Privacy by Design applied in Practice and the Consequences for System Developers

SARA ERVIK
Privacy by Design applied in Practice and the Consequences for System Developers

SARA ERVIK
Abstract

Providing privacy for users is an important matter, data is processed to an increasing extent including sensitive personal information. It is a liability for organizations to take responsibility for the privacy of their users. Organizations are required by law to handle personal information in accordance to General Data Protection Regulation (GDPR). But there is a gap between the legal requirements and the technical solutions. The framework Privacy by Design (PbD) presents guidelines to include privacy in a system but lacks concrete implementations. This paper investigates how PbD can be applied to a system and how it impacts the system development. The study adopts the approach of Colesky, Hoepman and Hillen to apply Privacy by Design in Practice. This was used to develop a system model with consideration of the privacy of users as well as functional requirements and the needs of system developers. The evaluation showed a positive attitude among system developers towards the proposed system model implementing PbD. The system developers estimated that the proposed system model would introduce a slight decrease in productivity but believed the positive aspects of applying privacy would outweigh the disadvantages.
Sammanfattning

# Contents

1 Introduction 1
   1.1 Research Question ........................................ 2
   1.2 Research Gap ............................................... 2
   1.3 Scope ...................................................... 3
   1.4 Outline .................................................... 4

2 Background 5
   2.1 Privacy ...................................................... 5
   2.2 Anonymity and Pseudonymity ................................. 5
   2.3 Privacy by Design .......................................... 6
   2.4 Privacy-Enhancing Technologies ............................ 7
         2.4.1 De-identification .................................. 7
         2.4.2 k-anonymity ....................................... 8
         2.4.3 Privacy-Preserving Aggregation .................... 9
   2.5 Role-Based Access Control ................................ 9
   2.6 General Data Protection Regulation ....................... 9
   2.7 Related work .............................................. 10

3 Methods 16
   3.1 Literature study ........................................... 16
   3.2 Prerequisites & Requirements ............................... 17
         3.2.1 System model ....................................... 17
         3.2.2 Data Survey and Interviews ....................... 18
         3.2.3 Daily need ........................................ 18
         3.2.4 Exceptions .......................................... 19
   3.3 Designing the new system model ........................... 20
   3.4 Evaluation ................................................ 21
         3.4.1 Comparison with other PbD frameworks ............ 21
         3.4.2 Impact on System Developers ..................... 21
3.4.3 User Privacy ........................................ 21

4 Applying PbD to the system .................................. 23
  4.1 Minimise ............................................. 24
  4.2 Abstract ............................................. 25
  4.3 Separate ............................................. 26
  4.4 Hide .................................................. 27
  4.5 Inform, Control, Enforce & Demonstrate ................. 28
  4.6 The proposed system model ................................ 29

5 Results .................................................. 32
  5.1 Comparison with other PbD frameworks .................. 32
  5.2 Impact on System Developers ............................ 33
    5.2.1 Productivity ..................................... 34
    5.2.2 Ease of Use ..................................... 35
  5.3 User Privacy .......................................... 37

6 Discussion ............................................... 39
  6.1 Ethical and societal considerations ..................... 39
  6.2 Critical Evaluation ................................... 40
    6.2.1 Technical Limitations ........................... 41
    6.2.2 Data Quality ................................... 42
  6.3 Future work ........................................... 43

7 Conclusions ............................................ 44

Bibliography ............................................. 45
Acronyms

**API**  application programming interface. 42

**ENISA**  European Union Agency for Network and Information Security. 13

**EU**  European Union. 1, 6, 9

**GDPR**  General Data Protection Regulation. 1, 4–6, 9, 12, 16, 24

**OECD**  Organisation for Economic Co-operation and Development. 3

**PbD**  Privacy by Design. 1–4, 6, 7, 10, 12, 16, 17, 20, 21, 23, 29, 32–36, 39, 43

**PET**  Privacy-Enhancing Technologies. 3, 7, 16, 39

**RBAC**  Role-Based Access Control. 9, 27, 29–31, 34, 35, 37, 41, 44

**UN**  United Nations. 39
Chapter 1

Introduction

Data is a powerful tool and could be an organization's greatest asset and advantage against competitors. But like any great power it comes with great responsibility. On the one hand, data is beneficial for understanding and developing better products. On the other hand, handling personal user data can intrude on the privacy of the individual.

The rapid development in Big data have resulted in a situation where organizations have access to and are trusted with large amount of personal user data. Historically, the individual user possessed few possibilities to exercise his or her right. One step towards a more equal power relationship is through legal regulations. In May 2018, the General Data Protection Regulation (GDPR) [1] was taken into force and this has strengthened the requirements on data handling for organizations with users from European Union (EU).

Efforts have been made to develop systems with privacy in mind. Privacy by Design (PbD) and related research areas are working towards providing technologies and methods to design privacy-preserving systems. But legal regulations for privacy alone are not sufficient, there needs to be technical solutions to support the requirements.

In a data driven organization, data is a necessary tool for many employees in their daily work. This might be particularly relevant for system developers, a job function closely tied to data. Any changes related to data, may it be storage or restrictive access will not pass the system developers unnoticed. With tasks dependent on data, modifications to the data handling could have consequences on the way of working for system developers.
Organizations and specifically system developers find themselves in a dual situation. Privacy for users is important, it is a human right [2] and legal requirement from the [1]. At the same time, system developers need to perform tasks associated with the role which could require user data.

This study is investigating the relationship between these two ambitions; privacy for users and utilizing system developers capacity in the system. The first goal is to preserve the users’ privacy within an organization. Secondly, an organization strive to take advantage of the data and to make for their system developers to perform their job efficiently in the system. To achieve this, the study explored how a privacy-preserving system can be designed applying PbD with consideration to both system developers and the users of the service. The key aspect for the study was to evaluate how a system applying PbD in this way would affect system developers.

1.1 Research Question

How can PbD be applied to a system and what are the consequences for system developers working in such a system?

1.2 Research Gap

This project shares a common ground with several studies. In Section 2.7 related work is presented in the area of Privacy by Design in Practice. The methodologies of the related work differ as well as the elements considered in the evaluations. The contribution of this project is to apply PbD to a system model followed by an investigation of what the consequences would be for system developers working in a system designed in this way.

This project aims to gather an understanding of the state of the art in privacy architectures and applying PbD to a system. This has been studied before, literature studies [3, 4] have been performed and proof of concepts [5] have been used as a method to model and evaluate privacy. The novelty of this work is the focus on system developers. This project evaluates the attitudes of system developers and how they would be affected in a system applying PbD with the approach of Colesky, Hoepman and Hillen.
In the related research, most works evaluate the privacy of the solution and lack the view of system developers. As mentioned, the experience of the developers is a major focus and contribution for this project. The paper [6] on how developers make privacy decisions has a scope that is quite different to this project. But the similarity of these two projects is the evaluation, where both focus on the system developers. The findings made by Ayalon, Toch, Hadar and Birnhack [6] could be combined with the conclusions from this study to get a better understanding of system developers and their relation to privacy.

It has been recognized before [7] that there is a need for Privacy-Enhancing Technologies (PET) that can obtain wide usage in the industry. The research [8] performed by Goldberg in 2008 aimed to distinguish what is needed from a PET to achieve wide spread. The conclusions from this project could be used to contribute to the goal and further identify factors that determine if PET would become successful in the industry today.

Another key metric considered in this project is the data quality. Hoepman mentions data quality in the paper Privacy Design Strategies from 2012 [9]. The design strategies are derived from the ISO 29001 framework and the Organisation for Economic Co-operation and Development (OECD) guidelines. Data quality is included in both of these guidelines but has been discarded in the design principles in the paper with the motivation that it "is not a privacy related issue"[9]. Regardless of data quality being a privacy related issue or not, high data quality is a requirement in this project. This requirement provides another angle and could lead to new insights in the field of PbD.

### 1.3 Scope

The hypothesis is that a solution to preserve privacy of users can be applied to a system model in such a way that the system developers can perform their daily tasks with tolerable trade-offs.
The desirable outcome is a privacy level high enough to make it difficult to identify a unique individual from any stored data that the system developers would have access to in the proposed system model. There would occur situations where user data would be needed access. For this reason, data can not be fully anonymized or encrypted in a way that is non-reversible. The privacy of users shall be preserved in such a way that the system developer cannot identify the data subject from the available dataset, unless special conditions apply where data needs to be de-anonymized.

The second part of the evaluation is the potential trade-offs for system developers working in the proposed system model applying PbD. The optimal outcome would be a system where employees can perform their daily tasks without being hindered. In addition, data access shall be adapted to fit the role of each employee, taking data minimization into account. Furthermore, the quality of data is a crucial factor. The data analysis needs to produce detailed, accurate and reliable results. A requirement is that results can be verified against the data subject if needed.

The requirements on the system are fulfilled if the system meets the privacy level set out by the GDPR and the system developers can perform their tasks.

1.4 Outline

This report is divided into 7 chapters. In Chapter 2, the reader is introduced to the theoretical framework for the report together with related work. Chapter 3 explains the methodology of the project, initiated with a literature study followed by a data survey to gather knowledge about prerequisites and requirements and finally interviews were conducted to evaluate the system model. The design process of the proposed system model is explained as well as the method for evaluation. Chapter 4 goes into depth about the design process of applying PbD to the system model by presenting it step by step. The results from the evaluation are presented in Chapter 5. Chapter 6 provides a discussion about the work, including ethical and societal considerations, critical evaluation as well as suggestions for future work. Lastly, Chapter 7 states the conclusions from the work.
Chapter 2

Background

2.1 Privacy

The concept of privacy covers several areas. To mention a few, privacy can be viewed from the aspects of territorial privacy and bodily privacy. For clarification, privacy in this project refers to the informational privacy of individuals. Informational privacy can be defined as the right to informational self-determination. In other words, this means that individuals own the right to regulate information about themselves and determine how, when and to what extent this information is shared to others [10]. One aim of the GDPR is to perform a power shift of data; to transfer the data ownership from the organization to the individual.

2.2 Anonymity and Pseudonymity

Anonymity refers to a user not being identifiable. More specific, within a set there needs to be subjects with potentially the same attributes ensuring the user can not be distinguishable within the anonymity set [11].
Pseudonymity is an option to anonymity to enable translation from pseudonym to the user’s identity. The technique ensures that a user under pseudonymity can use a service or research without disclosing his or her identity. However, the advantage with pseudonyms over fully anonymization is that a pseudonym can be translated into the identity of the user if required. This is usually performed under certain circumstances, for example in services where the user can not be fully anonymous due to laws where the user needs to be hold accountable for his or her actions [12].

There are several approaches for organizing pseudonyms. The pseudonym can be generated by the owner, so-called self-generated pseudonyms. It can be dependent on the current role the user is performing; role pseudonyms. Another method is cryptographic pseudonyms where the identity data is encrypted and the result becomes the pseudonym. In order to translate the pseudonym to the identity, it needs to be decrypted with the decryption function and decryption key [12].

### 2.3 Privacy by Design

The concept of PbD was developed and first presented in 1995 with Ann Cavoukian in the forefront [13]. It consists of seven principles, which are presented in Table 2.1. The digital world has gone through big changes since 1995. Today we are facing another world with social media, digital solutions for everyday services and other information sharing technologies. Despite the changes in technology, many of the privacy challenges remain the same. The principles in PbD can still be applied and are very much relevant. As recent as 2018, the EU released GDPR which implements PbD [1].

<table>
<thead>
<tr>
<th>1</th>
<th>Proactive not reactive; preventative not remedial</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Privacy as the default setting</td>
</tr>
<tr>
<td>3</td>
<td>Privacy embedded into design</td>
</tr>
<tr>
<td>4</td>
<td>Full functionality – positive-sum, not zero-sum</td>
</tr>
<tr>
<td>5</td>
<td>End-to-end security – full lifecycle protection</td>
</tr>
<tr>
<td>6</td>
<td>Visibility and transparency – keep it open</td>
</tr>
<tr>
<td>7</td>
<td>Respect for user privacy – keep it user-centric</td>
</tr>
</tbody>
</table>

Table 2.1: The seven principles of PbD [13]
The principles provide useful guidelines but lacks concrete implementations. Research has been made to translate the principles to concrete technical implementations and so called engineer PbD. A selection of work in this area is presented in Related research.

2.4 Privacy-Enhancing Technologies

Technologies that are protecting and enhancing user privacy can be classified as PET [10]. This makes PET a broad definition and includes a lot of different technologies. Initially, PET research focused on anonymous communication in emailing and browsing. Today, the field has expanded into several other domains where privacy has become important [14]. Social media, Internet of Things, Big data, and Artificial Intelligence are some fields that recently have caught interest in the PET community. Not to mention legal changes, for example GDPR that might affect the research in PET.

2.4.1 De-identification

De-identification is a technique where data is cleaned from personal information. This is common to perform before conducting statistical research. By doing this, the data can be used for analysis and preventing it from being connected with the owner’s identity. The de-identification process includes generalization of attributes and removing certain data fields [15].

A vulnerability with the de-identification method is the potential risk of re-identification. Perfect de-identification means that it is no longer possible to identify the data subject from the treated data. However, assuring perfect de-identification is almost impossible to achieve in reality [12, 15]. Data that appear anonymous can be used in combination with other information to uniquely identify the individual.
2.4.2 k-anonymity

In the process of de-identifying of data, personal information that can be used to uniquely identify an individual is considered. This include explicit identifiers like name and security number. It is although important to also recognize the quasi-identifiers. These are the attributes that can be used in combination to uniquely identify an individual. For example, gender, birth date and zip code are quasi-identifiers [16]. Recognizing quasi-identifiers is not trivial. Even anonymous movie ratings have been used as quasi-identifiers to de-anonymize a dataset [17].

The technique k-anonymity is targeting the risk of re-identification with quasi-identifiers. Within a dataset, there needs to be at least k occurrences with the same quasi-identifiers to satisfy k-anonymity. Table 2.2 shows an example of k-anonymity on a fictive dataset depicting food allergies. To achieve k-anonymity, data entries can either be removed or added to the dataset, alternatively the quasi-identifier can be generalized.

<table>
<thead>
<tr>
<th>Gender</th>
<th>Birth year</th>
<th>Zip code</th>
<th>Food allergy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>1985</td>
<td>122 67</td>
<td>Wheat</td>
</tr>
<tr>
<td>Female</td>
<td>1985</td>
<td>122 67</td>
<td>Wheat</td>
</tr>
<tr>
<td>Male</td>
<td>1998</td>
<td>659 10</td>
<td>Egg</td>
</tr>
<tr>
<td>Male</td>
<td>1998</td>
<td>659 10</td>
<td>Egg</td>
</tr>
<tr>
<td>Female</td>
<td>1962</td>
<td>340 20</td>
<td>Milk</td>
</tr>
<tr>
<td>Female</td>
<td>1962</td>
<td>340 20</td>
<td>Peanut</td>
</tr>
</tbody>
</table>

Table 2.2: Quasi-identifiers QI={Gender, Birth, Zip code} and k=2

The implementation of k-anonymity in real situations has proven to be challenging. Real datasets are often high dimensional, meaning that it contains a lot of data and potential quasi-identifiers. This impose difficulties for carrying out k-anonymity in an effective matter [17].
2.4.3 Privacy-Preserving Aggregation

In certain situations, Privacy-Preserving Aggregation can be an efficient method for collecting data and assuring privacy. The technique collects and stores data from a group rather than an individual. The method is purpose-specific, which limits the possibilities to reuse the data for other purposes. This inflexibility causes challenges when performing big data analysis with complex machine learning algorithms [15].

2.5 Role-Based Access Control

There exists multiple access control policies to manage and restrict access to a system or database. One approach of access control is Role-Based Access Control (RBAC) [18]. In RBAC are the system users assigned to appropriate roles. A role is a job function or title within an organization. Permissions are associated with a predefined role and not the user. This enables a user to enter different roles depending on the current work task. Contrary to permissions associated with the user, RBAC provides more flexibility for events such as individuals changing jobs internally or leaving the organization. The roles tend to change less frequently than the employer filling the job function it represents [18].

2.6 General Data Protection Regulation

GDPR is a law that came into effect 25th of May 2018. It restricts collection, storing and processing of personal data. The law protects all individuals in the EU, regardless where the organization processing the data is located [1].
CHAPTER 2. BACKGROUND

The regulation concerns personal data. This means that data which is not personal is not covered by the law. Personal data is defined to be “information relating to an identified or identifiable natural person” [1]. There is also a considerable difference between personal data and sensitive data (also referred to as special categories of personal data). The European regulation states that personal data revealing the following is sensitive: racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, biometric data for the purpose of unique identification, health, sex life or sexual orientation [1]. The processing of sensitive data is prohibited unless certain requisites apply.

2.7 Related work

Much of the research related to this project originate from the concept PbD, presented in the 1990’s. As mentioned in the Background, PbD provides abstract principles but lacks concrete methodologies. The principles has since been elaborated in other research and resulted in a sub field; Privacy by Design in Practice or also called Engineering Privacy by Design.

The following works have in different ways made contributions to use Privacy by Design in Practice for system development. The studies are presented by publication date, beginning with the latest.

Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering

Y.S. Martin and A. Kung.

The study [19] from 2018, recognizing the need for methods and tools to integrate data protection principles in the software development process. The authors suggest to implement privacy into existing tools, already used by software engineers instead of creating new tools. The study focuses on description of the problem and motivates the need for a solution. The study does not offer a solution to the problem but there is a follow-up research planned with the aim to provide this. The prerequisites of the study are very much alike to this project, therefore the outcomes of the two works could potentially be compared and complement one another.
Privacy Architectural Strategies: An approach for Achieving Various Levels of Privacy Protection

M. Alshammari and A. Simpson.

The paper [20] aims to define an approach to take design decisions with privacy in mind. Rather than developing new technologies and methods, it focuses on the process of selecting a suitable solution among existing options. The study from 2018 resulted in an approach that the authors illustrate with a case study. Briefly, the approach is to first gather an overview of the problem, this includes risk analysis and deriving the protection goals. The process is followed by identification of architectural tactics which lay the foundation for selecting appropriate design patterns and then PETs. When these steps are finalized, it remains to identify the architectural strategies.

The approach from Alshammari and Simpson’s the research served as inspiration when deciding on the methodology for this project.

How Developers Make Design Decisions about Users’ Privacy: The Place of Professional Communities and Organizational Climate

O. Ayalon, E. Toch, I. Hadar, and M. Birnhack.

In 2017, Ayalon, Toch, Hadar and Birnhack performed the research [6] to gain understanding on how developers make design decisions about the users’ privacy. The authors conducted an online survey among developers to distinguish what influences the decision making the most. The results indicate that organizational privacy climate has bigger effect than legal implications. It also showed that the developers’ personal experiences as an end-user and perceived privacy had an impact on their decision making. In addition to these aspect, developers tended to discard privacy architectures that deviate from existing frameworks.
Applying Privacy by Design in Software Engineering  
- An European Perspective

K. Bernsmed.

The paper [21] from 2016 looks at the current state of the art of Privacy by Design in Software engineering and analyzes what impact the GDPR has on this process. The main contribution of the project, besides providing the state of the art is a self-assessment method for PbD. The self-assessment method includes four viewpoints. The viewpoints are firstly acknowledge privacy in the organization, followed by appropriate privacy policies and by building privacy in and finally enabling end-user-control.

A Critical Analysis of Privacy Design Strategies

M. Colesky, J.H. Hoepman, and C. Hillen.

Colesky, Hoepman and Hillen published the paper [4] in 2016 with the aim to further bridge the gap between system development in practice and legal requirements for data protection, considering the GDPR. The paper was published four years after Hoepmans work on Privacy Design strategies [9]. The major outcome from the extensive privacy pattern literature review was tactics. The authors identified the need for an additional abstraction level between privacy strategies and privacy patterns and developed tactics to fulfill this need. The tactics are presented in Figure 2.3 and use the privacy design strategies presented by Hoepman in 2012 [9] for classification. One suggestion for future work mentioned by the authors is investigation of the strategies and tactics in practice, this could be achieved doing case studies.

The privacy design strategies and tactics by Colesky, Hoepman and Hillen were considered when designing the system model in this project. Chapter 4 presents each strategy in detail and how it can be applied in a system.
CHAPTER 2. BACKGROUND

<table>
<thead>
<tr>
<th>Minimise</th>
<th>Hide</th>
<th>Separate</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude</td>
<td>Restrict</td>
<td>Distribute</td>
<td>Summarize</td>
</tr>
<tr>
<td>Select</td>
<td>Mix</td>
<td>Isolate</td>
<td>Group</td>
</tr>
<tr>
<td>Strip</td>
<td>Obfuscate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destroy</td>
<td>Dissociate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inform</th>
<th>Control</th>
<th>Enforce</th>
<th>Demonstrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply</td>
<td>Consent</td>
<td>Create</td>
<td>Audit</td>
</tr>
<tr>
<td>Notify</td>
<td>Choose</td>
<td>Maintain</td>
<td>Log</td>
</tr>
<tr>
<td>Explain</td>
<td>Update</td>
<td>Uphold</td>
<td>Report</td>
</tr>
<tr>
<td></td>
<td>Retract</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3: Privacy design strategies by tactics from the paper [4]

Privacy and Data Protection by Design – from policy to engineering


The article [3] published by the European Union Agency for Network and Information Security (ENISA) in 2014 provides a state of the art on Privacy by Design with focus on technical implementation. The authors have summarized existing work in the field from that time. The article provides guidelines on how a system can be designed with privacy in mind and which privacy design patterns, strategies and techniques that are available. One key finding from the research was that traditional engineering approaches generally ignored privacy and data protection features. It was also verified that although the research area is active, it needs future work on privacy by design in practice as well as enforcement of compliance with the data protection and privacy regulatory. The authors recommend future work on privacy engineering, providers of software development tools need to provide privacy-supporting components and include privacy in standardization processes. They also encourage policy makers to support and promote privacy-friendly services.
Privacy Design Strategies

J.H. Hoepman.

In 2012, Hoepman published the paper [9] on design strategies to support Privacy by Design in the software development process. The eight privacy design strategies are minimise, hide, separate, aggregate, inform, control, enforce and demonstrate. These were developed from analysis of the data protection legislation from the time. In the paper, the design strategies are validated against two different models of ICT systems. The design strategies have become popular in the field and are frequently referred to in privacy research. Hoepman was four years later a co-author of the study A Critical Analysis of Privacy Design Strategies together with Colesky and Hillen [4].

Engineering Privacy by Design

S. Gürses, C. Troncoso, and C. Diaz.

By conducting two case studies, the research [5] from 2011 models how data minimization can be applied in a system. The study is within the field of Privacy by Design in practice and focuses specifically on data minimization. Two different case studies were chosen to reflect the wide usage of data minimization. The first case was an e-Petitions system where techniques that rely on anonymity could be used. The second case study case an electronic Toll Pricing system where identity is required and other techniques of data minimization need to be considered. The outcome from the case studies was generalized and resulted in five activities to implement Privacy by Design. The five steps are: functional requirements analysis, followed by data minimization, modelling attackers, threats and risks, multilateral security requirements analysis and lastly implementation and testing of the design.
**Engineering Privacy**

S. Spiekermann and L. F. Cranor

Above providing state of the art from the time, the paper [22] from 2009 presents a three-sphere model for user privacy concerns mapped to operations; data transfer, storage, and data processing. The authors describe two approaches on engineering privacy. One approach is privacy-by-policy which would require less intrusion on the existing system architecture and instead focus on implementing privacy principles and guidelines. The other approach mentioned is privacy-by-architecture where the privacy concerns are targeted from the perspective of system architecture. These two approaches provide different characteristics. The authors argue that privacy-by-architecture generally provides higher user privacy, but privacy-by-policy is mostly used by businesses because it does not risk to interfere with the business model.
Chapter 3

Methods

The research was conducted in multiple steps. The methodology has similarities with Alshammari and Simpson’s approach to achieve privacy [20]. The methodology used for this research was a literature study followed by a process to gather knowledge about the problem. A survey was conducted together with complementary interviews to understand what system developers need and to define general requirements for the system model. Thereafter, the privacy approach was applied to the system model considering the needs of system developers and functional requirements. Finally, the methodology to evaluate the solution consisted of individual interviews with system developers from a focus group.

3.1 Literature study

The first part of this project was a literature study. The purpose was to gather knowledge in the related fields and an overview for the state-of-art. The areas for the literature research were privacy, PbD, Privacy by Design in Practice, PET, GDPR and related subfields. The broad scope for the literature research resulted in several technologies that were used to narrow down the potential solution.
3.2 Prerequisites & Requirements

The second part of the research focused on establishing prerequisites and requirements. The main objectives were to identify what system developers need as well as requirements for the system model. The methodology used was a survey about data need complemented with interviews with system developers.

3.2.1 System model

The theoretical system model in Figure 3.1 provides a foundation to apply PbD. The system model represent an overview of an architecture with two environments; one production environment and one test environment.

![Figure 3.1: A theoretical model of a system architecture](image)

The system model includes an interface for the user to communicate with the service. In the model, the backend code would be hosted on a cloud computing platform and communicate with the production database. The production database would contain necessary data needed for the service, including user data.

The test environment shall support the system development and testing. In this system model, the test environment is a reflection of the production environment but with a separate backend code and a separate database.
3.2.2 Data Survey and Interviews

The design of the solution would rely heavily on the needs of system developers that would work in the system. Determining the data need with certainty is not trivial. The methodology used to establish the data need was firstly a formal survey and additional interviews with system developers. The results are based on estimations and generalizations made by the participants.

The participants of the survey had different roles in the IT field, including testers, managers, customer support agents etc. The particular group of interest in this survey was employees with the role system developer. The total number of participants in the survey was 69. Out of these, the number of system developers was 11. In addition to the survey, complementary interviews were conducted with employees with the role system developer.

The survey and interviews aimed to answer the following questions:

- What data do system developers need?
- In what format would the data be needed? Non-anonymized, pseudonymized, anonymized etc.
- Why and in which situations is data needed?
- How frequently would data be needed?

The need for data can be categorized into two groups; daily need and the need in an exception. It was identified from the data survey that the system developers would need access to certain data with higher frequency, representing the daily need. It was also anticipated that situations would occur requiring data beyond the daily need, considered as exceptions.

3.2.3 Daily need

The primarily role of the questioned system developers would be to develop and maintain the product and features within it. The activities within this are considered daily tasks. The role of the system developer could include other activities above this. One such example is technical support for users, which is not considered a daily task. This is covered in Section 3.2.4.
Type of data

The user data would contain various types of information. Depending on the purpose and functionality of the application, different information would be stored and the system developer might need to access certain information to greater extent than other. The type of data needed by system developers on a daily basis would be the data related to the main functionality. The results from the survey and interviews showed that data needed with most frequency would be data associated with the core functionality of the service.

Format of data

When developing new features or making changes to existing code, the system developers need to test and verify the changes. This activity require data with linkability property. The requirement is that the test data reflects the target user group and includes variations within the data. By this requirement, it is not a necessity for system developers to test on real user data, but the data needs to be a realistic representation of the user group. When developing new features and editing functionality, the system developers need to have a divergent dataset to cover a wide range of scenarios and edge cases. This is a crucial requirement for test data.

3.2.4 Exceptions

An exception refers to a situation where the system developer would need to perform a duty that is not considered a daily task and this might require more data than what is included in the daily need. The system developers were asked to motivate in which situations they need access to user data and why it is necessary. The most common scenario mentioned was when system developers work with support cases from users. This situation requires access to real data and debugging with specific requirements on the data.
Type of data

The participants of the survey were presented to categories of data (e.g. payment information, device information, demographic information) and the majority of categories would be seldom needed. The individual answers showed that system developers need access to different types of data with varying frequency. The results from the data survey and interviews indicate that most categories of data would be needed only occasionally and not by all system developers. Therefore, the system developers would not need access to a full dataset from a user, rather a subset which is dependent on the specific need for the moment.

Format of data

In the daily work, the system developers need linkability in the data. This would also be true for many exceptions. Linkability was mentioned as a necessity when system developers handle bugs in the system. This activity requires the system developer to be able to reproduce, investigate and verify fixes to bugs related to specific use cases. Pseudonymity could provide linkability in the dataset without revealing the identity of the data subject. It would also occur exceptions where system developers need knowledge about the identity of the data subject. This information would be needed infrequently and associated with situations where users of the service seek support.

3.3 Designing the new system model

The system model needs to consider multiple aspects. The expressed needs from system developers shall be fulfilled and the required level of privacy shall be achieved. These requirements were applied to the system model presented in section 3.2.1. The method to apply PbD to the system model was based on findings from the literature survey. As described in section 2.3, PbD offers principles to achieve privacy but lacks concrete guidelines on how to implement the principles. The research in the field of Privacy by Design in Practice aims to fill this gap. This project considered the approach described by Colesky, Hoepman and Hillen [4]. The process of designing the new system model was based on the eight privacy design strategies first defined by Hoepman [9] and later refined with the addition of tactics [4].
The process of applying Colesky, Hoepman and Hillen’s approach on Privacy by Design in Practice to the system is described in detail in the section 4.

3.4 Evaluation

The evaluation is mainly considering two aspects. One aspect is the general experience of the developers. The other focus for evaluation is how well privacy is preserved in the system. The system after applying PbD with the design strategies by Colesky, Hoepman and Hillen is also evaluated against other PbD frameworks.

3.4.1 Comparison with other PbD frameworks

The approach by Colesky, Hoepman and Hillen is just one guidance to apply PbD to a system. As presented in Related work, multiple studies aim to contribute with a concrete methodology to apply PbD. The system model presented in this study after applying PbD with the approach by Colesky, is validated against other methodologies.

3.4.2 Impact on System Developers

Individual interviews were used as the methodology to evaluate what the consequences would be for system developers to work in the proposed system model. The interviews were performed with a focus group consisting of 10 system developers. The participants were asked about their general attitude towards the proposed system model, if they would be able to perform daily tasks and the anticipated impact on productivity.

3.4.3 User Privacy

To evaluate if the privacy requirements were met, formal analysis and privacy risk analysis were applied to the proposed system model. The method included firstly to identify the expected work flow of the system developer in the proposed system model. The data survey and interviews provided knowledge about how the system would be used and the expected information flow.
An analysis of the proposed system model was performed to identify potential information leaks, threats and privacy risks. The system developers participating in the interviews did also contribute by predicting potential risks associated with the proposed system model.
Chapter 4

Applying PbD to the system

The authors Colesky, Hoepman and Hillen conducted an extensive literature review in the year 2016, considering around 100 existing privacy patterns. The patterns were categorized by strategies and a main contribution of the paper is tactics. The eight strategies are: minimise, hide, separate, abstract, inform, control, enforce och demonstrate. The following section demonstrates how the system architecture would be modelled after applying PbD with guidance of these privacy design strategies and tactics.
4.1 Minimise

The design strategy *minimise* would be applied in the initial step where data is collected from the user. Minimisation includes the tactics *exclude, select, strip* and *destroy*. The data survey performed indicated what data is needed for system developers to perform their job. To minimise the collection of data, the data survey would serve as a guidance on what data to collect and not. The Figure 4.1 illustrates the process. After the minimization process, the system only collects and stores personal data where the purpose can be stated. In addition to being a privacy design strategy by Hoepman, this is also a requirement by GDPR.

![Figure 4.1: Model of minimization process](image)
4.2 Abstract

Once the stored personal data have been minimised, abstraction would be applied to further enhance privacy. The strategy abstract includes the tactics summarize and group. The database in the system would first be grouped by type of data, e.g. payment information, device information, demographic information. Secondly, the data within each of the groups would be summarized in one field. This would enable to present data on a higher, less detailed level. To illustrate this, an example would be demographic data. On a low level is street address or possibly coordinates, which is unique to every individual (or limited to a small group of individuals sharing the same address). This can be summarized into a higher level value like continent. In between these levels of details, information can be presented ranging from country, city and zip code. Figure 4.2 models this abstraction of data. The outcome from applying abstraction to the system would be a database where data is grouped into categories which have a field with a summarized, high level value.

![Figure 4.2: Model of abstraction process](image-url)
4.3 Separate

The design strategy *separate* includes the tactics *distribute* and *isolate*. The theoretical system model which serves as a foundation has one production database and an additional database in the test environment. When applying the design strategy separate, this architecture can be further brought apart to meet different needs. The test environment could be used by employees with different roles. These groups would have different needs for data. Therefore, when applying the strategy separate, the test database was made into two versions where one would be accessible for system developers and contain synthetic data. The second database in the proposed system model would contain psuedonymized user data for other groups than system developers that are reliant on real data. This is explained in Figure 4.3.

![Figure 4.3: The two versions of the database in test environment.](image)

The separation strategy can be further applied by isolation and distribution on a more detailed level. This is partly achieved when the strategy abstract is applied and the data is categorized and therefore separated into groups.
4.4 Hide

It was recognized from the data survey that there is a distinction between the daily need and the need in an exception. The data that is rarely accessed can not be deleted from the database since it is still needed, although it is not accessed by system developers on a daily basis. The solution is to apply the design strategy hide, which gathers the tactics restrict, mix, obfuscate and dissociate. By applying this strategy to the system model, specifically obfuscate where encryption is one technique, the two problems are solved; data is available when needed and unnecessary exposure of data is avoided.

The strategy hide would include both approaches privacy-by-architecture and privacy-by-policy. Obfuscate would have an effect on the architecture of the system by implementing encryption techniques and encrypt the fields which are not included in the role access. Another method to obfuscate is to not use real data. In the system model, this would be applied in the test environment by having fictive personas with data instead of real system users when possible.

The privacy-by-policy implementation would regard the access control. The access and possibility to decrypt the fields in the system would be restricted by RBAC.

![Figure 4.4: Model of process for exceptions when user data is needed](image)
The roles would be defined based on the findings from the data survey. The need for user data was classified by daily need and exceptions. This could be represented in a default role to cover the daily need for a system developer and a process for handling exceptions. The exceptions would be handled by separation privilege. It was identified from the data survey that exceptions would mostly originate from support tickets where system developers perform debugging. In these situations, the default system developer role access would not be sufficient. Additional access would be granted to a system developer when assigned to the task in the planning tool. In Figure 4.4, the process of granting access to another role is modelled.

### 4.5 Inform, Control, Enforce & Demonstrate

The strategies *inform*, *control*, *enforce* and *demonstrate* are process or policy oriented. These privacy design strategies have for this reason not had a major influence on the architectural design of the proposed system model. Instead, the strategies strive to achieve privacy-by-policy rather than privacy-by-architecture.

*Inform* is a strategy including three tactics *supply*, *notify* and *explain*. The strategy states that the data subject shall be informed of what data is stored, why and how it is stored, who has access to it in addition to when it will be destroyed. This information would be shared with the user of the application.

*Control* holds the tactics *consent*, *choose*, *update* and *retract*. This strategy ensures that the data subject can take actions on the information given to her. In the proposed system model, it would include the possibility for the user to review, update and retract the consent at any time. The user shall also be able to choose what specific data she wants to share and for what purpose. There would be a difference made between sharing data for use of the service and the purpose of research.

*Enforce* includes three tactics which are *create*, *maintain* and *uphold*. This refers to acknowledging privacy in an organization by having a privacy policy and ensuring it is maintained and followed. In the proposed system model it would mean having regular trainings for employees in privacy and data handling to inform and make sure the policy is followed. The policy would be kept up to date by regularly reviews with stakeholders.
Demonstrate is the strategy which covers the tactics audit, log and report. The RBAC implemented in the proposed system model could serve as an audit functionality. The distribution of roles would be based on the planning tool where the system developer would be assigned to a task. This can be used to track the data access back to an individual and the purpose for accessing data. To benefit from this, the access would need to be audited and misuse reported. Historical accesses indicate what data that is needed and what is not and this information could be used to take actions to restrict or remove data.

4.6 The proposed system model

This section summarizes the process of applying PbD to the theoretical system model described in section 3.2.1. The proposed system model after applying PbD is pictured in Figure 4.5. The design strategies developed by Hoepman and redefined with tactics by Colesky, Hoepman and Hillen are open to interpretation. The proposed system model represents one result of applying PbD with this approach but could potentially be designed differently.

Considering the expected data needed by system developers daily, the test data needs to be representative of the user group but it does not need to be real user data. This means that system developers can work with a test database populated with synthetic data.
The test environment with fictive data covers the expected daily need by system developers. In case of an exception and the system developer needs to access real data, this would be performed with RBAC. The system developer would enter the required role and be granted access to a limited part of the production database with enough information to solve the task and not more. The process of distributing role access is explained in Figure 4.5. An exception could be when developing a new feature or debugging and issue where it is critical that the input data reflect real users and cover a wide variation. For this, the developer would gain access to a role with authority to read data from multiple users in a pseudonymized or anonymized format. A different kind of exception could be a support case where a user seeks support and requires the system developer to analyze the specific dataset. This exception would give the developer possibility to enter a role which could access the data of a single user in an identifiable way. There are two factors; the number of data entries accessible and the level of anonymization. These two factors need to be taken into account when defining the roles and most importantly, distributing access to the roles.

Figure 4.5: Model of the system after applying the privacy approach
A key point of the architecture in the proposed system model is the separation and abstraction to the database. The database in the proposed system model would be grouped based on category and summarized to a generalized, high level value when possible. This enables RBAC to be specific and not disclosing redundant information to the task or more identifiable information than necessary.
Chapter 5

Results

5.1 Comparison with other PbD frameworks

Bernsmed highlights the need for a simplified way for organizations to adopt PbD. The study [21] looks into existing PbD approaches, among them Colesky, Hoepman and Hillen. It proposes four key viewpoints of PbD which translate into a self-assessment method. The four viewpoints are Acknowledge privacy in the organization, Appropriate privacy policies, Building privacy in and Enabling end-user control. The proposed system model was developed focused on architecture, although there are design strategies focused on privacy-by-policy. Naturally, the viewpoints resembles the design strategies of Colesky, Hoepman and Hillen. The proposed system model does not present the privacy policies in detail and could be a reason for not fulfilling the viewpoints in the self-assessment method. It is possible that a system designed with Bernsmed’s approach would lead to a bigger focus on policies over architecture.
Gürses, Troncoso and Diaz demonstrate their privacy containing five generalized steps to implement Privacy by Design in practice [5]. The steps are Functional Requirements Analysis, Data Minimization, Modelling Attackers, Threats and Risks, Multilateral Security Requirements Analysis and finally the Implementation and Testing of the Design. How the system would look applying this approach is somewhat difficult to tell because of the requirements. Considering the most basic functional requirements, the solution would be completely different to an application where the wide spectra of functionality is a requirement, potentially influenced by business interests. Overall, the methodology resembles the process of this study. Gürses, Troncoso and Diaz focus on data minimization, which is included as a strategy. Although it is possible that a system designed with the approach of Gürses, Troncoso and Diaz would result in data minimization to a greater extent.

In summary, the approaches strive towards the same end goal, to adopt the seven principles of PbD. They have also influenced one another in different way. The approaches focus on different aspects. The proposed system model contains parts of both Gürses, Troncoso and Diaz approach as well as Bernsmed’s viewpoints.

### 5.2 Impact on System Developers

The two major focuses for evaluating the proposed system model were the estimated impact on productivity of system developers as well as general attitudes towards working in the proposed system model.
5.2.1 Productivity

The requirement for productivity was that system developers can still do their work in the system. The participants in the interviews were united in the opinion that it would be possible to do the work of a system developer in the proposed system model. The model enables the system developers to work efficiently and enhances user privacy by separating the two needs; daily need which could be fulfilled by synthetic data and the exceptions when real user data is necessary. By providing the data needed daily in an easily accessible way, the impact on productivity is kept to a minimum. The procedure for exceptions would potentially have an impact on productivity but this would occur less frequent.

When the system developers where asked how the system model would affect productivity, the most common conception was that the productivity would be slightly lower in the proposed system model compared to the theoretical system model. It was believed that the productivity would be most affected in the transition process of applying PbD in this way. The majority of the participating system developers thought that the decrease in productivity would be unnoticeable over time. It was also mentioned a possibility for the opposite effect; that productivity would increase due to the limitation of manual access control and clear ownership.

The RBAC was anticipated as the biggest influence on system developers way of work. The setup of the access control was the greatest concern and potentially biggest contributor to decreased productivity. The concerns mentioned were the workload caused by administration of RBAC. The system developers expressed it to be challenging to identify beforehand what data that is needed to perform a task. This is particularly difficult for support tasks requiring debugging. Another problem related to this would be if access to roles would require permission by an authority. The system developers predicted a negative impact on productivity if data access needs to be approved by an authority. This could result in multiple requests and approvals if the needed data is difficult to distinguish and needs to be redefined.
The system developers were asked in the interviews how strict they thought the RBAC shall be and the answers were divided. How the RBAC is implemented in the proposed system model is intended to reduce the manual administration of roles and rely on the planning tool to determine the needed role for the current task. There was although a difference of opinion among system developers if the needed access shall be defined and granted by an authority. One suggestions from the system developers was that an authority could be the manager of the employee requesting access. Another proposal was that the product owner is responsible for the requested dataset manage the access to it. But it was also expressed that having RBAC without the need to grant access through an authority would be better from an efficiency perspective, relying on the audit functionality. Although it might be a worse alternative from a privacy perspective if anyone could set and update the needed role.

5.2.2 Ease of Use

The interviews revealed an interesting aspect on the anticipated difficulties and frustration in the proposed system model. The system developers were asked if the proposed system model would introduce frustration or in any other way make it less pleasant to perform daily tasks in comparison to the theoretical system model without privacy actions. The majority of the questioned system developers declared that they would not be bothered themselves but believed that other system developers could have a negative attitude towards the system applying PbD in this way. The individual answers indicate that most system developers would be positive towards working in the proposed system model. There could be several reasons behind these answers. One possibility is that it exists a misconception and prejudice that the general view towards PbD is negative among system developers. It could also have been an effect of non-anonymous interviews in the evaluation that resulted that restrained the system developers from expressing their own opinions. Another potential explanation is that the system developers questioned had different responsibilities, where some worked more with data analysis very dependent on user data. Whereas some developers never accessed user data and in the interviews expressed themselves as if they would be in the position of working with user data.

The system developers were united in realizing the trade-offs in comparison to the theoretical system model but most expressed it to be negligible compared to the positive effects or privacy.
The proposed system model could contribute with positive effects on the usability as well. System developers declared that the privacy measurements in the proposed system model would provide safety when working in the system. The architecture and processes in the proposed system model ensures a low risk for system developers to accidentally read data and find out unintended information. The proposed system model have fictive test data and the production database separated. It also enables granular access to view only the needed subset of user data rather then being presented with all content of the database. The theoretical system model before applying PbD would rely heavily on the individual system developer to take responsibility for the privacy of users. The functionalities introduced in the proposed system model support system developers in handling user data in a privacy-preserving matter.
5.3 User Privacy

The analysis of the expected information flow in the proposed system model and the interviews performed disclosed a number of privacy risks.

Firstly, RBAC is a risk for abuse. This applies to the scenarios if the access control is too strict but also if it is too loose. The risk with having an access control that is very strict, meaning only a few people have authority to provide access to roles, would be that the restriction is not used as intended. If the process is too complicated, difficult or time consuming, there is a risk of granting full access to reduce the work of figuring out the amount needed and potentially having to update it later. In the opposite situation, if the restriction is too weak, it is easy to exploit. Although it can be argued that if the planning tool is used as an audit function, this might provide the required privacy measurements. The distinction between having an access control that is too strict or too weak needs to be made by each organization. This would be dependent on the required level of privacy and organizational culture.

Secondly, there would be micro services holding a fraction of the database. There is a risk that a third party application or micro service could be used by system developers to bypass the direct restriction to production database.

As Spiekermann and Cranor express [22], privacy can be achieved by policies and architectures. The architecture of the proposed system model has been showed to include risks of abuse. These privacy risks could be tackled with policies. As part of the policy oriented design strategies are enforce and demonstrate. They are focused on privacy from the perspective of the controller and authority. The strategies are open for interpretation, enforce includes the tactics create, maintain and uphold privacy policies. But how the policies shall be defined is not discussed. Demonstrate contributes with tactics audit, log and report that are concrete tools for verifying that privacy policies are followed. But it leaves out details such as how it should be implemented in order to minimize the risks of violations.
It all comes down to the human factor. Colesky, Hoepman and Hillen do not mention the human factor in their study [4] and how the privacy design strategies would affect the privacy decisions made by system developers. It does include the strategy enforce which emphasize the importance of having privacy policies and a clear communicated priority of privacy. This aligns with the findings from the study by Ayalon [6] showing that organizational culture had a bigger influence on developers decisions on privacy than legal regulations.
Chapter 6
Discussion

6.1 Ethical and societal considerations

In any work, it is important to consider the ethical consequences and impact on society. The research field for this project is privacy, which is a fundamental human right in the United Nations (UN) Declaration of Human Rights [2]. This project intends to contribute to protect the privacy of users of a service by providing guidance and information for organizations. The work of this project highlights the trade-offs for system developers when working in a system applying PbD by the approach described by Colesky, Hoepman and Hillen. These trade-offs can be derived by an organization to make estimations on factors such as the economical cost of implementing PbD to the system. The conclusions of this investigation are valuable input for discussions the positive aspects and potential sacrifices of PbD.

The work from this study is valuable for any organization that handle personal information, it is of extra relevance to organizations managing sensitive personal information. In the long-term, the work from this study could contribute to better privacy solutions. It is of importance in the field of PbD to study how systems are used and the consequences by applying PbD. The conclusions from this study could provide insights about the needs and attitudes of the system developers. These findings could be valuable for the future development of PET.
This project highlights the risk with data analysis but there are many potential benefits of analysing data. Organizations make use of data in many ways that lead to positive results, such as financial gain and to become more sustainable. This can be done by analysing customers’ behavior, understand how their products are used and can be improved from a usage perspective as well as financial and sustainable viewpoint. Data analysis provides a tool for organizations to find their biggest impact and take meaningful actions towards becoming more sustainable.

Another ethical issue of this work is handling of personal information for the study. The data survey and the individual interviews were not performed anonymous. It was communicated to the participants that their answers would be collected non-anonymous. The answers from individuals have and will only be shared as results in aggregated form in consideration of the privacy of participants.

6.2 Critical Evaluation

This study was performed during a time period of 20 weeks. The methodology included a data survey and individual interviews. A total number of 69 persons working in the IT field participated in the data survey. The individual interviews were performed with 11 system developers. The definition of a system developer is not always clear. The role system developer could include different task, employees could belong to multiple departments and have multiple responsibilities. The distinguishing of department and work role was made by the author, with guidance by the work descriptions. The subjective factor remains and it is therefore possible that the statistics from the data survey could differ if the classification of departments and roles was made differently. In addition to this, the number of participants is low. These aspects could be improved by conducting a research in a larger scale.
For the evaluation, system developers from one organization were interviewed. These system developers expressed that assisting users with support is included among their tasks. The system developers anticipated that most exceptions when additional data is needed is debugging support tickets. The responsibilities of a system developer could vary depending on the organization and field of business. An organization of a smaller size could possibly result in a wider range of task for the developer in opposition to a bigger organization that might have a specialized technical support unit. The proposed system model was designed considering the tasks expressed by system developers in the data survey. The tasks included in the role of the system developer is a factor that could differ and lead to different results.

The evaluation of this work were based on the expected experience of system developers. Future research is needed to investigate the real experience of working in the proposed system model. Also, the interviews were conducted non-anonymous which could affect the answers of the system developers.

### 6.2.1 Technical Limitations

There are technical challenges involved with implementing the proposed system model in practice. The major difficulties anticipated are to enable granular access to a database and the technical solution for RBAC.

Many organizations in the industry today use cloud computing platforms to run their applications and store databases. This makes them rely on the functionality of these products. An SQL database supports the possibility to grant access to specific fields in the database but a relational database on the other hand might require bigger changes to the architecture. The specific settings for the cloud computing platform might also introduce challenges for achieving granular access.
The Figure 4.4 illustrates a workflow where access to the production database is granted through the planning tool. This functionality assumes that there would be a program between the planning and the database that fetches information in the planning tool and make changes to role access. There are multiple requirements for this technical solution to work. Firstly, there needs to be technical support to fetch data from the planning tool. The popular tools in the industry today provide application programming interface (API) for integration with other products. Access to the database could be managed by a configuration file.

The system developers where asked in the individual interviews if they believed the proposed system model and processes would be technically possible to implement. All the questioned system developer believed it to be possible, although complicated. The biggest anticipated obstacles would be the investment in time and cost as well as the willingness to replace certain tools and products if needed.

### 6.2.2 Data Quality

One aspect in the evaluation was the data quality. In the proposed system model, the developers would mainly work with fictive data. This would be more privacy-preserving than if the developers would use real user data for development. The test data would be generated to reflect the characteristics of the user group. But the risk of not covering edge cases, not having diverse data and introducing biases are greater with fictive data than real user data.

The data quality of this product would be important for two reasons. Firstly, the development of the product. The system developers mentioned the need for a divergent dataset when developing and testing new features, as well as debugging. The second reason for data quality being important is to provide reliable data for statistical purposes and research.
In the process of designing the proposed system model, k-anonymity was considered to be implemented. But due to the negative impact on data quality, k-anonymity was discarded. There would be a risk that k-anonymity introduces biases into the data that could result in false conclusions in research. By adding, removing or generalizing data entries to achieve k-anonymity, there would be a risk of unconsciously favouring certain characteristics. Since high data quality was a requirement for this product, it was decided to not apply k-anonymity in the proposed system model. k-anonymity could be a potential addition to further increase the privacy if the data quality in a system can be compromised.

6.3 Future work

The results from this study are dependent on the specific needs and requirements of the participating system developers. Although the results of this study can be generalized, an extended study would be needed to provide higher accuracy. Future research is needed to be performed in a larger scale, including multiple organizations in varies fields and sizes which could provide higher accuracy in generalization.

Another future step would be to perform the implementation of the proposed system model and working processes in a real environment. This would be interesting from an engineering aspect and also for the evaluation of the real working experiences. This study relies heavily on assumptions by the system developers and estimations of how it would be to work in the proposed system model. An actual implementation would be needed to evaluate the system model in practice to verify or reject the results produced in this study. In addition to the evaluation of the results, it would also be needed to investigate the technical solution. Performing the steps to implement the system model could disclose technical limitation that affects the system architecture.

This study evaluated the consequences for system developers working in the proposed system model applying PbD in the approach described by Colesky, Hoepman and Hillen [4]. Another possibility in this field would be to investigate the effects on other roles in an organization. The customer support agents in an organization naturally handle personal user data. This work role would be an interesting focus group when evaluating the consequences of working in a system applying PbD with this method.
Chapter 7

Conclusions

The proposed system model was designed with the approach of Privacy by Design in Practice developed by Colesky, Hillen and Hoepman. It consists of the eight privacy design strategies presented by Hoepman [9] and the complementary tactics [4]. From the description of the proposed system model and no implementation, the system developers predicted how it would be to perform their job in the system.

The system developers acknowledge the need for a privacy-preserving system. The performed interviews indicate that the system developer generally have a positive attitude towards the proposed system model. It is anticipated that the proposed system model would decrease productivity slightly. The system model is also expected to bring some frustration and additional workload related to administration of RBAC. Still the system developers predict that the negative consequences will be minor and acceptable trade-offs in comparison to the positive aspects of achieving higher privacy. This was expressed in comparison to the theoretical system model without any privacy actions.
Bibliography


[2] UN General Assembly. “Universal declaration of human rights”. In: **UN General Assembly (1948).**


