [Sho14; Sio+18; WSJ20]. To address these challenges, new variations have been developed over time.

One such variant is the interaction-based LINDDUN, which focuses on interactions between sources and destinations rather than analyzing each DFD element individually [Sio+18]. The interaction-based approach reduces the number of iterations because the interactions are fewer than the individual elements [Sio+18]. In addition, it helps identify threats that might be missed in larger contexts when interactions are not considered [Sio+18]. The corresponding mapping table is shown in table 4.2.

Another variant is LINDDUN GO. LINDDUN GO is a lightweight, collaborative approach that prioritizes usability while incorporating the

core process and privacy knowledge of traditional LINDDUN [WSJ20]. It reduces the overall effort and expertise required for threat modeling [WSJ20]. To apply LINDDUN GO, only three components are needed: a system sketch, knowledge of LINDDUN, and the LINDDUN privacy threat cards[1].

Unlike many other threat modeling card games, LINDDUN GO cards provide comprehensive information. Each card includes a title, hot spot, threat source, description, elicitation questions, examples, consequences, additional information, and a card identifier, designed to help non-experts understand the specific threat types. A comparison of the LINDDUN GO cards with those from a well-known STRIDE-based card game called Elevation of Privilege[2] is shown in figure 4.2.

1: Link to LINDDUN threat cards: https://web.archive.org/web/20240813021444/https://downloads.linddun.org/linddun-go/default/v240118/go.pdf (last accessed 28.11.2024).

2: Link to Elevation of Privilege threat card: https://web.archive.org/web/20240909155801/https://www.microsoft.com/en-us/download/details.aspx?id=20303 (last accessed 28.11.2024).



(a) LINDDUN GO card

(b) Elevation of Privilege card

Figure 4.2: Comparison of LINDDUN GO and Elevation of Privilege with representative cards.

The gameplay of LINDDUN GO is simple: a player draws a card, reads its description and identifies potential threats based on the system representation. Other players can suggest additional threats. When no more threats are identified, the card is discarded and it is the next player's turn. The game ends when all the cards have been played.

The LINDDUN threat modeling framework was ideal for the Explanym project because of its privacy focus. The collaborative approach of LINDDUN GO was appropriate given the diverse perspectives and limited privacy expertise of the Explanym project team.

4.4 System Representation

A DFD was chosen for the system representation. While a simple system sketch would have sufficed for the LINDDUN workshop [WSJ20], the

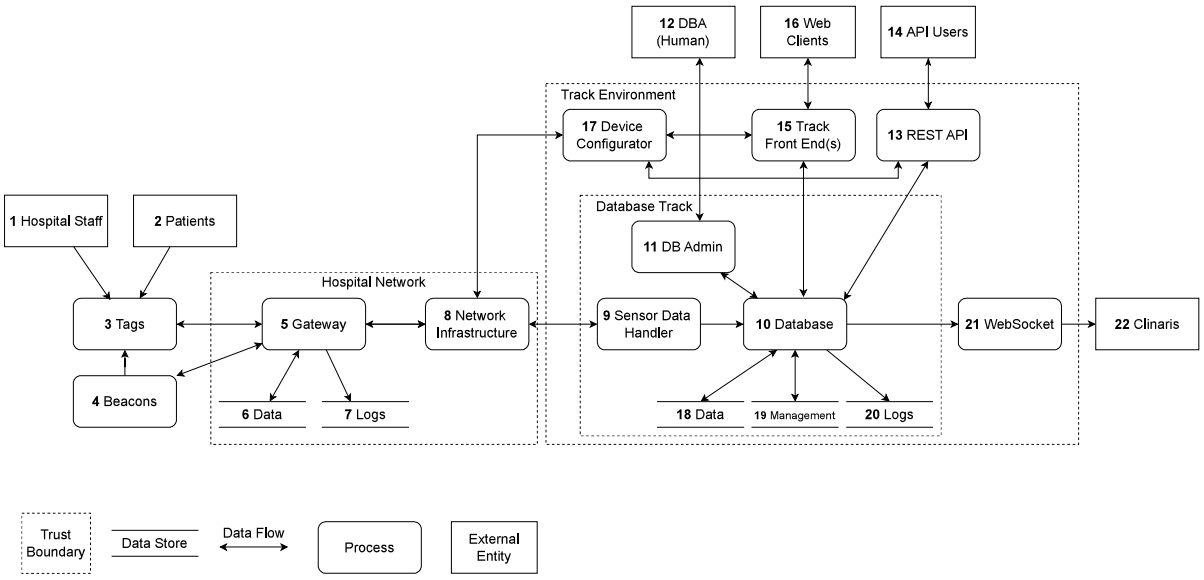


Figure 4.3: The DFD used for the LINDDUN Go workshop. Adapted from Shostack [Sho14].

DFD was chosen because it is the standard in threat modeling for its clarity and usefulness [Sho14]. The simplicity of the DFD –using only six symbols compared to UML's 80, as noted by Shostack [Sho14]– makes it accessible to those unfamiliar with this type of diagram. The DFD was created based on the defined scope (see section 4.2), reviewed with the Explanym team, and revised accordingly. The final DFD is shown in figure 4.3.

At the bottom of the diagram, the entities are listed by name and numbered for easy reference during discussions. The DFD begins on the left, where Hospital Staff (1) and Patients (2) generate mobility data by interacting with Tags (3). The Beacons (4) constantly broadcast advertising messages. When a Tag (3) detects an advertisement, the data is sent to the Gateway (5). Notably, there is no trust boundary here because the communication is public and not encrypted.

The Gateway (5) performs several functions: storing Logs (7) for debugging, caching Data (6) when the network is unavailable, and forwarding data via the Network Infrastructure (8) to the Sensor Data Handler (9). The Sensor Data Handler (9) forwards data to the Database (10), which stores the Data (18) itself, but also contains Management (19) configurations, and generates Logs (20). A Database Administrator (DBA) (12) can access the database using a DB Admin (11) account.

Data is accessible through the Track Front End(s) (15) for Web Clients (16) and through the REST API (13) for API users (14). The Database (10) forwards data to the external Clinaris (22) entity using a WebSocket (21).

The Device Configurator (17) allows remote configuration of Gateways (5), Tags (3) and Beacons (4). It can be accessed by API Users (14) via the REST API (13) or by Web Clients (16) via the Track Front End(s) (15).

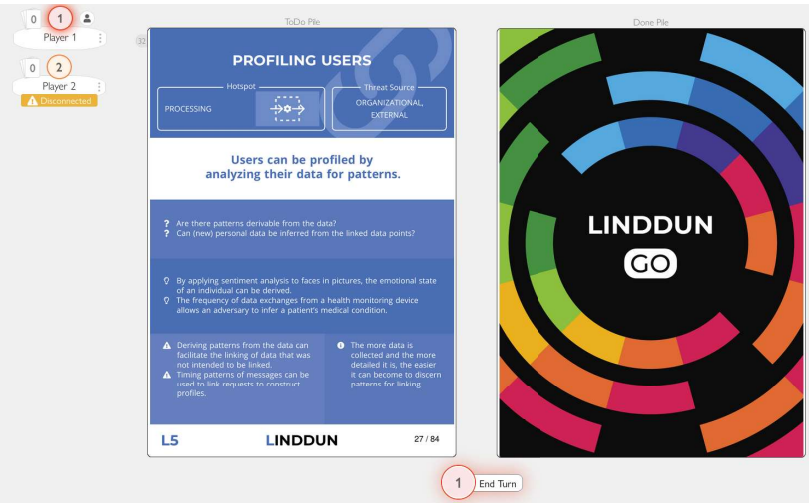


Figure 4.4: Game board for LINDDUN GO.

4.5 Execution and Results

Seven participants attended the digital workshop: two from MOBI Group, two from Safectory, two from Clinaris, and one from HTK. The digital format was chosen to maximize participation. Unfortunately, no representative from the PSI group was available. To include their perspective and expertise, the threats identified during the workshop were later reviewed with the PSI group, keeping the LINDDUN framework in mind.

The workshop began with a one-hour presentation on LINDDUN, introducing general threat modeling concepts, followed by an overview of the LINDDUN privacy threat modeling framework. The presentation, based on Kim Wuyts' materials[3], served as a template for the LINDDUN section.

Following the presentation, participants engaged in a two-hour session of the LINDDUN GO card game, during which the moderator documented the threats identified. To encourage active participation beyond screen sharing, the playingcards.io platform [4] was used to create a virtual card board beforehand. The custom board, shown in figure 4.4, features players on the left, a deck of cards, a discard pile, and a button to end the round. To improve accessibility, each threat card is assigned a page number. This allows users to easily locate and zoom in on the corresponding card in a accompanying PDF, ensuring readability even when resolution issues arise.

During the workshop, it was observed that certain vectors, such as tracking API Users (14), Web Clients (15), and the DBA (12), had been largely overlooked in regard of privacy issues. While including them in the analysis could provide valuable insights, the primary focus of the Explanym project is on anonymizing data in the context of mobility. Therefore, these threats were excluded from the scope.

The workshop led to the identification of the following ten threats:

3: Slides and a recording of Kim Wuyts' presentation are available at https://web.archive.org/web/20250107155408/https://cif-seminars.github.io/#linddun-go-a-lightweight-approach-to-privacy-threat-modeling-by-dr-kim-wuyts (last accessed 07.01.2025).

4: https://playingcards.io/ (last accessed 29.04.2025).

Threat #1 – Excessive Collection of Information by the Tracking System

DFD element(s): Device Configurator (17), Beacons (4), Tags (3), and Gateway (5).
LINDDUN categories: Data disclosure and non-compliance.
Description: Over-collection of information can occur due to an excessive number of beacons, too frequent location updates, or additional unnecessary metadata being transmitted. This results in more data being collected than necessary.
Resolution: Included in PPMs and discussed in chapter 7; see requirement FR #3.

Threat #2 – Lack of Support for Data Subjects' Rights

DFD element(s): –.
LINDDUN categories: Unawareness and non-compliance.
Description: The application does not implement essential rights of data subjects, such as the right to object, the right to be forgotten and the right to access personal data.
Resolution: Partially included in PPMs and discussed in chapter 7; see requirement FR #3.

Threat #3 – Risk of Re-Identification Through Additional Attributes

DFD elements: Data (18), Track Front End(s) (15), and REST API (13).
LINDDUN categories: Linkability and identifiability.
Description: Allowing additional attributes to be added to tags or beacons may increase the risk of re-identification of individuals, depending on the nature of the attributes.
Resolution: Included in PPMs and discussed in chapter 7; see requirement FR #3.

Threat #4 – Unintentional Employee Surveillance

DFD element: Data (18).
LINDDUN categories: Linkability, unawareness, and non-compliance.
Description: Combining collected tracking data with external information, such as shift schedules, allows unintended inferences about work processes and individual employee performance, facilitating unwarranted surveillance.
Resolution: Included in PPMs and discussed in chapter 7; see requirement FR #3.

Threat #5 – Identification via Movement Patterns

DFD elements: Data (18), Beacons (4), and Tags (3).
LINDDUN categories: Linkability and identifiability.
Description: Analysis of movement patterns from individual tags can reveal identifiable information about people.
Resolution: Included in PPMs and will be discussed in chapter 7, see requirement FR #3.

Threat #6 – Non-Repudiation Risk from Notification Feature

DFD element: –.
LINDDUN category: Non-repudiation.
Description: If a notification feature such as SMS alerts is implemented, users will be required to provide personally identifiable information (e. g., a phone number). Receiving a text message associated with the tracking system may indicate that the user was at a particular location (e .g., hospital). This creates a non-repudiation risk because it leaves an auditable record linking the person to a specific time and place that the user cannot later plausibly deny. This can raise privacy concerns in sensitive contexts.
Resolution: Out of scope. The notification feature was not part of the VAC pump use case and is therefore not included.

Threat #7 – Identifiable User Requests in the Tracking System

DFD elements: API Users (14), and REST API (13).
LINDDUN categories: Linkability and identifiability.
Description: Adding features that allow tracked individuals to interact with or access their data may inadvertently result in identifiable user requests, compromising user anonymity.
Resolution: Threat can be mapped to requirement FR #5.

Threat #8 – Insufficient Transparency for Data Subjects

DFD element: –.
LINDDUN category: Unawareness.
Description: Data subjects lack adequate information about how their data is processed and who has access to it, leading to potential non-compliance with transparency obligations.
Resolution: Threat can be mapped to requirement NFR #2.

Threat #9 – Sharing of Data with External Parties

DFD elements: Clinaris (22), and API Users (14).
LINDDUN categories: Data disclosure and non-compliance.
Description: Data is shared with additional services or external parties beyond what is strictly necessary.
Resolution: Out of scope. The prototype does not forward data to external entities, and no sharing mechanism is implemented (see section 8.1). However, the software design can be easily extended to support data sharing with external entities.

Threat #10 – Unencrypted and Detectable Communication

DFD elements: Beacons (4), Tags (3), and Gateway (5).
LINDDUN categories: Data disclosure and detectability.
Description: Unencrypted communication between beacons, tags, and the gateway is both detectable and vulnerable to eavesdropping.
Resolution: Out of scope. However, this threat will be discussed in section 4.6.

In response to feedback from a PSI group member, three previously identified privacy threats were addressed and have since been revised for clarity and completeness. These updated versions are already reflected in the threat list above. In addition to these revisions, the feedback also resulted in the identification of a new privacy threat, which is presented below.

Threat #11 – Incomplete or Ineffective Anonymization

DFD elements: Beacons (4), Tags (3), Gateway (5), Network Infrastructure (8), Sensor Data Handler (9), Database (10), and Database (10).
LINDDUN categories: Data disclosure, Identifiability, and Linkability.
Description: Anonymization mechanisms may not work correctly or consistently, resulting in incomplete or ineffective privacy protection.
Resolution: The PPMs were manually verified for this work, with more details provided in chapter 7.

Potential security threats were also identified, but are not explored further in this work. For future consideration, these include access control at the gateway (due to possible physical access), system availability in the event of partial or complete failure, the impact of a DDoS attack, and risks posed by insider threats with authorized access. Some of these issues are discussed in section 4.6.

4.6 Security Considerations

While the primary focus of this work is on privacy, this section outlines relevant security threats to the asset tracking system presented in chapter 3. The system uses BLE for communication between gateways, beacons, and tags. As discussed in Threat #10, this communication is unencrypted, which presents inherent threats.

Improvements in BLE security, such as improved security in pairing mechanisms and address randomization in newer protocol versions, help mitigate some vulnerabilities [Cäs+22]. Nevertheless, BLE remains vulnerable to various attacks, as highlighted in more comprehensive surveys on security of Bluetooth [Has+18] and BLE [Cäs+22].

At a higher level, security threats in indoor positioning systems must also be considered. Sartayeva and Chan [SC23] classify indoor positioning methods into categories, including proximity-based methods, where location is inferred based on the presence or signal strength of nearby beacons. Based on this classification, the asset tracking system described in chapter 3 can be categorized as proximity-based, specifically using the landmarking variant –where the nearest beacon is assumed to represent the current location of a tag.

The following types of attacks are relevant to proximity tracking systems [SC23]:



Figure 4.5: Ubertooth One Bluetooth sniffer.

- *Resource Draining*: Overloading the system by introducing rogue beacons that constantly transmit signals and degrade performance.
- *Malicious Node Attack*: An attacker clones the identity of a beacon or physically manipulates the placement of the beacon.
- *Jamming Attack*: A Denial of Service (DoS) attack that disrupts BLE communication by flooding the channel with noise.
- *Relay Attack*: Signals from a distant beacon are relayed to the vicinity of a tag, misleading the system as to the tag's location.
- *Replay Attack*: Legitimate signals are captured, possibly modified, and later replayed to spoof the presence of a tag near a beacon.

In the scope of this work, a replay attack was successfully performed on the asset tracking system. The test was conducted outside clinical settings using a hardware testbed consisting of a gateway, three tags, and three beacons.

The attack setup used the Ubertooth[5] wireless development platform in conjunction with the Ubertooth One (see figure 4.5) usb antenna to capture BLE advertisements. To analyze the communication between the devices, the network protocol analyzer Wireshark[6] was used to capture Bluetooth traffic. The resulting traces were examined and revealed that beacon broadcasts were easily identified due to their static MAC addresses. In contrast, tags often randomize their MAC addresses, making them more difficult to trace consistently.

Using this data and the `hcitool`[7] command-line utility on Linux, fake advertisement packets were crafted and transmitted. The tags accepted these packets, falsely indicating new locations. This confirms the replay attack was effective, and it is likely that relay attacks would also be feasible under similar conditions.

In a non-field test environment, a security threat analysis, such as STRIDE, is recommended to identify and mitigate potential threats. Additionally, there are existing approaches aimed at improving security for the use of the iBeacons protocol [CMO21; Mar+16].

5: https://github.com/greatscottgadgets/ubertooth (Last accessed: 30.04.2025)

6: https://www.wireshark.org/ (last accessed: 30.04.2025)

7: https://linux.die.net/man/1/hcitool (last accessed 01.04.2025).

5 Requirements

This chapter outlines the constraints, functional requirements, and non-functional requirements that form the foundation for implementing the anonymization system in chapter 6. The theoretical framework is based on *Mastering the Requirement Process* by Robertson and Robertson [RR12], while the practical input is derived from Explanym project documents and the LINDDUN GO workshop.

5.1 Constraints

This section outlines the project and solution constraints for developing the anonymization system for the asset tracking system. Constraints are conditions that the product must meet and should ideally be established prior to defining the requirements [RR12].

5.1.1 Project Constraints

Project constraints refer to limitations related to time and budget [RR12]. These constraints set limits on what can be accomplished within the given resources and time frame. For this project, the following project constraint was defined based on the scope of a master's thesis.

- A working prototype is to be delivered within six months.

5.1.2 Solution Constraints

Solution constraints define mandatory design choices or specific solutions that the system must incorporate [RR12]. These constraints can result from technical dependencies, regulatory requirements, or organizational standards.

- The system shall be compatible with Safeactory's Track Infrastructure.
- The system shall only use open source libraries.

5.2 Functional Requirement

Functional requirements specify the actions a system must perform, typically expressed as verbs [RR12]. Ideally, these requirements should remain

technology-independent to ensure flexibility [RR12]. Each functional requirement comprises two main components:

- *Description:* A concise, preferably one-sentence, statement that clearly describes what the system must do [RR12].
- *Rationale:* An explanation by the author of the requirement as to why the requirement is necessary [RR12].

In addition, a *fit criterion* can be included to ensure that the requirement is measurable. A fit criterion provides clear, objective benchmarks against which the implementation can be tested, reducing ambiguity and aiding validation [RR12].

The following subsections list the functional requirements defined for this project.

FR #1 Apply PPMs to incoming data

Description: The system shall be able to apply PPMs to incoming data. Figure 5.1 shows the implementation.
Rationale: Protecting incoming data ensures that sensitive information is anonymized from the moment it enters the system.
Fit Criterion: At least one implementation exists that applies PPMs to incoming data, and its correct placement in the system pipeline is verifiable.

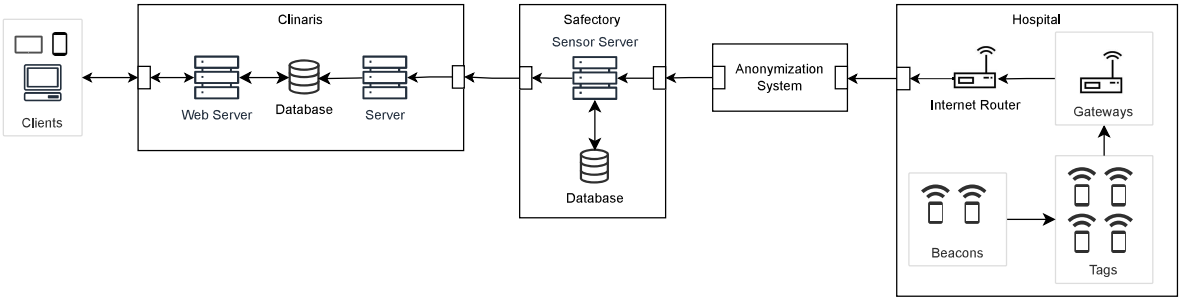


Figure 5.1: Asset tracking system with anonymization for incoming data.

FR #2 Apply PPMs to outgoing data

Description: The system shall be able to apply location privacy mechanisms to outgoing location data. Figure 5.2 shows the implementation in a client-server architecture.
Rationale: Ensures that sensitive data shared externally remains protected and compliant with privacy regulations.
Fit Criterion: At least one implementation exists that applies PPMs to outgoing data, and its correct placement in the system pipeline is verifiable.

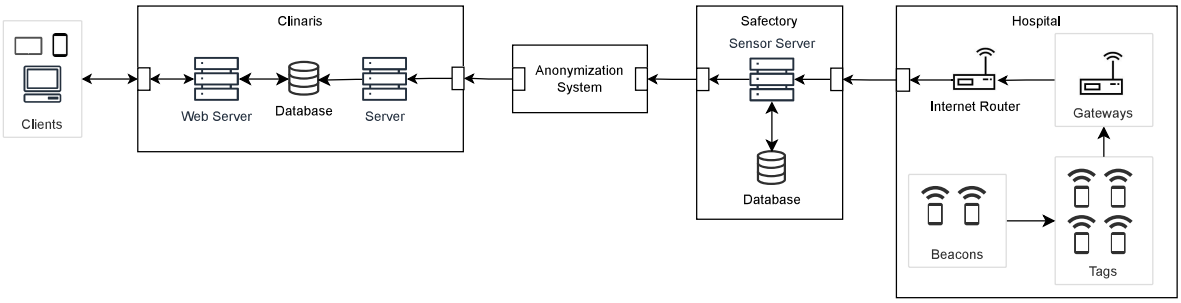


Figure 5.2: Asset tracking system with anonymization for outgoing data.

FR #3 Support for applying PPMs

Description: The system shall be able to use different location privacy preserving mechanisms.
Rationale: Allows flexibility to adapt to different use cases and privacy requirements in different contexts.
Fit Criterion: The system supports at least one location privacy preserving mechanisms.

FR #4 Include Context Information for PPMs

Description: The system shall be able to incorporate contextual information when applying PPMs.
Rationale: Contextual information ensures that PPMs are applied appropriately, based on the specific situation and data sensitivity.
Fit Criterion: The system integrates relevant contextual information (e. g., location, time, or environmental factors) to determine the application of PPMs.

FR #5 Ensure Anonymity in User Requests

Description: The system shall allow non-administrative users to access data through the front-end without logging, forwarding, transmitting, or storing any information that could identify the user.
Rationale: Logging or transmitting user-identifiable data during front-end access could compromise the anonymity of non-admin users.
Fit Criterion: For all fron-tend requests by non-admin users, system logs and request payloads must not include user-identifiable information (session ID, authentication token, or user agent string). This shall be verified through log inspection and network traffic analysis in at least one test scenario.

Together, these functional requirements define the essential capabilities of the system needed to implement and manage privacy-preserving mechanisms effectively.

5.3 Non-Functional Requirements

Non-functional requirements describe the qualitative attributes or how well aspects of a product, typically expressed as adjectives [RR12]. These include categories such as usability, performance, security, and legal [RR12]. Non-functional requirements are crucial to a system's success and are structured in the same way as functional requirements (see 5.2). The following boxes define the non-functional requirements defined for this project.

NFR #1 Scalability

Description: The anonymization system shall be able to process large volumes of data.
Rationale: This ensures that the system remains efficient and effective as the asset tracking system grows.
Fit Criterion: The system must successfully pass a 24-hour stress test simulating high-frequency data input at maximum expected load, without service interruption or critical errors.

NFR #2 Anonymization Transparency

Description: The anonymization system shall provide users with clear information on how data is anonymized, including the methods used and the extent of anonymization applied.
Rationale: Transparency in the anonymization process enhances user trust and allows them to understand how their data is being handled, potentially promoting compliance with privacy regulations.
Fit Criterion: The system shall include a user-accessible interface or documentation that describes the anonymization methods and the scope of the data transformation, and this content must be validated for clarity and completeness by user review or expert assessment.

NFR #3 Integration with Existing Asset Tracking Systems

Description: The anonymization system shall integrate with existing asset tracking infrastructure.
Rationale: Integration is essential to extend the functionality of the system across different environments, maximising its utility and enhancing data protection in different asset tracking scenarios.
Fit Criterion: By the end of the implementation phase, the system shall integrate with at least one asset tracking system.

NFR #4 Support for Diverse Use Cases

Description: The anonymization system shall support various use cases for asset tracking in different settings.
Rationale: Flexibility in use cases enhances the system's applicability and effectiveness in different environments.
Fit Criterion: By the end of the implementation phase, the system shall support at least one additional use case.

The non-functional requirements presented above define essential quality attributes that complement the system's core functionality. They aim to support the anonymization system in functioning as intended while promoting reliable performance, effective scalability, and sustained trustworthiness and legal compliance across diverse operational contexts.

6 Implementation of Priv Proxy

This chapter introduces PrivProxy, which was developed based on the requirements in chapter 5. It begins by outlining the implementation approach, including how the system fits into the broader landscape of PPMs. This is followed by an overview of the system architecture and its components. The chapter then details the technologies used, rule logic and state management, and the front-end design. A practical implementation is demonstrated using the VAC pump use case introduced in section 3.4. The chapter concludes with a discussion of implementation challenges. PrivProxy is available on GitHub[1] under the Apache 2.0 license.

1: https://github.com/B4n4n4j0e/PrivProxy (last accessed 04.05.2025).

6.1 Implementation Approach

The name PrivProxy reflects the purpose and functionality of the application: "Priv" stands for privacy, and "Proxy" indicates the routing of data through an intermediary rather than direct transmission. This proxy allows various PPMs to be applied, including LPPMs. Because LPPMs are built into PPMs, any discussion of PPMs inherently includes LPPMs.

According to the LPPM categorization of Primault et al. [Pri+19], PrivProxy can be classified as a trusted third-party system that implements real-time, rule-based PPMs. Its architecture is considered trusted because, as an external component, it has access to data from all users. However, because it is open source, PrivProxy can be deployed by various stakeholders in the VAC pump use case, such as the hospital, Safectory, or Clinaris. The most effective deployment would be within the hospital to ensure that only data processed by PPMs is stored.

PrivProxy uses a rule-based approach, identified by Primault et al. as the most flexible type of LPPM, capable of integrating all other LPPM categories (see section 2.3) [Pri+19]. In PrivProxy's first development iteration, this rule-based approach formed the basis for the implementation of the PPMs.

A key concept in the Explanym project was state-based tracking. In this approach, the tracking behavior of an asset, whether it is fully tracked, not tracked, or partially tracked (e. g. with an obfuscated location), depends on the assets current state. This led to the introduction of a state management

component in the second iteration of PrivProxy, which allows rules to be applied based on asset states.

The design of PrivProxy is based on the requirements in chapter 5, with section 7.3 detailing how these requirements are met. The following section describes the system architecture and design of PrivProxy.

6.2 Design and Architecture of Priv Proxy

PrivProxy was designed with an event-driven architecture that incorporates microservice principles. The system operates on events, making an event-driven architecture the natural choice. Most components function as independent services, following a microservice approach. This ensures scalability, flexibility, and the ability to independently develop, deploy, and scale each service as needed. The 6.1 diagram shows the event components (producers, consumers, and Event Processing Agent (EPA)) with brief descriptions below.

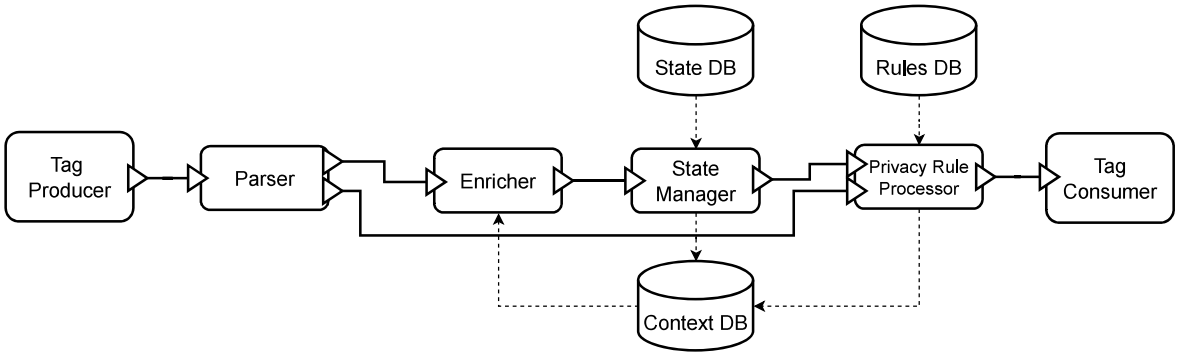


Figure 6.1: Event Processing Network for PrivProxy.

There are two implementations of the *Tag Producer*. The first, `websocket-service`, connects to the Track infrastructure (see section 3.2) via WebSocket and supports four types of notifications: status, devices, events, and positions. The status type updates asset status (e. g., PING), the devices type notifies of asset updates and configuration changes, the events type triggers notifications for events (e. g., asset entering or leaving a geofence), and the positions type sends location data, including iBeacon protocol information, geofence details, beacon attributes, and activation levels (indicating button presses, heartbeats, or position updates). This implementation only handles position notifications. Upon WebSocket connection, Track sends all current position data for each tag to PrivProxy before streaming position updates.

The second implementation, `route-request-service`, provides historical location data via Track's REST API for testing PrivProxy and its PPMs. The data mirrors the content of position notifications.

The *Parser* is the first EPA in PrivProxy. It converts data into a compatible `JSON` format and filters items based on whether they require additional data. If the key `priv_proxy` exists, the data is sent directly to the

Privacy Rule Processor; otherwise, it is passed to the Enricher for further data retrieval. Currently, the parser functionality is integrated into the `websocket-service` and `route-request-service` to reduce communication overhead, but it could be separated into a new service for full microservice compliance.

The data items are converted by the Parser into the following schema :

- *id:* Unique tag identifier.
- *time:* Timestamp for processing.
- *beacon_id:* ID of the beacon the tag has passed.
- *latitude:* Latitude of beacon location.
- *longitude:* Longitude of beacon location.
- *heartbeat:* Indicates whether the data item is a heartbeat, required for the heartbeat rule.
- *priv_proxy:* Contains context information for further processing.
- *attributes:* A miscellaneous field for remaining key-value pairs.

Next is the *Enricher*, which is implemented as `enrich-tag-service`. For each incoming data item, the Enricher fetches additional context information from a context database. There are two back-end implementations for the context database:

- *Track-based implementation:* This version uses the Track infrastructure, accessing and storing context data through a REST API. It also supports state management and capabilities.
- *Local database implementation:* This version uses a local database to retrieve context information, but does not support state management or features.

The track-based implementation goes a step further by retrieving not only basic tag attribute data, but also geofence-related information. This includes details such as how many other active assets are in the same geofence as the tag. This information, along with the tag's context and state data, is used later by the State Manager and Privacy Rule Processor.

The *State Manager*, introduced in the second iteration of PrivProxy, is implemented as the `tag-state-service`. It uses its own State DB to store states and allowed state transitions. The Context DB serves as temporary storage for counters and timestamps. It enables state transitions based on additional conditions, such as specific time intervals between timestamps. When a state changes or a counter is updated, the Context DB is updated accordingly. For a detailed explanation of state management, refer to section 6.4.

The *Privacy Rule Processor*, implemented as `rule-processor-service`, retrieves and sequentially applies rules to a data item. If a rule requires an update to the Context DB, it is sent. The Privacy Rule Processor also removes any additional information added by the Enricher or State Manager. For a detailed explanation of rule processing, see the 6.4 section.

The last stage is the *Tag Consumer*. Currently, there is only one Tag Consumer implementation: the `statistic-service`, which stores the output of PrivProxy as a file. The Tag Consumer receives the data item after all privacy rules have been applied by the Privacy Rule Processor. For production use, a custom Tag Consumer must be developed to forward the processed data to its intended destination. A detailed description of the states is provided in section 6.4.

In addition to the components described above, a web application was developed that consists of two parts: the `webapp-service-backend` and the `webapp-service-frontend`. The back-end (`webapp-service-backend`) provides a REST API for accessing and managing the Rules DB, State DB and Context DB. It also provides a Swagger[2] UI for interactive API documentation. The front-end (`webapp-service-frontend`) provides a user-friendly interface that allows users to interact with the system.

2: https://swagger.io/ (last accessed 13.04.2025).

The application serves both regular users and administrators. Regular users can enter a tag ID to view the privacy rules applied to their tracked asset. Administrative access is required to modify database entries, including adding, updating or deleting rules, state, and context information.

For more details on the web application, see the 6.5 section.

6.3 Project Structure and Technologies

Before describing the individual components and technologies in detail, the overall structure and configuration options are outlined. Figure 6.2 shows the project structure, excluding default files (e. g. `.gitignore`, `README.md`) for brevity.

The `docker-compose.yml` file defines the composition of services, shared container volumes, and network configurations. Environment variables for the containers are specified in `.env` files in the conf folder, including settings such as connection retry limits, access credentials (e. g. for the track back-end), and the choice of back-end implementation. Currently there are two back-end implementation options that can be set in the `global.env` file.

The `messages.log` file contains the output of the statistics service, which reflects the data output processed by PrivProxy.

The `sqlite_db` folder contains a Dockerfile for creating a container equipped with an SQLite database. The core application resides in the services folder, where each service is defined in a separate subfolder. Each subfolder contains a Dockerfile that specifies the image structure and the corresponding service code.

With the structure and configuration in place, the following paragraphs describe the individual components and the technologies that enable their functionality.

The application consists of several interconnected Docker [3] containers,

3: https://www.docker.com/ (last accessed 23.03.2025).

```
PrivProxy
├── docker-compose.yml
├── conf
│   ├── global.env
│   └── ...
├── database
│   └── sqlite_db
├── logs
│   └── messages.log
└── services
    ├── rule-processor-service
    ├── enrich-tag-service
    ├── route-request-service
    ├── statistic-service
    ├── tag-state-service
    ├── webapp-service-backend
    ├── webapp-service-frontend
    └── websocket-service
```

Figure 6.2: Directory structure of PrivProxy.

each handling a specific service. Docker is a platform for running applications in lightweight, isolated environments called containers. Unlike virtual machines, Docker does not virtualize an entire operating system, but instead shares the host operating system kernel, reducing overhead. It enables resource throttling (CPU, memory, and I/O), supports diverse environments, and simplifies the deployment of complex systems. Using `docker compose`, multiple containers, networks, and shared volumes are defined in a `docker-compose.yml` file. This file describes how the application should be structured and how its services interact, allowing Docker to automatically configure and manage them. Instead of manually running commands, developers specify the desired setup and Docker makes sure everything runs accordingly. This technology was a natural fit for the chosen architecture.

Python[4] was used for all services except `webapp-service-frontend`. This choice was driven by developer's expertise and Python's rich ecosystem of libraries for accessing databases, message queues, and handling HTTP communications, among others. Although there were no language constraints for developing the application, Python was chosen as the primary programming language. However, the application's modular architecture ensures that the services can be adapted or extended in other languages if required.

4: https://www.python.org/ (last accessed 23.03.2025).

SQLite[5] is a lightweight, serverless, and self-contained SQL database engine. Unlike traditional databases, it does not require a separate server process, but instead reads and writes directly to disk files. This simplicity makes it an ideal choice for development environments. SQLite was chosen for this project because of its minimal overhead, which allows for easy local storage. Three SQLite databases are used: one for storing rules, used by the `webapp-service-backend` and `rule-processor-service`; one for

5: https://www.sqlite.org/ (last accessed 23.03.2025).

user management in the webapp-service-backend;one for state management in the tag-state-service; and one for the local implementation of the Context DB, used by the rule-processor-service, webapp-service-backend, and enrich_tag-service. Since no official and verified DockerFile for SQLite could be found on DockerHub, a custom DockerFile was created, which installs SQLite on Linux Alpine. The DockerFile can be found in the project's sqlite_db directory.

SQLAlchemy[6] is used by all services that interact with SQLite databases in the project. SQLAlchemy is a Python SQL toolkit and ORM that maps Python classes to database tables, simplifying database operations using Python syntax. SQLAlchemy streamlines entity management and query writing. While SQLite can be accessed directly via SQL statements, SQLAlchemy provides the flexibility to switch to other databases with minimal code changes. This is especially beneficial for future scaling, such as migrating to a more robust database in production or taking advantage of geospatial capabilities. SQLAlchemy also integrates well with libraries such as FastAPI for efficient session management. In addition, the asyncio[7] extension is installed to enable asynchronous database operations for improved performance.

6: https://www.sqlalchemy.org/ (last accessed 23.03.2025).

7: https://docs.sqlalchemy.org/en/20/orm/extensions/asyncio.html/. last accessed 23.03.2025

FastAPI[8] is a Python web framework for building APIs, known for its high performance, automatic OpenAPI documentation, asynchronous processing, and robust data validation. In this project, the webapp-service-backend uses FastAPI to provide a REST API for managing rules and retrieving tag states in the webapp-service-frontend. FastAPI was chosen for its efficiency in handling concurrent requests, ease of development with built-in documentation, and support for both public and authenticated endpoints. Its asynchronous capabilities ensure responsive interactions, making it well suited for fast data processing.

8: https://fastapi.tiangolo.com/ (last accessed 23.03.2025).

RabbitMQ[9] is a messaging broker that facilitates communication between system components. It supports multiple protocols and provides routing and filtering capabilities. In this project, RabbitMQ manages the flow of tag data items between components, decoupling message producers and consumers to ensure scalability and fault tolerance. RabbitMQ uses the AMQP 0-9-1 protocol, where messages are published to exchanges and routed to one or more queues based on bindings. Consumers can either subscribe to queues or fetch messages on request.

9: https://www.rabbitmq.com/ (last accessed 23.03.2025).

The project uses the default exchange with four queues: enrich_queue, state_queue, rule_processing_queue, and export_queue. The message flow starts with parsers such as the websocket-service or the route-request-service, which parse incoming messages and publish them to enrich_queue for enrichment, or directly to rule_processing_queue if no enrichment is required. The enrich-tag-service, subscribed to enrich_queue, enriches the data and publishes it to state_queue. The tag-state-service, subscribed to state_queue, updates the state and forwards the data to rule_processing_queue. Finally, the rule-processing-service processes the data and publishes the result to export_queue, which is consumed by the statistic-service.

The official RabbitMQ Docker image[10] with the RabbitMQ management

10: https://hub.docker.com/_/rabbitmq (last accessed 23.03.2025).

plugin is used. This image includes an admin dashboard, simplifying debugging and monitoring. In the future, the RabbitMQ container without the management plugin could be used to reduce the image size if the UI is no longer necessary.

To interact with RabbitMQ, the `aio-pika` library[11] is used. This library enables asynchronous communication with RabbitMQ using the AMQP 0-9-1 protocol and integrates with Python's asyncio, allowing asynchronous queue processing. This asynchronous approach improves overall system performance by aligning with the asynchronous nature of other components. Additionally, RabbitMQ benefits from a large community and various community plugins that can further enhance the system, particularly for implementing additional PPMs. For example, the rabbitmq-delayed-message-exchange plugin[12] enables message delays in exchanges, which could be useful for time-based PPMs. RabbitMQ's message queuing system contributes to the system's modularity, making it easier to scale, maintain, and extend individual components without disrupting overall operations. This flexibility is essential for accommodating future growth and optimizing performance.

11: https://github.com/mosquito/aio-pika (last accessed 23.03.2025).

12: https://github.com/rabbitmq/rabbitmq-delayed-message-exchange (last accessed 23.03.2025).

Vue.js[13] is a JavaScript framework for building web applications, especially single page applications (SPAs). Vue.js enables the creation of reusable components, improving modularity and maintainability. In addition, extensions such as Vuex for state management, vue-router for navigation, and Vuetify for pre-built UI components (e. g., navigation bars, data tables) were installed to enhance functionality. Vue.js was chosen for this project because of the developer's previous experience with it, which allowed for quick implementation and efficient realization of ideas. Its modularity and flexibility made it a strong fit for the front-end requirements of the application. The framework was used for the `webapp-service-frontend`.

13: https://vuejs.org/ (last accessed 23.03.2025).

6.4 Rule Logic and State Management

Initially, the PPMs were applied based on rule logic only. To simplify the process, improve clarity, and incorporate feedback from the Explanym team, a state component was introduced to allow state management and support state-based rules.

Each state contains the following fields: State, Type, Allowed Previous States, Time Threshold, Trigger Count, Beacon IDs, Geofence IDs, and Priority.

- *State* is the name of the defined state.
- *Allowed Previous States* lists valid previous states for a transition, or "All" to allow transitions from any state.
- *Type* maps the state to a specific asset type. Each state can only be defined once per type.
- *Time Threshold* defines the minimum time interval required between two consecutive timestamps for the trigger count to be incremented.

- *Trigger Count* sets the number of valid timestamp intervals (according to the time threshold) required before a state transition is triggered.
- *Beacon IDs / Geofence IDs* specify the location-based triggers for state transitions. These IDs indicate presence in specific areas.
- *Priority* resolves conflicts when multiple states share the same beacon or geofence ID. States with higher priority take precedence. If priorities are equal, beacon-based states will override geofence-based states.

An example of how states are defined and configured in the front-end is shown in figure 6.8.

The combination of Time Threshold and Trigger Count prevents premature state transitions when a tag enters a location for a short period of time. For example, to ensure that an *In Stock / Ready* state is not assigned until an asset has actually been in the storage for ten minutes, a Time Threshold of five minutes and a Trigger Count of two can be configured. This means that the system must detect two time stamps that are at least five minutes apart –indicating sustained presence– before updating the state.

All state-related data is stored in a dedicated database (State DB), including tables for states, associated beacons and geofences, and state transitions.

Based on the state management, the system applies PPMs based on rules. These rules determine how the tag data is processed. Each rule has the following fields:

- *Rule Name:* A unique identifier for the rule. Nine predefined rules are currently available and are explained below.
- *Rule Type:* Rules can be defined at three levels –Global, Type, and Asset. Global rules apply to all tags, type rules apply to tags of a specific type, and asset rules apply to individual tags. During execution, global rules are applied first, followed by type rules and then asset rules.
- *Type:* When defining a type rule, this field specifies the tag type to which the rule applies (consistent with the type definitions used for states).
- *State:* Optionally, a rule can be associated with a state. In this case, the rule will only apply if the tag is of the specified type and is currently in the specified state.
- *Content:* This field contains the rule-specific configuration, such as thresholds, states, or beacon/geofence IDs. The structure of this content depends on the rule type and is described in detail in the following section.
- *Custom Description:* A custom text used to document the purpose or logic of the rule. This description is displayed in the front-end for transparency.
- *Priority:* Within each rule level (global, type, asset), rules are executed in descending order of priority.

An example of how rules are defined and configured in the front-end is shown in figure 6.7.

The first six rules (Allow Filter Rule, Deny Filter Rule, Allow Tag Filter Rule, Deny Tag Filter Rule, and Heartbeat Rule) can operate independently of the state manager. However, without the State Manager, state-based linking cannot be used. The last three rules (Track By State Allow Rule, Track By State Deny Rule, and Geofence Threshold Rule) require the State Manager and cannot work without it.

The following sections describe each rule in detail. The core logic of each rule is implemented in a `apply(tag)` function, which is shown along with code snippets that illustrate its behavior. The variable `self.content` contains rule-related information.

- The *Allow Filter Rule* filters out tag data from beacons that aren't actively allowed. For example, it can be used to track tags only in a specific part of the clinic. The allowed beacon IDs are stored in `self.content`. The implementation is shown in the listing 6.1.

  Listing 6.1: Pseudocode for Allow Filter Rule.

```
1 IF tag["beacon_id"]  NOT IN self.content:
2       RETURN tag
3 ELSE:
4       RETURN NULL
```

- The *Deny Filter Rule* filters out tag data from beacons that have been banned, such as those in sensitive locations where sensitive information could be inferred. The set of prohibited beacon IDs is stored in `self.content`. Unlike the Allow filter rule, the implementation differs at line1, where it checks if the beacon ID is in the set.

  Listing 6.2: Pseudocode for Deny Filter Rule.

```
1 IF tag["beacon_id"] IN self.content:
2       RETURN tag
3 ELSE:
4       RETURN NULL
```

- The *Allow Tag Filter Rule* recursively filters data from the tag data item. The set of keys to be filtered is stored in `self.content`. The listing 6.3 shows the function `filter_tag` called with `keys=self.content`, `tag=tag` and `inverse=TRUE`.

  In the lines 2 to 5, `priv_proxy` is added to the keys to keep, as the data may still be needed by other rules. The function `filter_dict(data)` is defined in lines 7 to 11. This function calls itself recursively for dictionary entries, checking if a key exists in the `keys` set. If it does, it keeps the key, otherwise it returns `NULL`.

  The lines 13 to 15 check if single elements of the tag schema (see 6.2) are present in `keys`. If not, they are set to `NULL`. Line 17 calls `filter_dict` to process `tag["attributes"]`, but only if it is a dictionary. The last step (lines 20 to 23) checks if there is any data left in the tag. If so, the tag is returned, otherwise `NULL` is returned.

Listing 6.3: Pseudocode for the filter_tag function.

```
FUNCTION filter_tag(keys: SET, tag: DICTIONARY, inverse: BOOLEAN = FALSE) ⟷
    -> DICTIONARY OR NULL:
      IF NOT inverse:
            ADD "priv_proxy" TO keys
      ELSE:
            REMOVE "priv_proxy" FROM keys (if it exists)

      FUNCTION filter_dict(data):
            IF data IS A DICTIONARY:
                  RETURN { key: filter_dict(value) FOR EACH (key, value⟷
                       ) IN data
            IF (key IN keys) == inverse }
                        RETURN data

      FOR EACH field IN ["id", "time", "beacon_id", "latitude", "longitude⟷
           ", "", "attributes"]:
            IF (field IN keys) ≠inverse:
                  tag[field] = NULL

      IF "attributes" IN tag AND tag["attributes"] IS A DICTIONARY:
            tag["attributes"] = filter_dict(tag["attributes"])

      IF ALL tag[field] IN [NULL, "", {}] FOR EACH field IN ["id", "time",⟷
           "beacon_id", "latitude", "longitude", "heartbeat", "⟷
           attributes"]:
            RETURN NULL

      RETURN tag
```

- The *Deny Tag Filter Rule* works similarly to the Allow Tag Filter Rule, but the `filter_tag` function (6.3) is called with `inverse=FALSE`, i. e. it removes all keys from the tag data that are present in `self.content`. For a detailed explanation of the filtering process, see the Allow Tag Filter Rule description.
- The *Track Filter Rule:* ensures that only tag data items explicitly marked for tracking are processed. If a tag is not marked, it will be ignored. The rule does not store any data in `self.content`. The implementation is simple: it checks if the `track` exists in the `priv_proxy` dictionary and if its value is `TRUE`. If both conditions are satisfied, the tag is returned; otherwise, `NULL` is returned.

Listing 6.4: Pseudocode for the Track Filter Rule.

```
IF "priv_proxy" IN tag:
      IF "track" IN tag["priv_proxy"] AND LOWERCASE(tag["priv_proxy"]["⟷
           track"]) = "true":
            RETURN tag
RETURN NULL
```

- The *Heartbeat Rule* automatically marks an asset for tracking when it sends multiple consecutive heartbeat signals. This requires the tag data item to support heartbeat transmission.

  This rule helps identify assets that haven't moved for a while, suggesting they may be out of use or lost. If a heartbeat is detected and tracking is triggered, the asset´ can be located. The threshold for triggering

tracking is defined by `self.content[0]`. The pseudo code for this rule is shown in 6.5.

The lines 1-8 check if the key `priv_proxy` exists in the tag data item. If not, the system checks if the tag data item is a heartbeat. If it is, the following attributes will be initialized: `heartbeat_counter` is set to one, and `heartbeat_last_updated` is assigned the tag's timestamp before being sent to the Context DB. If the tag is not a heartbeat, it will be returned without an update request.

The lines 10-21 handle cases where the key `priv_proxy` exists. If the `heartbeat_counter` is already in `attributes` and is below the threshold (`self.content[0]`), it is incremented and `heartbeat_last_updated` is updated. If `heartbeat_counter` does not exist, it is initialized to one and `heartbeat_last_updated` is set to the timestamp of the tag. If the threshold is exceeded and `heartbeat_counter` exists, `heartbeat_counter` and `heartbeat_last_updated` are removed and `track` is set to `TRUE`.

If the tag is not a heartbeat but contains `priv_proxy`, the attributes `heartbeat_counter` and `heartbeat_last_updated` are removed to reset the heartbeat values in the Context DB. This prevents `track` from being set to `TRUE` when the heartbeat sequence is interrupted, e. g., by movement.

The `update` function updates the tag state with new attributes (see line 7 and 25). It expects `tag`, the tag data item; `updated_attributes`, changed attribute values; `time_key`, the last update timestamp to avoid duplicate processing; and `offset`, the minimum timestamp difference for updates.

Finally, the tag is returned.

The Heartbeat Rule is only effective when used in conjunction with the Track Filter Rule, which checks whether the track attribute is set to `TRUE` or `FALSE` and filters tags accordingly. To ensure that affected tags are not filtered out prematurely, the Heartbeat Rule should always run before the Track Filter Rule.

Listing 6.5: Pseudocode for Heartbeat Rule.

```
1  IF "priv_proxy" NOT IN tag:
2         IF tag["heartbeat"]:
3                attributes = {
4                       "heartbeat_counter": 1,
5                       "heartbeat_last_updated": tag["time"]
6                }
7                AWAIT updater.update_tag(tag=tag,updated_attributes=⟷
                       attributes,time_key="heartbeat_last_updated", offset=60⟷
                       )
8         RETURN tag
9
10 attributes = tag["priv_proxy"]
11 IF tag["heartbeat"]:
12        IF "heartbeat_counter" IN attributes AND attributes["heartbeat\⟷
               _counter"] ≤INTEGER(self.content[0]):
13                attributes["heartbeat_counter"] += 1
14                attributes["heartbeat_last_updated"] = tag["time"]
15        ELSE IF "heartbeat_counter" NOT IN attributes:
16                attributes["heartbeat_counter"] = 1
```

```
17              attributes["heartbeat_last_updated"] = tag["time"]
18      ELSE:
19              REMOVE "heartbeat_counter" FROM attributes
20              REMOVE "heartbeat_last_updated" FROM attributes
21              tag["priv_proxy"]["track"] = TRUE
22 ELSE:
23      REMOVE "heartbeat_counter" FROM attributes
24      REMOVE "heartbeat_last_updated" FROM attributes
25 AWAIT updater.update_tag(tag=tag, updated_attributes=attributes,time_key="↩
     heartbeat_last_updated", offset=60)
26 RETURN tag
```

The *Track By State Allow Rule* can only be defined as a type-level rule. It specifies a list of states in `self.rule_content`. If a tag data item matches the specified type and is currently in one of the listed states, the track attribute is set to `TRUE`. This rule is intended to be used in combination with the Track Filter Rule to be effective. The corresponding pseudocode is shown in Listing 6.6.

Listing 6.6: Pseudocode for Track By State Allow Rule.

```
1 IF "priv_proxy" IN tag:
2       IF "state" IN tag["priv_proxy"]:
3               IF tag["priv_proxy"]["state"] IN self.rule_content:
4                       tag["priv_proxy"]["track"] = TRUE
5 RETURN tag
```

- The *Track By State Deny Rule* is restricted to type-level use. It defines a list of states in `self.rule_content`. If a tag matches the specified type and is currently in one of these states, the track attribute is set to `FALSE`. Like its counterpart, this rule is only meaningful when used in combination with the Track Filter Rule. The corresponding pseudocode is shown in Listing 6.7.

Listing 6.7: Pseudocode for Track By State Deny Rule.

```
1
2 IF "priv_proxy" IN tag:
3       IF "state" IN tag["priv_proxy"]:
4               IF tag["priv_proxy"]["state"] IN self.rule_content:
5                       tag["priv_proxy"]["track"] = FALSE
6 RETURN tag
```

- The final rule is the *Geofence Threshold Rule*, which, like the previous two rules, is restricted to use at the type level. The threshold defined in `self.rule_content` specifies how many other active tags must be in the same geofence. Geofence information is provided by `tag["priv_proxy_geofence_ids"]` (see line 1), with the tag data item enriched by two components: the Enricher provides the number of tags per geofence, while the State Manager sets `active` to `TRUE` if the tag's current state is based on that geofence. A tag can be inside a geofence, but `active` is only `TRUE` if the tag's state is actually determined by it.

  If a geofence is active and the number of tags within it does not exceed the threshold, the `track` attribute is set to `FALSE`. This rule is only effective in combination with the Track Filter Rule, which will filter out the tag accordingly.

Listing 6.8: Pseudocode for Geofence Threshold Rule.

```
geofences = tag["priv_proxy_geofence_ids"] OR EMPTY LIST
threshold = self.content[0]

FOR EACH geofence IN geofences:
       IF geofence["active"] IS TRUE AND geofence["count"] ≤ threshold:
              tag["priv_proxy"]["track"] = FALSE
              BREAK
RETURN tag
```

The interaction between rules and states defines the PPMs and how they are applied within PrivProxy. The rules are designed to be easily extensible, ensuring scalability as new privacy needs arise. The flexibility of the system allows for continuous improvement, with potential for various different applications. Section 6.6 illustrates the use of PrivProxy with the VAC pump use case, showing how it adapts to specific privacy requirements in practice.

6.5 Web Interface and User Interaction

The web application consists of a back-end and a front-end that together allow administrative users to manage PrivProxy rules and states, while allowing regular users to view the rules applied to their tracked assets. Originally designed to display tracking information, the front-end was later extended to include rules, state, and context management –an intuitive extension given its existing access to the Rule DB and Context DB.

When the back-end is launched, an Admin user is automatically created, with full administrative access to the web application. Regular users are limited to the Tag View, where they can enter a tag ID to view the tracking details of the associated asset. This view displays the rules applied to the tag, listed in the order of execution (see Figure 6.3 for the landing page, and Figure 6.4 for the tag overview after entering the tag ID).

Administrators have additional access to the Context DB, where they can view and modify a tag's key-value pairs. The Admin View allows editing of the tag's attributes (see Figure 6.6 for the Admin landing page and figure 6.5 for the Admin section of the Tag Overview).

The Rule View and State View allow administrators to manage the rules and states in the system. These views consist of tables where new entries can be created and existing entries can be edited or deleted. The Rule View is shown in Figure 6.7 and the State View is shown in figure 6.8.

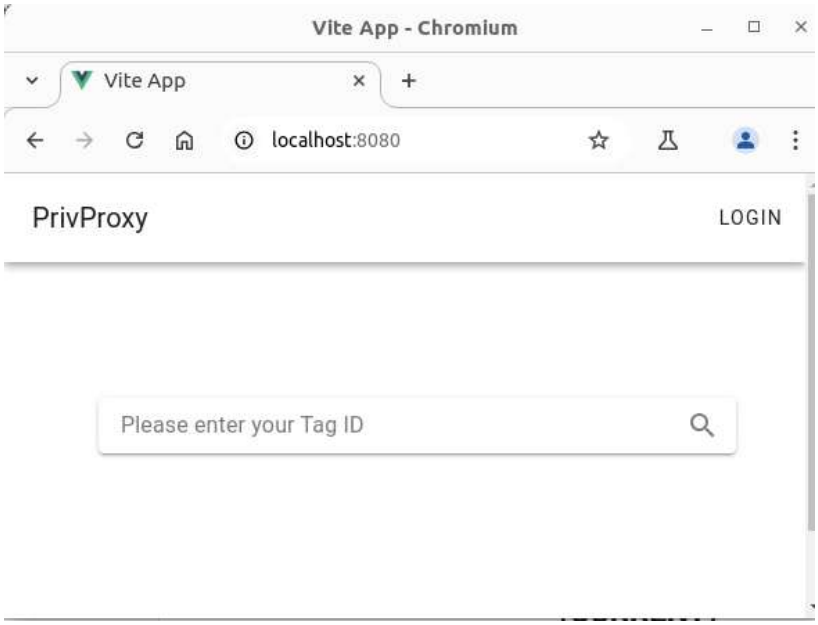


Figure 6.3: Landing Page (Tag Overview Page).

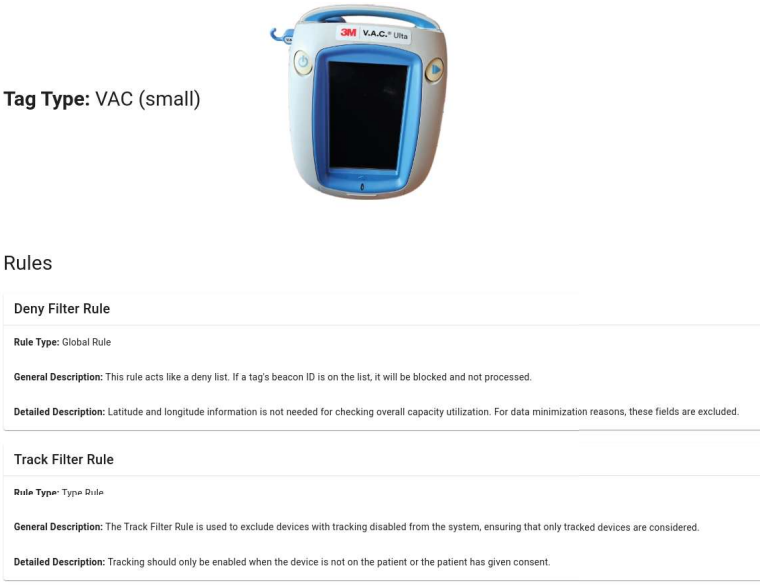
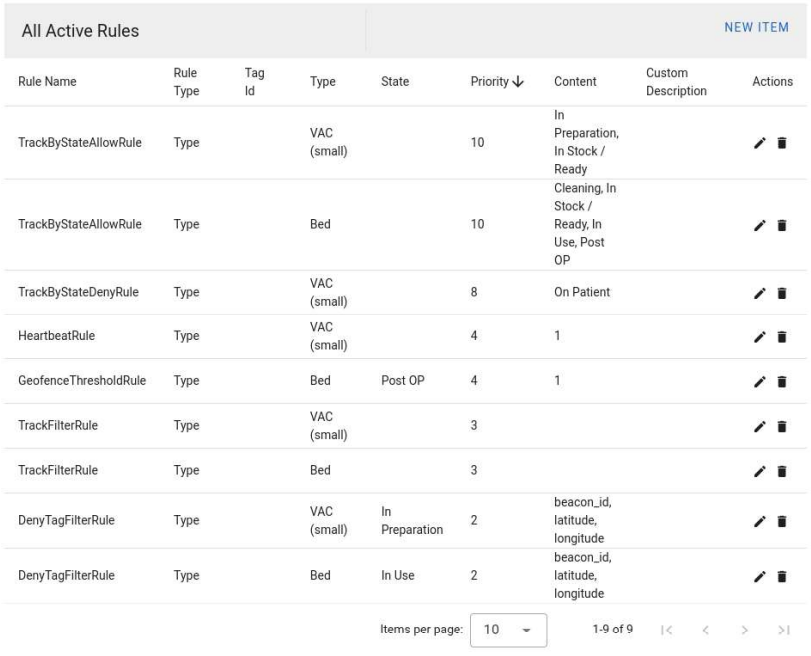


Figure 6.4: Tag overview page after entering the tag ID.



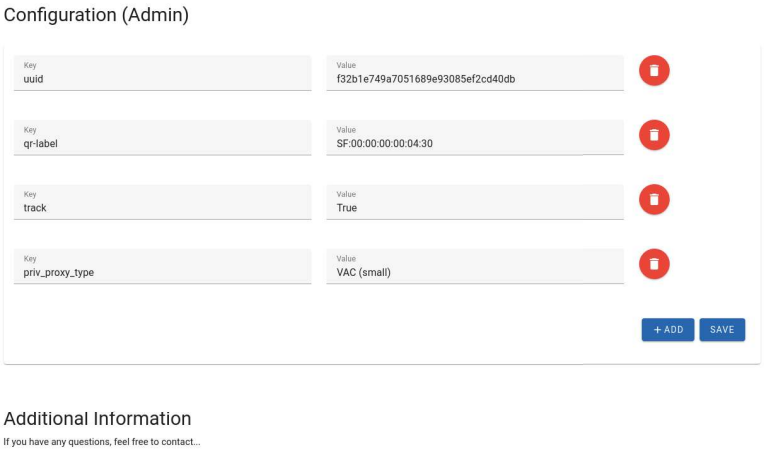


Figure 6.5: Admin section for editing tag attributes.

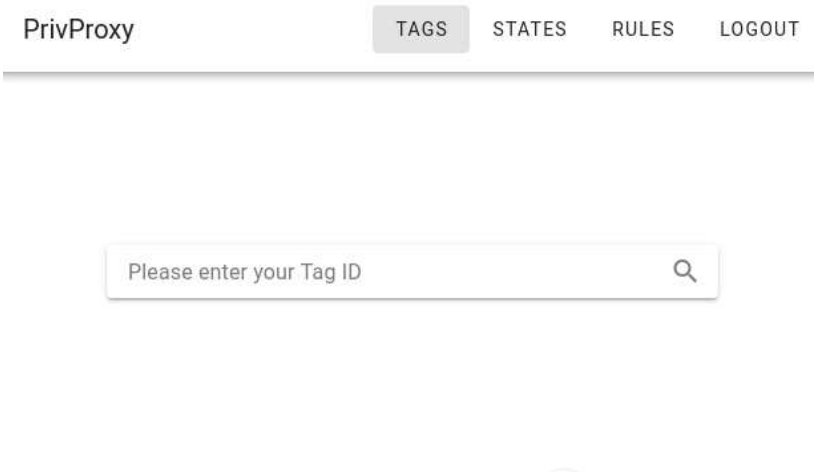


Figure 6.6: Access to the rules and state management page is restricted to Admin users only.

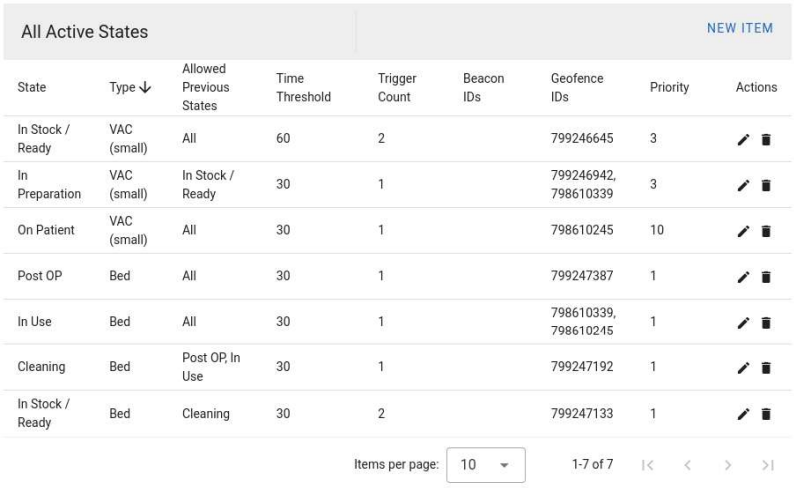


Figure 6.8: State administration page.

For more details on the technical implementation, refer to section 6.3.

6.6 Applying the Implementation: VAC Pump Tracking Scenario

The 3.4 section described the VAC pump use case. This section explains how PrivProxy was used to implement PPM in the VAC pump tracking scenario. Based on Müller and Halbritter's process analysis [MH24], insights from bi-weekly Explanym meetings, and the intended tracking goals –such as locating lost VAC pumps and monitoring inventory levels– a status-based tracking approach was developed.

The VAC pump should be trackable while in storage to verify availability. Tracking continues when the pump is removed from storage and is en route to the patient, but limited tracking information is provided during this time. Tracking stops when the pump reaches the patient and does not resume until the pump is returned to storage. To reduce the risk of lost assets, a location update is sent if the pump is stationary for an extended period of time. This does not restore full tracking.

Four states have been defined in PrivProxy to represent the life cycle of the VAC pump:

- *Unknown:* Initial undefined state. Once the VAC pump leaves this state, it cannot return.
- *Ready / On Stock:* Assigned when the pump is in storage and a certain number of detections occur within a defined time window, avoiding false positives from transient movement.
- *In Preparation:* This state can only be entered from the Ready / On Stock state when the pump is moved for use (e. g., during transport). If the VAC pump is later moved from the patient area to a preparation area, it will remain in the On Patient state.
- *On Patient:* This state is activated when the VAC pump enters the patient area. Once in this state, the VAC pump will not be tracked again until it returns to storage.

Figure 6.9 illustrates the states in PrivProxy. Each state requires presence in a specific area (e. g., inPatientArea). These areas are determined based on the system's knowledge of the VAC pump's location, derived either from beacon information or from previously defined s. *The Ready / On Stock* state is only triggered when a specified number of messages are received within a specified time window. The counter will only be incremented if the minimum time between messages has elapsed. If the VAC pump leaves the storage area before the threshold is reached, the counter is reset.

The mapping between Müller and Halbritter process states [MH24] and PrivProxy states is shown in figure 6.10.

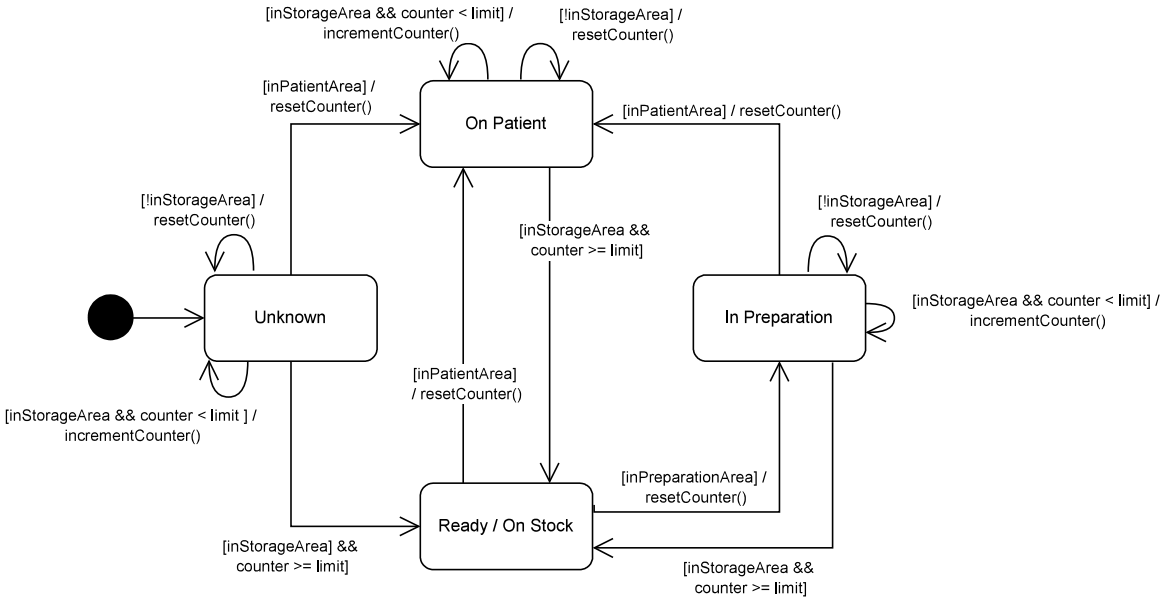


Figure 6.9: State Diagram for VAC pumps use case.

In combination with the previously defined states, the following rules enforce the desired tracking behavior, listed in descending order of priority (the first rule is executed first).

- *Track By State Allow Rule:* This rule enables tracking when the VAC pump is in the *In Preparation* or *Ready/On-Stock* state by setting the tracking flag to True.
- *Track By State Deny Rule:* This rule disables tracking when the VAC pump is in the *On Patient* state and sets the tracking flag to False.
- *Heartbeat Rule:* This rule requires two consecutive heartbeats to trigger tracking. When this threshold is met, the tracking flag is set to True, enabling tracking for the pump.
- *Track Filter Rule:* This rule checks the tracking flag in the tag's attributes. If the flag is True, the tracking data is passed; otherwise, it is discarded.
- *Deny Tag Filter Rule:* This rule filters out the beacon_id, latitude, and longitude fields in the tag if it is in the *In Preparation* state to prevent the exact location of the pump from being determined.

All rules are defined as type rules, such as *VAC (small)* in the PrivProxy system. The Deny Tag Filter Rule must also be assigned to the *In Preparation* state.

These states and rules ensure that the VAC pump is tracked only when necessary, protecting sensitive information while still allowing the necessary monitoring of its availability and location.

6.7 Challenges During Implementation

One of the first challenges in developing PrivProxy was determining how to retrieve and store contextual information for rule processing. While access to Track's REST API was available, access to the HIS was prohibited for privacy reasons, and access to the HPM® API was not provided. This left only two viable options: a local implementation or integration with Track.

To support testing and integrate data from the first field trial , a database dump from Track was requested but not provided. As a workaround, the `route-request-service` was developed to retrieve historical tag data. While this service is faster than the WebSocket-based approach, it may produce different rule evaluation results. Future iterations should aim to make its behavior more similar to the data flow of the WebSocket-based approach.

A major architectural change occurred in the second iteration with the introduction of the State Manager. The original version of PrivProxy operated on a purely rule-based model. To support state tracking, the rule data model had to be restructured to include state mapping. This change also required extended capabilities of the Context DB, as new rules depended not only on tag attributes, but also on geofence data. Since geofence access was only available in the track-based Context DB, it was chosen for the state handling proof of concept. A local version is still possible, but would require additional development.

Designing effective rules and states added another layer of complexity. Tools such as state diagrams (e.g., figure 6.9) helped identify logical inconsistencies early on. Given that rules and states are manually defined, a comprehensive understanding of their interactions and interdependencies is crucial –misconfigured priorities can disrupt tracking mechanisms and compromise privacy objectives. Rigorous testing is therefore essential to ensure correct implementation.

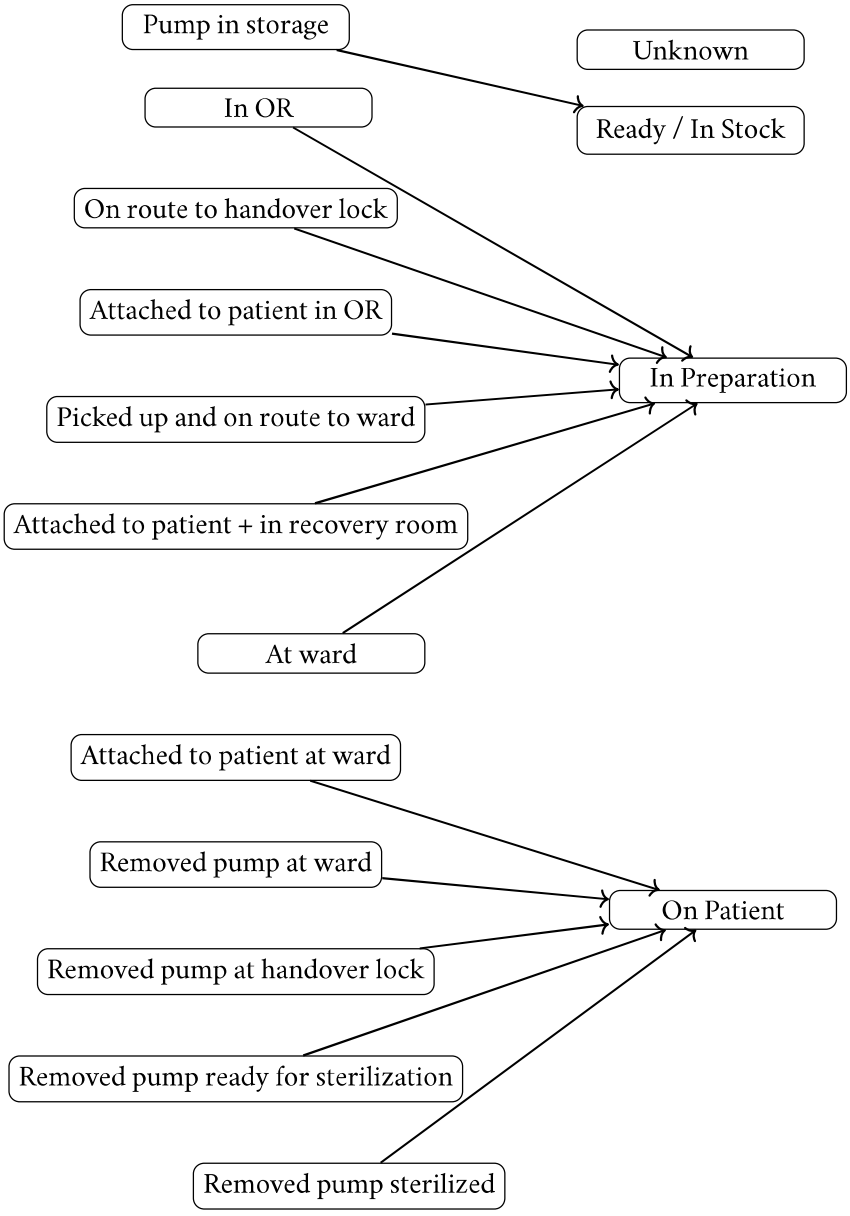


Figure 6.10: Mapping of Müller's states [MH24] (left) to the corresponding Priv Proxy states (right).

7 Evaluation

This section evaluates the results of this work. It first outlines the evaluation setup that provides the data for the privacy and utility metrics. The performance metrics are evaluated from a system-level perspective. The fulfilment of each requirement is then examined, followed by a discussion of the threats not directly addressed by the requirements. The chapter ends with a discussion.

7.1 Evaluation Setup

To generate test data for evaluation, the test scenario from section 6.6 was implemented. To simulate two parallel use cases, an additional hypothetical scenario involving beds was created for testing purposes only. The application was tested both in a controlled setup using three tags and three beacons, and in a real hospital environment using the existing tracking infrastructure. The following evaluation is based on the hospital data.

In the hospital environment a total of ten tags were used: seven representing VAC pumps and three representing beds. One of the VAC pump tags was configured to send a heartbeat every minute to test the Heartbeat Rule during this session. To ensure that the test focused on relevant data, temporary code was added to filter only the tags of the specified type, ignoring all other tag data items. Most of the time, the tags were transported together.

Geofences have been created for both use cases, including *Corridor 9* and *Patient Area 9*. For the VAC pumps, the geofences *VAC Prep (Dummy)* and *VAC Storage (Dummy)*. For the beds, *Bed Cleaning (Dummy)*, *Bed Storage (Dummy)*, and *Wake-Up Room (Dummy)* geofences have been created. The term dummy indicates that these geofences do not correspond to actual hospital rooms; other accessible rooms were used for testing due to restricted access to certain areas, such as the operating room.

As detailed in section 6.6, the VAC pump use case supports tracking within the storage areas and provides unprecise tracking during transport to the patient. Unprecise tracking discloses only the geofence name, while hiding detailed location data such as beacon ID, latitude and longitude. Upon entering the patient area, tracking is completely disabled and remains inactive until the pump is detected in the storage area for a

specified period of time –60 seconds in the simulated scenario. In the test scenario, the pump was first located in the *VAC Storage (Dummy)* area, then moved to the *VAC Prep (Dummy)* area, followed by *Corridor 9* and then *Patient Area 9*. It then returned to the *VAC Prep (Dummy)* area before finally returning to the *VAC Storage (Dummy)* area.

The bed use case is designed to evaluate an additional tag type and is based on hypothetical assumptions rather than empirical data or stakeholder input. In this scenario, similarly to the VAC use case, unprecise tracking is enabled in *Patient Area 9*, while standard tracking applies elsewhere. A transition to the *In Stock / Ready* state is only allowed if the bed tag was previously in the *Cleaning* state. For the *Post-OP* state, data transmission is only allowed if at least two bed tags are present in the area at the same time. During testing, the tags were transported alongside VAC pump tags from the previous scenario, with the geofences defined differently according to the bed use case. The bed tags were initially located in the *Bed Cleaning (Dummy)* area, moved to the *Bed Storage (Dummy)* area, passed through *Corridor 9* and entered *Patient Area 9*. It was then moved to the *Wake up room (Dummy)*, returned to the *Bed Cleaning (Dummy)* area and *Bed Storage (Dummy)* area, and finally a single bed tag was moved back to the *Wake up room (Dummy)* to test the Geofence Threshold Rule.

During the hospital test, data was collected from a live PrivProxy session where rules and states were applied and the output was saved to a file. All data was sent to the Track back-end as usual, as PrivProxy handled data after it had already been processed by Track. This setup allows a direct comparison between data processed with and without PPMs.

The comparison focuses on data collected by Track versus the live output from PrivProxy. figure 7.1 shows message traffic on the test day. Tests were conducted between 12:00 UTC and 13:35 UTC. The data generated earlier in the day resulted from preliminary testing in the small testbed. However, the analysis in section 7.2 considers only the period from 12:00 to 13:35.

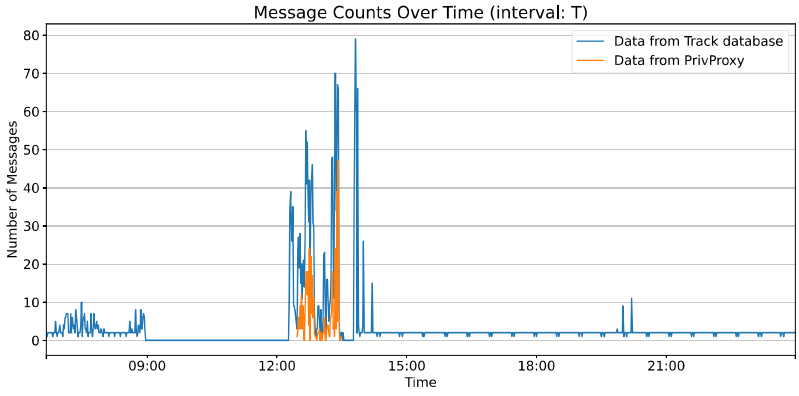


Figure 7.1: Message volume comparison between Track and PrivProxy.

7.2 Results

During the evaluation period, PrivProxy generated 765 messages. A time filter was applied to exclude outdated data caused by Track resending all positions upon reconnection, reducing the count to 474. After removing 45 duplicates, 429 relevant messages remained. Both the time filtering and duplicate removal were performed manually. In comparison, Track recorded 1,570 unique messages during the same timeframe highlighting the effectiveness of PrivProxy's rule-based filtering. For each use case, two diagrams were created: one showing data without applying PPMs (from 12:00:00 to 13:35:00 on the test day) and one showing data with PPMs applied.

The VAC use case results are shown in the bar chart 7.2 (with PPMs) and bar chart 7.3 (without PPMs). Without PPMs, 1,059 tag data items were recorded, with PPMs only 294. This reduction primarily results from filtering tag data items after they enter the patient area. Although tracking is generally allowed in the *VAC Storage (Dummy)* area, the PrivProxy configuration requires two consecutive messages from this area, separated by at least 60 seconds, before a tag is moved to the *In Stock / Ready* state. This configuration contributes to the observed difference in the number of tag data items. Tag data items were classified as either precise or unprecise. Unprecise items had location details –latitude, longitude, and beacon ID– removed, preventing exact positioning. In contrast, precise items included complete location data. The *No Geofence* category contains tag data for which no geofence could be assigned to the associated beacon. This occurred when tags were transmitting signals near beacons not in any geofence or when the beacon ID was 0, indicating no valid beacon. In the VAC use case, precise location data should only be transmitted from the *VAC Storage (Dummy) area*. However, 23 precise items appeared in the *No Geofence* category, and 6 in *Patient Area 9*. On verification, all 23 items in *No Geofence* had beacon ID 0, but their latitude and longitude matched the coordinates of the *VAC Storage (Dummy) area* –so acceptable. The 6 items *in Patient Area 9* were heartbeat messages from a tag configured to send one every minute. Their presence is valid, as the Heartbeat Rule was active for this use case.

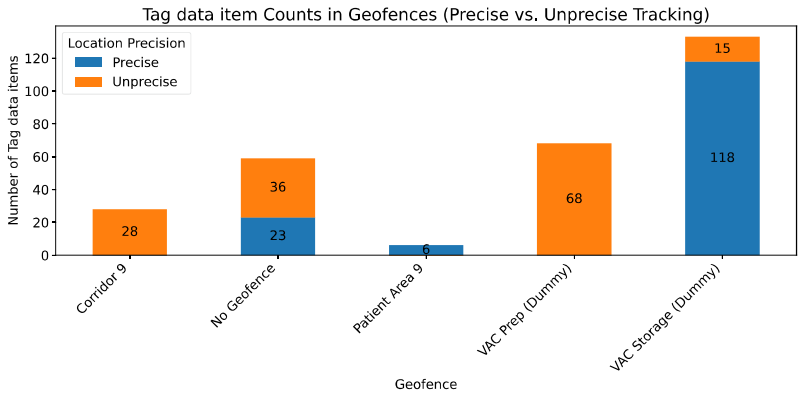


Figure 7.2: VAC Usecase with PPMs.

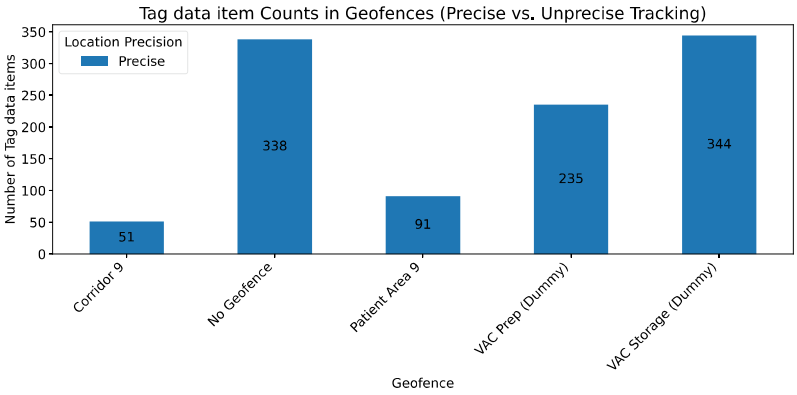


Figure 7.3: VAC Usecase without PPMs.

The bed use case results are shown in bar chart 7.4 (with PPMs) and bar chart 7.5 (without PPMs). Without PPMs, 511 tag data items were recorded; with PPMs, 135. The main difference comes from the fact that the test was started in the *Bed Cleaning (Dummy)* and *Bed Storage (Dummy)* areas, but to enter the *Cleaning* state, the tag data item has to be in the *Post OP* or *In Use* state before, which it wasn't. It had no state. So tracking started when *Patient Area 9* was visited, and the initial state could be set to *In Use*, which allowed any state to be the previous state. The intention was to track the beds from the beginning, it was not considered that entering the test scenario without an initial state would not be possible in the starting area. In contrast to the VAC pump use case, where tracking wasn't allowed when the patient area was visited, imprecise tracking was allowed in *Patient Area 9*. An additional test was the *Wake Up room (Dummy)*, which triggers the *Post OP* state. In the *Post OP* state, tracking is only allowed if there is at least one other tag of the same type (bed). The *Wake up room (Dummy)* was entered twice –first with all three bed tags, and later with only one. During the second entry, three tag events detected by the tracking system were not recorded by the PPMs due to the Geofence Threshold Rule, which requires at least two bed tags to be present in the *Wake up room (Dummy)* area for data transmission to occur.

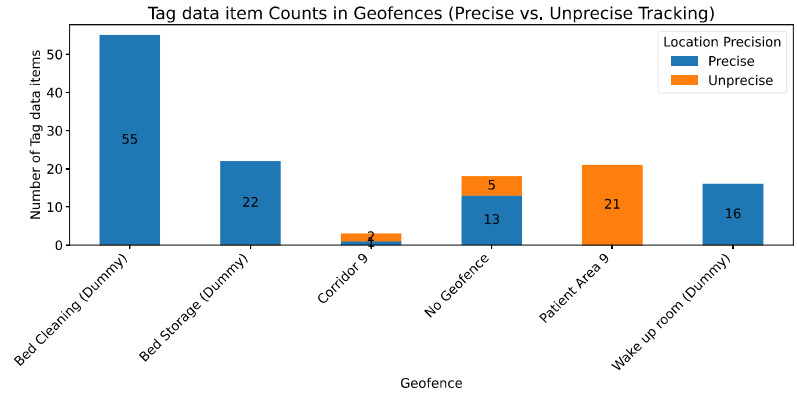


Figure 7.4: Bed Usecase with PPMs.

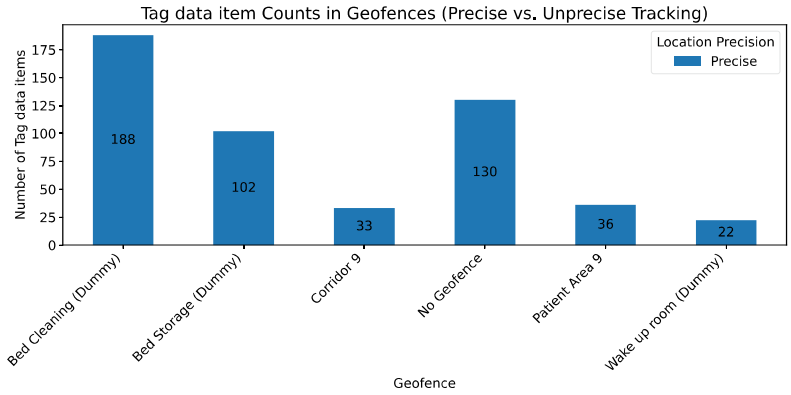


Figure 7.5: Bed Usecase without PPMs.

The results from the VAC and bed use cases demonstrate that PrivProxy's rule-based filtering not only reduces the number of transmitted tag data items but also minimizes the transmission of potentially user-identifying location information, thereby preserving privacy. Next, the performance characteristics of PrivProxy are evaluated. As discussed by Primault et al. [Pri+19], performance evaluations of LPPMs typically consider execution time, energy overhead, scalability, and communication overhead. In this work, execution time and energy overhead are not considered.

Execution time was excluded because it is largely determined by the latency between PrivProxy and the Track system, which depends on the external infrastructure. In addition, the development environment used in this project does not represent a realistic deployment scenario, making meaningful measurement of execution time impractical. Energy overhead is also not considered, as it is primarily relevant for LPPMs affecting mobile or battery-powered devices, where additional processing can lead to faster battery drain. PrivProxy, on the other hand, runs on stationary infrastructure and does not affect the power consumption of mobile devices. Scalability is covered in the relevant requirements section (section 7.3). Communication cost is the additional workload on the network caused by the use of LPPM. Depending on the rules configured, PrivProxy can reduce the number of messages sent. In the scenario evaluated, 765 messages (including duplicates) were sent with PPMs enabled, compared to 1,570 messages without. However, within PrivProxy, each tag message is passed internally through four components: from the parser to the enricher, from the enricher to the state manager, and from the state manager to the rule processor before the final message is sent. This internal message flow introduces additional communication overhead and can affect processing time. In addition to internal messaging, the PrivProxy can make up to four HTTP requests per tag, depending on the configuration. The Enricher can send two API requests: one to retrieve contextual information about the tag, and another to determine if the tag is currently inside a geofence. If the tag's state changes, the state manager sends an API request to update the state. In the case of heartbeat messages, the rule processor can also send an update request. In the scenario evaluated, the system did not introduce significant overhead, and the additional processing time was negligible. Nevertheless, the impact

of communication overhead should be re-evaluated in future test cases that more closely reflect real-world conditions.

To evaluate privacy and utility, the metric *Reduction in Observable Features (ROF)* was chosen. Although originally developed for load-hiding algorithms in smart metering applications [WE18], it can be meaningfully adapted to location data.

ROF quantifies how many observable events are hidden by a PPM. In this context, the events form a time series $\vec{T}$, and the feature mass $F(\vec{T})$ is a summary value, such as the number of events with a particular property.

It is defined:

- $F(\vec{T_Y})$: number of events with unprecise or no location data (i. e., filtered events)
- $F(\vec{T_{X^*}})$: number of events with full location data

The ROF metric is computed as:

$$priv_{ROF} \equiv \frac{F(\vec{T_Y})}{F(\vec{T_X^*})}$$

For the VAC use case:

- Total events: 1,059
- Events sent: 294
- Of the 294, only 147 included precise location data

Thus:

$$F(\vec{T_Y}) = 1{,}059 - 147 = 912$$
$$F(\vec{T_{X^*}}) = 147$$
$$priv_{ROF} = \frac{912}{147} \approx 6.20$$

For the bed use case:

- Total events: 511
- Events sent: 135
- Of the 135, only 107 included precise location data

Thus:

$$F(\vec{T_Y}) = 511 - 107 = 404$$
$$F(\vec{T_{X^*}}) = 107$$
$$priv_{ROF} = \frac{511}{107} \approx 4.78$$

These results illustrate the effectiveness of PrivProxy's rule-based filtering in reducing location data exposure across use cases.

7.3 Addressing Requirements and Threats

The functional requirements primarily assess whether PrivProxy delivers the intended functionality.

The first functional requirement, *FR* #1 – Apply PPMs to incoming data, is not fulfilled. To fulfill the requirement, the system should process data before it entered the track infrastructure via a tag producer and parser. Due to a lack of access, this could not be implemented in the current version of PrivProxy. However, the modular architecture has been designed with this requirement in mind and can be extended to support this use case once access is granted.

The second functional requirement, *FR* #2 – Apply location PPMs to outgoing data, is fulfilled. The system is correctly placed in the data pipeline, as incoming data is already enriched from Track on arrival. Although the configuration has not yet placed the PrivProxy between Safectory and Clinaris, this could be achieved by implementing a Tag Consumer accordingly.

The third functional requirement, *FR* #3 – Support for applying PPMs, is fulfilled. The primary approach is state-based tracking, which enables precise, imprecise, or no tracking depending on the tag's state. Additionally, the Geofence Threshold Rule allows for cloaking-based approaches.

The fourth functional requirement, *FR* #4 – Include Context Information for PPMs, is fulfilled. Contextual information, both from the tag data and external sources, is used to determine the application of PPMs. This functionality has been demonstrated in previous test cases and verified in smaller test settings.

The fifth functional requirement, *FR* #5 – Ensure anonymity in User Requests, is fulfilled. No cookies are set for non-admin users, as verified by the web development tool. The application does not access external web sites. In a production environment, the application logs at both the front-end and back-end must be manually checked to ensure that no user information is being stored.

The evaluation now shifts to non-functional requirements, which focus on system qualities rather than specific capabilities.

The first non-functional requirement, NFR #1 – Scalability, could not be evaluated. PrivProxy was designed with scalability in mind, allowing for the addition of more EPAs and asynchronous request handling to manage increased workloads. However, the system has not been tested under high load or in long-term production environments. Due to the lack of such tests, it cannot be confirmed that the system can reliably scale in practice.

The second non-functional requirement, NFR #2 – Anonymization Transparency, could not be evaluated. While a front-end has been created to provide users with information about individual rules for the PPMs, it has not yet been tested by users for clarity and completeness. However, the necessary components to deliver this information are in place.

The third non-functional requirement NFR #3 – Integration with existing asset tracking systems is fulfilled. In its current implementation, PrivProxy only works with Safectory's asset tracking system (Track), but the design of PrivProxy would allow it to be easily extended to work with other asset tracking systems.

The fourth non-functional requirement, NFR #4 – Support for Diverse Use Cases, is fulfilled. As previously shown in the performed test scenarios, by defining different tag types and mapping them to different states or rules, the system can handle multiple use cases for PPMs simultaneously.

In addition to the requirements analysis, the following privacy threats are addressed individually, as they fall within the scope of this work, but are not directly mapped to specific requirements.

- Threat #1 – Excessive Collection of Information by the Tracking System: This can be mitigated by filtering out specific attributes or entire tag data items.
- Threat #2 – Lack of Support for Data Subjects' Rights: This can be addressed by adding relevant information to the front end of PrivProxy.
- Threat #3 – Risk of Re-Identification Through Additional Attributes: External context (e. g., nurse notes, VAC ID) can introduce additional risks. PrivProxy can mitigate, but not fully eliminate this risk.
- Threat #4 – Unintentional Employee Surveillance: Specific areas can be excluded for certain devices to prevent unintentional surveillance.
- Threat #5 – Identification via Movement Patterns: This depends on the implemented rule set and requires long-term testing with various attack types. A good starting point is data minimization—only collect necessary data.
- Threat #11 – Incomplete or Ineffective Anonymization: Anonymization was effective in earlier tests, but manual checks were required to verify that PPMs work as expected.

Most functional and non-functional requirements have been met. Some remain open, primarily due to the absence of production-level testing or because they lie outside the current scope. Nevertheless, PrivProxy was designed with these requirements in mind and is easily extendable. It offers a solid foundation for addressing privacy threats, although further evaluation in realistic scenarios and testing against various attack scenarios is necessary to ensure effectiveness.

7.4 Discussion

This thesis developed and evaluated PrivProxy, a rule-based LPPM system tailored for asset tracking in healthcare settings. The goal was to demonstrate that PPMs can be dynamically applied without significant loss of utility. Evaluation scenarios confirmed this to some extent, in particular demonstrating the context-aware application of rules.

PrivProxy was inspired by PrivKit, a toolkit designed to evaluate and apply PPMs to data types such as face and location [Cun+24]. PrivKit allows privacy and utility trade-offs to be evaluated through extensible adapters, including data, PPM, attack, and metric adapters. While PrivKit works offline with predefined PPMs, PrivProxy builds on this concept by enabling online, rule-based PPMs tailored for asset tracking systems. Its modular architecture supports scalability, allowing for future expansion across different systems and environments. Although it currently supports only one back-end, the design ensures flexibility and extensibility, laying the groundwork for broader applicability beyond its initial use case.

The interplay between rules and state transitions in PrivProxy was effective, with state diagrams playing a key role in modeling different states and identifying potential contradictions or logic errors early on. By visualizing system behavior, the interactions between states and potential conflicts became easier to track. As noted by Primault et al. [Pri+19], rule-based systems require careful management to avoid problems such as conflicting rules, improper execution order, and unintended side effects. These challenges highlight the need for thorough testing to ensure that system flexibility does not compromise stability, reliability, and confidentiality.

The development of PrivProxy was supported by bi-weekly meetings and existing documentation, which provided a solid foundation for defining requirements. Additionally, the availability of building blocks from prior work and fellow students proved valuable. The LINDDUN GO threat elicitation workshop played a key role in refining PrivProxys requirements by offering a shared understanding of the asset tracking system and its privacy threats. Due to certain limitations, not all requirements could be fulfilled, but they provide a solid foundation for future improvements.

The evaluation was conducted using data generated within the hospital, simulating real-world conditions but not fully capturing the complexities and scale of ongoing operations. While this data provided valuable insights, long-term deployments are necessary to fully evaluate the system's performance in a live environment. Continuous use will help determine how different PPMs behave over time and how their effectiveness evolves. In addition, the chosen metric $-priv_{ROF}-$ indicates whether filtering has occurred, but does not reveal the effectiveness of the PPMs used. Alternative metrics, such as maximum tracking time [WE18], could provide a more meaningful understanding of long-term privacy impacts and a clearer assessment of how well the system protects user privacy over extended periods of time. The implemented PPMs, as well as the current

implementation of PrivProxy in general, do not support formal privacy guarantees such as differential privacy, nor do they allow evaluation against privacy models such as k-anonymity. However, the use cases can restrict tracking to specific processes and locations, helping to reduce potential privacy risks.

PrivProxy's front-end also provides a means to explain rule-based decisions, increasing transparency and understanding of how privacy-preserving actions are applied. This explainability is critical to building trust in the system, especially in sensitive environments such as healthcare. It should be noted, however, that this feature has not yet been tested with real users. User feedback will be critical for refining and validating the effectiveness of the explanations in real-world scenarios.

The key privacy challenges in mobility data science identified by Mokbel et al. [Mok+24] include the lack of addressing privacy threat models, the trade-off between data utility and privacy, and the need for explainability in privacy-preserving systems. All of these challenges have been addressed in this work. First, privacy threat modeling was conducted through the LINDDUN GO workshop, ensuring that the potential privacy risks are considered. Next, the trade-off between privacy and data utility can be managed through PrivProxy, which allows rules to be dynamically applied to adapt to different use cases. Finally, explainability has been improved by developing a front-end interface that allows users to understand how privacy-preserving mechanisms are applied.

In conclusion, PrivProxy serves as a successful proof-of-concept implementation and provides a solid foundation for the development of future online PPMs. Future work should focus on deploying the system in a live environment, evaluating its long-term performance, and continuously refining its privacy guarantees in response to real-world operational insights.

8 Conclusion

Balancing privacy and utility is challenging in sensitive environments like hospitals, where diverse use cases and high demand for data-driven insights exist. One such use case is hospital asset tracking. Within the Explanym research context, the goal is to track VAC pumps while preserving privacy, leading to the central question of this thesis:

How can privacy be preserved for hospital mobility data from VAC pumps, taking into account contextual factors?

The LINDDUN GO privacy threat modeling workshop (see chapter 4) identified key privacy threats in hospital asset tracking. These threats were used to refine requirements (see chapter 5), which directly informed the development of PrivProxy. PrivProxy enables the application of PPMs in an online system and is designed for easy extension, allowing for adaptability as new requirements emerge.

PrivProxy has been tested in a hospital setting using two use cases (see chapter 7), demonstrating that the rule-based approach effectively reduces the amount of data sent and allows for imprecise location transmission. While the current implementation does not yet offer provable privacy guarantees, it lays important groundwork toward achieving them in future iterations. The results are discussed in detail in section 7.4.

This work contributes to privacy-preserving systems for mobile applications in sensitive environments. PrivProxy, as an open-source solution, could be used in future field tests, offering a scalable and adaptable approach for managing sensitive mobility data.

8.1 Limitations

Some requirements could not be evaluated or implemented due to access restrictions. The application has only been tested in a local development environment and has not yet been tested in production. Because performance can vary significantly in distributed systems due to factors such as network latency, load balancing, and scalability, comparing performance metrics between development and production environments is essential to ensure expected behavior in production.

PrivProxy's rule and state design is highly dependent on domain expertise. Variations in device behavior, environmental factors, and edge cases

introduce challenges that limit the system's generalizability and require context-specific adjustments.

Security has not been a primary focus of this work. Although section 4.6 outlines relevant security considerations, they have not yet been implemented or evaluated. This includes understanding what internal or external processes can trigger state changes that could lead to potential misuse or unintended effects.

Finally, PrivProxy is designed for asynchronous processing. Although rules are executed sequentially per tag, tag messages are processed asynchronously within PrivProxy. This lack of guaranteed message order can lead to race conditions –for example, if concurrent tag messages attempt to update the same context DB entry, resulting in potential data loss. Future releases should implement control mechanisms to mitigate this risk.

8.2 Future Work

Future work could focus on implementing and evaluating open requirements. The identified threats can be further explored for hospital asset tracking systems. An important direction could be to automate the verification of anonymization and privacy mechanisms that are currently not automatically monitored.

The LINDDUN GO methodology provided a strong foundation for identifying privacy threats. Future work could extend LINDDUN to evaluate the PrivProxy itself. However, addressing security concerns, particularly with respect to system processes that trigger state changes, is critical. The STRIDE methodology could help identify potential vulnerabilities.

Expanding the technological scope of PrivProxy is another possibility. Adding support for additional protocols, such as iBeacon, would increase system compatibility.

Front-end improvements, such as better input validation and clearer error communication, could reduce problems and improve the user experience. Enhancing the PrivProxy front-end with lifecycle or state visualization could help users better understand PPMs over time and across contexts, improving explainability.

Introducing a rule logic checker would prevent conflicts when creating rules or states (e. g., preventing conflicting rules). In addition, the flow of data through the system could be optimized by not requesting context information for tags that don't need the data to apply the mapped rules. Evaluating the system in a real-world context over an extended period of time and testing different attack scenarios would provide valuable insights.

Finally, PrivProxy could also be extended to process anonymized and non-anonymized data streams in parallel, similar to PrivKit [Cun+24], but with PrivProxy operating in an online context. This would allow direct

comparisons and better evaluation of privacy mechanisms. Metrics such as $priv_{ROF}$ could help assess the impact of PPMs and guide adjustments.

While this work demonstrates the feasibility and potential of contextual, explainable, PPMs in hospital asset tracking, several limitations remain. Full integration into live hospital systems, long-term evaluation under real-world conditions, and broader applicability beyond the specific use case of VAC pumps are essential next steps. Nevertheless, this work lays a practical foundation for promoting privacy in healthcare mobility data and contributes to the ongoing development of transparent and user-centric anonymization approaches.

References

[Abo+99] GD Abowd et al. Towards a Better Understanding of Context and Context-Awareness. In: *Handheld and Ubiquitous Computing*. Ed. by HW Gellersen. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 304–307 (pages 20, 21).

[Ack+23] L Ackermann et al. Towards Anonymizing Intermodal Mobility Data for Smart Cities. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geo-Privacy and Data Utility for Smart Societies*. GeoPrivacy '23. Hamburg, Germany: Association for Computing Machinery, 2023, pp. 24–27. https://doi.org/10.1145/3615889.3628514 (page 3).

[AHR24] ER Agustina, AR Hakim, and K Ramli. Modeling Data Security and Privacy Threats for VANET using STRIDE and LINDDUN. In: *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)*. 2024, pp. 114–119 (page 35).

[AAC16] U Alegre, JC Augusto, and T Clark. Engineering context-aware systems and applications: A survey. In: *Journal of Systems and Software* 117:(2016), 55–83. https://www.sciencedirect.com/science/article/pii/S0164121216000467 (page 20).

[Ban+19] A Banerjee et al. Indoor Positioning Knowledge Model for Privacy Preserving Context-Awareness. In: *Service Research and Innovation*. Ed. by HP Lam and S Mistry. Cham: Springer International Publishing, 2019, pp. 50–64 (page 21).

[Baz+21] R Bazo et al. A survey about real-time location systems in healthcare environments. In: *Journal of Medical Systems* 45:(2021), 1–13 (pages 2, 19).

[Bra+23] Z Braiterman et al. Threat Modeling Manifesto. https://web.archive.org/web/20231201002657/http://www.threatmodelingmanifesto.org/ (Last accesed 24.11.2024). 2023 (page 33).

[CMO21] KJ Campos-Cruz, C Mancillas-López, and B Ovilla-Martinez. A Lightweight Security Protocol for Beacons BLE. In: *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. 2021, pp. 1–6 (page 44).

[Cäs+22] M Cäsar et al. A survey on Bluetooth Low Energy security and privacy. In: *Comput. Netw.* 205(C):(Mar. 2022). https://doi.org/10.1016/j.comnet.2021.108712 (page 43).

[CM08] K Chaudhuri and C Monteleoni. Privacy-preserving logistic regression. In: *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. NIPS'08. Vancouver, British Columbia, Canada: Curran Associates Inc., 2008, pp. 289–296 (page 15).

[Coh22] A Cohen. Attacks on Deidentification's Defenses. In: *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, #aug# 2022, pp. 1469–1486. https://www.usenix.org/conference/usenixsecurity22/presentation/cohen (page 12).

[Cor+10] G Cormode et al. Minimizing minimality and maximizing utility: analyzing method-based attacks on anonymized data. In: *Proc. VLDB Endow.* 3(1–2):(#sep# 2010), 1045–1056. https://doi.org/10.14778/1920841.1920972 (page 12).

[CMV21] M Cunha, R Mendes, and JP Vilela. A survey of privacy-preserving mechanisms for heterogeneous data types. In: *Computer Science Review* 41:(2021), 100403. https://www.sciencedirect.com/science/article/pii/S1574013721000435 (pages 8, 9).

[Cun+24] M Cunha et al. Privkit: A Toolkit of Privacy-Preserving Mechanisms for Heterogeneous Data Types. In: *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*. CODASPY '24. Porto, Portugal: Association for Computing Machinery, 2024, pp. 319–324. https://doi.org/10.1145/3626232.3653284 (pages 4, 79, 82).

[DAl+20] M D'Aloia et al. IoT Indoor Localization with AI Technique. In: *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*. 2020, pp. 654–658 (page 19).

[Den+11] M Deng et al. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. In: *Requirements Engineering* 16(1):(Mar. 2011), 3–32. https://doi.org/10.1007/s00766-010-0115-7 (pages 33–36).

[Die+24] K Dietz et al. The Missing Link in Network Intrusion Detection: Taking AI/ML Research Efforts to Users. In: *IEEE Access* 12:(2024), 79815–79837 (page 4).

[Dis24] DistriNet. Privacy Threat Trees. https://web.archive.org/web/20230626042655/https://linddun.org/threat-trees/. (Last accessed 07.01.2025). 2024 (page 36).

[Dwo06] C Dwork. Differential Privacy. In: *Automata, Languages and Programming*. Ed. by M Bugliesi et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12 (page 13).

[DR14] C Dwork and A Roth. The Algorithmic Foundations of Differential Privacy. In: *Found. Trends Theor. Comput. Sci.* 9(3–4):(Aug. 2014), 211–407. https://doi.org/10.1561/0400000042 (page 13).

[Ene+16] M Enev et al. Automobile driver fingerprinting. In: *Proceedings on Privacy Enhancing Technologies*:(2016) (page 1).

[EN10] O Etzion and P Niblett. *Event Processing in Action*. 1st. USA: Manning Publications Co., 2010 (page 22).

[EC16] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://data.europa.eu/eli/reg/2016/679/oj. (Last accessed 13.04.2025). 2016 (pages 8, 14).

[FKS16] K Fawaz, KH Kim, and KG Shin. Privacy vs. reward in indoor location-based services. In: *Proceedings on Privacy Enhancing Technologies*:(2016) (page 5).

[Gad+24] A Gadotti et al. Anonymization: The imperfect science of using data while preserving privacy. In: *Science Advances* 10(29):(2024), eadn7053. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.adn7053. https://www.science.org/doi/abs/10.1126/sciadv.adn7053 (pages 8, 9, 13–15, 18).

[GG24] M Gecer and B Garbinato. Federated Learning for Mobility Applications. In: *ACM Comput. Surv.* 56(5):(#jan# 2024). https://doi.org/10.1145/3637868 (page 18).

[Has+18] SS Hassan et al. Security threats in Bluetooth technology. In: *Computers & Security* 74:(2018), 308–322. https://www.sciencedirect.com/science/article/pii/S0167404817300615 (pages 26, 43).

[ISO19] ISO/IEC. *ISO/IEC TR 27550:2019(en) Information technology — Security techniques — Privacy engineering for system life cycle processes*. Technical Report. ISO/IEC, 2019 (page 35).

[Jär+19] K Järvinen et al. PILOT: Practical Privacy-Preserving Indoor Localization Using OuTsourcing. In: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2019, pp. 448–463 (page 5).

[Jin+23] F Jin et al. A Survey and Experimental Study on Privacy-Preserving Trajectory Data Publishing. In: *IEEE Transactions on Knowledge and Data Engineering* 35(6):(2023), 5577–5596 (page 4).

[Kar+19] H Karvonen et al. Experimental Performance Evaluation of BLE 4 Versus BLE 5 in Indoors and Outdoors Scenarios. In: *Advances in Body Area Networks I*. Ed. by G Fortino and Z Wang. Cham: Springer International Publishing, 2019, pp. 235–251 (page 19).

[KL22] EJ Kim and EK Lee. Performance Impact of Differential Privacy on Federated Learning in Vehicular Networks. In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. 2022, pp. 1–5 (page 18).

[Kim+15] S Kim et al. Indoor positioning system techniques and security. In: *2015 Forth International Conference on e-Technologies and Networks for Development (ICeND)*. 2015, pp. 1–4 (page 19).

[KS14] M Kohne and J Sieck. Location-Based Services with iBeacon Technology. In: *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*. 2014, pp. 315–321 (pages 27, 28).

[Kom+16] K Komai et al. Elderly person monitoring in day care center using Bluetooth Low Energy. In: *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*. 2016, pp. 1–5 (page 19).

[Kon+15] A Konstantinidis et al. Privacy-Preserving Indoor Localization on Smartphones. In: *IEEE*

Transactions on Knowledge and Data Engineering 27(11):(2015), 3042–3055 (page 5).

[LLV07] N Li, T Li, and S Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In: *2007 IEEE 23rd International Conference on Data Engineering*. 2007, pp. 106–115 (pages 11, 12).

[Mac+06] A Machanavajjhala et al. L-diversity: privacy beyond k-anonymity. In: *22nd International Conference on Data Engineering (ICDE'06)*. 2006, pp. 24–24 (pages 10, 11).

[Mar+16] PD Martin et al. Applications of Secure Location Sensing in Healthcare. In: *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. BCB '16. Seattle, WA, USA: Association for Computing Machinery, 2016, pp. 58–67. https://doi.org/10.1145/2975167.2975173 (page 44).

[MK18] YS Martin and A Kung. Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering. In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2018, pp. 108–111 (page 35).

[MRC20] L Meftah, R Rouvoy, and I Chrisment. Capturing Privacy-Preserving User Contexts with IndoorHash. In: *Distributed Applications and Interoperable Systems*. Ed. by A Remke and V Schiavoni. Cham: Springer International Publishing, 2020, pp. 21–38 (page 21).

[Mok+24] M Mokbel et al. Mobility Data Science: Perspectives and Challenges. In: *ACM Trans. Spatial Algorithms Syst.* 10(2):(July 2024). https://doi.org/10.1145/3652158 (pages 2, 80).

[Mül+22] KOE Müller et al. An Overview and Ontology of Privacy to Preserve Privacy in Ultra-Wideband Networks. In: *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*. 2022, pp. 2317–2324 (page 19).

[MH24] R Müller and J Halbritter. Identifying Data-Induced Risks in Hospital Asset Tracking. MA thesis. University of Bamberg, 2024 (pages 4, 29, 30, 66, 69).

[Mül20] T Müller. Crowd Beacons: Extending Static Beacons With Anonymous Dynamic Beacons for Context-Based Indoor Navigation. Bachelor's Thesis. University of Bamberg, 2020 (page 5).

[Nwe+22] LO Nweke et al. A LINDDUN-Based Privacy Threat Modelling for National Identification Systems. In: *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*. 2022, pp. 1–8 (page 35).

[Oze+23] R Ozeki et al. Balancing Privacy and Utility of Spatio-Temporal Data for Taxi-Demand Prediction. In: *2023 24th IEEE International Conference on Mobile Data Management (MDM)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2023, pp. 215–220. https://doi.ieeecomputersociety.org/10.1109/MDM58254.2023.00044 (page 5).

[Per+14] C Perera et al. Context Aware Computing for The Internet of Things: A Survey. In: *IEEE Communications Surveys & Tutorials* 16(1):(2014), 414–454 (page 20).

[Peš+18] S Pešić et al. Bluetooth Low Energy Microlocation Asset Tracking (BLEMAT) in a Context-Aware Fog Computing System. In: *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*. WIMS '18. Novi Sad, Serbia: Association for Computing Machinery, 2018. https://doi.org/10.1145/3227609.3227652 (pages 19, 21).

[Pri+19] V Primault et al. The Long Road to Computational Location Privacy: A Survey. In: *IEEE Communications Surveys & Tutorials* 21(3):(2019), 2772–2793 (pages 4, 9, 14, 16–18, 51, 75, 79).

[RBA17] F Rahman, MZA Bhuiyan, and SI Ahamed. A privacy preserving framework for RFID based healthcare systems. In: *Future Generation Computer Systems* 72:(2017), 339–352. https://www.sciencedirect.com/science/article/pii/S0167739X16301819 (page 19).

[RR12] S Robertson and J Robertson. *Mastering the Requirements Process: Getting Requirements Right*. 3rd. Addison-Wesley Professional, 2012 (pages 24, 36, 45, 46, 48).

[REE24] JLL Ruiz, ÁV Espinosa, and ME Estévez. Federated Learning Methodology for Indoor Location Systems in Multi-Layer Architecture. In: *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024, pp. 1–10 (page 19).

[SC23] Y Sartayeva and HCB Chan. A survey on indoor positioning security and privacy. In: *Comput. Secur.* 131(C):(Aug. 2023). https://doi.org/10.1016/j.cose.2023.103293 (pages 5, 43).

[Sho14] A Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014 (pages 33, 34, 37, 39).

[Shu+23] V Shubina et al. Acceptable Margin of Error: Quantifying Location Privacy in BLE Localization. In: *2023 International Conference on Localization and GNSS (ICL-GNSS)*. 2023, pp. 1–7 (page 19).

[SIG25] SIG. Bluetooth market update 2025. https://web.archive.org/web/20250429065628/https://www.bluetooth.com/2025-market-update/. (Last accessed 29.04.2025). 2025 (page 2).

[Sio+18] L Sion et al. Interaction-Based Privacy Threat Elicitation. In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2018, pp. 79–86 (pages 35–37).

[Swe02] L Sweeney. k-anonymity: a model for protecting privacy. In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10(5):(Oct. 2002), 557–570. https://doi.org/10.1142/S0218488502001648 (page 10).

[SPN15] ER Sykes, S Pentland, and S Nardi. Context-aware mobile apps using iBeacons: towards smarter interactions. In: *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering*. CASCON '15. Markham, Canada: IBM Corp., 2015, pp. 120–129 (page 21).

[Tog+08] R van der Togt et al. Electromagnetic interference from radio frequency identification inducing potentially hazardous incidents in critical care medical equipment. In: *JAMA* 299(24):(#jun# 2008), 2884–2890 (page 19).

[Van+16] T Van Haute et al. Performance analysis of multiple Indoor Positioning Systems in a healthcare environment. In: *International Journal of Health Geographics* 15(1):(Feb. 2016), 7. https://doi.org/10.1186/s12942-016-0034-z (page 19).

[Wac+24] T Waclawek et al. Crowd-counting technology within the Smart City context: understanding, trust, and acceptance. In: *Frontiers in Psychology* 15:(2024), 1423837 (page 4).

[WE18] I Wagner and D Eckhoff. Technical Privacy Metrics: A Systematic Survey. In: *ACM Comput. Surv.* 51(3):(#jun# 2018). https://doi.org/10.1145/3168389 (pages 15, 76, 79).

[War65] SL Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. In: *Journal of the American Statistical Association* 60(309):(1965), 63–69 (page 13).

[WBH19] C Winter, V Battis, and O Halvani. Herausforderungen für die Anonymisierung von Daten. In: *INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft*. Bonn: Gesellschaft für Informatik e.V., 2019, pp. 339–352 (page 14).

[WSJ20] K Wuyts, L Sion, and W Joosen. LINDDUN GO: A Lightweight Approach to Privacy Threat Modeling. In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2020, pp. 302–309 (pages 33, 35, 37, 38).

[Xie+18] L Xie et al. Differentially Private Generative Adversarial Network. In: *CoRR* abs/1802.06739:(2018). arXiv: 1802.06739. http://arxiv.org/abs/1802.06739 (page 15).

[XC09] T Xu and Y Cai. Feeling-based location privacy protection for location-based services. In: *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS '09. Chicago, Illinois, USA: Association for Computing Machinery, 2009, pp. 348–357. https://doi.org/10.1145/1653662.1653704 (page 5).

[Yu+21] D Yu et al. Gradient perturbation is underrated for differentially private convex optimization. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. IJCAI'20. Yokohama, Yokohama, Japan, 2021 (page 15).

[ZY24] X Zheng and Z Yanxiang. Advances in private training for production on-device language models. https://web.archive.org/web/20250420031012/https://research.google/blog/advances-in-private-training-for-production-on-device-language-models/. (Last accessed 26.04.2025). 2024 (page 14).

[ZTH18] A Zidek, S Tailor, and R Harle. Bellrock: Anonymous Proximity Beacons From Personal Devices. In: *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2018, pp. 1–10 (page 5).

[Zuo+21] Z Zuo et al. Data Anonymization for Pervasive Health Care: Systematic Literature Mapping Study. In: *JMIR Med Inform* 9(10):(Oct. 2021), e29871. https://medinform.jmir.org/2021/10/e29871 (page 4).

Disclosure

AI tools were used to improve the readability of this document.