

DISSERTATION | DOCTORAL THESIS

Titel | Title

An Intelligent Systems Approach for Supporting Privacy
Awareness in Agile Software Development

verfasst von | submitted by

Guntur Budi Herwanto

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Doktor der technischen Wissenschaften (Dr.techn.)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 786 880

Dissertationsgebiet lt. Studienblatt | Field of
study as it appears on the student record
sheet:

Informatik

Betreut von | Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Dr. Gerald Quirchmayr
o. Univ.-Prof. Dipl.-Ing. Dr. A Min Tjoa

Acknowledgements

First and foremost, I thank Allah SWT, the Most Gracious, the Most Merciful, for His infinite blessings, guidance, and strength throughout my research and study. Without His will and mercy, none of this would have been possible. I also express my deepest gratitude to His messenger, Rasulullah Muhammad (Peace be upon Him), whose teachings and example have been a constant source of inspiration and guidance. *Alhamdulillah* for all that has been granted and accomplished.

I would also like to express my heartfelt gratitude to my supervisors, Prof. Gerald Quirchmayr and Prof. A Min Tjoa, for accepting me as your student and providing guidance while allowing me the freedom to explore. Your trust, mentorship, and support have been invaluable throughout my doctoral journey.

To my wife, Annisa, who has been by my side from the beginning of this doctoral journey until the completion of the final words of this dissertation, thank you for your unwavering support and love.

I am also grateful to all my family in Indonesia, especially my father and mother, for their constant encouragement and prayers for completing my doctoral journey.

To my fellow Indonesian students, particularly my colleagues from the field of Informatics, I am grateful for the discussions we have shared throughout this journey.

To my Indonesian friends in Vienna, fellow travel buddies, Wapena community, I extend my heartfelt thanks for your companionship. Although I cannot mention each of you individually without making this acknowledgment excessively long, please know that your presence has been an integral part of my journey and will always be cherished.

I also want to thank my colleagues at Universitas Gadjah Mada. Thanks to Pak Mardhani, who introduced me to my first supervisor. Furthermore, I appreciate all my colleagues in the Department of Computer Science and Electronics for their encouragement during my time away from the university while pursuing my degree.

I am grateful to my colleagues at WU Vienna, particularly Prof. Marta Sabou and Fajar Ekaputra, for allowing me to join your research group during the final stages of my study. The time and experience I spent at TU Vienna and WU Vienna have been invaluable, and I will undoubtedly carry the lessons learned with me into my future endeavors.

It is worth noting that I have written this thesis in various locations, from the comfort of our apartment during the COVID-19 lockdown to the old libraries of the University of Vienna and the city library. I have also had the pleasure of writing while traveling through Europe which might have been a one-in-a-lifetime experience. Throughout this journey, I have encountered numerous kind people, including junior researchers, people from the conferences, and fellow doctoral students. I am deeply grateful to each of them for their contributions to my academic and personal growth.

Acknowledgements

I believe peer review is essential to the research process, especially for doctoral work. I thank all the anonymous reviewers of doctoral consortia, journals, papers, and conference friends for their valuable feedback.

Finally, I sincerely thank the Indonesia Endowment Fund for Education (LPDP) Indonesia for providing me with the scholarship that made this journey possible. Without their funding, I would not have had the opportunity to pursue my doctoral studies.

Abstract

Privacy by design principles have long set the standards for creating privacy-aware systems. Privacy engineering seeks to bridge the gap between privacy policies and developing systems or technologies. Although privacy engineering is integrated into software and system development life cycles, many methodologies are designed for Waterfall development. This poses a challenge since most software products are now built using agile methods, requiring quick iterations and not capturing the full system scope early on. This research addresses how to integrate privacy engineering into agile software development methodologies effectively.

Privacy can be understood through the lens of contextual integrity, which depends on context, actors, information types, and processing principles. This research aims to embed this definition into privacy engineering, addressing the challenges of agile privacy engineering by using an intelligent system approach.

Agile teams often struggle to identify privacy criteria within user stories and to model systems using data flow diagrams (DFDs). This can slow down security and privacy threat analysis processes. Additionally, there is a lack of concrete guidance for selecting the most suitable privacy patterns for specific scenarios, hindering the effective implementation of privacy measures in agile software development.

The intelligent system approach proposed in this study focuses on several key areas: automatically identifying privacy-related entities in user stories using Named Entity Recognition (NER) techniques, generating DFDs from agile requirements using Natural Language Processing (NLP) and large language models (LLMs), assisting in generating privacy requirements through requirements patterns, and recommending appropriate privacy design patterns using Learning to Rank (LeToR) algorithms. These techniques were evaluated through example implementations and empirical evaluations, demonstrating their feasibility and usefulness for both beginners and practitioners in privacy engineering.

This research contributes to both the application of machine learning and natural language processing techniques in requirements engineering and to the field of privacy engineering. The specific contributions include a comprehensive method for integrating privacy engineering into agile requirements supported by intelligent systems; techniques for identifying privacy-related entities, generating DFDs, and creating privacy requirements within agile contexts; and a recommendation system for selecting suitable privacy design patterns. Empirical evidence from students and experts demonstrates the practical benefits of these intelligent systems, highlighting their potential to streamline privacy requirements engineering in agile environments. This work emphasizes the need for automated tools in privacy engineering, offering a practical solution to bridge the gap in the rapid iterations of agile software development.

Kurzfassung

Die Grundsätze des "Privacy by Design" setzen seit langem die Standards für die Entwicklung datenschutzgerechter Systeme. Privacy Engineering versucht, die Lücke zwischen Datenschutzrichtlinien und der Entwicklung von Systemen oder Technologien zu schließen. Obwohl Privacy Engineering in die Lebenszyklen der Software- und Systementwicklung integriert ist, sind viele Methoden auf eine Wasserfallmodell-Softwareentwicklung ausgelegt. Dies stellt eine Herausforderung dar, da die meisten Softwareprodukte heute mit agilen Methoden entwickelt werden, die schnelle Iterationen erfordern und nicht den gesamten Systemumfang in einem frühen Stadium erfassen. Diese Forschungsarbeit befasst sich mit der Frage, wie Privacy Engineering effektiv in agile Softwareentwicklungsmethoden integriert werden kann.

Datenschutz kann durch die Linse der kontextuellen Integrität verstanden werden, die von Kontext, Akteuren, Informationsarten und Verarbeitungsprinzipien abhängt. Ziel dieser Forschungsarbeit ist es, diese Definition in das Privacy Engineering einzubetten und die Herausforderungen des agilen Privacy Engineering mit Hilfe eines intelligenten Systemansatzes zu bewältigen.

Der in dieser Studie vorgeschlagene intelligente Systemansatz konzentriert sich auf mehrere Schlüsselbereiche: automatische Identifizierung von datenschutzrelevanten Entitäten in Benutzergeschichten („User Stories“) unter Verwendung von Named Entity Recognition (NER)-Techniken, Generierung von Data Flow Diagrams (DFDs) aus agilen Anforderungen unter Verwendung von Natural Language Processing (NLP) und großen Sprachmodellen (LLMs), Unterstützung bei der Generierung von Datenschutzerfordernungen durch Requirement Patterns und Empfehlung geeigneter Datenschutz-Designmuster unter Verwendung von Learning to Rank (LeToR)-Algorithmen. Diese Techniken wurden anhand von Beispielimplementierungen und empirischen Auswertungen evaluiert, um ihre Machbarkeit und Nützlichkeit sowohl für Anfänger als auch für Praktiker im Privacy Engineering zu demonstrieren.

Diese Forschungsarbeit leistet einen Beitrag sowohl zur Anwendung von Techniken des maschinellen Lernens und der Verarbeitung natürlicher Sprache im Requirements Engineering als auch zum Bereich des Privacy Engineering. Zu den spezifischen Beiträgen gehören eine umfassende Methode zur Integration von Privacy Engineering in agile Anforderungen, die von intelligenten Systemen unterstützt werden, Techniken zur Identifizierung von datenschutzrelevanten Entitäten, zur Erstellung von DFDs und zur Erstellung von Datenschutzerfordernungen in agilen Kontexten sowie ein Empfehlungssystem zur Auswahl geeigneter Datenschutz-Design-Patterns. Empirische Belege von Studierenden und Experten zeigen den praktischen Nutzen dieser intelligenten Systeme und unterstreichen ihr Potenzial, die Entwicklung von Datenschutzerfordernungen in agilen Umgebungen zu rationalisieren. Diese Arbeit unterstreicht den Bedarf an automatisierten Tools für das

Kurzfassung

Privacy Engineering und bietet eine praktische Lösung, um die Lücke in rascher Abfolge stattfindenden Iterationen der agilen Softwareentwicklung zu schließen.

Contents

Acknowledgements	i
Abstract	iii
Kurzfassung	v
List of Tables	xiii
List of Figures	xv
List of Algorithms	xix
I. Foundations	1
1. Motivation and Overview	3
1.1. Motivation	3
1.2. Research Questions	4
1.3. Research Method	5
1.4. Thesis Overview	7
1.5. Contributions and Publications	9
2. Background	11
2.1. Privacy	11
2.2. Requirements Engineering	12
2.2.1. Agile Requirements Engineering	12
2.2.2. User Stories as Requirements	13
2.3. Privacy Engineering	14
2.3.1. Privacy Requirements Engineering	14
2.3.2. Privacy Design Engineering	15
2.3.3. Privacy Engineering in Agile Software Development	16
2.4. Intelligent Systems in Software Engineering	16
3. The Current State of the Art in Privacy Requirements and Design Engineering	19
3.1. Introduction	19
3.2. Related Literature Studies	21
3.3. Methodology	22
3.3.1. Research Questions	22

Contents

3.3.2. Search Strategy	23
3.3.3. Inclusion and Exclusion Criteria	24
3.3.4. Study Selection	24
3.4. RQ-Rel-1: The Characteristics of the Studies	25
3.5. RQ-Rel-2: Common Processes Shared Among Methodologies	29
3.6. RQ-Rel-3: Specific Tasks, Techniques, Work Products, and Resources in Each Process	31
3.6.1. Defining Privacy Goals	31
3.6.2. Characterizing the System	33
3.6.3. Privacy Risks and Requirements Elicitation	35
3.6.4. Privacy Risks or Requirements Prioritization and Validation	37
3.6.5. Strategy and solution selection	39
3.7. RQ-Rel-4: The Sequence of Processes in Practice	40
3.7.1. Identifying Initial Steps in Privacy Requirements Engineering	41
3.7.2. Distinguishing Features of Privacy Requirements Engineering Meth- odologies' Sequences	41
3.7.3. Typical Sequences Observed in Privacy Requirements Engineering Approaches	42
3.8. RQ-Rel-5: Towards a Holistic Process	43
3.8.1. A Holistic Process	44
3.8.2. Challenges to a Holistic Process	45
3.9. Closing the Gap: Integrating Intelligent Systems in Agile Software Devel- opment Processes	47
3.10. Threats to Validity	48
3.11. Chapter Summary	48
3.12. The Method Used in the Literature Review	50
3.13. The Number of Sequences of the Process	57

II. Development of an Intelligent System for Privacy Requirements and Design Engineering 59

4. Contributions to Privacy Requirements Engineering	61
4.1. Introduction	61
4.2. Background and Related Work	63
4.2.1. Privacy Requirements Engineering	63
4.2.2. Natural Language Processing Applied to User Story Analysis	65
4.3. The Proposed PRE Model	66
4.3.1. Elicitation of User Stories	67
4.3.2. Identification of Privacy Entities	68
4.3.3. Data Flow Analysis Based on User Stories	70
4.3.4. Data Flow Diagram Generation using Large Language Models	75
4.3.5. Privacy Requirements Generation	76
4.3.6. Guided Threat Modeling	81

4.3.7. Contextualized Privacy Requirements	82
4.3.8. Prioritization of Privacy Requirements	82
4.4. Evaluating the Feasibility of the Developed NLP-based Approach	84
4.4.1. Evaluation of Disclosure Detection	84
4.4.2. Evaluation of Named Entity Recognition	87
4.4.3. Evaluation of Automated DFD Generation	88
4.4.4. Evaluation of Automated DFD Generation with LLMs	90
4.5. Discussion	94
4.6. Threats to Validity	95
4.7. Chapter Summary	97
5. Contributions to Privacy Design Engineering	99
5.1. Introduction	99
5.2. Background and Related Work	100
5.3. Recommending Privacy Design Patterns	101
5.3.1. Datasets	102
5.3.2. Privacy Requirements Classification	104
5.3.3. Learning-to-Rank Privacy Design Patterns	105
5.4. Privacy Design Patterns: Results and Analysis	106
5.4.1. Experimental Setup	106
5.4.2. Model Performance	108
5.4.3. Feature Importance of LeToR	108
5.4.4. Pilot Example	111
5.5. Threats to Validity	112
5.6. Chapter Summary	112
III. Implementation of the PrivacyStory Tool and Evaluation of the Approach	113
6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories	115
6.1. Architecture of the Tool	115
6.1.1. Quality Check of User Stories	118
6.1.2. Disclosure Detection	118
6.1.3. Detecting Privacy-Related Entities	118
6.1.4. Generating Data Flow Diagram	118
6.1.5. Generating Privacy Requirements	119
6.1.6. Guided Threat Modeling	119
6.1.7. Privacy Design Patterns Recommendation	119
6.2. Workflow of the Tool	120
6.3. Chapter Summary	121

7. Application of the Intelligent System	125
7.1. The CamperPlus Project	125
7.2. Automated Quality Check of User Stories	126
7.3. Detecting Privacy Disclosures	127
7.4. Identification of Privacy Entities	127
7.5. Data Flow Diagram Generation	128
7.6. Privacy Requirements Generation	129
7.7. Prioritization of Privacy Requirements	131
7.8. Recommendation of Privacy Design Patterns	131
8. Empirical Evaluation	133
8.1. Experiment Planning	133
8.1.1. Goals	133
8.1.2. Participants	134
8.1.3. Experimental Materials	134
8.1.4. Tasks	137
8.1.5. Experimental Design	138
8.2. Execution	139
8.2.1. Preparation	139
8.2.2. Procedure	140
8.2.3. Deviations from the plan	141
8.2.4. Analysis	141
8.3. Analysis: Student Feedback	142
8.3.1. Descriptive Statistics	142
8.3.2. Qualitative Feedback	142
8.4. Analysis: Practitioners Feedback	145
8.4.1. Demographics of the Practitioners	145
8.4.2. Descriptive Statistics	146
8.4.3. Qualitative Feedback	149
8.5. Results and Discussion	152
8.5.1. Goal Evaluation	152
8.5.2. Findings and Implications	153
8.5.3. Threats to Validity	154
8.6. Chapter Summary	155
9. Conclusion	157
9.1. Summary	157
9.2. Answers to the Research Questions	158
9.3. Comparison to the State of the Art	160
9.4. Reflection on Contribution and Implication	161
9.5. Basis for Future Work	162
9.5.1. Expanding Intelligent Systems in the Holistic Process of Privacy Engineering	163

9.5.2. Exploring Agent-Based Large Language Models for Privacy Engineering	163
9.5.3. A More Robust Empirical Analysis with Practitioners	164
9.6. Concluding Remarks	165
Bibliography	167
A. Appendix	187
A.1. Practitioners Feedback Form	187

List of Tables

1.1. Thesis Outline	8
3.1. Process 1: Defining Privacy Goals	32
3.2. Process 2: Characterizing the System	34
3.3. Process 3: Privacy Risks and Requirements Elicitation	36
3.4. Process 4: Privacy Risks or Requirements Prioritization and Validation . .	38
3.5. Process 5: Strategy and Solution Selection	39
3.6. Privacy Requirements and Design Engineering Processes and Tasks	41
3.7. Lists of Privacy Requirements Engineering Methods reviewed in our study. In the paper where the abbreviation for the method was not available, the author(s) name was used instead.	50
3.8. Tabular display illustrating links from initiating nodes (rows) to receiving nodes (columns). The number is determined by the number of methods that support that connection. The darker color aids in highlighting the most frequent connection.	58
4.1. Data Augmentation Example for Named Entity Recognition to Identify Privacy-Related Entities	70
4.2. Privacy Requirements Mapping	78
4.3. The overview of the data set used in our evaluation [39]. The Ann column denoted the number of user stories annotated by one of the privacy-related entities (either data subject, processing, or personal data attributes). . . .	85
4.4. Privacy Disclosure Detection Evaluation	87
4.5. Experimental Results on Named Entity Recognition	88
4.6. DFD Evaluation based on syntactic and semantic correctness	90
4.7. The Median Scores of Completeness (Cm) and Correctness (Cr)	93
4.8. Correlation Coefficients between Evaluator	93
5.1. Privacy Pattern Example: Who's Listening	101
5.2. Examples of privacy requirements for the three data sets: The first row is an example of a hard goal (unlinkability) and the second row is an example of a soft goal (transparency)	103
5.3. Performance of Learning-to-Rank Algorithm	108
5.4. Recommendation based on LINDDUN [40] Use Cases. The number after privacy requirements indicates the 'id' of the use case in the source. . . .	111
7.1. Example of User Story Quality Check	126
7.2. Entities extracted from Camperplus Project	127

List of Tables

7.3. Privacy Requirement Generation Result	130
7.4. User Stories and Corresponding Privacy Patterns	132
8.1. Technology Acceptance Model (TAM) Analysis Results	143
8.2. Most Useful Features of the Privacy Tool	144
8.3. Demographics of Participating Practitioners	145
8.4. Overall TAM Component Scores, Reliability, and Standard Deviation . . .	146
8.5. Perceived Usefulness (PU) Items	147
8.6. Perceived Ease of Use (PEOU) Items	147
8.7. Attitude Toward Using (ATU) Items	148
8.8. Behavioral Intention to Use (BI) Items	148
8.9. Most Useful Features of the PrivacyStory Tool According to Practitioners	149
8.10. Profile of the participants	150
8.11. TAM Component Scores (Students and Practitioners)	152

List of Figures

1.1. Design Science Research Methodology	6
3.1. The workflow of selecting articles for systematic review based on the PRISMA methodology [126].	22
3.2. RQ-Rel-1.1: Scientific Publication Trend in Privacy Requirements and Design Engineering Research	26
3.3. RQ-Rel-1.2: The Requirements and Design Focus	26
3.4. RQ-Rel-1.3: The Method Orientation	27
3.5. RQ-Rel-1.4: The use of Computer-Aided Tools	28
3.6. RQ1.5: The Mention of Agile	28
3.7. The distillation process begins with the 50 papers selected, followed by four distillation steps. Several selected papers use visual representations to aid in comparative analysis. An example shown in the Figure are based on LINDDUN [40], ISPM [12] and PDPbD [130]	29
3.8. Tasks and subtasks of the privacy requirements and design engineering process	30
3.9. This set of methods presents six starting points for the privacy requirements engineering process.	42
3.10. Network diagram depicting the workflow of privacy requirements and design processes as derived from literature. This directed graph shows processes that precede others. Thicker edges and darker colors indicate processes that are adapted by more methods.	43
3.11. Structure of a Holistic Approach of Privacy Requirements and Design Engineering.	44
4.1. Overview of the proposed workflow for privacy requirements engineering. The white box indicates that the process is performed manually, and the blue box indicates that the process is performed automatically with the support of the NLP module in the yellow box.	66
4.2. The architecture of the Privacy Disclosure classification.	68
4.3. The architecture of the named entity recognition model originally introduced in [70].	71

List of Figures

4.4.	The Workflow of the Automated Data Flow Diagram Generation [71]. The main contribution is highlighted in purple, and the use of other work is highlighted in green. The Robustness Diagram to the Data Flow Diagram can be seen in the connection to the work of Gilson et al. described in detail in Section C.2. The "Process Reason/Value" is explained in Section C.3 and composed of the existing Robustness Diagram from Gilson into DFD. The generation of the diagram is explained in Section C.4.	72
4.5.	The Transformation Rule from RD to DFD [71]	74
4.6.	The method for creating a Data Flow Diagram (DFD) starts with user requirements. From there, it optionally groups functionality before using prompts in LLMs. The output of the prompts is in CSV format, which is then imported into draw.io. This process results in a generated, editable DFD.	76
4.7.	Prompt to Generate the CSV syntax of the Data Flow Diagram.	77
4.8.	DFDs Generated from Large Language Models	92
5.1.	Practical example showcasing the recommendation of PrDPs utilizing the learning-to-rank approach.	102
5.2.	Establishing LeToR ground truth data.	104
5.3.	Sum of Importance	109
5.4.	Average of Importance	109
5.5.	Global Feature Importance (+hadamard, concat)	110
5.6.	Local Explanation Summary (+hadamard, concat)	110
5.7.	Global Feature Importance (baseline)	110
5.8.	Local Explanation Summary (baseline)	110
6.1.	The three screenshots of the PrivacyStory tool. (1) The first screenshot on the top shows a project dashboard with several action buttons and the User Story table. (2) The second screenshot on the bottom left shows an example of an automatically generated DFD along with the DFD triples. (3) The last screenshot on the bottom right shows an example of the unlinkability requirements generated from the first triple.	116
6.2.	Component Diagram of Privacy Story	117
6.3.	Guided Threat Modeling	119
6.4.	Solution Recommendation	120
6.5.	Sequence Diagram of Privacy Story of Privacy Story from User Story to Triples	122
6.6.	Sequence Diagram of Privacy Story from Triples to Privacy Design Patterns	123
7.1.	The Example of Named Entity Recognition on User Stories.	128
7.2.	The Data Flow Diagram for user story:"As a parent, I want to be able to track my child's activity and schedule at camp, so that I can have peace of mind."	129

7.3. The Data Flow Diagram that generated from three user stories that has "child" as data subject	129
8.1. Web interface that participants can view to understand the evaluation steps they need to go through.	135
8.2. The Online Course of Privacy by Design in Indonesian Language (Bahasa). 136	
8.3. The example of a quiz that must be attempted to get to the next module. 136	
9.1. Comparison of Processes among the State of the art. For reference on the codification, refer to Table 3.6	160

List of Algorithms

1.	Processing Reason	74
----	-----------------------------	----

Part I.

Foundations

1. Motivation and Overview

In this introductory chapter, the key components of the thesis are outlined, including the motivation behind the study, the research questions addressed, the research method used, and an overview of the subsequent chapters. Section 1.1 details the motivation for the research. Section 1.2 introduces the main research questions guiding the approaches. In Section 1.3, the design science research method is explained. Section 1.4 provides an overview of the thesis structure, helping the reader navigate and understand the overall theme. Finally, Section 1.5 presents the specific contributions of the research and highlights the key publications related to this thesis.

1.1. Motivation

As we navigate the digital era, understanding privacy has become more complex and multifaceted. In early 2012, an angry father stormed into a Target store in Minneapolis, demanding to speak to a manager. His teenage daughter had received coupons for baby clothes and cribs, which he found inappropriate and disturbing. The store manager apologized, but the father called back a few days later, embarrassed. After speaking with his daughter, he realized Target had been correct—she was indeed pregnant, a fact the store had deduced from her purchasing patterns before her own family knew [77]. This incident underscores a significant transformation in our perception of privacy in the digital age. Historically, privacy was viewed as a sanctuary from unwanted intrusion [56]. However, as technology propels us into an increasingly connected world, our traditional notions of privacy are evolving.

Digital interaction networks can construct narratives based on user activities like likes, purchases, and check-ins. These narratives provide personalized insights but also incorporate a surveillance aspect, enabling entities to observe and monitor our preferences, actions, and aspirations. Nissenbaum introduces the concept of "privacy in context," arguing that our concern lies not with the story being told but with who has access to it and how it is disseminated [120]. Privacy, therefore, is not just an individual narrative but a social one, where information flow is perceived through the lens of context. The Target case vividly illustrates this point. It wasn't the data itself that was troubling, but the context and the narrator—Target, a retail store—rather than someone more personal. This highlights the importance of context in privacy, suggesting that privacy encompasses the various aspects of our lives and the individuals who have access to them, and the way they can be accessed.

Digital products and services are typically created in the software format, and the software development industry faces significant concerns and challenges in addressing

1. Motivation and Overview

privacy issues. To address these concerns, "Privacy by Design" has been introduced as a guiding framework. This approach aims to integrate privacy aspects into products and services from the outset. Privacy Engineering has emerged as a research framework to systematically incorporate these principles into organizational and technical measures [60]. Seven foundational principles underpin Privacy by Design, ensuring that privacy is embedded into the software development process right from the beginning. Privacy engineering is thus integrated into the software development life cycle (SDLC), including the requirements and design phase. The requirements phase can, therefore, be referred to as privacy requirement engineering. Many privacy requirements engineering methodologies rely heavily on a waterfall-style approach, which can be time-consuming and not suited to the agile pace that much of the industry now follows. Researchers have studied these challenges and have clearly articulated the conflicting nature of agile software development (ASD) and privacy engineering [93].

The agile turn makes modeling privacy threats or designing a privacy-aware system more challenging [51]. According to a study on legal compliance within agile teams, the teams did not know how to identify privacy principles in user requirements [26]. They believe it is challenging to incorporate privacy considerations into the development process [26]. To overcome this, researchers are developing methods to suit the agile process [94] or proposing the lean process on privacy engineering [178]. Tool support is also proposed to capture privacy requirements in an agile situation [133].

Intelligent software engineering (ISE) has been a significant component of ASD [136]. Perkusich [136] refers to ISE as "the application of intelligent techniques to software engineering." In addition, they define an intelligent technique as "a technique that explores data (from digital artifacts or domain experts) for knowledge discovery, reasoning, learning, planning, natural language processing, perception or supporting decision-making." The implementation of ISE aims to help better manage and even accelerate the agile process, including software requirements and design [136]. Advances in intelligent techniques, such as natural language processing (NLP), have accelerated the adoption of intelligent techniques in requirements engineering [143, 176]. The potential of ISE, however, has not been realized to assist privacy engineering in ASD. Empirical research on agile teams found that they prefer to be assisted with techniques and tools to perform privacy requirement elicitation [26]. They also have difficulties capturing privacy in textual user stories [26]. There is a clear opportunity to bridge the gap between advancing IS techniques such as NLP and privacy requirement engineering [1]. We hypothesize that IS-enabled tools will be able to overcome these challenges. To the best of our knowledge, there is still a lack of research on implementing IS in privacy engineering.

1.2. Research Questions

The primary goal of this thesis is to assess how effectively an intelligent system can improve the process of establishing privacy requirements and the design engineering process for agile requirements. To achieve this objective, it is broken down into five research questions:

RQ1: What is the current state of privacy engineering research?

Various privacy engineering methods have been proposed by researchers in the field. By identifying common tasks, techniques, work products, and resources among these methods, we can better understand potential areas for improvement. This will also provide a solid foundation to explore further or implement privacy requirements and design engineering.

RQ2: Which aspects of the privacy requirements and design engineering process can benefit from intelligent systems?

Building on the insights from RQ1 and the core motivations behind this research, we aim to pinpoint which parts of the privacy engineering process are particularly time-consuming or complex and therefore could benefit the most from the implementation of intelligent systems.

RQ3: How can an intelligent system help to specify the privacy requirements from an agile requirements artifact?

A critical step in the privacy engineering process is the extraction of privacy requirements. We seek to investigate various intelligent system techniques that can guide privacy engineers more effectively in Privacy Requirements Engineering (PRE) within the context of agile requirements.

RQ4: How intelligent system can help to streamline the process from the privacy requirement into a suitable privacy design pattern?

There are multiple approaches available to privacy engineers, such as Privacy Enhancing Technologies and Privacy Design Patterns. However, the wide range of options can be overwhelming, making it difficult for engineers to choose the best solution for specific privacy requirements. We aim to explore how intelligent systems can assist in this selection process.

RQ5: To what extent are practitioners willing to use intelligent systems in the privacy requirements engineering process?

We aim to assess the practical benefits and effectiveness of intelligent systems in real-world scenarios related to privacy requirements and design engineering. Our goal is to evaluate their utility in supporting the agile development process.

1.3. Research Method

The primary goal of this research is to develop an Intelligent System (IS) that assists agile software teams in integrating privacy considerations into their projects. Effective tool support is essential for this integration. We employ the Design Science Research

1. Motivation and Overview

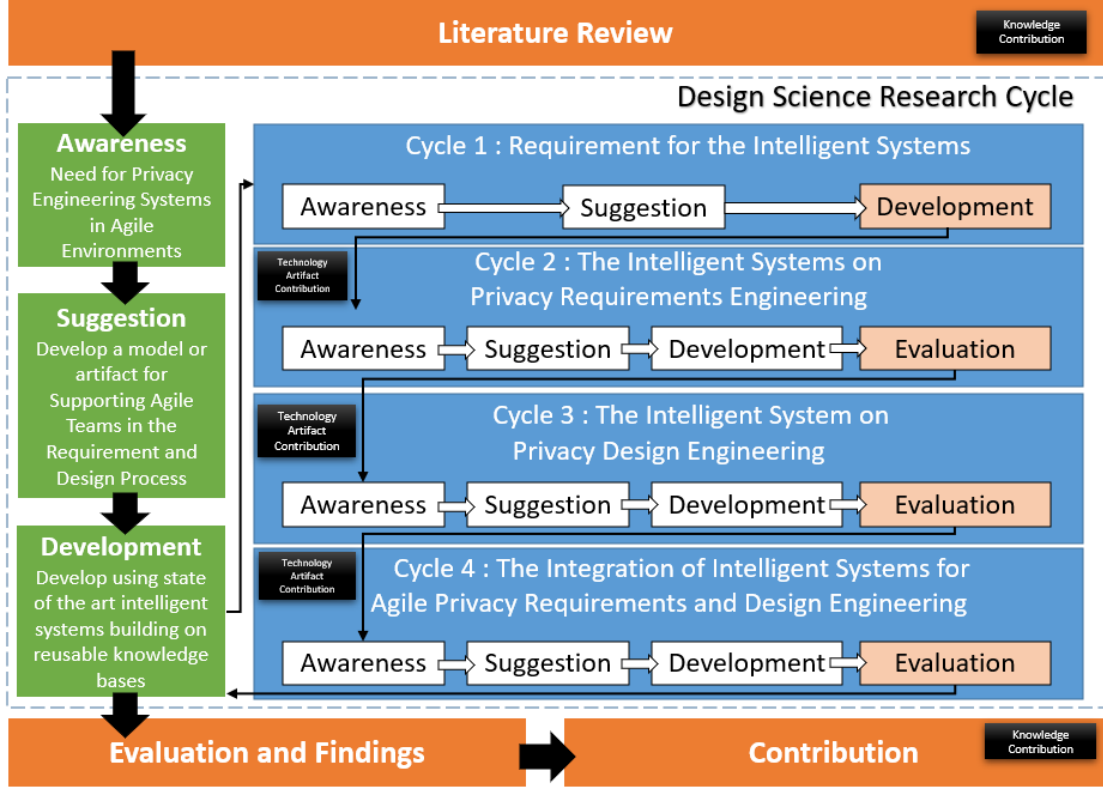


Figure 1.1.: Design Science Research Methodology

(DSR) framework for fostering innovative technological solutions [76, 165], as depicted in Figure 1.1.

DSR is a dynamic and iterative approach, promoting continuous engagement with the problem context [165]. It emphasizes the generation of new knowledge through the systematic evaluation and refinement of the designs, contributing both to academic literature and practical implementations.

The DSR framework is structured around multiple iterative cycles, with the principal cycle marked in green and the IS design sub-cycle in blue. These cycles encompass stages such as Awareness, Suggestion, Development, and Evaluation, progressively deepening our expertise in this domain [161]. Each cycle targets specific research questions, enhancing our overall knowledge.

1. Cycle 1: Requirements for the Intelligent Systems

- **Awareness:** Many existing privacy engineering methods are not suited for agile environments. We identify potential areas where Intelligent Systems can support privacy engineering based on agile requirements.
- **Suggestion:** Conduct a systematic literature review and standardize the process within privacy engineering.

- Development: Create standardized processes and define specific requirements for IS in key areas of privacy engineering.

2. Cycle 2: Intelligent Systems for Privacy Requirements Engineering

- Awareness: Identify specific privacy engineering requirements for IS, building on the findings from Cycle 1.
- Suggestion: Design a machine learning model.
- Development: Implement the model, focusing on natural language processing (NLP).
- Evaluation: Test the model with standard metrics of machine learning to determine its practical feasibility.

3. Cycle 3: Intelligent System for Privacy Design Engineering

- Awareness: Address the needs for privacy design engineering in IS, refining the outcomes from Cycle 2.
- Suggestion: Design another machine learning model.
- Development: Implement the model, with an emphasis on recommendation systems.
- Evaluation: Evaluate the model using standard metrics of information retrieval approach to verify its applicability.

4. Cycle 4: Integration of Intelligent Systems for Agile Privacy Requirements and Design Engineering

- Awareness: Recognize the need for an easy-to-use tool for privacy engineering.
- Suggestion: Develop a tool that integrates the machine learning models from Cycles 2 and 3.
- Development: Implement the tool as a web-based application for user ease.
- Evaluation: Conduct empirical testing with experts and students.

1.4. Thesis Overview

The structure of this thesis consists of three main parts and a total of nine chapters. An overview of this structure is provided in Table 1.1. This table also maps out the specific contributions and research questions addressed in the thesis.

The first part, *Foundations*, includes the first three chapters. Chapter 1 introduces the motivation and outlines the research questions. Chapter 2 offers a comprehensive background, explaining crucial terms and definitions referenced in later chapters. Chapter 3 reviews the literature, focusing on the process of privacy requirements and design engineering, and identifies potential gaps that intelligent systems might address.

1. Motivation and Overview

Table 1.1.: Thesis Outline

Outline	Contributions	Research Questions
Part I. Foundations		
Chapter 1. Motivation and Overview		
Chapter 2. Background		
Chapter 3. State of the Art	<i>The Process of Privacy Requirement and Design Engineering</i>	RQ1, RQ2
Part II. The Intelligent System		
Chapter 4. Contributions to Privacy Requirements Engineering	<i>A Named Entity Recognition based approach for Privacy Requirements Engineering, A model for automatically generating Data Flow Diagrams from User Stories</i>	RQ3
Chapter 5. Contributions to Privacy Design Engineering	<i>A Model for Extracting Privacy Requirements Patterns from User Stories based on NLP techniques</i>	RQ4
Part III. Evaluation		
Chapter 6. PrivacyStory: Tool Support for Streamlining Privacy Requirements and Design Engineering	<i>PrivacyStory: An NLP based tool for supporting Privacy Requirements Elicitation and Selection of Privacy Design Patterns</i>	
Chapter 7. Evaluation and Case Study		
Chapter 8. Empirical Evaluation	<i>Empirical Validation of Models and Supporting Tools</i>	RQ5
Chapter 9. Conclusion		

The second part presents our main contributions regarding the integration of *intelligent systems* in privacy requirements engineering, discussed in Chapter 4, and privacy design engineering, elaborated upon in Chapter 5.

The third part *evaluates* the proposed intelligent system. Chapter 6 introduces the tools developed to incorporate the intelligent system, which are used for empirical evaluation. In Chapter 7, we present a pilot example using an open-source user story to demonstrate the expected output of our method and tool. Chapter 8 details the empirical evaluation conducted with students and experts. Finally, Chapter 9 provides the conclusions, answers the research questions of this thesis, and offers remarks for future work.

1.5. Contributions and Publications

This research has produced several significant contributions, each documented in a series of publications where I am the lead author. These contributions, along with the titles of the publications and the chapters in which they are detailed, are summarized below:

Process Definition for Privacy Requirement and Design Engineering A structured process for Privacy Requirement and Design Engineering has been established and is explored in-depth in Chapter 3.

Herwanto, Guntur Budi, Fajar J. Ekaputra, Gerald Quirchmayr, and A. Min Tjoa. "Towards a Holistic Privacy Requirements Engineering Process: Insights from a Systematic Literature Review." *IEEE Access* (2024).

Integrating Contextual Integrity in Privacy Requirements Engineering: A Study Case in Personal E-Health Applications

Intelligent System on Privacy Requirements Engineering Three notable contributions to privacy requirements engineering are: 1. Development of a Named Entity Recognition-based approach for Privacy Requirements Engineering, enabling the automatic detection of privacy-related entities within user stories. 2. Development of a method that facilitates the automatic generation of Data Flow Diagrams from user stories. 3. Construction of Privacy Requirements Patterns from user stories, enabling automatic generation by leveraging intelligence from Information Systems. These contributions are detailed in Chapter 4.

Herwanto, Guntur Budi, Gerald Quirchmayr, and A. Min Tjoa. "A Named Entity Recognition-Based Approach for Privacy Requirements Engineering." 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW). IEEE, 2021.

Herwanto, Guntur Budi, Gerald Quirchmayr, and A. Min Tjoa. "From User Stories to Data Flow Diagrams for Privacy Awareness: A Research Preview." International Working Conference on Requirements Engineering: Foundation for Software Quality. Cham: Springer International Publishing, 2022.

Herwanto, Guntur Budi, Gerald Quirchmayr, and A. Min Tjoa. "Leveraging NLP Techniques for Privacy Requirements Engineering in User Stories." *IEEE Access* (2024).

1. Motivation and Overview

Intelligent System on Privacy Design Engineering Formulation of a model that facilitates the derivation of privacy design pattern recommendations based on identified privacy requirements, as discussed in Chapter 5.

Herwanto, Guntur Budi, Gerald Quirchmayr, and A. Min Tjoa. "Learning to Rank Privacy Design Patterns: A Semantic Approach to Meeting Privacy Requirements." In International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 57-73. Cham: Springer Nature Switzerland, 2024.

PrivacyStory Tools and Its Empirical Evaluations Introduction of PrivacyStory, an NLP-based tool designed to assist in Privacy Requirements Elicitation and Privacy Pattern Design, along with rigorous empirical validation of the models and tools. The tool is described in detail in Chapter 6, with the evaluations presented in the subsequent chapters.

Herwanto, Guntur Budi, Gerald Quirchmayr, and A. Min Tjoa. "PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories." 2022 IEEE 30th International Requirements Engineering Conference (RE). IEEE, 2022.

Additionally, I have published a doctoral consortium paper outlining the research question of this thesis and an experimental paper on generating DFD as part of the intelligent system. Both papers are single-authored.

Herwanto, Guntur Budi. "An Intelligent Systems Approach for Supporting Privacy Awareness in Agile Software Development." REFSQ Workshops. 2022.

Herwanto, Guntur Budi. "Automating Data Flow Diagram Generation from User Stories Using Large Language Models." In 7th Workshop on Natural Language Processing for Requirements Engineering. 2024.

2. Background

This chapter provides an overview and basic definitions of the key concepts discussed in this thesis. Section 2.1 begins by presenting our chosen definition of privacy. Next, Section 2.2 explores agile software development, focusing on its requirements. Section 2.3 then introduces Privacy Engineering. Finally, Section 2.4 discusses Intelligent Systems and their relevance to software engineering.

2.1. Privacy

Privacy is a multifaceted concept interpreted differently across various disciplines. This thesis is grounded in Helen Nissenbaum’s concept of privacy as contextual integrity [120]. Nissenbaum argues that privacy is not just about controlling information but is deeply connected to the norms and expectations within specific contexts.

Nissenbaum’s theory highlights the importance of context, suggesting that the appropriateness of information flow depends on the situation’s nature. Thus, privacy concerns not only what information is shared but also with whom, how, and under what circumstances. This perspective contrasts with traditional views of privacy that focus primarily on individual control or secrecy.

In the framework of contextual integrity, several entities are crucial in determining how privacy functions within any context. First, the **actors** include the senders, receivers, and subjects of information [120]. Senders disclose or transmit data, receivers accept it and are responsible for its subsequent use or storage, and the subject is the individual or entity the information pertains to. These roles can overlap, especially in personal exchanges. Second, the **information type** describes the nature or category of the information being communicated. Different information types, such as health records or financial details, have distinct privacy implications. Lastly, the **processing** principles dictate the constraints, norms, or rules that guide how the information is used, disseminated, or stored. These entities collectively determine whether a particular information flow maintains or violates the contextual integrity of a situation. For example, in a medical context, the sender might be a doctor, the receiver a specialist, the subject a patient, the information type medical records, and the transmission principles might include patient consent and legal regulations.

Nissenbaum’s approach allows for a more nuanced understanding of privacy by considering social norms and expectations. By recognizing that privacy depends on specific contextual parameters, her theory can analyze complex privacy issues in various domains such as technology, healthcare, finance, and social interactions. This sets the stage for the subsequent chapters that will explore the application of privacy in context within the

2. Background

specific scope of this thesis.

2.2. Requirements Engineering

Requirements Engineering (RE) is a critical initial step in the software design process, focusing on understanding and defining the functionality and constraints of a software system. RE involves eliciting needs from stakeholders, modeling those needs into clear specifications, rigorously analyzing and validating the models, and managing their evolution throughout the software lifecycle [32]. The importance of RE cannot be overstated, as it establishes the foundation for a successful software project by ensuring that the final product meets user needs, functions correctly, and integrates seamlessly within its operating environment. Notably, RE is closely intertwined with privacy by design, as understanding and incorporating privacy requirements from the outset is essential for developing software that respects user data and fosters trust [29].

However, traditional RE practices often face challenges due to communication gaps between developers and stakeholders, leading to misunderstandings and software that fails to meet expectations [48]. Furthermore, the desire for comprehensive upfront requirements gathering can result in "overscoping," burdening projects with unnecessary complexity and hindering their ability to adapt to changing needs [82]. Agile requirements engineering addresses these issues by embracing iterative development, continuous stakeholder feedback, and a focus on delivering value incrementally. Agile RE acknowledges that requirements evolve over time and prioritizes flexibility and collaboration to navigate the dynamic landscape of software development [153].

2.2.1. Agile Requirements Engineering

Agile Software Development is a paradigm that emphasizes flexibility, collaboration, and customer-centricity. Distinct from the traditional waterfall model, where each development phase follows in strict order, Agile methodologies, such as Scrum or Kanban, encourage iterative progress, team collaboration, and continual feedback [105].

The term "agile requirements engineering" is used to define the "agile way" of planning, executing and reasoning about requirements engineering activities [97]. Agile RE diverges from traditional approaches by emphasizing adaptability, collaboration, and continuous refinement of requirements. Instead of attempting to capture all requirements upfront, agile RE recognizes that requirements are likely to change as the project progresses and new insights emerge. This iterative approach allows for a more responsive and flexible development process, ensuring that the software remains aligned with evolving user needs and business goals [144].

Key practices in agile RE include:

- **User Stories:** Agile RE often employs user stories to capture requirements in a concise, user-centric format. User stories describe a specific functionality from the perspective of an end-user, focusing on the value it provides [34].

- **Continuous Stakeholder Involvement:** Agile RE emphasizes close collaboration between the development team and stakeholders throughout the project. Regular feedback loops ensure that requirements are continually refined and validated [153].
- **Prioritization and Incremental Delivery:** Agile RE prioritizes requirements based on their value to the user and the business. By delivering high-priority features incrementally, agile RE enables faster time-to-market and allows for early user feedback [144].
- **Embrace Change:** Agile RE acknowledges that change is inevitable in software development. By maintaining a backlog of requirements and regularly reviewing and adjusting priorities, agile RE can effectively respond to changing needs and market conditions [82].

By adopting agile RE practices, software development teams can improve communication, reduce waste, and deliver software that better meets user needs. However, successfully implementing agile RE requires a shift in mindset and a commitment to collaboration, transparency, and continuous improvement.

2.2.2. User Stories as Requirements

In agile software development, user stories have emerged as a powerful and efficient method for capturing user requirements. Contrasting with traditional, highly detailed requirement specifications, user stories concisely encapsulate specific user needs. This approach promotes a shared understanding and facilitates ongoing collaboration between the development team and stakeholders throughout the development process [34].

To ensure clarity and consistency in user story formulation, Cohn [34] recommends using a semi-structured template:

As a [type of user], I want [an action] so that [a benefit/a value].

This template consists of three core components:

1. **User Role:** This element identifies *who* benefits from the described functionality, often representing the end-user or a specific persona with distinct needs and behaviors.
2. **Function/Action:** This part focuses on *what* the user wants to achieve, describing the desired action or feature from their perspective using clear and concise language.
3. **Reason/Benefit:** Perhaps the most crucial aspect, this element explains *why* the user desires this functionality, clarifying the value proposition or the problem they aim to solve.

Consider the following example:

2. Background

As an **online shopper**, I want to be able to **filter products by size** so that
I can quickly find products that fit me.

In addition to the core template, user stories often include **acceptance criteria**, which provide testable conditions that must be met for the story to be considered complete [34]. These criteria might specify details such as the range of sizes to include in the filter, the display format of filtered results, or error handling for unavailable sizes. Research supports the effectiveness of user stories in agile environments. Lucassen et al. [98] found a positive sentiment towards user stories within the software development community and emphasized the importance of structured approaches. Their study highlights how using templates for consistent formatting and adhering to quality guidelines (e.g., the INVEST mnemonic) enhances the benefits of user stories, leading to improved productivity and work quality. By focusing on user needs, promoting collaboration, and providing flexibility for iterative development, user stories have become a cornerstone of agile requirement practices.

2.3. Privacy Engineering

Privacy engineering is the process of systematically integrating privacy into the software development life cycle (SDLC) and business processes. Privacy engineering consists of a collection of principles, methods, tools, and best practices for enhancing privacy [41]. It embodies the principle of "privacy by design", where the main goal is to ensure that its principles are integrated into products and services from the very beginning. Privacy engineering aims to cover all aspects of the SDLC, but in this research, we only focus on the requirements and design steps. In this chapter, we will introduce the general concept of privacy requirements and design. Since we believe that a more detailed understanding of this concept is crucial, we will present the process in more detail in the following sections.

2.3.1. Privacy Requirements Engineering

Privacy Requirements Engineering (PRE) can be defined as the systematic approach to eliciting, analyzing, documenting, and managing privacy-specific requirements in software systems. There are numerous techniques for designing privacy requirements. Notario et al. [122] considered it from two perspectives: a goal-oriented and a risk-based perspective. The goal-oriented approach focuses on extracting principles and establishing them as the system needs to meet. The goal of privacy or data protection can be derived from privacy principles and legal requirements. Hansen et al. [65] defined six data protection goals, which consists of *Confidentiality*, *Integrity*, *Availability*, *Unlinkability*, *Transparency*, and *Intervenability*. The Confidentiality, Integrity, and Availability (CIA) Triad is a commonly used framework for evaluating and defining security requirements, and is considered a cornerstone of information security. [152]. In the context of data protection, the CIA Triad can be redefined to meet specific requirements, such as privacy and data protection

regulations [111]. These six goals, along with the additional goals, are used by Meis and Heisel [111] to develop a PRE method under the ProPAn method.

Meanwhile, the risk-based approach was more concerned with the threat that could arise from the movement of assets and data in the system. Privacy threat modeling is one discipline that uses a risk-based approach to address privacy. Deng et al. [40] present LINDDUN, a comprehensive privacy threat modeling technique. LINDDUN is a mnemonic that represents a potential privacy concern in a system under consideration: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of Information, Unawareness, and Non-compliance. Prior to analyzing the threat, the analyst should provide a data flow diagram of the system (DFD). Each component of the DFD can then be linked to multiple threats in LINDDUN.

Privacy Impact Assessment (PIA) is also part of the risk-based approach. As with LINDDUN [40], the risk-based approach should begin with system characterization, which often includes asset identification and system modeling. Oetzel et al. [123] presented a seven-step process for eliciting privacy threats and documenting the PIA. This process can also be viewed as goal-oriented, as its second step incorporates the use of privacy goal catalogs, which provide a set of predefined privacy goals to guide the PIA process and ensure that all relevant privacy considerations are taken into account. Thus, it is important to recognize that an effective PRE strategy should incorporate both a goal-oriented and a risk-based approach.

Combining risk-based and goal-based approaches can be of greater benefit [121]. The goal-oriented approach, which is easier for a novice privacy engineer to follow, is the first step in reducing uncertainty in privacy engineering. The remainder of the potential risk is then addressed using a risk-based approach. The combination of goal-oriented and risk-based approaches provides a more comprehensive approach to PRE. P-STORE [10], the latest PRE which combines both approaches begins by establishing and prioritizing privacy objectives based on organizational goals. Prioritization is necessary because of the lack of ability to meet all privacy goals simultaneously [65]. For example, if we value confidentiality goals, unlinkability goals will be overridden [111]. Once the goal has been specified, the requirements engineer can select the critical asset that will form the basis of the privacy threat analysis. These threats are mapped and prioritized in LINDDUN to become the ultimate privacy requirement to be addressed.

2.3.2. Privacy Design Engineering

Privacy Design Engineering builds upon privacy requirements engineering by transforming privacy requirements into concrete design elements and workflows. This ensures that the software actually incorporates privacy principles into its design [78]. Several approaches facilitate this integration. One approach involves the direct incorporation of privacy-enhancing technologies, such as differential privacy and homomorphic encryption, or the application of privacy design patterns [37,40]. Another approach emphasizes the adoption of privacy-aware architecture [139], which involves structuring and organizing the software system with a primary focus on privacy. This architectural approach guarantees that the structural components, their interactions, and the guiding principles of the system design

2. Background

prioritize the protection and confidential handling of user data.

2.3.3. Privacy Engineering in Agile Software Development

An approach that relies on extensive planning, such as ProPAN [111], is considered unsuitable for ASD [111]. The fundamental difficulty for ASD, such as Scrum, is the lack of system-wide planning. The PRIPARE method proposes a managerial approach to integrate its approach with ASD. The first is through an incremental approach, which means that privacy planning is only considered in its current iteration. In addition, privacy planning should be included prior to the start of the sprint. PRIPARE also considered including privacy in a dedicated sprint. In contrast to PRIPARE, Peixoto [132] aims to provide tool support to agile teams by guiding them through the specification of privacy requirements. However, extensive knowledge of privacy is required to fill out the form provided by the tools.

Agile methodologies encourage frequent communication between teams. This can be achieved in a variety of ways, one of which is the use of card games, an engaging and interactive approach [150, 171]. An example of this is Threat Poker [150] and LINDDUN GO [171], tools that empower Scrum teams to identify potential threats during the game and subsequently assess the level of risk and the corresponding effort required to counteract these threats [150]. This card game could also serve as an educational instrument, particularly for those team members with limited security and privacy expertise [171]. However, this strategy presents a challenge in that each team member needs to possess the requisite knowledge of potential threats associated with a given user story. In addition, practitioners caution that these tools may not be appropriate for more complex microsystems, highlighting the need for careful tool selection and adaptation based on the project's unique context and requirements.

The current literature presents several limitations in addressing PRE problems in ASD. Despite the comprehensive approach offered by ProPAN [111], it still requires a huge workload of careful planning and expertise to follow its plan-driven steps. LINDDUN needs a significant amount of time for modeling the system [40], also in-depth privacy expertise is needed to specify accurate privacy criteria [132]. The lightweight approaches offered by Threat Poker and LINDDUN GO may not provide a comprehensive view of the system, especially for complex systems. These limitations emphasize the need for a more comprehensive and automated approach to PRE, one that leverages the power of intelligent systems to enhance the capabilities of human experts and streamline the process of eliciting, analyzing, and mitigating privacy risks.

2.4. Intelligent Systems in Software Engineering

Perkusich et al. [136] provide a clear definition of intelligent systems in the context of software engineering. These systems leverage "intelligent techniques" that harness data from digital sources or human experts to support various processes like knowledge discovery, reasoning, learning, planning, natural language processing, perception, and

decision-making [136]. Prominent examples of such intelligent techniques include natural language processing (NLP) [1], information retrieval (IR), data mining, fuzzy logic, and various machine learning methods.

Rather than being fully autonomous, intelligent systems in software engineering often blend human and artificial intelligence [172]. While AI assists and sometimes automates tasks, reducing human effort, the expertise of humans remains essential. Human knowledge frequently guides the design of AI systems, and the outcomes generated by AI typically undergo human review. This interplay between human and artificial intelligence creates a feedback loop, enhancing both. The synergy between human expertise and intelligent tools is further underscored by cooperative testing and analysis processes involving a blend of human-tool and human-human interactions in software engineering.

One specific application of intelligent software engineering techniques is using NLP to enhance the handling of user stories. The consistent structure of user stories allows NLP to assist with tasks such as creating models, identifying ambiguities and defects, understanding the structure of the story, and promoting traceability [143]. For example, Ahmed et al. [3] applied NLP to identify quality attributes from user stories by using regular expressions to match the user story pattern with quality attributes. However, this study did not consider security and privacy criteria. In terms of security, NLP has been used to help analysts uncover security-related attributes in product backlogs, as shown in the study by Galster et al. [50].

The use of NLP in detecting privacy-related information in user stories was demonstrated by Casillo et al. [27] and Herwanto et al. [70]. They utilized NLP to assess the potential disclosure of personal data in user stories and its compliance with privacy requirements. Privacy-relevant keywords and entities, such as personal data, data subjects, and processing entities, were considered in the analysis [27, 70]. Herwanto et al. [70] also employed NLP techniques, specifically Named Entity Recognition, to extract privacy entities from user stories. They noted the potential for further privacy analysis through the examination of user stories. Nevertheless, none of this work is integrated into the full process of Privacy Requirements Engineering (PRE).

System characterization is also a main challenge when conducting privacy impact assessments [123] or privacy threat modeling [171]. Although not explored specifically in the security and privacy domain, the automatic modeling of the system by NLP sourced from user stories has been investigated by some researchers [143]. Data Flow Diagram (DFD) is the main model used in privacy threat modeling [40]. Recent work by Herwanto explores the possibility of generating a DFD directly from the text of user stories [71]. Aside from DFD, transforming user stories into conceptual models is one of the tools explored by NLP [59]. Due to the time-consuming nature of system characterization, an automated modeling process is essential for agile teams. These studies have not yet been integrated into the PRE process. Our aim is to streamline these processes and integrate them into the full cycle of PRE to elicit a comprehensive list of privacy requirements. Additionally, we provide a tool that allows for human intervention in cases where inaccuracies in the NLP process may occur.

3. The Current State of the Art in Privacy Requirements and Design Engineering

This chapter presents the state of the art in privacy requirements engineering. A review of 50 papers was conducted, common tasks and sequences were analyzed, and a typical sequence that could be enhanced with an intelligent system was identified. This work has already been published in a journal article [68].

3.1. Introduction

As technology increasingly pervades our daily lives, safeguarding privacy in the design and development of systems and software becomes of utmost importance [160]. The widespread use of technology and the internet has resulted in a vast accumulation, retention, and processing of personal information. Therefore, it is crucial to implement measures that secure this sensitive data. Failure to consider privacy in system design can have severe consequences, including loss of trust, reputational damage, and legal penalties [45]. Therefore, organizations must take necessary measures to ensure that privacy is protected and duly considered in the design and development of systems that collect and process personal data [45]. To address this challenge, privacy engineering has emerged as a research framework that aims to provide methods, tools, and frameworks for incorporating privacy into organizational and technical measures [156]. This involves integrating privacy considerations into the software development life cycle (SDLC), from the requirement gathering phase, through to the final implementation of the system [156].

The process of privacy engineering in the software development life cycle (SDLC) begins with identifying and specifying privacy requirements [11, 45]. This phase presents significant challenges due to the intricate nature of privacy laws and the task of striking a balance between privacy and other competing requirements. Effective tools and techniques are necessary for designing and implementing privacy-preserving systems [24, 111]. Privacy engineering encompasses a range of approaches, including using regulations to define privacy requirements [22] and analyzing organizational assets for privacy needs [13]. System diagrams can be particularly useful in identifying and addressing privacy concerns, as well as in threat modeling and implementing a risk-based approach [111].

The goal of privacy requirements engineering is to ensure a clear definition and understanding of a system's privacy requirements, integrating them into the system's design and development [24]. This often involves conducting risk analyses, threat analyses, and privacy impact assessments (PIAs) to identify and mitigate potential privacy risks [87, 123].

3. The Current State of the Art in Privacy Requirements and Design Engineering

Following the identification and definition of privacy requirements, the next step is to design solutions that meet these requirements, a phase known as privacy design engineering [78]. This may include selecting and implementing specific technologies or approaches, such as encryption or anonymization, to protect user privacy and data [24]. Additionally, a risk-based approach helps align privacy designs with the principles of risk management [40, 123].

Despite various practice methods, there is still no unified standard or common procedure that combines these methods into a single, comprehensive process. Establishing a standardized process for privacy engineering, especially during the requirements and design phase, is essential. This standardization would assist development teams in ensuring a complete approach to privacy in their system design.

Previous research on privacy engineering processes, particularly by Caiza et al. [24], focuses on the reusable aspects of privacy engineering, including the process itself. However, the study doesn't provide a clear sequence of steps that can be easily followed, specifically in the areas of privacy requirements and design processes. This gap in comprehensive and accessible guidelines is the primary motivation for our study. Specifically, we aim to answer the following *main research question*:

What are the shared processes across different methodologies in privacy engineering, and can these processes be integrated to create a comprehensive holistic to privacy requirements engineering?

To answer this research question, we conducted a systematic literature review (SLR) to analyze the process of the method in depth. Based on the results of the SLR, we provide two main contributions:

1. *A Holistic Process for Privacy Requirements Engineering*: This process includes a detailed sequence of tasks, techniques, work products, and resources for each phase of the process.
2. *Insights into future research directions*: First, the characteristics of the privacy requirements engineering study landscape are highlighted, and based on this information and the holistic process, the challenges for future research directions are presented.

The first contribution is geared towards practitioners who aim to implement a comprehensive approach to privacy requirements engineering. The second contribution extends beyond the practical application to explore the broader landscape of privacy requirements engineering research. Furthermore, both of these contributions are beneficial for research in privacy requirements engineering. By providing a well-defined process, it lays a foundation for process benchmarking and comparison. Researchers can use this framework to examine existing privacy requirements engineering methodologies, identify gaps, and explore areas for improvement.

The structure of this chapter is organized as follows: Initially, Section 3.2 reviews existing literature relevant to our study. Next, Section 3.3 describes our systematic

literature review (SLR) methodology, including the formulation of our research questions (RQs). These RQs are explored in detail across five distinct sections: Section 3.4 covers the characteristics of the study, Section 3.5 explores the common processes identified, Section 3.6 delves into specific details of these processes, Section 3.7 discusses the typical sequence of these processes, and Section 4.5 presents a holistic overview of the process along with its challenges. Section 4.6 addresses potential validity threats to our study. Finally, this chapter concludes in Section 4.7, where we summarize our key findings.

3.2. Related Literature Studies

The field of privacy engineering has garnered significant attention since its introduction [61]. As a result, multiple systematic literature reviews have been conducted on the subject. Three distinct literature studies have been identified, each with its own focus.

The first study attempts to define Privacy by Design (PbD), identify privacy goals, and examine the various approaches to this topic [47]. This study provides a coherent definition of PbD, lists the privacy goals discussed in the literature, and presents various tools, methods, models, and patterns for PbD implementation. The study shows that the existing literature often discusses PbD theoretically. The study shows no holistic, universally applicable solutions, which this chapter aims to uncover.

Various methodologies within privacy engineering contribute to a body of knowledge that includes methods, tools, privacy knowledge bases, models, documentation, and more, all aimed at helping engineers design privacy-friendly systems [24]. The systematic mapping of Caiza et al. [24] highlights privacy patterns, reference models, and methodologies as the most frequently recurring knowledge bases. The authors suggest that there is an urgent need for further research and the creation of replicable components for designing systems with privacy in mind. They highlight the significance of automated tools for wider industrial implementation.

The above discourse on the second study leads to the third study, which examines the automated methods used in the privacy engineering [1] requirements domain. Aberkane et al. [1] conducted a systematic mapping study in which they investigated the automation of regulatory and compliance of GDPR using natural language processing (NLP). Their results point to potential ways to bridge the existing gap between these domains, specifically by using NLP methods to automate manual RE tasks, with a focus on linking GDPR and RE. These findings motivate our first set of research questions.

The literature reviewed underscores the need for universally applicable, comprehensive solutions. Therefore, this thesis focuses on defining the universal privacy requirements engineering process based on a systematic literature review, a comprehensive review of literature that has not been previously conducted. Simultaneously, we will provide a granular technique, work products, and resources that can be utilized in every process. Our objective also includes deriving a coherent sequence for the process based on the literature we've reviewed. This convergence of insights provides a new perspective not found in previous literature studies.

3.3. Methodology

In this section, we review the details of our systematic literature review, following the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) criteria [126]. We highlight the research questions we aim to answer and explain the steps we took to collect and assess relevant literature.

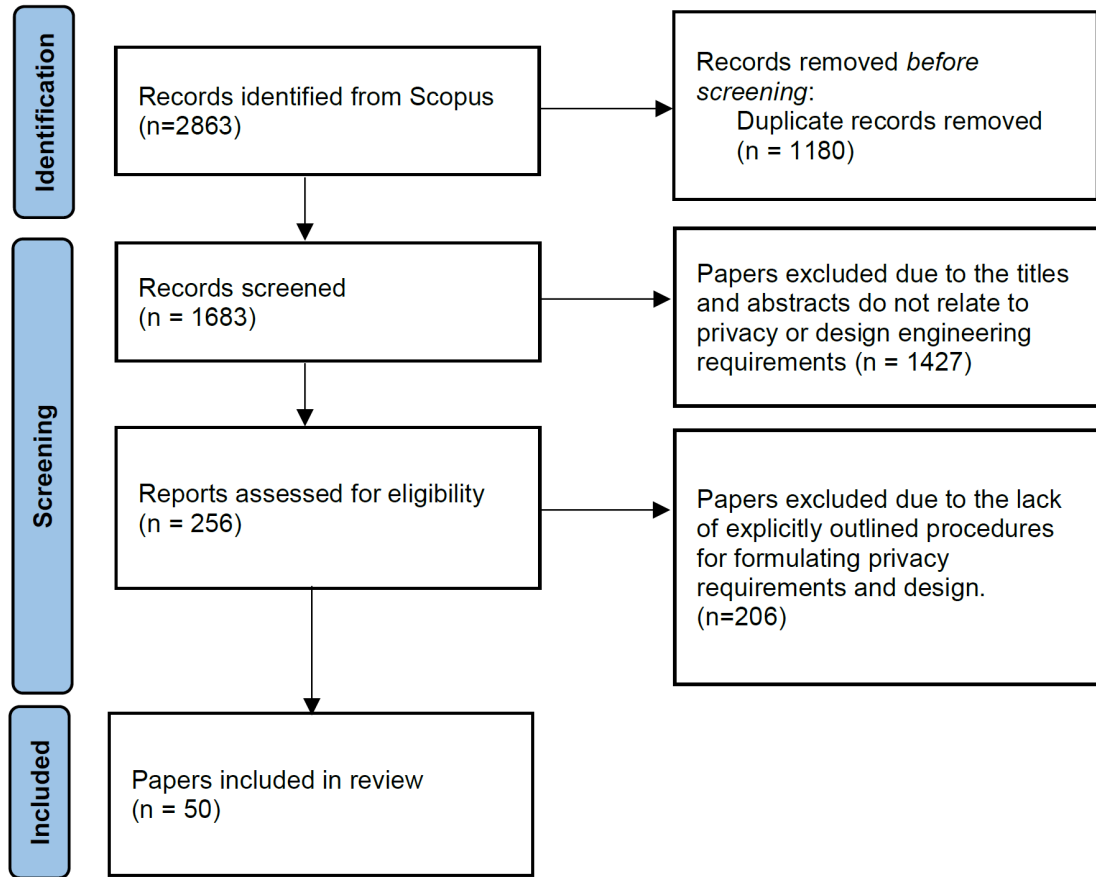


Figure 3.1.: The workflow of selecting articles for systematic review based on the PRISMA methodology [126].

3.3.1. Research Questions

Our literature study is particularly focused on addressing these specific research questions:

RQ-Rel-1 The Characteristics of the Studies:

RQ-Rel-1.1 How has the scientific publication in privacy requirements and design research evolved?

RQ-Rel-1.2 Do these studies concentrate exclusively on requirements, or do they also explore design elements?

RQ-Rel-1.3 What primary methodology is adopted in privacy engineering research?

RQ-Rel-1.4 Do the papers reference the utilization of computer-assisted tools?

RQ-Rel-1.5 Is there any indication of agile methodologies being applied in the studies?

RQ-Rel-2 Which processes are commonly shared among these methodologies?

RQ-Rel-3 Which are the specific tasks, techniques, work products, and resources associated with each identified process?

RQ-Rel-4 What sequence do these processes follow in practice?

RQ-Rel-5 What does the sequence look like for a holistic privacy requirements engineering process?

The research questions prefixed by RQ-Rel indicate that this set of research questions is in our design science methodology's "Relevance" cycle. The first set of research questions seeks to reveal general trends in privacy requirements engineering methods (RQ-Rel-1.1, RQ-Rel-1.2, RQ-Rel-1.3) and the challenges researchers encounter when employing computer-based tools (RQ-Rel-1.4) and integrating agile methodologies (RQ-Rel-1.5). The output of RQ-Rel-2 might be critical in implementing a standardized approach that ensures coherence and compatibility among distinct systems and control environments. RQ-Rel-3 delves into the details of individual processes. Understanding the details enables the identification of best practices, key resources, and effective techniques crucial in creating a comprehensive standard. The goal of RQ-Rel-4 is to examine the privacy requirements of engineering workflow, potentially revealing which researcher-followed sequences indicate agreement with its significance. It can also determine whether a standardized sequence is viable or should be adapted to specific contexts. Finally, RQ-Rel-5 aims to synthesize the findings from the previous research questions to propose a holistic privacy requirements engineering process that incorporates the identified best practices, techniques, and sequences.

3.3.2. Search Strategy

The literature search for our study was primarily carried out using the Scopus indexing database, which offers a broad and trustworthy collection of research pertinent to our study objectives [14]. Choosing Scopus as our main data source is a methodologically robust decision, ensuring access to a wide and relevant dataset, notably including works from IEEE and Springer. This decision is supported by Scopus' established reputation for high-quality and exhaustive content coverage, making it an ideal tool for the initial phase of our literature review process [14].

3. The Current State of the Art in Privacy Requirements and Design Engineering

The search strategy was developed and implemented using the following keywords and Boolean operators: ("privacy" OR "data protection" OR "GDPR") AND ("engineering" OR "requirement" OR "design" OR "architecture"). The inclusion of the keyword GDPR is motivated by its significant impact on generating a large amount of research and its recognition as a fundamental law for data protection. The keyword "architecture" targets the privacy design aspect of the overall software design, which is typically in the software architecture domain.

Our search was specifically targeted at keywords in the title, using the Scopus API through the "Publish or Perish" software¹. This approach allowed us to focus our literature review on relevant titles. We also aimed for a wide range of years of publication between 1970 and 2023.

3.3.3. Inclusion and Exclusion Criteria

We used several inclusion and exclusion criteria for the methodological selection of relevant papers. This allowed us to ensure that the literature included in our review was highly relevant to our research objectives. The rationale and specifics of these criteria are explained in more detail below:

Inclusion Criteria

(1) The paper presents a method for eliciting requirements and designing a privacy-friendly software system. **(2) The paper is a primary study**, which offers sources of the research, providing methodology of privacy engineering and findings without being reinterpreted through secondary analyses. **(3) The paper is written in English.** Due to linguistic capabilities and to ensure uniformity in our analysis, we chose to focus on papers written in English.

Exclusion Criteria

(1) Sole Emphasis on Proposing Privacy-Enhancing-Technologies: While privacy-enhancing technologies are vital, our primary interest was in the privacy engineering processes. Papers merely introducing such technologies without detailing the architectural or requirement aspects weren't within our review's scope. **(2) Papers that simply listed privacy requirements without explaining the reasoning or derivation process.** **(3) Papers on secondary or tertiary research findings:** We prioritized primary studies, avoiding potential biases from aggregated or interpreted studies.

3.3.4. Study Selection

We collected all the studies from our search, eliminating any duplicates, and uploaded them to the annotation tools which presented their titles and abstracts. Inclusion/exclusion

¹<https://harzing.com/resources/publish-or-perish>

3.4. RQ-Rel-1: The Characteristics of the Studies

criteria were applied to identify applicable studies. In cases where the decision could not be determined solely based on the title and abstract, the full text was reviewed.

A total of 2863 studies were identified from our search queries, of which 1683 remained after eliminating duplicates. A preliminary review of titles and abstracts reduced the number of studies to 256. It is noteworthy that most of these studies focused on privacy-enhancing solutions, which was not the main focus of our systematic literature review. Further review led to the identification of 50 studies that extensively discussed requirements engineering processes and presented privacy-aware design.

The first author conducted the study selection to ensure consistency and streamline the evaluation process. The result is further validated through an independent review of selected studies by the other authors, ensuring thoroughness and reliability. The approach was supported by the study protocol summarized in Figure 3.1. All discrepancies were jointly addressed, ensuring a rigorous and credible selection outcome.

3.4. RQ-Rel-1: The Characteristics of the Studies

We identified five key characteristics to highlight potential areas for future research: These are **(1) Annual Research Trends**: This aims to track the shifting focus within the field, particularly how changes in the regulatory landscape can impact research directions. **(2) Focus on Requirements and Design Processes**: Evaluate the depth and scope of the privacy requirements and design engineering processes. **(3) Process Orientation**: Examining whether the approach is geared towards goal-setting, risk assessment, or other methodologies, thereby underscoring the methodological variety in the field and its practical consequences. **(4) Use of Computer-Assisted Tools**: Emphasized in prior studies [123, 139] for their crucial role in managing the growing complexity of systems. [123, 139] given the increasing complexity of systems, and **(5) Flexibility Towards Agile Methodologies**: Considering previous critiques on the field's adaptability to the quick and flexible nature of agile development practices [62, 93].

RQ-Rel-1.1: How has the annual trend in privacy requirements and design research evolved?

The temporal analysis of the selected studies, illustrated by the publication trend over the years in Figure 3.2, shows different phases of research interest. The first years, from 2002 to 2013, show little interest with a limited number of publications. This is followed by the years from 2014 to 2017, with modest but steady research activity. However, from 2018 onwards there has been a notable increase in research publications. This increased interest coincides with the announcement and subsequent implementation of GDPR in 2016 and 2018 respectively. It is plausible that the GDPR, as a key regulation in the field of data protection and privacy, has prompted the academic community to engage more intensively with data protection-related topics. In the diagram, the year 2022 is highlighted as the last year. However, this analysis includes a study from the beginning of 2023, as the literature review was conducted in this period and we excluded it only for this annual distribution analysis.

3. The Current State of the Art in Privacy Requirements and Design Engineering

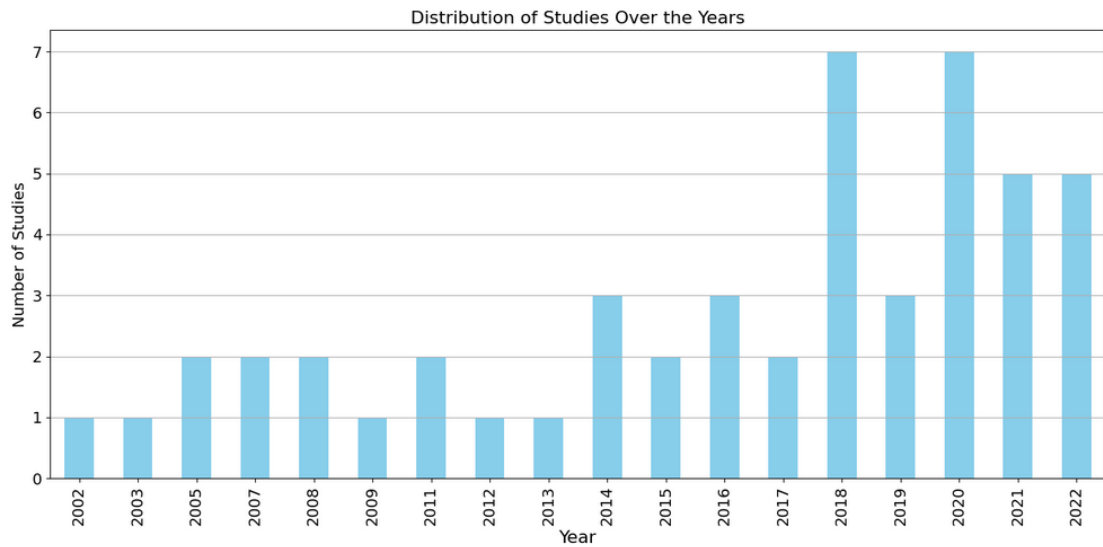


Figure 3.2.: RQ-Rel-1.1: Scientific Publication Trend in Privacy Requirements and Design Engineering Research

RQ-Rel-1.2: Do these studies concentrate exclusively on requirements, or do they also explore design elements?

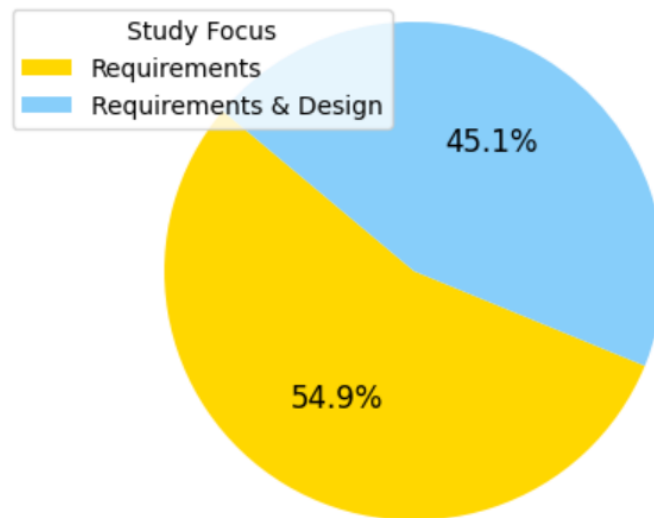


Figure 3.3.: RQ-Rel-1.2: The Requirements and Design Focus

The paper's content analysis reveals that a significant portion of the research focuses on identifying privacy requirements (54.9%). This highlights the need to precisely define, comprehend, and validate these requirements to establish a strong basis for the following

3.4. RQ-Rel-1: The Characteristics of the Studies

stages of system development. This guarantees the consistent implementation of privacy principles from the beginning.

Nevertheless, there is also a noteworthy set of studies that adopt a more comprehensive approach, encompassing both requirements and design aspects (45.1%). These studies not only focus on privacy requirements but also proceed with their implementation into tangible designs, encompassing technical measures [40,90], system architecture [138], and organizational measures [122].

RQ-Rel-1.3: What primary methodology is adopted in privacy requirements engineering research?

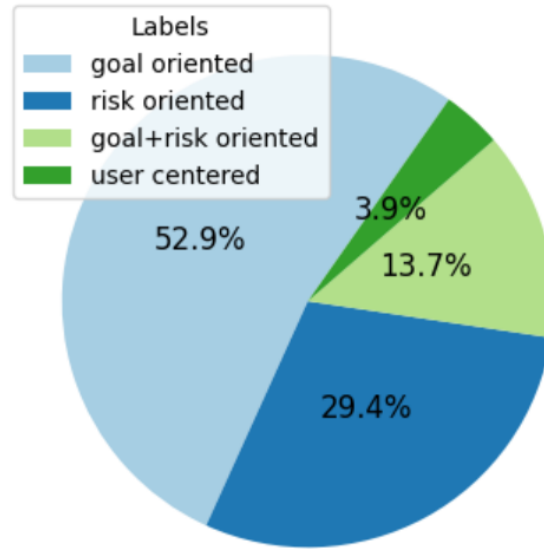


Figure 3.4.: RQ-Rel-1.3: The Method Orientation

The literature review provides insights into various approaches adopted for ensuring privacy. A significant portion, 52.9%, are guided by specific privacy goals stemming from either regulations, standards, or stakeholder expectations. Conversely, 29.4% employs a risk-based strategy, concentrating on identifying and mitigating privacy risks in relation to the existing system's features and assets. An integrated approach is used by 13.7%, merging both goal-oriented and risk-centric methods to leverage the benefits of clearly defined objectives while preemptively addressing potential threats. A small fraction, 3.9%, doesn't fit neatly into the goal or risk categories. These approaches are distinct in that they engage users directly to gather requirements or solutions, leading us to define them under the umbrella of a user-centered approach [83,117,142]. The distribution of these methodologies is illustrated in Figure 3.4.

RQ-Rel-1.4: Do the papers reference the utilization of computer-assisted tools?

The use of computer-aided tools was mentioned in 21.6 percent of the studies, while the

3. The Current State of the Art in Privacy Requirements and Design Engineering

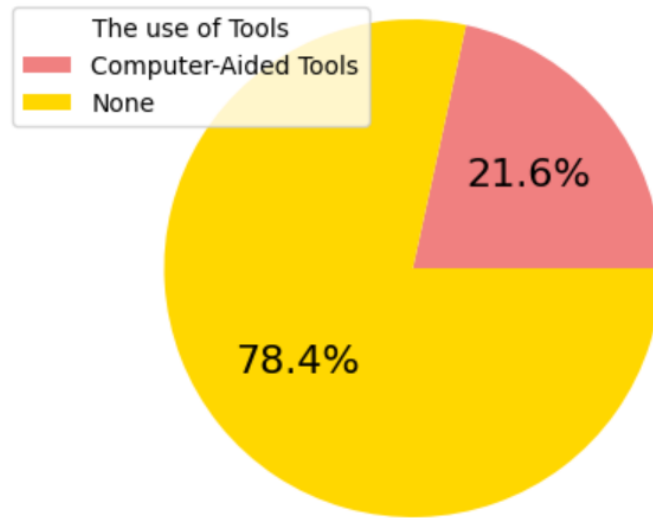


Figure 3.5.: RQ-Rel-1.4: The use of Computer-Aided Tools

majority did not. It indicates that the inclination to use tools exists but is underutilized. These computer-aided tools can range from web forms [107, 123, 131, 163] to modeling tools [49, 107, 112], or even use natural language processing techniques [71, 72, 157]. The distribution of the use of computer-aided tools is depicted in Figure 3.5.

RQ1.5: Is there any indication of agile methodologies being applied in the studies?

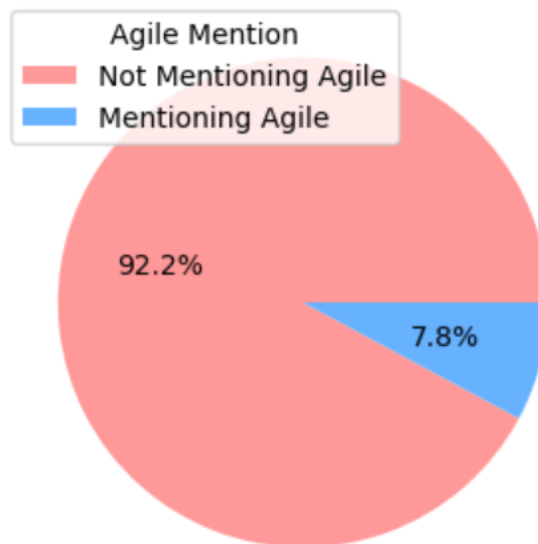


Figure 3.6.: RQ1.5: The Mention of Agile

3.5. RQ-Rel-2: Common Processes Shared Among Methodologies

Privacy requirements engineering adaptations often adhere to the traditional "waterfall" approach and lag in keeping pace with agile methodologies. It is noteworthy that merely 7% of the studies give attention to the agile approach. Within this limited scope, privacy requirements engineering methods can be integrated into agile by employing agile artifacts [70] or by incorporating privacy requirements within the agile sprints [122]. The distribution of the mentions of agile is depicted in Figure 3.6.

3.5. RQ-Rel-2: Common Processes Shared Among Methodologies

We discovered 40 privacy requirements engineering methodologies discussed in the 50 most relevant papers of our study, suggesting that some methods are examined in multiple publications. We delved deeper into these methodologies to understand the specific process they use to establish privacy requirements and designs. The distillation process is depicted in Figure 3.7.

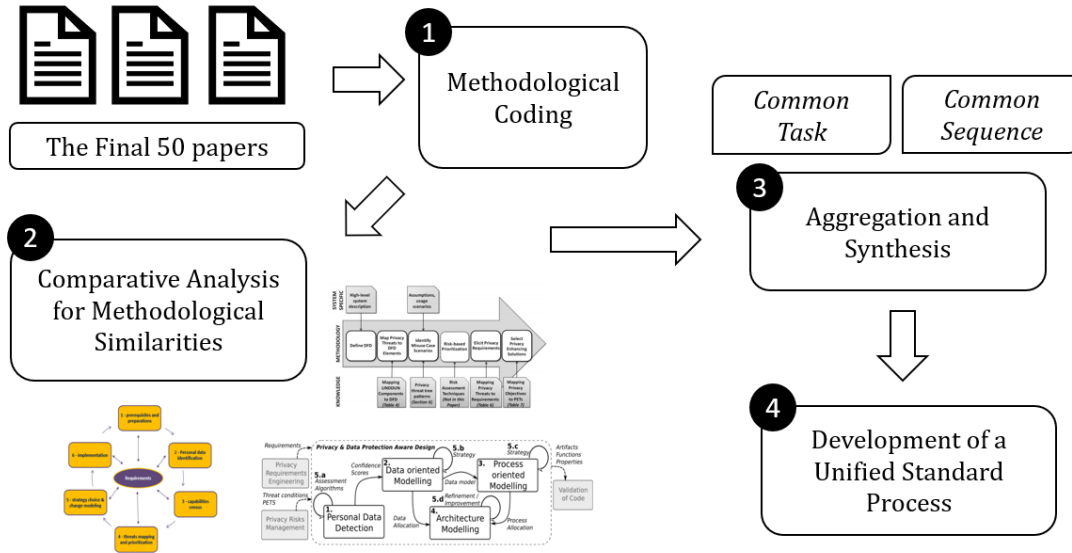


Figure 3.7.: The distillation process begins with the 50 papers selected, followed by four distillation steps. Several selected papers use visual representations to aid in comparative analysis. An example shown in the Figure are based on LINDDUN [40], ISPM [12] and PDPbD [130]

The analysis started with a comprehensive review of the 50 selected papers. Each paper was read carefully, and the methodologies were coded, with special attention paid to the main figures or diagrams provided in the paper. These visuals are essential for quickly comparing methodologies between papers. Similarities and recurring tasks

3. The Current State of the Art in Privacy Requirements and Design Engineering

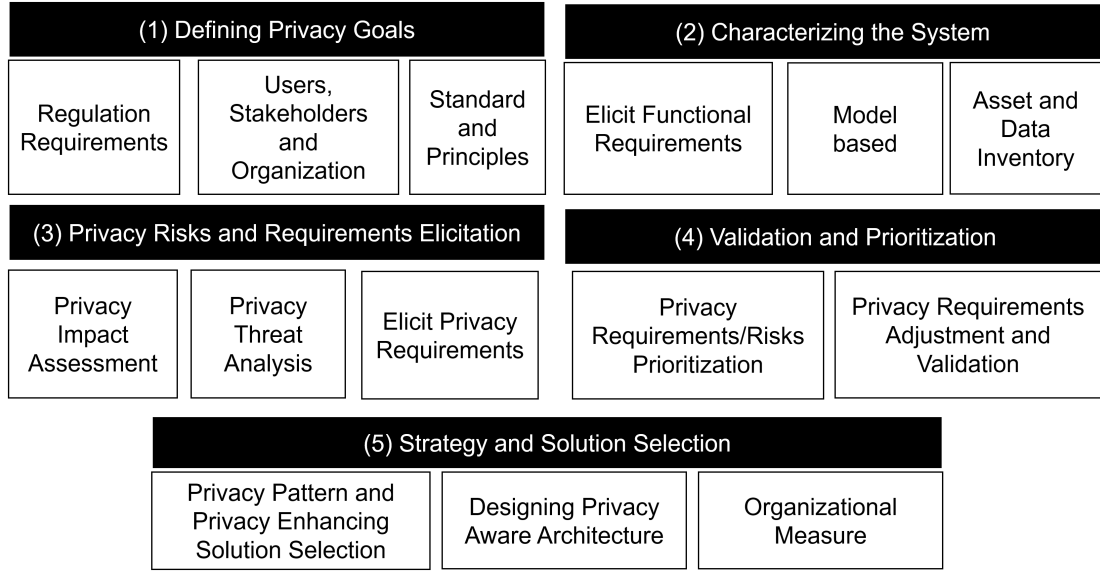


Figure 3.8.: Tasks and subtasks of the privacy requirements and design engineering process

across the methodologies were noted, and they were grouped to highlight these patterns. Furthermore, the sequence of tasks was documented in the order presented in the paper. The subsequent steps involved synthesizing these findings into a coherent framework. A preliminary framework was created, outlining the major tasks and sub-tasks, drawing from the commonalities of the methodologies. This framework was iteratively refined to ensure it was comprehensive, logical, and applicable to various privacy requirements engineering scenarios.

Based on our synthesis, we identified five main tasks that are essential to privacy requirements and design engineering. For each of these main tasks, we identified distinct approaches and actionable sub-tasks that will help in completing the main tasks.

These five tasks are part of the privacy requirements and design engineering process and represent the typical sequence followed by the methodologies. While not all methods adhere to this sequence, we aim to highlight the most common approach suggested in privacy requirements engineering studies. Less common methods, like user-based surveys or solution-based approaches, will be elaborated upon in the discussion section.

Each main process comprises several tasks, indicating numerous methods to execute a particular process. Figure 3.8 illustrates the five primary processes and their related tasks. The following section explains the particular tasks taking place within each process in detail.

3.6. RQ-Rel-3: Specific Tasks, Techniques, Work Products, and Resources in Each Process

In explaining the process of privacy requirements and design engineering, we use the privacy engineering metamodel (PEM), as introduced by Martin et al. [104]. The PEM establishes a solid framework for understanding and integrating diverse privacy requirements engineering methodologies. The metamodel comprises four key components: (1) Tasks, (2) Techniques, (3) Work Products, and (4) Resources.

The fundamental units of work in the privacy engineering process [104] are referred to as **Tasks**. These tasks represent the specific activities that need to be carried out to achieve privacy-related objectives. We defined these tasks in a similar manner to the process illustrated in Figure 3.8. Furthermore, we also aim to identify the techniques, work products, and resources necessary to establish the complete privacy engineering process. The methods or tools utilized in carrying out tasks within the privacy engineering process are known as **Techniques**. These techniques can be either manual or automated and are designed to guide effective and efficient task completion. The tangible results or artifacts that stem from executing these tasks and applying these techniques are referred to as **Work Products**. These outputs can take various forms, including documents, diagrams, software, or hardware components. Work products serve as evidence of completed tasks and can be used to measure progress and success in the process of privacy engineering. The inputs and assets needed to apply techniques and perform tasks in privacy engineering are referred to as **Resources**. These can be human, financial, technical, or informational resources. The proper conduct and support of privacy engineering activities rely on adequate resources.

3.6.1. Defining Privacy Goals

Most privacy engineering approaches begin by defining privacy goals that the organization wants to achieve or must meet due to regulatory requirements. This first process can be further divided into three tasks: (1) deriving goals from legal requirements, (2) from stakeholders (users, interest groups, and organization), and (3) from standards and principles. The overview of the tasks, techniques, work products, and resources for defining privacy goals can be seen in Table 3.1.

Regulation Requirements

In this task, privacy goals are determined based on relevant regulations and legal requirements. Analysis of regulations such as the General Data Protection Regulation (GDPR) is conducted to pinpoint specific privacy requirements the system must adhere to. These requirements can be converted into privacy goals which will direct the system design process.

Technique. The techniques can be broadly categorized as translating natural language expressions into more structured and formal models that allow for better understanding and application. Simple analyzing [78, 128, 140, 163] the intent of the text to identify

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.1.: Process 1: Defining Privacy Goals

Task	Technique	Work Products	Resources
Regulation Requirements	Analysis of regulations, Semantic Parameterization [22], KORA [148], Grounded Theory Framework [124]	Specific privacy goals and requirements [22, 124, 128, 148, 163], Traceability matrix	Regulatory Legal Documents (HIPAA [22], GDPR [20, 148], etc.)
Users, Stakeholders, and Organization	Workshops [6, 167], Interviews [11], Questionnaires [52, 53], STS-ml [151]	Stakeholder goal tables [167], Sample questions [6, 117]	Feedback from users [117], stakeholders [11], organization members [167]
Standard and Principles	Systematic approach [78, 122], Structured Interviews, Focus Groups [107]	Relevant standards and principles, Derived privacy goals [123, 163]	Privacy standards documentation (OECD Guidelines, Privacy by Design, FIPPs, ISO/IEC 27701 [123, 163])

the generic template that can be reused. A more formalized analysis using Semantic Parameterization [22] is utilized to obtain formal models from natural language statements. KORA (Concretization of Legal Requirements) [148] is employed, along with a grounded theory framework [124] to categorize each statement in the laws and legislations as a definition, right, obligation, or constraint [22]. The regulation can also be extended to stakeholder analysis to assign what rules apply to a particular stakeholder group [22].

Work Products. The primary work product in this process is the identification of specific goals and specific privacy requirements [22, 124, 128, 148, 163] stemming from relevant laws and regulations [20] and stakeholder hierarchy according to the regulation [22]. A traceability matrix can be used to link the targets to the articles of the regulation to support traceability. Furthermore, a structured approach and techniques for deriving objectives and stipulations from regulations have been established [22, 124].

Resources. Regulatory Legal Documents, including HIPAA [22], Spanish Personal Data Protection Law [163], NDPR [124], and GDPR [20, 148], serve as reference materials.

Understanding Users, Stakeholders, and Organization

In this task, the goal is to derive privacy objectives by considering the needs and expectations of users, stakeholders, and the organization. To achieve this, various data collection methods, such as surveys, interviews, or focus groups, can be employed to gather insights from these groups about their concerns and preferences regarding privacy. The gathered information is then utilized to establish privacy goals that align with the

3.6. RQ-Rel-3: Specific Tasks, Techniques, Work Products, and Resources in Each Process

needs and expectations of all involved parties.

Techniques. Defining the privacy goals according to the specific application context [90] together with stakeholders. To gather input from the stakeholders, the following methods can be used: workshops [6, 167], meetings [11], interviews [11], working groups [167], questionnaires [52, 53], or brainstorming sessions [11, 167]. A participatory approach by directly engaging with the community can also be implemented [117]. STS-ml (Socio-Technical Security Modeling Language) can be used to represent and analyze the objectives of the stakeholders [52, 53, 151].

Work Products. The work products generated from these methods include sample questions for stakeholders [6, 117] and a template for stakeholder goal tables [167].

Resources. The collected feedback comes from users [117], stakeholders [11], and members of the organization [167].

Standard and Principles

In this task, privacy goals are derived from established standards and principles for privacy protection. This can involve analyzing standards such as ISO/IEC 29100 (Privacy Framework) [78] or principles such as Privacy by Design to identify best practices for protecting user privacy. These standards and principles can then be used to define privacy goals that guide the system design process.

Technique. Systematic approach based on catalogs for transforming high-level privacy principles into specific, actionable goals [78, 122]. Structured Interviews and Focus Groups can also be used to agree on the definitions in standards [107].

Work Products. A compiled list of relevant standards and principles for privacy protection and a set of privacy goals derived from these sources [123, 163].

Resources. Existing privacy standards and principles documentation such as The OECD Guidelines, Privacy by Design principles, The Fair Information Practice Principles (FIPPs), ISO/IEC 27701 [123, 163].

3.6.2. Characterizing the System

Characterization of the system provides a clear understanding of the scope and intricacies of the system and forms the basis for subsequent processes. Several privacy requirements engineering processes directly start from characterizing the system, while others continue from defining privacy goals. This step can be divided into three sub-processes: (1) Elicit Functional Requirements, (2) Asset and Data Inventory, and (3) Model-based. The overview of the tasks, techniques, work products, and resources for characterizing the system can be seen in Table 3.2.

Elicit Functional Requirements

This task involves the elicitation of functional requirements, which are crucial for defining the system's desired behavior and its individual components. Eliciting functional

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.2.: Process 2: Characterizing the System

Task	Technique	Work Products	Resources
Elicit Functional Requirements	Textual specifications [49, 70], NLP techniques [1], Questionnaires [53, 131], Co-creation tools [167]	Functional requirements list [49], Privacy-related entities [70]	Stakeholder goals
Asset and Data Inventory	Data processing analysis [138], Guided questionnaire [131, 164], Focus group discussions [142], Surveys, Interviews [11, 107]	Assets and data inventory list [138], Data processing activities [6]	Data audit forms [13], Current database, Data flow diagrams
Model-based Approach	Data Flow Diagrams [40, 100], STS-ml [53], SecBPMN2, Secure Tropos, JTrust, Misuse Case Diagrams [102, 107], UML, BPMN [12], Goal modeling, Problem Frames, TOGAF-based modeling [111]	Diagrams illustrating software artifact interrelationships	System description from user or stakeholder requirements

requirements provides a foundation for identifying potential privacy risks and allows for the formulation of privacy requirements.

Technique. The use of textual specifications [49] helps articulate functional requirements clearly. Simple textual specification facilitates effective customer communication, particularly when dealing with agile requirements like user stories [70, 72]. This also opens the use of natural language processing (NLP) techniques applied in textual requirements representation [1]. Requirements can be synthesized from the questionnaire [52, 53, 131], co-creation tools [167] or other methods of eliciting privacy goals from the previous steps.

Work Products. A list of functional requirements written in textual representation that represents the system’s intended behavior [49]. Additionally, develop a set of privacy-related entities that support the next steps in the privacy engineering process [70].

Resources. Utilize stakeholder goals to guide the elicitation of functional requirements, considering the specific needs and preferences of system users and other relevant parties.

Asset and Data Inventory

This task requires creating a detailed list of all personal data, sensitive personal data, and related identifiers, either within a specific area or across the entire information system. While this cataloging is often done during business process redesigns, especially for companies with pre-existing data, it’s also suitable for new systems. This provides a thorough understanding of the types of personal data being processed and how they are

3.6. RQ-Rel-3: Specific Tasks, Techniques, Work Products, and Resources in Each Process

used.

Technique. The process begins by analyzing the organization's data processing activities [138]. This involves pinpointing the various data categories and assets in play [13, 103, 130, 131, 140]. To generate a comprehensive list of these activities, one can employ tools such as a guided questionnaire [131, 138, 164] or even conduct focus group discussions [142, 167]. Following this, the collection of assets and data inventory is facilitated through surveys and stakeholder interviews [6, 11, 107].

Work Products. A list of assets, data used by the system, as well as data processing activities [138]. This data can be structured in a semantics-based format [6].

Resources. A certain type of data audit form can be a primary tool for this process [13]. Additionally, existing resources like the current database or documentation of data flow diagrams should be accessible.

Model-based Approach

The model-based approach involves creating graphical models, to represent the processing of personal data within a system. These models offer a visual representation of the data flow through the system, making it easier to identify potential privacy risks and requirements. This approach is particularly useful for characterizing and understanding complex systems.

Technique. Various modeling approaches includes Data Flow Diagrams [40, 71, 100, 103, 130], STS-ml (Socio-Technical Security Modeling Language) [52, 53, 151], SecBPMN2 [151], Secure Tropos (SecTro) [151], JTrust [151], Misuse Case Diagrams [100, 102, 107], UML diagram [12, 128], BPMN [12, 44], Goal modeling diagrams [134], Problem Frames [111], or TOGAF-based modeling [12].

Work Products. Corresponding diagrams mentioned in the technique illustrate the interrelationships of various software artifacts within the systems.

Resources. System description derived from user or stakeholder functional requirements.

3.6.3. Privacy Risks and Requirements Elicitation

Building on the insights from the previous steps ensures a more informed approach to pinpointing privacy risks and requirements that are accurate, relevant, and actionable. The identification process of these privacy risks and requirements can be further divided into three tasks: (1) Privacy Impact Assessment, (2) Privacy Threat Analysis, and (3) Privacy Requirements Elicitation. The overview of the tasks, techniques, work products, and resources for privacy risks and requirements elicitation can be seen in Table 3.3.

Privacy Impact Assessment

In this task, a Privacy Impact Assessment (PIA) is conducted to identify and assess potential privacy risks associated with the system. This can involve analyzing the system's data processing activities, data flows, and other components to identify potential risks to

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.3.: Process 3: Privacy Risks and Requirements Elicitation

Task	Technique	Work Products	Resources
Privacy Impact Assessment	Analytical queries [86], Structured methodologies [33, 123]	PIA report detailing privacy risks and mitigation [123]	System documentation, System characterization, Methodology documentation [123]
Privacy Threat Modeling and Analysis	Data Flow Diagrams [40, 100], Misuse cases [102], Threat tree patterns [40], Privacy attack analysis [11], Card games for threat identification [171]	Threat tree patterns [40], Misuse cases [102], Threat catalog [40]	Common threats list, Threat trees [40] ²
Elicit Privacy Requirements	Patterns for privacy requirements [72, 112], Requirements from privacy threats [11, 40], Requirements templates [112, 163], User interaction analysis [128], Literature reviews [83], Contextual integrity norms [117]	Privacy requirements patterns [49, 112, 163], Contextual integrity rules [117]	Privacy goals [65], Legal and regulation sources [128, 163], Threat types [40], Participatory research methods [117]

user privacy. The results of the PIA can be used to inform the selection of appropriate privacy controls and measures to mitigate identified risks.

Technique. The assessment leverages analytical queries tailored to the system’s features [86]. This helps in discerning the impact or gaps concerning privacy issues [13]. Moreover, structured methodologies, like those proposed by Oetzel and Spiekermann [123] or the PIA-CNIL approach by Commission Nationale de l’Informatique et des Libertés [33], are employed.

Work Products. The culmination of this assessment is the PIA report. This document encapsulates the recognized privacy risks, detailing their possible mitigation [123].

Resources. Vital resources for this phase include system documentation, system characterization, and specific methodology documentation [123].

Privacy Threat Modeling and Analysis

This task differs from PIA in that it uses a special threat modeling approach influenced by security methodologies [154]. Here, potential privacy threats are accurately identified and then linked to specific system components.

Technique. Analyze from Data Flow Diagrams [40, 100] and misuse cases [102], detailing the threat with threat tree patterns [40], privacy attack analysis [11], and using

3.6. RQ-Rel-3: Specific Tasks, Techniques, Work Products, and Resources in Each Process

card game for identify the threat [171]

Work Products. Threat tree patterns [40], misuse cases [102] and threat catalog [40] identified.

Resources. A list of common threats and threat trees [40]³.

Elicit Privacy Requirements

In this task, privacy and data protection requirements are derived from the previous steps, e.g., from the defined data protection goals, system characterization, or the results of the privacy impact assessment or privacy threat modeling.

Technique. One widely used method is to use patterns for privacy requirements [72, 112]. Another approach emphasizes deriving requirements from identified privacy threats [11, 40] or directly from established privacy goals [174]. The use of a requirements template has been shown to be beneficial as it promotes the reuse of commonly accepted requirements [13, 112, 163]. In addition, a thorough analysis of user interactions can provide insight into specific privacy needs [128]. Some methods include conducting literature reviews to identify privacy concerns related to known applications [83]. Finally, placing requirements in the context of contextual integrity norms provides another valuable perspective [117].

Work Products. A list of reusable privacy requirements patterns from the perspective of regulation [49, 163], privacy principles and goals [112], and user interaction [128]. Moreover, it may also not be a singular pattern, but community-specific privacy norms that take the form of contextual integrity rules [117].

Resources. Privacy goals [65], legal and regulation source [128, 163], threat types [40], Participatory research methods [117].

3.6.4. Privacy Risks or Requirements Prioritization and Validation

The results derived from the preceding stages of privacy requirements elicitation and threat modeling might not always be accurate. Consequently, a phase to refine and validate these results is essential. Moreover, due to the potential vastness of these results, a structured prioritization process becomes crucial. This includes systematically evaluating and arranging the identified privacy threats and risks based on their probable occurrence and potential consequences. The overview of the tasks, techniques, work products, and resources for privacy risks or requirements prioritization and validation is depicted in Table 3.4.

Privacy Requirements Adjustment and Validation

Several privacy requirements engineering methods and tools rely on pipeline processing to generate the privacy requirements, thus it needs to be adjusted or validated.

Technique. Stakeholder or Peers validation [13, 53, 107, 163], expert judges [83], Scenario-based inspection methods [11], applying validation rules and conditions through

³<https://linddun.org/threat-trees/>

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.4.: Process 4: Privacy Risks or Requirements Prioritization and Validation

Task	Technique	Work Products	Resources
Privacy Requirements Adjustment and Validation	Stakeholder/Peers validation [53, 107, 163], Expert judges [83], Scenario-based methods [11], Validation rules [112], Fagan inspection [107], VACCINE [117], Requirements graph [167]	Approved privacy requirements report [11, 112], Formal privacy requirements [163]	Stakeholder expertise [163], Validation rules and conditions [112]
Privacy Requirements/Risk Prioritization	DREAD analysis [11, 159], Analytical Hierarchy Process (AHP) [107], Stakeholder Feedback, Dependency analysis [53]	Visual aid for threat prioritization [11]	Risk evaluation formula [11]

each privacy requirements [112]. The more formal Fagan inspection [107] or VACCINE [117] can also be employed to help in identifying ambiguities, contradictions, or missing requirements. The completeness of the system in terms of requirement fulfillment can also be viewed through the requirements graph [167]. Finally, resolve conflicts among requirements and negotiate with stakeholders to reach a consensus on the final set of requirements [83].

Work Products. Report on privacy requirements that approved [11, 112] and agreed upon stakeholders [83]. This can be in the form of formal privacy requirements [163].

Resources. Stakeholder expertise [163], Rules and conditions of privacy requirements adjustment and validation [112].

Privacy Requirements/Risk Prioritization

Privacy requirements, threats, and risks must be prioritized based on their potential ramifications on the system and its users.

Technique. The risk value, such as DREAD (an acronym for Damage, Reproducibility, Exploitability, Affected Users, and Discoverability) [11, 159], is calculated to help in this prioritization process [11, 100]. Higher risk values indicate the potential for more serious harm and are accorded greater priority [11, 107]. Analytical Hierarchy Process (AHP) can also be used to rank the requirements by considering all criteria such as risk value [107]. Stakeholder Feedback and Dependency analysis can also prioritize the requirements [52, 53].

Work Products. a practical visual aid that helps stakeholders understand the urgency and importance of each threat, such as ordered table [11].

Resources. A formula allowing for a consistent and quantitative evaluation of threats, making risk comparison and prioritization more straightforward [11].

3.6.5. Strategy and solution selection

Strategy and solution selection is the next step after validating and prioritizing privacy risks or requirements. The selected risks to be solved and requirements to be met can then be moved to the design or solution steps. This can be divided into three tasks: (1) privacy pattern and privacy-enhancing technology (PET) selections, (2) designing privacy-aware architecture, and (3) organizational measures. The overview of the tasks, techniques, work products, and resources for strategy and solution selection is depicted in Table 3.5.

Table 3.5.: Process 5: Strategy and Solution Selection

Task	Technique	Work Products	Resources
Privacy Pattern and PET Selections	Category defining [78], Expert knowledge [13, 102], Mapping table [5, 6, 12, 13, 40, 90, 123, 128], Semantics-based recommendation [6]	Mapping table from threats to PETs and patterns [6, 13, 40, 90, 128, 140]	Privacy Patterns, PETs catalog [36], Privacy expert consultation [13]
Designing Privacy-Aware Architecture	Architecture tactics [158], Enterprise architecture patterns [20, 162], First Order Approximation [142]	Data Governance Service (DGS), User Personal Data Access Service (PDAS) [158], Compliance architecture [20]	Reference Architectures (ArchMate [20, 162]), Architecture modeling languages (SysML BDD and IBD [130])
Organizational Measures	Identification of privacy targets [123], Standards and guidelines assessment	Mapping table of privacy targets to measures [123]	ISO/IEC 27000 standards [123]

Privacy Pattern and PET selections

In this task, appropriate technical measures such as privacy-enhancing technologies (PETs) [40] and countermeasures are selected to mitigate the identified threats and satisfy the privacy requirements.

PETs (Privacy-Enhancing Technologies) can be converted into standardized design patterns by examining their attributes, including context, problem, and forces. This process involves selecting the attributes that most effectively align with the system's specific privacy requirements, resulting in a consistent format for implementing privacy solutions [109].

Technique. Defining the category of the pattern's solution [78]. After that, choose the PETs and privacy patterns based on experts knowledge [13, 102] or from the mapping table between privacy requirements/threats to the corresponding solutions [5, 6, 12, 13,

3. The Current State of the Art in Privacy Requirements and Design Engineering

40, 90, 123, 128] or based on recommendation system from semantics-based knowledge [6]. This can be categorized in further detail into a group of threats [40] or design strategies whether it is related to data or process [103, 130] or related to specific requirements in regulation [162].

Work Products. Mapping table from privacy threats to PETs and Privacy Patterns [6, 13, 40, 90, 128, 140].

Resources. Privacy Patterns catalog and PETs catalog [36] for mapping and privacy expert [13] for making a final decision.

Designing privacy-aware architecture

In this task, the focus is on designing an architecture that supports the selected PETs and countermeasures while also meeting other relevant criteria. This involves defining the system architecture in a way that best supports the identified privacy-related processes [12, 20, 90, 122, 130, 140, 158, 162] and compliance [139].

Technique. To ensure privacy in architectural designs, employ tactics such as the separation of roles, audit logging, access control, data revelation, data retention, data encryption, and multifactor authentication [158]. Beyond these measures, incorporating enterprise architecture patterns provides an added layer of security [20, 162]. Acknowledging the limitations in perfectly implementing all behavioral or privacy requirements, the use of First Order Approximation is also advocated in one study [142]. This approach helps in enhancing the effectiveness of the deployed privacy and security measures.

Work Products. A set of architecture designed for enabling data privacy such as Data Governance Service (DGS) and User Personal Data Access Service (PDAS) [158] or complying with regulation [20].

Resources. Reference Architectures and Models such as ArchiMate [20, 162], or architecture modeling language such as SysML BDD and IBD [130].

Organizational Measures

In this task, organizational measures are selected to support the technical measures and ensure compliance with relevant privacy regulations. This can include measures such as management, administrative, and accountability [123].

Technique. To select appropriate organizational measures, organizations should first identify the privacy targets [123] or requirements [122] relevant to their operations.

Work Products: Mapping table of privacy targets to organizational measures [123].

Resources: Standards and guidelines for information security management. The best known are the ISO/IEC 27000s [123].

3.7. RQ-Rel-4: The Sequence of Processes in Practice

For clarity and standardization of the sequence, we have assigned color codes to each task. The mapping of the task to the color codes can be seen in Table 3.6. These color codes facilitate the mapping of each task in a network diagram for every methodology

addressed. We subsequently generate a network diagram for each method, all of which are included in the Appendix.

Table 3.6.: Privacy Requirements and Design Engineering Processes and Tasks

Process	Code	Task
Defining Privacy Goals	GR	Regulation Requirements
	GO	Users, Stakeholders and Organization
	GS	Standards and Principles
Characterizing the System	CF	Elicit Functional Requirements
	CM	Model-Based
	CA	Assets and Data Inventory
Privacy Risks and Requirements Elicitation	PA	Privacy Impact Assessments
	PT	Privacy Threat Analysis
	PR	Elicit Privacy Requirements
Validation and Prioritization	VP	Prioritization of Privacy Requirements/Risk
	VA	Adjustment and Validation
Strategy and Solution Selection	SP	Privacy Patterns and PETs selection
	SA	Designing Privacy-Aware Architecture
	SO	Organizational Measure

3.7.1. Identifying Initial Steps in Privacy Requirements Engineering

Our analysis reveals six potential approaches to initiating the privacy requirements engineering process: it can either start with the "Defining Privacy Goals" or directly begin with the "Characterizing the System". We also indicate that the goal-based method predominantly originates from "Defining Privacy Goals", whether from GS, GO, or GR. Conversely, the risk-based method typically begins with "Characterizing the System", either through CA, CF, or CM. Figure 3.9 illustrates examples of methods corresponding to each starting point.

3.7.2. Distinguishing Features of Privacy Requirements Engineering Methodologies' Sequences

Expanding on Figure 3.9, we can observe the distinctive starting point of the process and the unique approaches of each method. Specifically, the STRAP method [86], is unique in its mention of an iterative process, a feature not explicitly stated in others, although it remains a possibility. A notable distinction is the work of Breaux and Anton [22], which focuses extensively on the analysis of regulatory texts for privacy requirements. In contrast,

3. The Current State of the Art in Privacy Requirements and Design Engineering

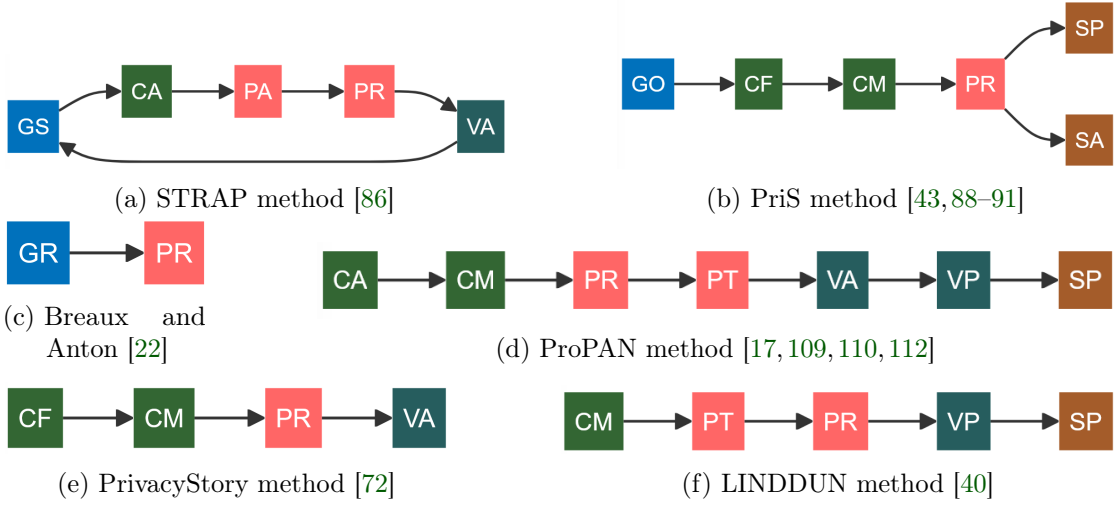


Figure 3.9.: This set of methods presents six starting points for the privacy requirements engineering process.

the ProPAN method, encompassing several papers [17,109,110,112], demonstrates the broadest scope of coverage. The figures also show the use of models like Data Flow Diagrams to characterize the system in PrivacyStory [72], ProPAN [17,109,110,112], and LINDDUN [40], which serve as foundational tools for defining privacy requirements and/or threats. The figure further indicates that while half of the methods offer strategies and solutions, the other half conclude once privacy requirements are defined or validated. Lastly, the PriS method employs architecture as a key strategy for addressing privacy requirements [43,88–91].

3.7.3. Typical Sequences Observed in Privacy Requirements Engineering Approaches

Figure 3.10 consolidates the information from 40 different process approach diagrams, allowing us to identify the most common sequences in these processes.

Privacy Requirements Elicitation (PR) serves as the focal point in the process flow, acting as both a destination and a starting point for various processes. Most processes can be transitioned directly into PR, with the majority of these transitions occurring after the identification of privacy threats. Following PR, the most common outgoing processes involve the adoption of privacy patterns and PET solutions. Additionally, PR often leads to either the adjustment of privacy requirements, supported by 7 studies, or the requirements prioritization, supported by 5 studies.

The predominant sequence observed is moving from the Elicitation of Privacy Requirements (PR) to the Selection of Privacy Patterns and PETs (SP), with this pattern occurring 9 times. This trend suggests a standard approach where, upon defining privacy requirements, there is a direct move towards adopting privacy patterns and

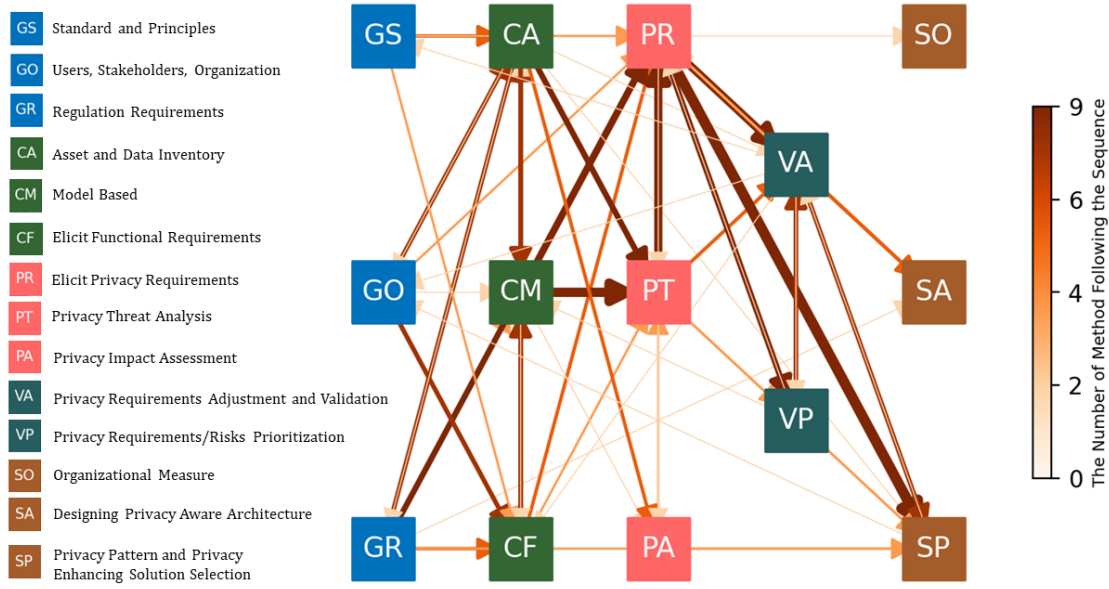


Figure 3.10.: Network diagram depicting the workflow of privacy requirements and design processes as derived from literature. This directed graph shows processes that precede others. Thicker edges and darker colors indicate processes that are adapted by more methods.

PETs [13, 43, 122, 128, 131, 148, 157, 167, 174]. It implies that the processes of validating and adjusting privacy requirements are often bypassed in favor of immediately implementing standardized solutions and technologies to fulfill these requirements. However, methodologies like LINDDUN [40] and ProPAN [112] follow the PR to SP sequence, but only after involving Verification or Validation (VP/VA) steps.

The second most frequent sequence involves transitioning from Model-Based Characterization (CM) to Privacy Threat Analysis (PT), with this pattern occurring 8 times. This suggests that after a system is characterized using various models, the next recommended step is to conduct a privacy threat analysis on that model [12, 40, 100, 102, 103, 123, 134, 151]. In this scenario, the model provides essential context and structure for systematically identifying potential privacy threats. The complete sequence of each method we studied in this thesis is depicted in Table 3.7.

3.8. RQ-Rel-5: Towards a Holistic Process

After examining the privacy requirements and design process, we are evaluating whether a holistic approach can be achieved by addressing each element of the process as shown in Figure 3.8. Consequently, we propose a hypothetical process that encompasses all potential aspects of the process along with its challenges.

3. The Current State of the Art in Privacy Requirements and Design Engineering

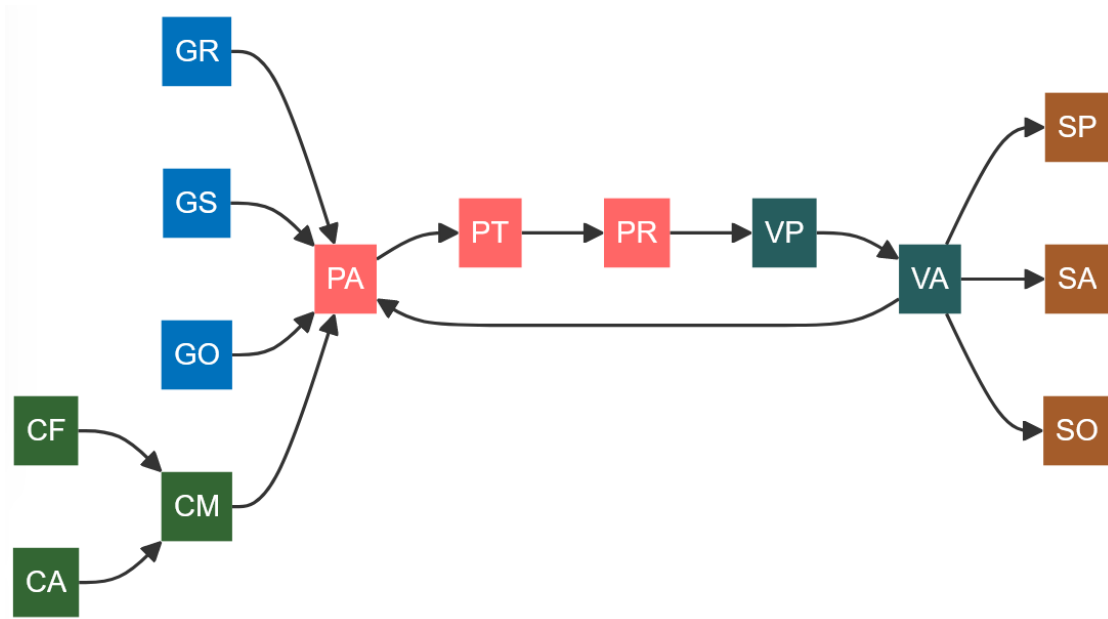


Figure 3.11.: Structure of a Holistic Approach of Privacy Requirements and Design Engineering.

3.8.1. A Holistic Process

Figure 3.11 illustrates a conceptual approach to the holistic process. The approach begins with two simultaneous branches of process. The first branch initiates with a three-goal approach: focusing on regulation compliance (GR), adhering to international privacy standards (GS), and considering user or stakeholder needs (GO). Concurrently, the second branch delves into system characterization, drawing insights from both functional and non-functional requirements (CF), as well as asset and data inventories (CA). Expressing these characteristics in an accessible modeling language could facilitate a better understanding of the system’s architecture among stakeholders (CM). These two branches can function independently, with different teams working on them, and their outcomes are integrated to serve as the foundation for subsequent steps.

Subsequent processes include Privacy Impact Assessment (PA), Privacy Threat Analysis (PT), and Privacy Requirements Elicitation (PR), which are distinct but interrelated. The process starts with a (PA), evaluating the privacy objectives set earlier and identifying those with the highest potential impact if not met. This is followed by a (PT), which pinpoints specific threats and vulnerabilities. Finally, privacy requirements elicitation (PR) leverages the insights from (PT) to establish clear and actionable privacy requirements.

Integrating both goals and risks during the initial phase of product development can be overwhelming for developers. To manage this, prioritization becomes essential, focusing on the impact and likelihood of privacy requirements (VP). Additionally, it’s essential for stakeholders to consistently validate and adjust (VA) these requirements to ensure

alignment with the initial goals, as well as with evolving standards and regulations.

The final step in this process involves selecting specific technical and organizational measures that align with the privacy requirements. This selection process includes adopting suitable privacy patterns or Privacy Enhancing Technologies (PETs) (SP), according to specific requirements. Equally important is the development of a Privacy Aware architecture (SA), which integrates privacy considerations into the design. On the organizational side (SO), measures such as policies, procedures, and training play a critical role. These organizational measures are geared towards supporting the privacy objectives of the organization, ensuring a comprehensive approach to privacy across the entire project.

The quick reference guide for this holistic process is published in an open repository⁴ for practitioners to use.

3.8.2. Challenges to a Holistic Process

The defined process may present challenges that we identify based on what has been mentioned in the literature we investigated. We broadly categorized it into four main themes: (1) Time and resources (2) Tools and Automation (3) Agile Requirements.

Time and Resources

The process of privacy requirements engineering requires significant time and resources, as noted in numerous studies [84, 171]. This process entails detailed system characterization, which involves identifying assets and visually mapping out the system's process [171]. Additionally, it may involve comprehensive techniques such as Privacy Impact Assessments [84, 123].

To address this, researchers have introduced lightweight models. Examples include LINDDUN GO [171], which is a lightweight version of LINDDUN [40], and PDP-ReqLite [49], a simplified version of ProPAN [109]. However, we argue that these lighter models might compromise the comprehensiveness we previously emphasized.

Another solution for this challenge suggests the advantages of creating automated tools to decrease required time and resources [160]. Nevertheless, in several privacy requirements engineering implementations, this has yet to be explored [177, 179], which presents another challenge.

Tools and Automation

The review of the literature highlights a clear need for specialized tools and automation in the field of privacy requirements engineering. The complexity involved in Privacy Impact Assessments, for example, has made the role of such tools indispensable [123].

Focusing on tools, there are a few notable ones. The Privacy Requirements Elicitation Technique Tool (PRET) aids in integrating the Security Quality Requirements Engineering (SQUARE) methodology [107]. Another tool is designed for quantitative threat modeling,

⁴<https://github.com/gunturbudi/privacy-requirements-engineering>

3. The Current State of the Art in Privacy Requirements and Design Engineering

specifically to assess security and privacy risks in attack tree elements [100]. There's also a significant tool that assists developers in creating privacy specifications for agile environments [134], distinguished by its provision of auditable resources and its empirical validation [131].

In the context of automation, its value in the software engineering community is well-recognized [2]. However, automation's application in privacy requirements engineering is still nascent [177, 179]. Automation can effectively describe privacy target systems, set goals, choose technical measures, and evaluate results [177, 179].

Many of the tools discussed do not use automation systems. ProPAN [112] and its simplified version, PDP-ReqLite [49], automate the generation of privacy requirements using pattern-based approaches, but they require initial manual input. In contrast, PrivacyStory [70, 72] and EPICUREAN [157] employs Natural Language Processing (NLP) for extracting privacy requirements. Additionally, RMCM-V [102] conducts automatic conformance checks using a predefined model template. We believe that integrating automation throughout these processes into a comprehensive computer-assisted tool [72, 112] would likely boost the adoption of the privacy requirements engineering approach.

Compatibility with Agile Methodologies

Most privacy requirements engineering processes we reviewed follow a linear approach that is not suitable for the iterative and adaptable nature of agile software development [93]. In an agile environment, the system specification is never considered final, and it can be subject to changes at every iteration. This may result in alterations to the target functionality, the data processed, or the services implemented in the system. This poses challenges for privacy requirements engineering as conventional sequential methodologies may not be adaptable to these changes [93]. The ProPAN approach explicitly acknowledges its incompatibility with agile environments [109].

On the other hand, the PRIPARE framework [122] is known for its adaptability within agile settings, emphasizing the critical role of sprint planning. It suggests three specific strategies to integrate privacy and security considerations into agile workflows. The first strategy is the incremental approach, which focuses solely on the privacy issues relevant to the current sprint. Although practical, this approach risks overlooking the broader system context. Secondly, the Sprint Zero approach prioritizes crucial data protection efforts from the beginning, establishing a solid foundation without fully anticipating future system developments. Additionally, dedicating entire sprints to privacy and security (Privacy and Security Sprints) ensures comprehensive attention to these areas, but it could disrupt regular development cycles. However, the practical effectiveness of these strategies has yet to be empirically proven.

To address the integration of privacy in agile environments, some strategies have been proposed from an agile requirements perspective. Tools that aid agile teams in identifying privacy requirements are considered a viable solution [131, 134]. Several methods also incorporate the iterative nature of agile [86]. Moving forward, there is a movement towards automated analysis of agile artifacts and system characterization automation [70, 71].

3.9. Closing the Gap: Integrating Intelligent Systems in Agile Software Development Processes

The agile methodology prioritizes adaptability, teamwork, and user-focused development. Some techniques for privacy requirement engineering emphasize the active involvement of users in the development process [117]. Other approaches focus on resolving user issues identified through feedback and review [83]. However, there is still an unresolved problem of privacy requirements engineering methods regarding empirically validating these techniques in real agile situations.

3.9. Closing the Gap: Integrating Intelligent Systems in Agile Software Development Processes

The integration of privacy requirements engineering into agile software development methodologies has been a significant challenge for organizations [62]. Our findings from RQ-Rel-1.5 indicate that privacy requirements engineering adaptations have not kept pace with the rapid evolution of agile methodologies, with only a limited number of studies addressing the integration of privacy into agile processes. Furthermore, RQ-Rel-1.4 reveals that despite the inclination to use computer-aided tools, they remain underutilized in the field of privacy engineering. To address these challenges, we analyzed the types of intelligent systems that could support agile teams in the process of privacy requirements engineering and design.

Recent research has highlighted the struggles faced by agile teams in identifying privacy criteria within user stories [26]. Participants in a study conducted by Oetzel et al. [123] expressed concerns regarding the time and labor required to create detailed system characterizations. Similarly, Wuyts et al. [171] noted that the resources needed to model systems using data flow diagrams (DFDs) could potentially slow down security and privacy threat analysis processes.

Our findings from RQ-Rel-1.2 indicate that most research in this area focuses on identifying privacy requirements. However, there is a lack of concrete guidance for selecting the most suitable privacy patterns for specific scenarios. This gap in knowledge hinders the effective implementation of privacy measures in agile software development.

These findings suggest that the most promising areas to address in the context of this thesis, include:

1. **CF** *Identifying privacy-related entities within agile requirements*: Developing intelligent systems that can automatically detect and highlight privacy-related entities in user stories and other agile requirements documentation could significantly reduce the time and effort required by agile teams to identify privacy criteria.
2. **CM** *Generating data flow diagrams*: Intelligent systems capable of automatically generating DFDs based on agile requirements could streamline the process of system modeling and facilitate more efficient security and privacy threat analysis.
3. **PR** *Generating privacy requirements*: Intelligent systems that can analyze agile requirements and generate relevant privacy requirements could help ensure that pri-

3. The Current State of the Art in Privacy Requirements and Design Engineering

vacy considerations are consistently addressed throughout the software development lifecycle.

4. **SP** *Selecting appropriate privacy design patterns*: Developing intelligent systems that can recommend suitable privacy design patterns based on specific project requirements and scenarios could provide agile teams with much-needed guidance and support in implementing effective privacy measures.

While other areas, such as automatically eliciting requirements from regulations or prioritizing requirements, are also important, they fall outside the scope of this thesis and will be considered for future research.

3.10. Threats to Validity

In conducting our systematic literature review, we identified potential threats to the validity of our findings, which can be classified into internal and external threats.

Internal Threats. This study focused on specific databases and search terms and excluded non-English language studies, which may have limited the scope of research included in our findings. By focusing on Scopus we potentially missed some research papers which are not indexed there, because they were not published in mainstream journals or conferences, such as IEEE, ACM, and Springer. Although we chose Scopus for its recognized standards and its ability to compile data from various sources, this approach still presents internal challenges in maintaining the objectivity and completeness of our review. Additionally, the act of interpreting and synthesizing data from the chosen studies can introduce a level of subjectivity that might affect our analysis and outcomes.

External Threats: The dynamic nature of privacy issues means our review might not capture the most recent trends and methodologies. This evolving landscape poses a challenge to maintaining the relevance of our findings over time. The specificity of our focus on privacy requirements engineering and the diversity of methodologies applied in the studies we examined may restrict the applicability of our findings across different domains and settings. This limitation affects the broader applicability of our conclusions.

By identifying these internal and external threats, we aim to acknowledge the limitations of our review and highlight areas for future research to address.

3.11. Chapter Summary

This chapter presents a systematic literature review that aims to understand commonalities among various privacy requirements engineering methodologies. We analyzed 50 papers systematically and identified 40 distinct methodologies. From this, we have identified five key steps that are commonly found in privacy requirements engineering: (1) defining privacy goals, (2) characterizing the system, (3) eliciting privacy risks and requirements, (4) validating and prioritizing privacy requirements, and (5) selecting appropriate strategies and solutions. Each step includes a range of tasks and techniques that lead to the creation of work products equipped with the necessary resources.

Our findings aim to establish a complete and sequential process flow for privacy requirements engineering. This process begins with two parallel paths: the first addresses compliance with regulations, international privacy standards, and stakeholder needs. The second path focuses on system characterization, which includes both functional and non-functional elements and data assets. The following steps include Privacy Impact Assessment, Threat Analysis, and Requirements Elicitation. These steps focus on identifying and defining privacy requirements and threats. This model highlights the significance of prioritizing and continuously adapting privacy requirements from the early stages of development. The final phase entails selecting technical and organizational measures, creating a Privacy-Aware architecture, and implementing policies and training to integrate privacy considerations throughout the project lifecycle. The aim is to provide privacy engineers, requirements engineers, and other stakeholders with a clear, step-by-step process equipped with relevant techniques.

We have identified three major challenges in adopting this approach: (1) dedicating specific time and resources, (2) employing necessary tools and automating processes, and (3) integrating with agile software development methodologies. Furthermore, the holistic framework we propose does not mean that a one-size-fits-all approach must be adopted, which may not address the particular privacy concerns or operational constraints that arise in each project.

Lastly, we analyzed the type of intelligent system required to support agile teams in the process of privacy requirements engineering and design. Our findings suggest that the most promising areas for research, particularly in the context of this thesis, include: (1) Identifying privacy-related entities within agile requirements, (2) Generating data flow diagrams, (3) Generating privacy requirements, and (4) Selecting appropriate privacy design patterns. In the following chapter, we will go into detail about the intelligent system components approach we used to address these areas of focus.

3.12. The Method Used in the Literature Review

Table 3.7.: Lists of Privacy Requirements Engineering Methods reviewed in our study.
In the paper where the abbreviation for the method was not available, the author(s) name was used instead.

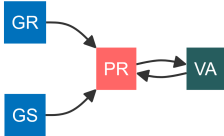
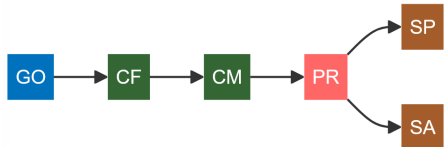
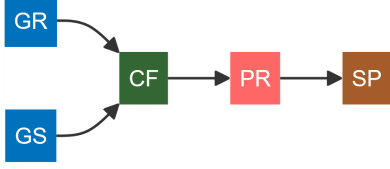
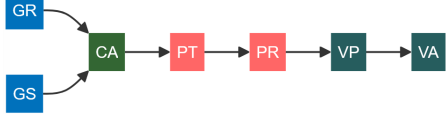
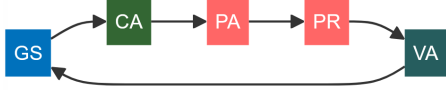
	Method	Description
1	SIREN [163]	The process model is based on a catalog of reusable requirements and specific requirements related to a project, and consists of steps such as gathering, analyzing, developing, and evaluating. 
2	Privacy Interface Analysis [128]	The method involves developing a detailed description of the application, examining HCI requirements, scrutinizing generic privacy solutions, organizing solutions according to use cases, and implementing the interface according to the requirements document. 
3	Breaux and Anton [22]	A methodology for extracting formal descriptions of rules that govern stakeholder actions from policies and regulations, which combines the Goal-based Requirements Analysis Method (GBRAM) and a process called Semantic Parameterization for acquiring formal models from natural language statements. 
4	PriS [43, 88–91]	Security and privacy requirements engineering framework that incorporates privacy requirements early in the system development process by using privacy process patterns to model identified privacy-related processes and identify proper implementation techniques. 
Continued on next page		

Table 3.7 – continued from previous page

	Method	Description
5	Privacy-Aware Engineering [174]	The method involves integrating privacy artifacts in a compatible, compliant, and verifiable way throughout the software development lifecycle to ensure appropriate privacy controls are implemented and maintained in IT systems.  <pre> graph LR GR[GR] --> CF[CF] GS[GS] --> CF[CF] CF --> PR[PR] PR --> SP[SP] </pre>
6	LINDDUN [40]	Privacy threat modeling methodology that provides systematic support for the elicitation and mitigation of privacy threats in software systems through a three-step process of modeling the system, eliciting threats, and managing threats.  <pre> graph LR CM[CM] --> PT[PT] PT --> PR[PR] PR --> VP[VP] VP --> SP[SP] </pre>
7	SQUARE + PRET [107]	A method that modifies the Security Quality Requirements Engineering (SQUARE) process to incorporate privacy considerations, aided by a Privacy Requirements Elicitation Technique (PRET) tool, which uses a questionnaire to determine system information and suggest suitable privacy requirements from its database.  <pre> graph LR GR[GR] --> CA[CA] GS[GS] --> CA[CA] CA --> PT[PT] PT --> PR[PR] PR --> VP[VP] VP --> VA[VA] </pre>
8	STRAP [86]	A four-step method involving analysis, refinement, evaluation, and iteration to systematically identify, mitigate, and evaluate privacy vulnerabilities in system design, and ensure ongoing adaptation to changes and new features while maintaining user privacy.  <pre> graph LR GS[GS] --> CA[CA] CA --> PA[PA] PA --> PR[PR] PR --> VA[VA] VA --> GS </pre>
9	ProPAN [17, 109, 110, 112]	Systematic approach for identifying privacy threats during the requirements analysis of software systems using problem frame models.  <pre> graph LR CA[CA] --> CM[CM] CM --> PR[PR] PR --> PT[PT] PT --> VA[VA] VA --> VP[VP] VP --> SP[SP] </pre>

Continued on next page

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.7 – continued from previous page

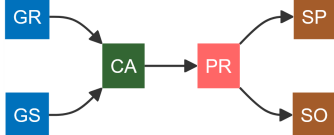
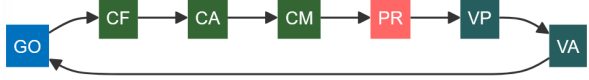
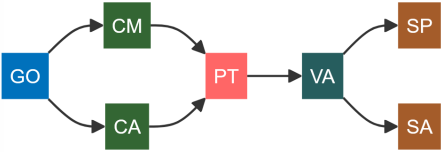
	Method	Description
10	Quantitative Threat Modeling (QTMM) [100]	A methodology that quantifies the security and privacy risks of each element of an attack tree to make objective design decisions about privacy-respecting systems. 
11	PREprocess [142]	Framework for developing privacy-aware applications. It consists of two processes: a bottom-up (Design) process and a top-down (Evaluation) process. 
12	Privacy Design Strategies [78]	Process for deriving privacy design strategies from legal requirements, resulting in eight data-oriented and process-oriented strategies: MINIMISE, HIDE, SEPARATE, AGGREGATE, INFORM, CONTROL, ENFORCE, and DEMONSTRATE. 
13	Oetzel and Spiekermann [123]	A seven-step Privacy Impact Assessment (PIA) methodology for addressing privacy issues in the design or upgrade phase of an IT system. 
14	PRIPARE [122]	A systematic methodology for privacy engineering combines current best practices, incorporating both a goal-oriented and risk-based approach in the analysis phase, and emphasizes the creation of privacy-compliant architectures in the design phase. 
15	Gharib [52, 53]	A holistic approach for analyzing privacy requirements through four primary steps - elicitation, classification, prioritization, and validation, helping developers determine privacy requirements and identify potential interface design solutions for their projects. 
Continued on next page		

Table 3.7 – continued from previous page

	Method	Description
16	SPACE [167]	The methodology combines Co-Creation processes and Security-and-Privacy-by-Design (SPbD) methodologies to ease the elicitation and analysis of security and privacy requirements, and is structured into four phases: Scenario Definition, Input Data Identification, Stakeholders' Goals and Data of Interest Analysis, and Threats and Feared Events Investigation.  <pre> graph LR GO[GO] --> CA[CA] CA --> CF[CF] CF --> PT[PT] PT --> PR[PR] PR --> SP[SP] </pre>
17	Islam et al. [83]	Empirical research approach to identify and validate social network user privacy concerns, and using the PREview approach, it elicits and negotiates requirements based on these concerns, thereby generating and validating social network user privacy requirements.  <pre> graph LR GO[GO] --> CF[CF] CF --> PR[PR] PR --> VP[VP] VP --> VA[VA] </pre>
18	GuideMe [13]	A six-step approach designed to aid organizations in understanding and complying with GDPR's data protection obligations, involving steps such as data audit, gap analysis, planning and preparation, plan review, execution, and evaluation.  <pre> graph LR CA[CA] --> GR[GR] GR --> PR[PR] PR --> SP[SP] SP --> VA[VA] VA --> SP </pre>
19	RMCM [102]	A use case-driven method for specifying and analyzing security and privacy requirements of multi-device software ecosystems in a structured manner, including steps for requirements elicitation, mitigation schemes specification, artifacts analysis, refinement, and practical application to an industrial healthcare project.  <pre> graph LR CM[CM] --> PT[PT] PT --> VA[VA] VA --> SP[SP] </pre>
20	PDP4E [103]	A method that integrates privacy and data protection principles into software and systems engineering disciplines such as risk management, requirements engineering, model-driven design, and software/systems assurance, with the aim of systematically achieving regulatory compliance and upholding data protection principles.  <pre> graph LR GO[GO] --> CM[CM] GO --> CA[CA] CM --> PT[PT] CA --> PT PT --> VA[VA] VA --> SP[SP] VA --> SA[SA] </pre>

Continued on next page

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.7 – continued from previous page

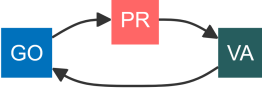
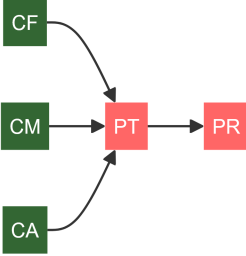
	Method	Description
21	Mir et al. [117]	A participatory design process for data-centric applications where privacy policies are formed through community involvement in the research/design process, enabling communities to define and enforce their own contextual, ethical use of data.  <pre> graph LR GO[GO] --> PR[PR] PR --> VA[VA] VA --> GO </pre>
22	Peixoto and Silva [134]	A framework for specifying privacy requirements using goal-oriented modeling languages, outlining 14 essential privacy modeling capabilities that need to be addressed by requirements modeling languages to improve privacy specification, thereby offering a structured approach to privacy requirements in the initial stages of software development.  <pre> graph LR CF[CF] --> PT[PT] CM[CM] --> PT CA[CA] --> PT PT --> PR[PR] </pre>
23	KORA [148]	A method to identifies 74 reusable technical requirements from the GDPR to ensure software products processing personal data are legally compliant via a four-step process involving the definition of legal requirements, determination of legal criteria for IT systems, derivation of technical requirements, and creation of technical design proposals.  <pre> graph LR GR[GR] --> PR[PR] PR --> SP[SP] </pre>
24	EAM [20]	Approach that maps General Data Protection Regulation (GDPR) compliance principles into a structured business architecture, enhancing understanding and implementation of regulatory requirements within an organization.  <pre> graph LR GR[GR] --> PR[PR] PR --> SA[SA] </pre>
Continued on next page		

Table 3.7 – continued from previous page

	Method	Description
25	EPICUREAN [157]	Approach that simplifies the privacy settings of IoT applications in line with GDPR through a recommender-based, three-phase system, which analyzes and annotates data sources, trains users to specify privacy requirements and uses collaborative filtering to offer personalized privacy setting suggestions, thereby enhancing user privacy while optimizing application functionality.  <pre> graph LR CA[CA] --> GO[GO] GO --> PR[PR] PR --> SP[SP] </pre>
26	PDP-ReqLite [49]	A lightweight method that aids engineers in eliciting GDPR and privacy requirements in system and software projects by using two modeling artifacts (RDFD and PID), a protocol ensuring GDPR coverage, and tool support for automatic generation of candidate requirements, all demonstrated via a Smart Grid case study.  <pre> graph LR GR[GR] --> CM[CM] CM --> PR[PR] PR --> VA[VA] </pre>
27	DEFEND [138, 139, 164]	GDPR-compliant data privacy governance platform that uses a seven-step methodology to facilitate organizations in identifying and managing their data privacy and security requirements, performing self-assessments, threat analyses, and continuous risk assessments.  <pre> graph LR GR[GR] --> CF[CF] CF --> CA[CA] CA --> PA[PA] PA --> PT[PT] PT --> PR[PR] PR --> VA[VA] VA --> SA[SA] VA --> CF </pre>
28	SePTA [151]	A method that aids the requirements engineering process by utilizing four modeling languages (STS-ml, SecBPMN2, Secure Tropos (SecTro), and JTrust) to analyze security and privacy requirements and validate trust relationships in a system, covering stages from early requirements definition to late-stage security analysis and threat identification.  <pre> graph LR GO[GO] --> CF[CF] CF --> CM[CM] CM --> PT[PT] PT --> SP[SP] SP --> GO </pre>
29	P-STORE [11]	A methodology that guides the elicitation and validation of privacy requirements in software development by defining privacy goals, identifying and prioritizing stakeholders, assets, and potential privacy attacks, categorizing and evaluating risks, and ultimately producing a privacy requirements specification document.  <pre> graph LR GO[GO] --> CA[CA] CA --> PT[PT] PT --> VP[VP] VP --> PR[PR] PR --> VA[VA] </pre>

Continued on next page

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.7 – continued from previous page

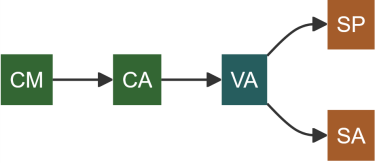
	Method	Description
30	PrivacyStory [70–72]	A tool that assists development teams in conducting privacy requirements engineering by providing an end-to-end privacy requirement generation from a set of user stories.  <pre> graph LR CF[CF] --> CM[CM] CM --> PR[PR] PR --> VA[VA] </pre>
31	PDPbD [130]	A model-based methodology that supports the integration of GDPR provisions during design-time activities through five phases: personal data detection, data-oriented modeling, process-oriented modeling, architecture modeling, and privacy assessment and validation, illustrated with a case study in the automotive domain.  <pre> graph LR CM[CM] --> CA[CA] CA --> VA[VA] VA --> SP[SP] VA --> SA[SA] </pre>
32	Stefanova and Dimov [158]	A method that ensures GDPR-compliant privacy in service-based software systems by analyzing privacy requirements, applying architectural tactics like role separation and data encryption, introducing two additional infrastructure services (Data Governance Service and User Personal Data Access Service).  <pre> graph LR GR[GR] --> SA[SA] </pre>
33	Teixeira et al. [162]	A methodology that utilizes enterprise architecture patterns to achieve GDPR compliance, involving steps such as identifying relevant entities, defining use cases, creating pattern templates, retrieving and organizing patterns, applying them to a case study, and evaluating their impact on GDPR compliance.  <pre> graph LR GR[GR] --> CA[CA] CA --> SP[SP] </pre>
34	Alkhariji et al. [5]	A methodology that involves surveying and analyzing PbD schemes and privacy patterns, synthesizing this knowledge into a comprehensive model, applying it to real-world IoT designs in healthcare, and identifying research challenges to construct an intelligent privacy assistant that can support software developers in creating privacy-conscious designs.  <pre> graph LR GR[GR] --> SP[SP] </pre>
Continued on next page		

Table 3.7 – continued from previous page

	Method	Description
35	ISPM [12]	An approach that aims to ensure privacy by design in information systems, involving prerequisite preparations, personal data identification, capabilities census, threats mapping and prioritization, strategy choice and changes modeling, and implementation of privacy-enhancing technologies.  <pre> graph LR GO[GO] --> CA[CA] CA --> CM[CM] CM --> PT[PT] PT --> VP[VP] VP --> SP[SP] </pre>
36	Diamantopoulos and Karyda [44]	A four-step methodology for incorporating privacy-by-design principles into the redesign of business processes, encompassing the identification of personal data, security, and privacy analysis, optional Data Protection Impact Assessment (DPIA), and the application of privacy-preserving patterns in the redesign of business processes.  <pre> graph LR CF[CF] --> CA[CA] CA --> CM[CM] CM --> PA[PA] PA --> PT[PT] PT --> SP[SP] </pre>
37	Olukoya [124]	A method to extract privacy and security requirements from legal texts, privacy policies, and system documentation, analyze them, and then map these requirements to software system's design properties to verify compliance with relevant laws and regulations.  <pre> graph LR GR[GR] --> CF[CF] CF --> PR[PR] </pre>
38	PCM [131]	A method that integrates with agile software development practices to specify privacy requirements, identifying and classifying personal information, defining trust relationships between actors, specifying privacy constraints, risks, and mechanisms, all during the early stages of software development.  <pre> graph LR CF[CF] --> CA[CA] CA --> PT[PT] PT --> PR[PR] PR --> SP[SP] </pre>
39	Plappert et al. [140]	A systematic approach to designing a privacy-aware electrical/electronic (E/E) architecture for electric vehicles that aligns with GDPR requirements, by translating these legal requirements into technical ones and implementing privacy-securing measures.  <pre> graph LR GR[GR] --> CA[CA] CA --> PA[PA] PA --> SP[SP] </pre>
40	PARROT [6]	A semantic web-based personal assistant to guide software engineers on Privacy by Design practices during the design phase of IoT systems, utilizing a process of information gathering, analysis, ontology development, and evaluation.  <pre> graph LR GR[GR] --> CF[CF] CF --> CA[CA] CA --> SP[SP] </pre>

3.13. The Number of Sequences of the Process

3. The Current State of the Art in Privacy Requirements and Design Engineering

Table 3.8.: Tabular display illustrating links from initiating nodes (rows) to receiving nodes (columns). The number is determined by the number of methods that support that connection. The darker color aids in highlighting the most frequent connection.

	GR	GO	GS	CF	CM	CA	PA	PT	PR	VP	VA	SP	SA	SO
GR	0	0	0	3	2	4	0	0	5	0	0	2	1	0
GO	0	0	0	4	1	4	0	0	2	0	0	0	0	0
GS	0	0	0	2	0	3	0	0	1	0	0	0	0	0
CF	0	0	0	0	4	4	0	2	3	0	0	0	0	0
CM	0	0	0	0	0	1	1	8	6	0	0	0	0	0
CA	1	1	0	1	4	0	3	5	2	0	1	1	0	0
PA	0	0	0	0	0	0	0	2	1	0	0	2	0	0
PT	0	0	0	0	0	0	1	0	7	2	3	2	0	0
PR	0	0	0	0	0	0	0	1	0	5	7	9	3	1
VP	0	0	0	0	1	0	0	0	1	0	4	2	0	0
VA	0	1	1	1	0	0	0	0	1	1	0	4	3	0
SP	0	1	0	0	0	0	0	0	0	0	1	0	0	0
SA	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SO	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Part II.

Development of an Intelligent System for Privacy Requirements and Design Engineering

4. Contributions to Privacy Requirements Engineering

This chapter introduces three key contributions to privacy requirements engineering: (1) a Named Entity Recognition model for identifying privacy-related entities, (2) a model for automatically generating data flow diagrams from user stories, and (3) a model for automatically generating privacy requirements. This work has been published in workshops [67,70,71] and a journal [75].

4.1. Introduction

The principles of "privacy by design" and "privacy by default" have a long-standing tradition [28] and are today obligatory by legislation. These principles are most prominently highlighted in the European Union's General Data Protection Regulation (GDPR), which comes into force in 2018. In this context, it is essential to raise awareness among developers and ensure that systems are built in an a-priori privacy-aware way. Since the introduction of the GDPR, the integration of privacy aspects into the software development process has become a key concern and a major challenge for the industry, especially for digital product developers. To address this challenge, privacy engineering has emerged as a research framework that focuses on integrating data protection into organizational and technical measures [60].

Privacy engineering bridges the gap between legal obligations, privacy policies, organizational policies, and the realization of systems or technologies under development [41]. Privacy engineering must be integrated into the software development lifecycle (SDLC), starting with requirements analysis, design, and implementation [122]. The specific activity in the requirements phase can be referred to as Privacy Requirements Engineering (PRE). These activities include analyzing assets, identifying data subjects, and drawing data flow diagrams to determine the necessary data protection and privacy requirements. The popular approaches to eliciting privacy requirements are conducting a risk analysis [87], threat analysis [40], and privacy impact assessment (PIA) [123]. These analyses are conducted in the early stage of development. However, many privacy engineering methodologies depend heavily on a plan-driven approach that can be time-consuming and not tailored to the agile speed that much of the industry is currently taking [21]. This agility falls under agile software development (ASD) methodologies such as Scrum, which typically cannot capture the full complexity of the system at an early stage [119]. The turn from a plan-driven approach to ASD means adopting new ways to make development faster and more efficient [62]. Consequently, modeling privacy threats or designing a

4. Contributions to Privacy Requirements Engineering

privacy-friendly system becomes more challenging [51]. The steps in PRE methods such as assets identification and modeling the system to identify privacy threats could take an immense amount of time [171], preventing the iterative nature of the agile approach in changing requirements [109]. Researchers have studied these challenges [51, 93], and clearly state the conflicting nature of agile and privacy engineering.

The agile software development methodology promotes user stories to elicit functional requirements (FRs). As a result, user stories have become the primary form of requirement elicitation, representation, and documentation in ASD [168]. Despite their simplicity, user stories have been shown to effectively refine customer requirements and make them more detailed and precise [168]. Typically, customer requirements are expressed as FR and do not encompass non-functional requirements (NFRs), such as security and privacy. Eliciting NFRs requires expertise to be defined and are often specified in separate sessions to ensure their completeness [7]. However, this approach can result in NFR being treated as an afterthought, potentially leading to technical debt or even system failure [7].

Defining NFR early on aligns with the principles of privacy by design. The use of user stories has been proposed as a means to facilitate privacy requirements engineering (PRE) in agile situations [15, 122]. However, relying solely on user stories for NFR can become an issue [85], highlighting the importance of considering functional requirements (FR) in conjunction with NFR, especially when it comes to PRE. Addressing NFRs in conjunction with FRs is essential for successful identification and implementation of NFR in ASD [85]. In the context of privacy requirements, this means that agile teams need to be able to identify privacy criteria in FRs, with research showing that agile teams still struggle to identify privacy criteria in user stories [26]. The use of computerized tools has been suggested to assist agile teams in identifying NFRs [7, 125, 145]. In the domain of PRE, several tools have been proposed in literature [111, 132]. Despite the availability of tools, the manual work involved, such as identifying actors, personal data, and completing numerous forms, can hinder the utilization of these tools to their full potential. We very much believe that incorporating an intelligent system (IS), particularly machine learning focused on natural language processing (NLP), can help agile teams to significantly reduce this manual workload [3, 136, 173].

The importance of incorporating IS into ASD has been highlighted by various studies investigating the benefits of utilizing NLP-based models in user stories [136, 143]. These studies have shown the potential benefits of using NLP for PRE activities, such as defect detection and visual model generation [143]. However, a systematic mapping study has revealed that NLP solutions have yet to be fully incorporated into proposed PRE methods [1]. Our research aims to fill this gap by exploring the use of NLP for a comprehensive pipeline of PRE activities.

We present a set of NLP solutions aimed at streamlining various activities within the PRE process. These solutions aim to automate manual and time-consuming tasks, such as identifying assets and creating data flow diagrams from functional requirements, and ultimately reduce the huge burden of eliciting privacy requirements for agile teams. Our approach aims to facilitate the integration of IS in the complete pipeline of privacy requirements by utilizing user stories as the primary input and user stories as the final

output. As the process is nearly fully automated, the minimal effort of eliciting NFR alongside FR strongly supports privacy by design principles as a triggering facilitator for agile teams to avoid leaving NFR until later stages [85], thereby supporting privacy by design principles. In this chapter, we present several key contributions:

- We present a novel approach to enhance the PRE process by incorporating NLP-based intelligent systems and user stories as the primary input, thus simplifying the integration of PRE into agile workflows.
- We present a collection of privacy requirements presented as user stories, offering a set of patterns for eliciting privacy requirements in an agile development environment.

The core content of the chapter is structured as follows: Section 4.2 discusses recent approaches to privacy requirements engineering and natural language processing in user stories. Section 4.3 describes our proposed approach for incorporating NLP solutions into privacy requirements engineering. Section 4.4 discusses the reliability of the NLP approach in terms of performance compared to the gold standard. Section 4.5 discusses the limitations and potential impact of our proposed approach. Section 4.6 shows the threats to the validity of our approach. Finally, Section 4.7 summarizes the results of our work and discusses potential future work.

4.2. Background and Related Work

This section describes the privacy requirements engineering process based on recent work [10, 40, 111, 122]. We then discuss the natural language processing techniques that can be used to facilitate the application of privacy requirements engineering.

4.2.1. Privacy Requirements Engineering

There are numerous techniques for designing privacy requirements. Notario et al. [122] considered it from two perspectives: a goal-oriented and a risk-based perspective. The goal-oriented approach is centered on extracting principles and establishing them as the need for the system to meet. The goal of privacy or data protection can be derived from privacy principles and legal requirements. Hansen et al. [65] defined six data protection goals, which consists of *Confidentiality*, *Integrity*, *Availability*, *Unlinkability*, *Transparency*, and *Intervenability*. The Confidentiality, Integrity, and Availability (CIA) Triad is a commonly used framework for evaluating and defining security requirements, and is considered a cornerstone of information security. [152]. In the context of data protection, the CIA Triad can be redefined to meet specific requirements, such as privacy and data protection regulations [111]. These six goals, along with the additional goals, are used by Meis and Heisel [111] to develop a PRE method under the ProPAN method.

Meanwhile, the risk-based approach was more concerned with the threat that could arise from the movement of assets and data in the system. Privacy threat modeling is one discipline that uses a risk-based approach to address privacy. Deng et al. [40] present

4. Contributions to Privacy Requirements Engineering

LINDDUN, a comprehensive privacy threat modeling technique. LINDDUN is a mnemonic that represents a potential privacy concern in a system under consideration: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of Information, Unawareness, and Non-compliance. Prior to analyzing the threat, the analyst should provide a data flow diagram of the system (DFD). Each component of the DFD can then be linked to multiple threats in LINDDUN.

Privacy Impact Assessment (PIA) is also part of the risk-based approach. As with LINDDUN [40], the risk-based approach should begin with system characterization, which often includes asset identification and system modeling. Oetzel et al. [123] presented a seven-step process for eliciting privacy threats and documenting the PIA. This process can also be viewed as goal-oriented, as its second step incorporates the use of privacy goal catalogs, which provide a set of predefined privacy goals to guide the PIA process and ensure that all relevant privacy considerations are taken into account. Thus, it is important to recognize that an effective PRE strategy should incorporate both a goal-oriented and a risk-based approach.

Notario et al. [122] combine risk-based and goal-oriented approaches under the PRIPARE methodology. The goal-oriented approach, which is easier for a novice privacy engineer to follow, is the first step in reducing uncertainty in privacy engineering. The remainder of the potential risk is then addressed using a risk-based approach. The combination of goal-oriented and risk-based approaches provides a more comprehensive approach to PRE. P-STORE [10], the latest PRE which combines both approaches begins by establishing and prioritizing privacy objectives based on organizational goals. Prioritization is necessary because of the unworkability to meet all privacy goals simultaneously [65]. For example, if we value confidentiality goals, unlinkability goals will be overridden [111]. Once the goal has been specified, the requirements engineer can select the critical asset that will form the basis of the privacy threat analysis. These threats are mapped and prioritized in LINDDUN to become the ultimate privacy requirement to be addressed.

An approach that relies on extensive planning, such as ProPAn [111], is considered unsuitable for ASD [111]. The fundamental difficulty for ASD, such as Scrum, is the lack of system-wide planning. The PRIPARE method proposes a managerial approach to integrate its approach with ASD. The first is through an incremental approach, which means that privacy planning is only considered in its current iteration. In addition, privacy planning should be included prior to the start of the sprint. PRIPARE also considered including privacy in a dedicated sprint. In contrast to PRIPARE, Peixoto [132] aims to provide tool support to agile teams by guiding them through the specification of privacy requirements. However, extensive knowledge of privacy is required to fill out the form provided by the tools.

Agile methodologies encourage frequent communication between teams. This can be achieved in a variety of ways, one of which is the use of card games, an engaging and interactive approach [150, 171]. An example of this is Threat Poker [150] and LINDDUN GO [171], tools that empower Scrum teams to identify potential threats during the course of the game and subsequently assess the level of risk and the corresponding effort required to counteract these threats [150]. This card game also can serve as

an educational instrument, particularly for those team members with limited security and privacy expertise [171]. However, this strategy presents a challenge in that each team member needs to possess the requisite knowledge of potential threats associated with a given user story. In addition, practitioners caution that these tools may not be appropriate for more complex microsystems, highlighting the need for careful tool selection and adaptation based on the project's unique context and requirements.

The current literature presents several limitations in addressing PRE problems in ASD. Despite the comprehensive approach offered by ProPAN [111], it still requires a huge workload of careful planning and expertise to follow its plan-driven steps. LINDDUN needs a significant amount of time for modeling the system [40], also in-depth privacy expertise is needed to specify accurate privacy criteria [132]. The lightweight approaches offered by Threat Poker and LINDDUN GO may not provide a comprehensive view of the system, especially for complex systems. These limitations emphasize the need for a more comprehensive and automated approach to PRE, one that leverages the power of intelligent systems to enhance the capabilities of human experts and streamline the process of eliciting, analyzing, and mitigating privacy risks.

4.2.2. Natural Language Processing Applied to User Story Analysis

A user story is a brief description of a user's requirements in agile software development. It follows a semi-structured template, recommended by Cohn [34], to ensure clear communication between the development team and the customer. A user story is often utilized to express functional requirements, but it can also be used to express non-functional requirements, such as regulatory requirements [15, 119].

The consistent structure of user stories also allows NLP to assist with tasks such as creating models, identifying ambiguities and defects, understanding the structure of the story, and promoting traceability [143]. Ahmed et al. [3] applied NLP to identify quality attributes from user stories. This was performed by using regular expressions to match the user story pattern with quality attributes. However, the study [3] did not consider security and privacy criteria. In terms of security, NLP has been used to help analysts uncover security-related attributes in product backlogs, as shown in the study by Galster et al. ([50]).

The use of NLP in detecting privacy-related information in user stories was demonstrated by Casillo et al. [27] and Herwanto et al. [70]. They utilized NLP to assess the potential disclosure of personal data in user stories and its compliance with privacy requirements. Privacy-relevant keywords and entities, such as personal data, data subjects, and processing entities, were considered in the analysis [27, 70]. Herwanto et al. [70] also employed NLP techniques, specifically Named Entity Recognition, to extract privacy entities from user stories. They noted the potential for further privacy analysis through the examination of user stories. Nevertheless, none of this work is integrated into the full process of PRE.

System characterization is also a main challenge when conducting privacy impact assessment [123], or privacy threat modeling [171]. Although not explored specifically in the security and privacy domain, the automatic modeling of the system by NLP that

4. Contributions to Privacy Requirements Engineering

is sourced from user stories has been explored by some researchers [143]. Data Flow Diagram (DFD) is the main model used in privacy threat modeling [40]. The recent work from Herwanto explores the possibility of generating a DFD directly from the text of user stories [71]. Aside from DFD, transforming user stories into conceptual models is one of the tools explored by NLP [59]. Due to the time-consuming nature of system characterization, an automated modeling process is essential for agile teams. These studies also have not been integrated into the PRE process. Our aim is to streamline these processes and integrate them into the full cycle of PRE to elicit a comprehensive list of privacy requirements. Additionally, we provide a tool that allows human intervention in cases where inaccuracies in the NLP process may occur.

4.3. The Proposed PRE Model

This section presents our proposed PRE approach and the corresponding ML-based NLP model assigned for each PRE activity. Figure 4.1 illustrates the workflow of our model, which is divided into four main steps. The workflow shows that automation is carried out alongside human tasks performed by agile teams, as indicated by the white box in the figure. The automation tools are represented by the blue box, and the primary engine of these tools is the NLP models, depicted in the yellow box. These NLP models will be discussed in the following sections. Our approach incorporates the use of machine learning (ML) to perform automated quality check, detection of privacy disclosure and generation of data flow diagrams in user stories. While we recognize that a certain degree of human interaction will always be necessary due to the ambiguity of natural language, ML can help analysts by highlighting privacy-related sections of user stories.

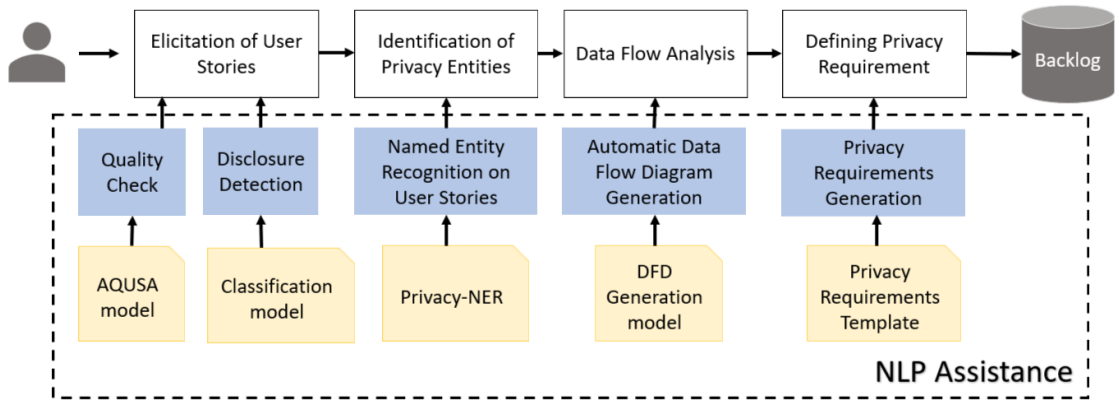


Figure 4.1.: Overview of the proposed workflow for privacy requirements engineering. The white box indicates that the process is performed manually, and the blue box indicates that the process is performed automatically with the support of the NLP module in the yellow box.

4.3.1. Elicitation of User Stories

Our approach is centered on user stories as the primary artifacts of agile requirements engineering. Therefore, instead of using questionnaires and forms to collect information about the system in terms of assets and data flow, we aim to use NLP to collect information from the user stories. The initial phase of the approach focuses on writing out the functional requirement in the form of a well-formed user story. Agile teams do not need to deal with a complete picture of the system to create privacy requirements using our approach, according to the lean and iterative nature of ASD. These user story sets constitute the basis for the next process of the PRE approach.

Automated Quality Check of User Stories

The free-form nature of user stories can result in poor-quality user stories. This problem obviously affects our NLP engine, especially when generating data flow diagrams. As a result, an automatic quality check is necessary before proceeding to the next steps. Our approach works best on the well-syntactically structured user stories suggested by Cohn's template [34]. There are two well-known quality standards for user stories, called INVEST (Independent, Negotiable, Valuable, Estimable, Scalable, Testable) and QUS (Quality User Story) [97]. In this research, we use the latest approach, QUS, proposed by Lucassen et al. [97]. QUS proposes 13 criteria for user stories in terms of syntactic, semantic, and pragmatic quality. The authors also provide a system called AQUUSA¹ (Automatic Quality User Story Artisan), which uses an NLP model to automatically check the quality of user stories. AQUUSA can automatically check 5 of the 13 quality criteria proposed in QUS. These qualities are (1) well-formed, (2) atomic, (3) minimal, (4) unique, and (5) uniform. The first three qualities are associated with the syntactic quality of user stories, while the last two are associated with the pragmatic quality. These five automatic checks are considered sufficient for our approach. Therefore, we integrate AQUUSA into the first step of our approach.

Detecting Privacy Disclosures in User Stories

The functional requirement written in the user story may potentially contain personal data that is subject to further generation of privacy requirements. To reduce the burden on the analyst to perform analysis on each user story, we propose to automatically determine which user story should serve as the basis for generating privacy requirements. We assume that whenever a user story contains information about the processing of personal data, a potential for disclosure exists. Consequently, the automated model must be able to detect the presence of this assumption. Therefore, we develop a supervised text classification approach that classifies which requirements are eligible for further processing in the PRE processing.

Previous research on privacy is mainly based on a list or dictionary of verbs (such as "access" or "ensure") or nouns (such as "data" or "database") to identify the disclosure

¹<https://github.com/RELabUU/aqusa-core>

4. Contributions to Privacy Requirements Engineering

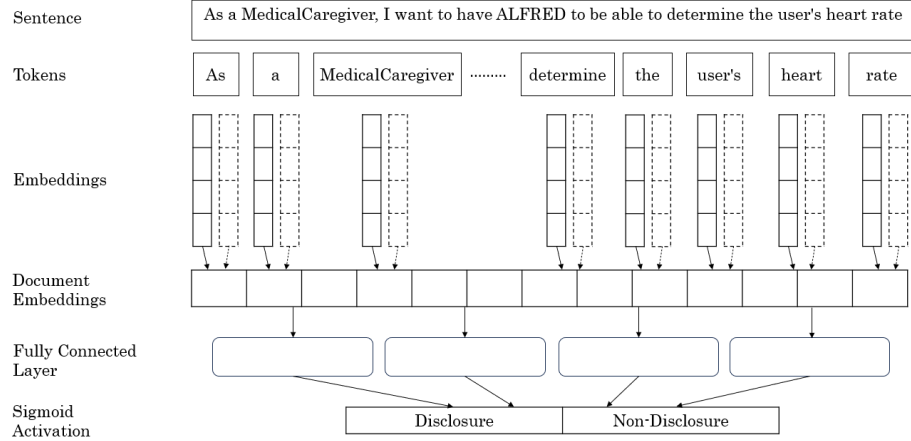


Figure 4.2.: The architecture of the Privacy Disclosure classification.

signals [27]. However, relying only on the existing dictionary could lead to false positives due to loss of context. A supervised machine-learning approach that relies on human labels can overcome this limitation. Since the user story dataset is limited, previous studies have used prior knowledge from other domains, such as social media text, and applied it to the user story text using transfer learning [27]. We intend to build on the concept presented in these previous studies by enhancing the privacy words based on the privacy vocabulary [127] and annotating the privacy-related entities by human experts on each of the user stories. In addition, by embedding semantics in context, the machine learning model can understand not only the word in the dictionary but also its synonyms.

The architecture of our privacy disclosure classification system is illustrated in Figure 4.2. The process begins by converting user stories into embedding vectors. These vectors are a blend of general word embeddings and contextual embeddings, providing a comprehensive understanding of the text's semantics. The embeddings are concatenated and then fed through a document embedding process, which could be either convolutional or recurrent, to identify and condense local textual patterns into a single document vector. This vector is then fed into a fully connected layer, which employs sigmoid activation to classify the user stories as either disclosure or non-disclosure. Additionally, our model includes a mechanism that transforms individual sentences from the user stories into document embeddings using transformer technology. Further details on the experiments will be discussed in subsequent sections.

4.3.2. Identification of Privacy Entities

Our approach focuses on the early identification of the data subject, the personal data, and the processing. In Article 4, Definition 1 of GDPR outlined personal data as "*any information relating to an identified or identifiable natural person ('data subject'). An identifiable natural person is one who can be identified directly or indirectly*". Then, in

Article 4, Definition 2, of the GDPR, the definition of processing is outlined as "*any operation or set of operations which is performed on personal data or on sets of personal data*". According to these definitions, we believe that identifying both the personal data and the data subject, as well as the processing that targets the personal data, are critical aspects of PRE. In our viewpoint, these entities can be spotted from the textual user requirements, including user stories. We propose an automatic detection of these entities through the use of Named Entity Recognition (NER). NER is an NLP task that involves identifying a collection of words or phrases associated with a specific Named Entity (NE) [116]. Therefore, we have built a named entity recognition (NER) system that targets three privacy-related entities: (1) personal data, (2) data subject, and (3) processing [70].

The NER model is treated as a form of supervised machine learning. As a result, the model requires ground truth data that has been manually labeled. The ground truth data is created by annotating the user story collection [39] with the three entities mentioned above. Specifically, I took responsibility for the annotation due to limited resources. To ensure validity, the annotation is done according to the guidelines. The primary source for the guideline is the Data Privacy Vocabulary (DPV), which is an established ontology for privacy entities. The ontology includes the list of personal data categories, processing categories, and data subject categories, which is highly applicable to clarify the decision to annotate our selected privacy-related entities. The annotated data is available in our open repository².

Due to the limited availability of annotated data in user stories, we utilize two data augmentation techniques: (1) synonym replacement and (2) mention replacement to enhance our training dataset. The method of synonym replacement involves substituting a given entity with a synonym. For example, the term "email address" in a user story could be replaced with its synonym "electronic mail address." This process relies on synonym databases like WordNET and the Paraphrase Database (PPDB).

The second technique, "Mention Replacement," involves substituting an entity with a different entity of the same category, randomly selected from our training data [38]. For instance, the Personal Data entity "consent forms" could be replaced with another Personal Data entity such as "privileged access". Unlike synonym replacement, where the meaning remains relatively the same, mention replacement introduces a variety of data within the same category. This method modifies the particular entity while maintaining the sentence's general structure, thus enhancing the dataset with varied yet pertinent instances. Table 4.1 contains an example of the augmentation process.

Alongside these methods, we incorporate non-user-story data sourced from the Data Privacy Vocabulary (DPV) ontology [127]. This involves integrating descriptions of various personal data categories from the ontology. We find this particularly beneficial for expanding the range of personal data entities covered in our dataset. The DPV ontology is specifically designed to standardize a broad spectrum of data privacy terminology, making it an ideal resource for enhancing our dataset with a more comprehensive array of privacy-related terms and concepts. The augmentation aims to improve recall, which

²<https://zenodo.org/doi/10.5281/zenodo.5801369>

4. Contributions to Privacy Requirements Engineering

is considered critical in automated tools for software engineering tasks [18].

Table 4.1.: Data Augmentation Example for Named Entity Recognition to Identify Privacy-Related Entities

Method	Story
Original	As a parent [Data Subject], I want to be able to see [Processing] which consent forms [Personal Data] I have submitted [Processing], so that I can know what I still need to do.
Mention Replacement	As a repository manager [Data Subject] I want to be able to see [Processing] which privileged access [Personal Data] I have share [Processing] so that I can know what I still need to do
Synonym Replacement (WordNet)	As a parent [Data Subject] I want to be able to see [Processing] which consent forms [Personal Data] I have submit [Processing], so that I can know what I still need to do
Synonym Replacement (PPDB)	As a parenting [Data Subject] I want to be able to see [Processing] which consent formats [Personal Data] I have achieved submitted [Processing], so that I can know what I still need to do

Once the ground truth data is established, we make use of the state-of-the-art feature representation BERT to represent the user stories. BERT stands for Bidirectional Encoder Representation from Transformers [42]. BERT was pre-trained on the Mask Language Modeling (Mask LM) task and the Next Sentence Prediction (NSP) task, which enables it to comprehend the context of a word based on the preceding and following words, commonly referred to as bidirectional context. Due to this capability, we chose BERT for our NER tasks. Our neural network model will receive the extracted feature representations, allowing it to learn the probability of sequences that appear as privacy-related entities in the user story. The main neural network model we employ is a Bidirectional Long Short-Term Memory (BiLSTM) network, which provides context from both the previous and subsequent words in the user story. This context is deemed crucial due to the dependence of entities on neighboring words. Finally, a Conditional Random Field (CRF) layer is added to incorporate sentence-level tag information rather than information at individual positions [79]. The architecture of the model depicted in Figure 4.3.

4.3.3. Data Flow Analysis Based on User Stories

Data Flow Analysis is an essential step in PRE as it highlights the flow of personal data in relation to functional requirements [111]. The privacy threat analysis framework, LINDDUN, uses Data Flow Diagrams (DFD) to visualize the system’s process model and data flow, enabling the identification of potential privacy threats and informing the creation of privacy requirements. DFDs are a concise and expressive modeling tool

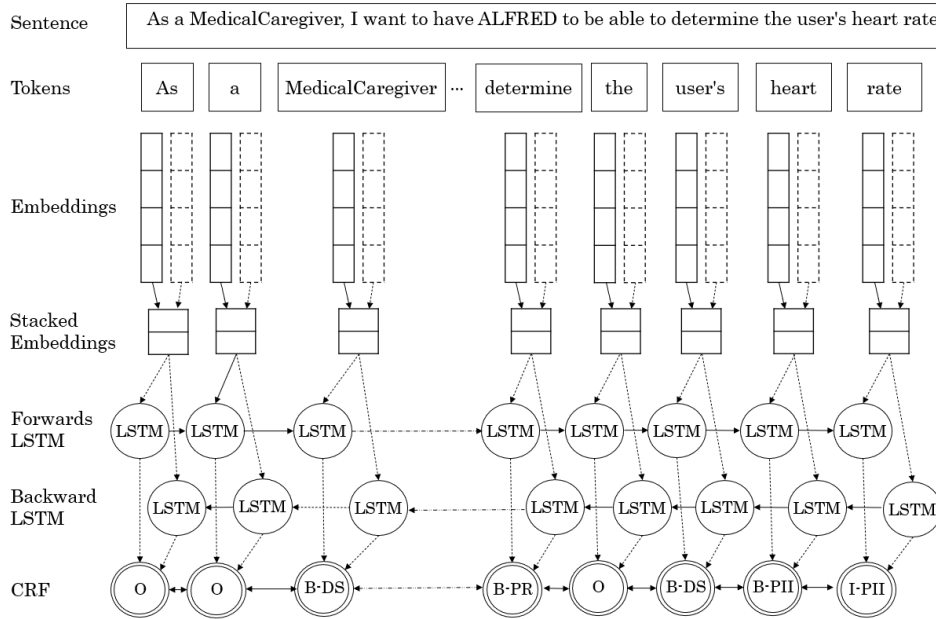


Figure 4.3.: The architecture of the named entity recognition model originally introduced in [70].

that is well-suited for the context of privacy threat analysis. However, the significant time and resources required to manually create DFDs pose a challenge for their practical use [123, 166, 171]. To address this, we have incorporated automation using NLP into our PRE approach. We have developed an NLP pipeline that automatically generates DFDs from textual user stories, as described in the research preview by Herwanto et al. [71]. In this section, we will elaborate on the work of Herwanto et al. [71] by focusing on the details of the DFDs, the transformation process from Robustness Diagrams (RDs) [54] to DFDs, and the improvement of the model by introducing an algorithm to transform the end parts of user stories, ensuring full coverage of information and preventing any missing information related to privacy entities. The workflow of automatically creating data flow diagrams can be seen in Figure 4.4. We publish our code and our results in the open repository³.

The Data Flow Diagram (DFD)

The DFD consists of four elements: processes, data flows, data stores, and external entities. In security analysis tools such as STRIDE [154], an additional element called the trust boundary is included. In order to ensure the accuracy of DFD, certain syntax rules must be followed when drawing them. There are 5 rules stated in Ambler [9]: (1) All processes must have at least one data flow in and one data flow out, (2) All processes should modify the incoming data, producing new forms of out-going data, (3) Each data

³<https://zenodo.org/doi/10.5281/zenodo.5801350>

4. Contributions to Privacy Requirements Engineering

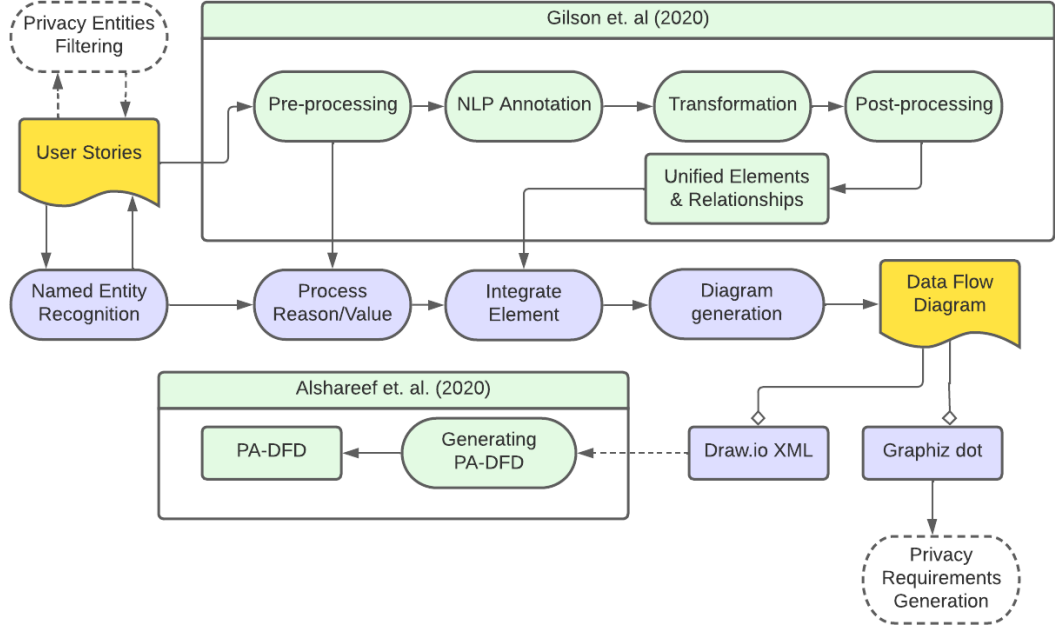


Figure 4.4.: The Workflow of the Automated Data Flow Diagram Generation [71]. The main contribution is highlighted in purple, and the use of other work is highlighted in green. The Robustness Diagram to the Data Flow Diagram can be seen in the connection to the work of Gilson et al. described in detail in Section C.2. The "Process Reason/Value" is explained in Section C.3 and composed of the existing Robustness Diagram from Gilson into DFD. The generation of the diagram is explained in Section C.4.

store must be involved with at least one data flow, (4) Each external entity must be involved with at least one data flow, and (5) A data flow must be attached to at least one process. These rules serve as a syntactic validation in our automation approach. We omit rule (2) for our approach due to the limitation of our information in the user story to be processed in the NLP model. Moreover, we are not focusing on the decomposition of the DFD level in this work, as we are generating entirely from the text of the user stories. The DFD which is used for threat modeling in both security [154] and privacy [40], also ignores DFD decomposition. The most important part of the Data Flow Analysis with DFD is that risk areas or hotspots can be identified. The NLP can help identify concepts and relationships needed to transform a collection of user stories into a DFD.

From RDs to DFDs

The approach to identifying concepts and relationships in user stories has been explored by Gilson et al. [54] to generate use case scenarios. The use case scenarios are depicted in

the robustness diagram (RD) described in Rosenberg and Stephens [149]. A robustness diagram can show a visual use case scenario diagram, showing the flow of the events. The objects in RD are grouped into actor, boundary, control, entity, and property. Due to the similarities of the RD with the DFD, we decided to adapt the model to our PRE pipeline. We use the model mainly to obtain the elements and relationships between the elements in the user stories. The elements may overlap with the entity from our NER model. Therefore, we use the information from the NER elements, such as the PII element, to indicate that the element as personal data, enabling visual hotspot detection directly from the DFD.

To generate the RD, the models separated the steps for generating concepts and relationships from user stories into four stages: (1) preprocessing; (2) NLP annotation; (3) transformation; and (4) post-processing [54]. The preprocessing steps are responsible for correcting pronoun and punctuation errors to improve the accuracy of further processes. After preprocessing, NLP annotation is carried out by part-of-speech (POS) tagging and dependency tree parsing⁴ [54]. Based on the knowledge of POS and dependency between the elements, rule-based transformations are applied to understand the linguistic characteristics of the user story. These rules will guide the grouping of the element along with their relationships. A total of 23 rules were used for transformations [54]. Some of the most important rules are explained in the original paper [54]. An example of a simple rule: if a role part contains a noun (N) or a proper noun (PRP), then the N or PRP becomes the actor's object. There are additional rules not explained in the paper, such as *conjunct rule* that aim to track multiple items or processes located around conjunction words. For example, the sentence "create a registration form for both staff and kids" will be captured as two separate processes, which are "create a registration form for staff" and "create a registration form for kids". Understanding the linguistic features allows the algorithm to categorize the words into the elements of the RD.

We have built a rule-based method that automatically transforms the RD into a DFD [71]. The graphical representation of the rule is depicted in the Figure 4.5. Once the RD has been successfully created, it can be converted directly into DFD. The *actor* in the RD can be directly converted to an *external_entity* in the DFD. The *control* in the RD can be converted to a *processing* in the DFD. While the *entity* in the RD can become a potential *data_store* in the DFD [71]. However, several elements need to be omitted, such as *boundary* element in RD. Before omitting, the *boundary* element in RD can become the indicator to connect the data flow between the *external_entity* and the *process*. The data flow between the *process* and *data_store* in the DFD can be converted from the connection between *control* and *element* in the RD. DFD does not allow data flow between *data_store*, although it is still possible for *element* in the RD to have an interaction. To solve this, we use string matching to find out which control this element comes from and connect them.

⁴<https://gitlab.com/mde4asd/ucscenario-gen>

4. Contributions to Privacy Requirements Engineering

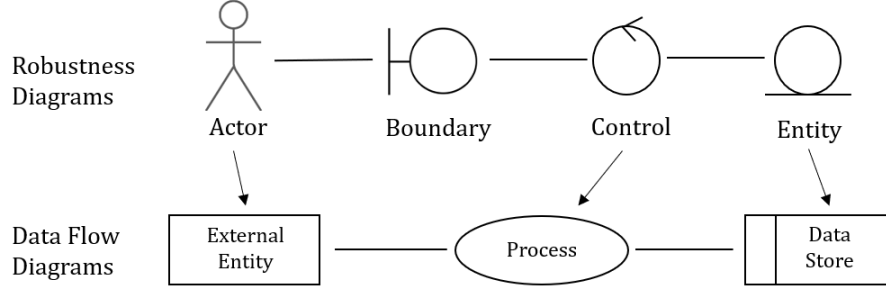


Figure 4.5.: The Transformation Rule from RD to DFD [71]

Algorithm 1: Processing Reason

```

1  ner1NER(1)
2  nerNER    input  : userStory
   output : actor, processing, dataStore

3  role, means, end = SplitUserStory(userStory)
4  if PII_Entity in
5  acrshortnerend then
6    processing ← GetVerb(end)
7    if processing = ∅ then
8      actor, processing, dataStore = ∅
9      return
10   end
11   if GetSubject(end) = "I" then actor ← GetSubject(role)
12   else if GetSubject(end) = "they" then actor ← GetSubject(means)
13   else actor ← GetSubject(end)
14   dataStore ← PII_Entity
15   return actor, processing, dataStore
16 else
17   actor, processing, dataStore = ∅
18   return
19 end

```

Incorporating the "so that" part of the User Story

One drawback of the RD in the approach by Gilson et al. [54] is that they omit the *end* part of the user story that usually starts with "so that". The *end* of the user story usually provides a reason or value that can describe a non-functional requirement. Moreover, our observation found that some user stories might contain valuable information such as personal data. However, we only want to focus when the end part of the user story indicates containing functional requirements. It can be characterized by the existence

of *processing* entity. Thus, we aim to capture this functional requirement and add it to DFD. We built a rule-based transformation described in Algorithm 1. The algorithm takes the user story as input and splits it to get the *end* part. We aim to capture three elements of the DFD based on the *end* part. These three elements can only be captured if only the *end* part contains *PII_Entity* and *processing* entity, which we are able to capture from the NER algorithm. Those two entities will become the *dataStore* and *processing*, respectively. The actor or *external_entity* will be decided by the subject in the *end* part as seen in the *GetSubject* function. There are three possible subjects; when they use "I," they refer to the subject mentioned in *role* of the user story. Meanwhile, when they use "they," the subject refers to the subject detected in the *means* part of the user story. Otherwise, the subject is detected from the *end* part of the user story. The output of this rule-based algorithm will then be integrated with the element and relationship produced by Gilson's approach [54].

Diagram Generation

The DFD can be produced in graphic format or draw.io format. The draw.io format enables the team to modify the DFD for further analysis. Optionally, the draw.io format transforms the DFD into a Privacy-Aware DFD (PA-DFD), putting privacy checks directly in the DFD notation [8]. However, the subsequent process does not require PA-DFD and instead relies solely on the basic DFD to generate privacy requirements.

4.3.4. Data Flow Diagram Generation using Large Language Models

We also saw the potential of using large language models (LLMs) to transform a set of user stories into a DFD. Figure 4.6 summarizes the workflow of the proposed approach.

The process begins when a development team defines a user story set. Since the DFD is intended to represent the data flow clearly, overcrowding it with too many elements or connections can lead to confusion and reduce its readability. Therefore, an optional grouping of functionalities is recommended. The diagram shows an example from the ALFRED project, taken from the user story dataset [39]. It is a virtual assistant that helps elderly people to stay active. The stories can be grouped into several functional groups, such as health and safety features, communication and social interaction, etc. Each functional group, identified by its thematic or functional similarity among user stories, is represented by a distinct Data Flow Diagram (DFD).

These groups of user stories can then be used as input to a prompt as shown in Figure 4.7. The prompt is divided into four main parts: Task Description, Detailed Instruction, User Stories Input, and Few-Shot Prompt. This structure is designed to sequentially guide the LLM through the process of understanding the task, the method of execution, the input to be transformed, and the format in which the output should be structured.

The task description introduces the objective for the LLM, focusing on the correct representation of DFD elements without delving into the specific syntax rules of DFDs. Adding detailed instructions aims to enhance the LLM's ability to abstract concepts and ensure compliance with syntax requirements. Abstracting information effectively is

4. Contributions to Privacy Requirements Engineering

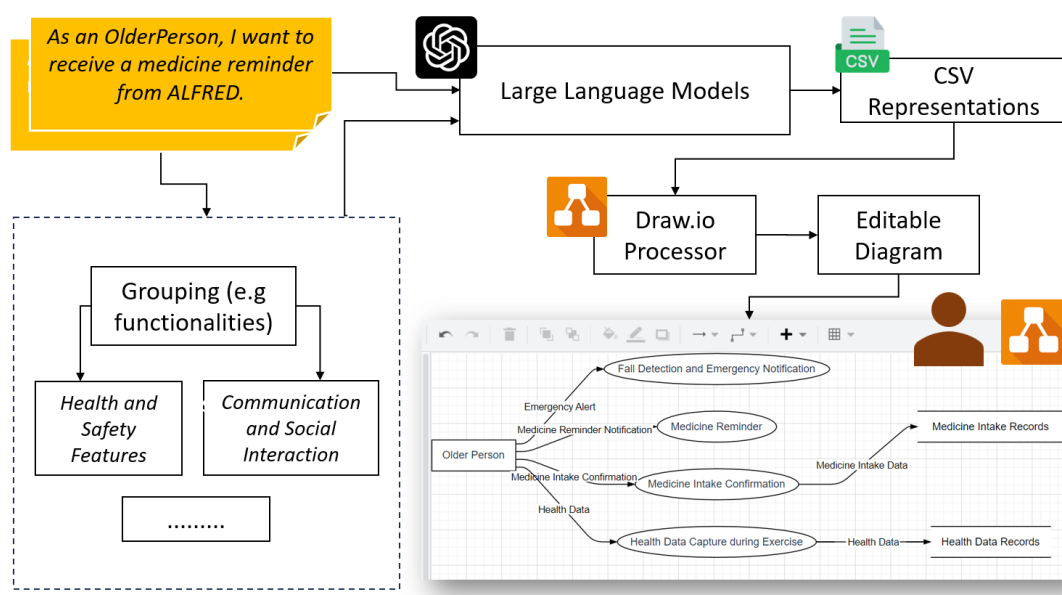


Figure 4.6.: The method for creating a Data Flow Diagram (DFD) starts with user requirements. From there, it optionally groups functionality before using prompts in LLMs. The output of the prompts is in CSV format, which is then imported into draw.io. This process results in a generated, editable DFD.

vital to prevent the LLM from treating user stories as separate entities. The LLM must identify similarities among processes, actors, and data stores, capturing the essential flow of information accurately. The Few-Shot Prompt section introduces a CSV template that outlines the mandatory sequence the LLM must follow, aligning with the predefined syntax for use in draw.io. It is argued that generating the specific syntax without any initial template (zero-shot generation) is impractical in this context, considering the specific syntax required by the custom draw.io CSV import.

The output generated in response to this prompt includes the requested CSV format and additional textual information. Due to the format of this output, users are advised to interact through a chat interface under human oversight.

Finally, the Draw.io CSV import feature is utilized by first defining the style of the DFD elements and connections to create a standardized DFD representation. One advantage of using draw.io for diagrams, as opposed to code-based diagrams [25], is its editable nature. This allows for human visual input, which can be beneficial when communicating with customers.

4.3.5. Privacy Requirements Generation

In this phase, the understanding of data subjects, personal data, processing, and their interconnection in the DFD is established as the minimum knowledge requirement for

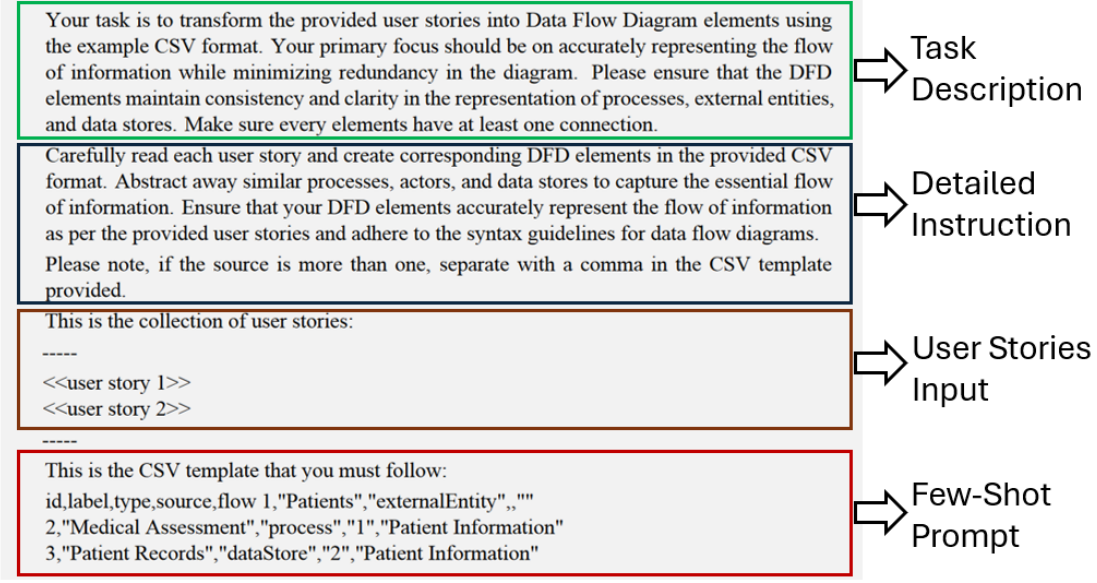


Figure 4.7.: Prompt to Generate the CSV syntax of the Data Flow Diagram.

recognizing potential privacy threats and requirements [40]. To build upon this knowledge, we propose an automated method for generating privacy requirements. Our approach is based on the best practice of using patterns to define NFRs in the ASD process [85]. Due to the commonness and universality of privacy requirements, we also propose using requirement patterns to define the template of the privacy requirement. The 12 categories of privacy requirements were selected from existing PRE approaches that have been evaluated and validated by the privacy research community. The first approach is ProPAN, which presents a comprehensive taxonomy of privacy requirements [111]. We also follow the refinements of the transparency [115] and intervenability [113] requirements that proposed in ProPAN. The second approach is the risk-based LINDDUN [40] which has demonstrated its practicality in real-world applications [171]. The detail of our selected privacy requirements with both approaches can be seen in Table 4.2.

We argue that all requirements, including privacy requirements, should be elicited in well-formed user stories to ensure traceability and clarity of user needs. We want to ensure that the user story conveys both the privacy requirement for each personal data processing and the achievable privacy values. Hussain et al. [81] discovered that value interception can be incorporated into user stories, specifically in the "so that" part. As a result, we took the approach of framing the privacy threat to privacy values as one that can be avoided if the requirement is implemented.

The patterns require three main elements in DFD that are connected to each other. The DFD consists of a set of external entity, which in our case could be considered as data subject $S = \{s_1, \dots, s_n\}$, a set of processing $P = \{p_1, \dots, p_n\}$ and a set of data store, which also could become a personal data store $D = \{d_1, \dots, d_n\}$. Triples are created whenever

4. Contributions to Privacy Requirements Engineering

Table 4.2.: Privacy Requirements Mapping

Privacy Requirement	LINDDUN [40]	ProPAN [111]
Confidentiality	✓	✓
Integrity	-	✓
Availability	-	✓
Unlinkability	✓	✓
Anonymity	✓	✓
Pseudonymity	✓	✓
Undetectability	✓	✓
Transparency	✓	✓
Intervenability	-	✓
Plausible deniability	✓	✓
Content Awareness	✓	-

there are data flows that connect the three entities or whenever data store and subject connect to the same processing. We formalized the triples as $t_i = \{s_i, p_i, d_i\}$. Within one DFD there will be a set of triples $T = \{t_1, \dots, t_n\}$. Each element of the triples will be subjected to 11 privacy requirements that will be explained in this section. Every privacy requirement will generate at least one user story. However, the requirements for Transparency and Intervenability can be more granular in that they lead to the creation of 2 and 3 distinct user stories, respectively. Thus, one triple will generate 14 possible privacy requirements $t_i = \{r_1, r_2, \dots, r_{14}\}$. The detailed user story pattern can be seen in each of the following sub-sections of privacy requirements.

Confidentiality

Confidentiality refers to the practice of maintaining the secrecy of personal data and preventing its disclosure to unwanted actors [111]. We took the privacy requirement confidentiality from Meis et al. [111] and modified it to become a user story. The confidentiality requirement (r_1) requires knowledge of the data subject, personal data, and processing.

confidentiality(s_i, p_i, d_i): As a s_i , I want the d_i data that is processed in p_i to be kept confidential, so that unwanted actors are unable to access it.

Integrity

In terms of privacy, integrity refers to the practice of maintaining the correctness and accuracy of personal data. The integrity requirement (r_2) is taken as inspiration from the ProPAN method [111].

integrity(s_i, p_i, d_i): As a s_i , I want the d_i data that is processed in p_i to be prevented from faults or unwanted actors, so that the consistency and correctness of the data is not compromised.

Availability

In terms of privacy, availability refers to maintaining the accessibility of the data. The availability requirement (r_3) also included the ProPAN method [111].

availability(s_i, p_i, d_i): As a s_i , I want the d_i data that is processed in p_i to be prevented from faults or unwanted actors, so that the consistency, correctness, and availability of the data are not compromised.

Unlinkability

Unlinkability is intended to avoid disclosing links between privacy-sensitive data outside the domain that share a common purpose and context [65]. The unlinkability requirement (r_4) requires knowledge of the data subject, personal data, and processing.

unlinkability(s_i, p_i, d_i): As a s_i , I want the d_i data that is used in p_i to be protected from being linked directly or indirectly to other personal data within or outside of our system so that an attacker cannot link it to the identity of the subject in d_i data.

Anonymity

Anonymity requirements refers to the inability of unwanted actors to identify whether a subject exists in the system or not [111]. The anonymity requirement (r_5) requires knowledge of the data subject, personal data, and processing.

anonymity(s_i, p_i, d_i): As a s_i , I want that the d_i to be anonymized (or pseudonymized) when performing p_i data, so that unwanted actors cannot directly or indirectly identify subject in d_i data.

Pseudonymity

The requirements for pseudonymity (r_6) refer to the practice of concealing one's true identity by using an alternative name, commonly known as a pseudonym [111]. The decision to anonymize or pseudonymize is made by the developers, who decide based on the utility of the personal data.

pseudonymity(s_i, p_i, d_i): As a s_i , I want that the d_i to be pseudonymized when performing p_i data, so that unwanted actors cannot directly or indirectly identify subject in d_i data.

4. Contributions to Privacy Requirements Engineering

Undetectability

Undetectability refers to the inability of unwanted actors to determine whether an item of interest (IOI) exists in the system or not [111]. We took inspired from Meis et al [111] that defined the requirement pattern for undetectability, and we reform it as the user story. The undetectability requirement (r_7) requires knowledge of the data subject, personal data, and processing.

undetectability(s_i, p_i, d_i) : As a s_i , I want unwanted actors to be unable to sufficiently distinguish whether or not d_i data is present, so that I can safely perform p_i .

Transparency

Transparency is focused on informing (r_8) data subjects about how and why their personal data is processed [111]. The transparency requirement requires knowledge of the data subject, personal data, and processing.

transparencyInform(s_i, p_i, d_i): As a s_i , I want to be informed and consented that the d_i data is used in p_i , so that I can exercise my rights when it is used outside of this context.

The transparency requirement (r_9) also mitigates the policy and non-compliance threat mentioned in LINDDUN. In addition, GDPR also mandated the right of access for the data subject in Article 15. Inspired by Bartolini [15], we formulate transparency to access the personal data as:

transparencyAccess(s_i, p_i, d_i): As a s_i , I want to download a copy of d_i data that is used in p_i at camp, so that I can check their correctness.

Intervenability

Intervenability requires the controller to implement procedures that permit data subjects to exercise control over the timing, manner, and purpose of processing their personal data [111]. In addition, Hansen et al. [65] mentions that intervenability includes the right of rectification and erasure of data, as mentioned in GDPR Article 16 and 17. Inspired by Meis and Heisel [111], we formulate three intervenability requirements that focus on the ability to modify data (r_{10}), delete data (r_{11}), and withdraw consent (r_{12}). All of them requires knowledge of data subject, personal data, and processing. Thus, the pattern can be formulated as:

intervenabilityModify(s_i, p_i, d_i): As a s_i , I want to be able to modify the d_i data that have been processed at p_i without undue delay, so that I can prevent the inaccuracy of data.

intervenabilityDelete(s_i, p_i, d_i): As a s_i , I want to be able to delete the d_i data that have been processed at p_i without undue delay, so that I can exercise my right.

intervenabilityWithdraw(s_i, p_i, d_i): As a s_i , I want to withdraw my consent on the processing of p_i to the d_i data, so that I can exercise my right.

Plausible deniability

Plausible deniability (r_{13}) refers to the ability to deny having undertaken an activity that other parties cannot confirm or disprove [40].

plausibleDeniability(s_i, p_i, d_i): As a s_i , I want to have the ability to deny performing p_i on d_i data, so that unwanted actors unable to accuse me of doing such a thing.

Content Awareness

Content Awareness refers to educating the user of the system to be aware of the privacy-sensitive data in their system, and educate them to not overly share their privacy-sensitive data outside of the system [40]. The pattern of the content awareness requirement (r_{14}) only requires the data subject and personal data parameter.

contentAwareness(s_i, d_i): As a s_i , I want to be informed that I should not share the d_i data outside of the platform, so that my privacy or data subject in d_i data is not compromised.

4.3.6. Guided Threat Modeling

To address potential gaps in privacy requirements, it is beneficial to integrate a risk-based approach with the generation of privacy requirements. This deepens the analysis of privacy risks that may arise.

In a single Data Flow Diagram (DFD), a set of parameters, or triples, can be identified as $T = \{t_1, \dots, t_n\}$. These triples form the basis for generating context-specific questions aimed at identifying potential privacy threats. Moving from examining individual data flows, as per traditional LINDDUN methods, to focusing on these distinct triples enables a more contextualized methodology for eliciting privacy threats. Each triple $t_i = \{s_i, p_i, d_i\}$ represents an independent unit of analysis within the system.

We have adopted the LINDDUN GO [171] methodology to derive questions for threat elicitation. These cards, each focusing on a specific privacy threat like Linkability or Identifiability, guide the creation of tailored questions. By contextualizing these questions, we enhance their relevance and accessibility for threat modelers.

For each triple t_i , a set of questions Q_i is developed, contextualized to $\{s_i, p_i, d_i\}$. The complete set of these elicitation questions across the system is denoted by $Q = \bigcup_{i=1}^n Q_i$.

For example, a question from the "Linkability" card:

4. Contributions to Privacy Requirements Engineering

"Is the data from different individuals distinguishable?"

Is reformulated to:

Q_{L2} : "Is the d_i data from s_i transmitted via p_i distinguishable?"

This makes the question more directly applicable to the modelers. Responses to each question are binary—Yes (Y) or No (N)—with a response function $R : Q \rightarrow \{Y, N\}$ that maps each question to its answer.

The threat identification process then includes:

- Defining potential threats $\Theta = \{\theta_1, \theta_2, \dots, \theta_r\}$.
- Associating subsets of threats $\Theta_i \subseteq \Theta$ with each triple based on the responses to its questions Q_i .
- Mapping each triple and its question responses to identified threats with the function $F : T \times Q \rightarrow 2^\Theta$.

When multiple triples generate a polynomial number of questions, it is crucial to assess which triples require thorough evaluation to manage the volume effectively.

4.3.7. Contextualized Privacy Requirements

The formulation of contextualized privacy requirements involves using identified threats and their responses. For each question answered with a Yes, a specific privacy requirement is derived based on the threats associated with that triple, as denoted by Θ_i , and its context t_i . This process is formalized with the function $P : T \times \Theta \rightarrow R_i$, mapping each triple to its customized privacy requirement.

For instance, a tailored privacy requirement might be articulated as:

$R_{L2}(t_i)$: "As an s_i , I want to ensure that data from different individuals stored in d_i and processed through p_i remains indistinguishable to protect individual identities."

This approach helps align the resulting privacy requirements closely with initial user stories, providing clear and contextual templates for each identified privacy threat.

4.3.8. Prioritization of Privacy Requirements

The generation of privacy requirements resulted in 14 possible requirements in a triple. We can consider all of the requirements to be privacy requirements because they process personal information. Nevertheless, analysts can choose which privacy requirements to select according to their privacy objectives. It is also possible that the selection of one privacy requirement overrides the other requirements.

Several studies have examined the conflict between security and privacy requirements [65, 111]. Hansen et al. [65] introduced the three pairs of conflicting privacy goals: (1)

Confidentiality and Availability, (2) Integrity and Intervenability, and (3) Unlinkability and Transparency. These six protection goals are present in our approach, as seen in the previous section.

To mitigate the conflict between these goals or requirements, it is first necessary to establish Role-Based Access Control (RBAC) corresponding to the primary roles of stakeholders within the system. As RBAC is presumed to be governed by the principal functionality or security aspect, it is not incorporated into our Privacy Requirements Elicitation (PRE) approach. Meis et al [111] introduced a concept of stakeholders and counterstakeholders to identify who can access certain personal data in a process. In our generated requirements, we use only unwanted actors to indicate the person outside the trust boundary.

RBAC effectively manages conflicting privacy requirements. For instance, Confidentiality and Availability can concurrently be set as a requirement, with specific roles designated for data access, and unwanted actors prevented from disrupting data access and availability. The data owner or the subject can solely perform Intervenability, ensuring data integrity. Transparency in r_8 does not give the data directly to the subject and thus does not conflict with Unlinkability. However, Transparency in r_9 may conflict with Unlinkability if the data subject is ignorant of potential linkability. Consequently, we supplement this with content awareness in r_{14} to enhance the data subject's awareness of privacy disclosure.

Nevertheless, there is a prioritization that can be made between the requirements. The unlinkability, anonymity, pseudonymity, undetectability and confidentiality requirement can potentially be overlapped with each other [111]. The analysts can choose how strong the protection of personal data is. According to Meis and Heisel [111], once personal data becomes confidential to unwanted actors, it should fulfill the unlinkability requirements. However, if the user desires stronger privacy, the user can enforce the unlinkability requirement along with the confidentiality requirements. According to Meis and Heisel [111], Unlinkability can also include undetectability, anonymity, and pseudonymity requirements. Thus, one of these requirements can be chosen for a particular processing of personal data. Pseudonymity is the weaker data protection level and should only be chosen if the other aforementioned requirements cannot be met.

Plausible deniability may conflict with transparency and intervenability. If a user desires plausible deniability in a particular process, there should be no traces of the data subject conducting a certain process. This would make transparency (r_7) and all intervenability requirements less relevant. Moreover, this could potentially compromise integrity since the trace of change could not be detected. Nevertheless, plausible deniability is usually desired for very sensitive operations such as anonymous sender and recipient messages or electronic voting. Thus, it strongly depends on the purpose of the processing itself [40].

The requirement of anonymity and pseudonymity are not synonymous. Anonymity means that the personal data cannot be associated with the data subject. Pseudonymity, on the other hand, means that personal data can still be associated with the data subject. Depending on the objective of the processing, the analysts can choose whether they would prefer to have the personal data anonymized or pseudonymized.

4. Contributions to Privacy Requirements Engineering

The content awareness requirement can be considered a soft privacy requirement [40], so it can be considered a lower priority compared to the other hard privacy requirement [40]. Finally, all selected and prioritized requirements can be included in the backlog of privacy requirements.

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

The evaluation of our approach focuses on the feasibility of applying the NLP model, including the ML component, in each step of privacy requirements engineering, especially the privacy disclosure detection model and automatic DFD generation. We did not evaluate the AQUA [97] since we used the same model from the original work. We will conduct a new Named Entity Recognition experiment [70], where we exclude CamperPlus from the training data, as we intend to use it in our pilot example. This also applies in the disclosure detection model where we exclude CamperPlus data.

4.4.1. Evaluation of Disclosure Detection

Disclosure Detection is the first stage in determining if a user story will be issued for developing privacy requirements. Therefore a high recall is desired to avoid missing potential privacy requirements [18]. We start by defining our ground truth and then move on to the experimental setup to find the best model for our PRE purpose.

Establishing the Ground Truth

Our approach involves treating disclosure detection as a binary text classification task, inspired by the work of Casillo et al. [27], who used text classification to identify user stories with potential disclosure. This model was initially trained on various online forums [108] to detect unintentional disclosure of personal information in user posts.

However, we recognized a limitation in the work of Casillo et al. [27], as the ground truth used for text classification was based on a computer-generated label. This label depended on two conditions: the presence of privacy-related words in a user story, based on a dictionary lookup, and the prediction results of the model trained on online forums. We believe that such ground truth generation isn't suitable for our purposes.

With this in mind, we looked for datasets where the ground truth is assigned by human experts. We chose Herwanto's Named Entity Recognition (NER) dataset [70], as detailed in table 4.3. Although this NER dataset uses the same user stories as Casillo et al. [27], each user story was manually labeled for the presence of privacy-relevant entities.

To adapt it to our text classification model, we developed a rule that a user story would be assigned a positive label, implying a privacy requirement, if any privacy-related entities were found within it. As shown in Table 4.3, this rule results in 38% of user stories receiving a positive label, establishing our ground truth.

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

Table 4.3.: The overview of the data set used in our evaluation [39]. The Ann column denoted the number of user stories annotated by one of the privacy-related entities (either data subject, processing, or personal data attributes).

Project	Description	Ann	Tot	%
Planning Poker	The first version of the PlanningPoker.com	46	53	87%
University of Bath	of The University of Bath’s institutional data repository.	51	53	96%
Zooniverse	A platform for citizen science that enables everyone to assist with research tasks.	26	60	43%
Federals Spending	Online portal for disseminating open data on US government spending.	17	98	17%
Loudon	Land management information system for Loudon Country, Virginia	30	58	52%
Recycling	An online portal to help people recycle their waste.	31	51	61%
Open Spending	Website intended to provide a transparent overview of government spending.	15	53	28%
FrictionLess	Platform for discovering data insights	20	66	30%
Scrum Alliance	The initial version of the Scrum Alliance Website	26	97	27%
NSF	The redesign of National Science Foundation (NSF) website’s	11	73	15%
DataHub	Data portal for searching, exchanging, and publishing data online	22	67	33%
MIS	Duke University’s management information system	18	68	26%
CASK	A unified integration platform for Big Data.	38	64	59%
Neurohub	A web-based platform that allows researchers to store their data and collaborate with colleagues.	49	102	48%
ALFRED	A platform that is intended to assist elderly individuals with limited mobility in daily tasks	64	138	46%
BADCamp	Platform for conference registration and management	24	69	35%
RDA-DMP	Software for machine-actionable data management plans	22	83	27%
ArchivesSpace	An open source archives information management platform for maintaining and offering online access to archives, manuscripts, and digital objects.	10	57	18%
Duraspace	Software for controlling and distributing digital content that is free and open source.	10	100	10%
RAC DAM	Platform for Archivist	24	100	24%
CULRepo	Digital content management system for Cornell University	35	115	30%
		635	1680	38%

Experimental Setup

We conducted nine different experiments focusing on text classification, to evaluate the performance of different models using different token and document embeddings. Our goal was to identify the best-performing models. The first eight experiments focused on standard text classification tasks.

For the token embeddings in scenarios 1 to 6, we used a stacked embedding strategy for the textual data: the GloVe model [135], a general word embedding technique, and advanced transformer-based language models, namely BERT [42] and RoBERTa [96]. For document embeddings, we used three different neural network architectures to embed documents: CNN (Convolutional Neural Network), GRU (Gated Recurrent Unit), and LSTM (Long Short-Term Memory). CNNs can effectively capture local dependencies and identify key phrases that contribute significantly to the meaning of a text. The GRU and LSTM architectures are designed to enhance the model's ability to capture sequential data dependencies. GRUs simplify the learning process by using a reduced number of gates compared to LSTMs. This streamlined structure makes GRUs less computationally intensive while maintaining their effectiveness in learning sequence dependencies.

For scenarios 8 and 9, we experimented with a pure transformer-based method, which directly transforms sentences into document embeddings without the intermediary of traditional sequence-based approaches.

The last scenario in our study uses the Named Entity Recognition (NER) method proposed by Herwanto [70]. In this approach, we aimed to identify entities in user stories, with a particular focus on the prediction of Personally Identifiable Information (PII) entities as determined by NER predictions.

All text classification experiments were performed using the Flair framework. Detailed information about the hyperparameters used in these experiments can be found in our repository ⁵.

Discussion of Evaluation Results

The evaluation results in Table 4.4 show the performance of different combinations of token and document embeddings in detecting privacy disclosures. The combination of GloVe with CNN (#1) showed a good balance between precision and recall, resulting in a high F1 measure of 75.06%. Interestingly, stacking GloVe with RoBERTa (#2) slightly improved the F1 score, indicating a marginal improvement in the balance between precision and recall.

In contrast, using LSTM and GRU for document embedding (#3, #4, #5, #6) resulted in a decrease in performance compared to CNN. This suggests that CNN may be more effective for such tasks. BERT and RoBERTa as Transformer embeddings (#7 and #8) showed consistent and comparable results, highlighting the effectiveness of Transformers-based models.

Notably, the NER-PII approach with transformers (#9) showed high recall but lower precision, indicating its strength in identifying relevant instances but with a trade-off in

⁵<https://zenodo.org/doi/10.5281/zenodo.10474910>

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

precision. This highlights the different effectiveness and trade-offs of different embedding techniques and document models in detecting privacy violations.

Based on the experiments, we propose to implement two separate models. The first model is based on scenario number 9, which has a high recall and would ensure that minimal user stories are missed during the privacy requirements engineering process. The second model is based on scenario number 2, which can be used when analysts want to prioritize the user stories to be addressed in the first iteration.

Table 4.4.: Privacy Disclosure Detection Evaluation

No	(Token) Embedding	Document Embedding	Precision	Recall	F1-measure
1	GloVe	CNN	79.40%	74.20%	75.06%
2	GloVe+Roberta	CNN	78.16%	74.21%	75.09%
3	GloVe	LSTM	70.17%	66.22%	66.52%
4	GloVe+Roberta	LSTM	76.51%	73.08%	73.48%
5	GloVe	GRU	71.53%	65.71%	65.83%
6	GloVe+Roberta	GRU	73.27%	67.73%	68.07%
7	BERT	Transformers	74.33%	69.13%	69.51%
8	RoBERTa	Transformers	73.48%	73.10%	73.21%
9	NER-PII	Transformers	44.5%	92.8%	60.1%

4.4.2. Evaluation of Named Entity Recognition

Our objective is to identify the most effective model for recognizing privacy-related entities in user stories. In terms of the experimental setup and establishing a reliable standard, we utilize the dataset provided by Herwanto, specifically the version that omits the CamperPlus data [70]. For training our model, we have chosen a learning rate of 0.1 and a mini-batch size of 32. We have set the patience parameter to 3 and the maximum number of epochs at 200, ensuring that training concludes due to a plateau in performance improvements rather than simply reaching the maximum number of epochs. The optimization of the model is conducted using Stochastic Gradient Descent. We employ the FlairNLP⁶ library for this training process. The evaluation result can be seen in Table 4.5.

Discussion of Evaluation Result

Table 4.5 presents the results of an experiment on Named Entity Recognition (NER) using two different models, BERT and RoBERTa, which have been previously recognized for their effectiveness in this domain [70]. The experiment involved the application of data augmentation techniques, specifically synonym replacement (aug-wordnet, aug-ppdb) and mention replacement (aug-mr), to enhance the models' performance.

Upon examining the results, it's evident that the mention replacement technique (aug-mr) significantly improves the performance of BERT models across most metrics.

⁶<https://github.com/flairNLP/flair/>

4. Contributions to Privacy Requirements Engineering

Table 4.5.: Experimental Results on Named Entity Recognition

		Data Subject			Processing			Personal Data		
		<i>Pr</i>	<i>Rec</i>	F_1	<i>Pr</i>	<i>Rec</i>	F_1	<i>Pr</i>	<i>Rec</i>	F_1
BERT	base	87.3	92.3	89.7	67.2	67.2	67.2	71.4	69.4	70.4
	aug-mr	90.7	94.2	92.5	67.7	71.9	69.7	73.9	73.2	73.6
	aug-wnet	86.3	93.0	89.5	64.7	75.8	69.8	69.9	65.6	67.7
	aug-ppdb	88.0	93.6	90.7	68.6	66.4	67.5	70.8	65.1	67.8
RoBERTa	base	85.1	94.9	89.7	67.4	77.3	72.0	66.2	64.6	65.4
	aug-mr	85.1	91.7	88.3	75.2	68.8	71.8	72.8	58.9	65.1
	aug-wnet	86.5	90.4	88.4	74.6	64.1	68.9	72.7	61.2	66.5
	aug-ppdb	85.4	93.6	89.3	74.0	68.8	71.3	71.9	66.0	68.8

Specifically, BERT with aug-mr outperforms its base and other augmented versions in terms of Precision (*Pr*), Recall (*Rec*), and F1-score (F_1) in all entity categories. For the RoBERTa model, however, the results indicate a different trend. Contrary to BERT, the RoBERTa base model outperforms its augmented versions, including the one with mention replacement. This is particularly evident in the Processing category, where the base version of RoBERTa achieves the highest scores in Precision, Recall, and F1-score. This suggests that the base RoBERTa model is already highly effective for NER tasks, and the mention replacement augmentation does not provide additional benefits in this case.

In terms of synonym replacement (aug-wnet (wordnet), aug-ppdb), the improvements are mixed and less consistent across both models. While some metrics show improvement, these are not as pronounced as those observed with mention replacement. For example, BERT augmented with aug-wordnet exhibits a slight decline in its performance in the Data Subject category compared to its base version.

4.4.3. Evaluation of Automated DFD Generation

The DFD is rendered fully automatically from the text of the user story. It can be rendered based on the combination of user stories and individual user stories. In this evaluation, we focus on automatically generating individual user stories.

Evaluation Metrics

The metric for evaluating DFD generation is based on the syntactic and semantic correctness of the generated DFD.

Syntactic correctness is defined as whether the elements of DFD is followed the legal rule defined by standard DFD notation [66]. We took the syntactic guidelines published by Ambler [9] to evaluate the syntactic correctness of our generated DFD. We determined the syntactic correctness by using two discrete values: whether it is syntactically correct (1) or not (0). Syntactically correct means that all elements of the DFD are present

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

and connected according to the legal rule defined in Ambler [9]. Syntactically incorrect, on the other hand, means that either one of the elements of the DFD is missing or the connection is not properly established.

According to Harel and Rumpe [66], the semantics of DFD can be defined in several ways based on the goal of DFD itself. As our DFD is meant to illustrate the data flow between data subject, processing, and personal data, from which the analyst can derive a potential privacy threat and requirements, we will concentrate on how easily the evaluator can comprehend the language and syntax in DFD. Since the DFD may contain several information, we determined using three discrete values: whether the DFD is semantically correct (1), partially correct (0.5), or not correct at all (0). Semantically correct means that the evaluator is able to understand the DFD by using the correct syntax and language in the sentences used in each DFD element and to locate the position of the personal data. Partially correct means that only half or more of the DFD elements are correct according to the definition of semantically correct. Meanwhile, semantically incorrect usually occurs together with syntactic incorrectness or error in the NLP model, especially the dependency parsing.

Evaluation Setup

The preliminary evaluation of the DFD generation has been explored by Herwanto [71]. The preliminary evaluation focuses on the CamperPlus project, will be explored in detail in Chapter 7 as the example for the overall approach. For a more comprehensive evaluation, we took all the projects in Dalpiaz's dataset [39] and focused on user stories with privacy-related entities. These user stories can be obtained from the same data we used in the NER evaluation and shown in Table 4.3. By focusing only on user stories that have privacy-related entities, we can align the evaluation with the semantic definition of the DFD.

Discussion of Evaluation Result

Table 4.6 shows the evaluation of our automatic DFD generation model. The first row shows the evaluation based on 635 user stories based on Table 4.3. Meanwhile, the second row eliminates some user stories that did not fulfill the quality check according to AQUASA automatic checking [97]. The results show that by fulfilling the quality proposed by AQUASA, our DFD generation can produce better syntactic and semantic quality. The best results show that 88,93% DFDs generated are syntactically correct, and 82,70% are semantically correct.

The results show that syntactic and semantic errors are still possible, even if the user story has passed the AQUASA quality check. The error is caused by NLP parsing, such as failure to recognize the dependent clause and negation. This error results in incomplete sentences or grammar, which normally occurs in the process part of the DFD.

We also found that several automatically generated DFDs still required human intervention. Some PII is not explicitly mentioned in the stories. Consider a user story "As a parent, I want to be able to message my child's counselors, so that I can voice my

4. Contributions to Privacy Requirements Engineering

Table 4.6.: DFD Evaluation based on syntactic and semantic correctness

Data	Number of Stories	Syntactic	Semantic
Priv	635	87.25%	80.79%
Priv + QC	497	88.93%	82.70%

concerns or check on my child's progress.". Our NER model identifies the message as processing but not PII, although it also functions as PII. Concerns and progress, however, are identified as PII, which implies the message itself. This example shows the drawbacks of our approach, which will obtain a literal word from the user stories, which sometimes requires further grouping and aggregation of the same attribute.

4.4.4. Evaluation of Automated DFD Generation with LLMs

We evaluate the feasibility of generating DFDs with various LLMs individually to determine which performs best at this task.

Experiment Setup

The study involved experimenting with four Large Language Models (LLMs). Among these, two were provided by OpenAI: GPT-3.5 Turbo and GPT-4 Turbo, accessible through the ChatGPT interface⁷. The other two open-source models, Llama2-70B⁸ by Meta AI and Mixtral-8x7B⁹ by Mistral AI, were accessed via the Together.AI¹⁰ chat playground.

For the empirical analysis, three software engineering instructors with experience teaching DFDs participated. They evaluated the DFDs generated by each method based on completeness and correctness on a scale from 1 to 5, where 5 represented the highest score. The LLMs used are pseudonymized to eliminate any bias resulting from the instructor's familiarity with the LLM's capabilities. Prior to the evaluation, a meeting was held to standardize the scoring methodology among the three evaluators and the author.

The evaluation criteria are as follows:

- **Completeness:** This metric assessed whether the DFD included all essential elements (processes, data stores, external entities, and data flows) for a comprehensive system description. Each component must be connected to at least one other component to avoid isolated elements.

⁷<https://chat.openai.com>

⁸<https://huggingface.co/meta-llama/Llama-2-70b-chat-hf>

⁹<https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1>

¹⁰<https://api.together.xyz>

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

- **Correctness:** This involved verifying that the DFD accurately reflected the system’s requirements, including logical data flow and consistent representation of external entities and data stores following the system’s external interactions. The goal was to ensure that the DFD faithfully represented the semantic content of the user stories.

From the open dataset [39], 17 projects were chosen, each featuring a variety of user stories. To keep the complexity of the DFDs at a manageable level for evaluation, five representative user stories from each project were selected. These stories were chosen to illustrate the system’s primary functions and demonstrate the effectiveness of LLMs in delineating data flows among various stakeholders, with a specific focus on stories involving multiple stakeholders.

Five out of the 17 projects were assessed collectively by the three evaluators to maintain a manageable workload for the evaluators. The evaluators then individually assessed three projects, while the author evaluated the remaining three. The projects selected for collective evaluation included Alfred, CamperPlus, Recycling, NSF, and Datahub. To assess the completeness and correctness of the generated DFDs, a 1-to-5 Likert scale was used, enabling evaluators to quantify their judgments regarding the completeness and correctness of each DFD.

Generating DFDs began with manually entering the user stories into a chat interface, following a specific prompt described in Figure 4.7. The responses containing the desired CSV format were then integrated into predefined Draw.io CSV templates and uploaded to Draw.io for editing. The platform’s auto-layout feature was used to organize the diagram elements, eliminating the need for manual adjustments. All materials produced, including CSV files, Draw.io configurations, and final DFD images, were documented and made available in a dedicated public repository¹¹.

Performance Results

The DFD example produced by this approach is presented in Figure 4.8. Table 4.7 displays the median completeness and correctness scores for each LLM method from the five collectively selected projects. Among these, GPT-4 is highlighted as the most accurate in generating DFDs, with the highest scores in completeness and correctness. This underscores GPT-4’s superior capability in interpreting and converting user stories into DFDs. Mixtral, an open-source model, demonstrates commendable performance by producing useful DFDs with a respectable degree of accuracy and completeness. It outperforms closed-source models like GPT-3.5, which display moderate to lower effectiveness. Llama2 received the lowest scores, especially in correctness, suggesting that it faces challenges in understanding the semantics of the user stories. These results can also answer the two research questions:

Answer to **RQ1**: GPT-4 outperformed other LLM for generating syntactically correct DFDs, scoring a median of 4.0 on a 1 to 5 Likert scale.

¹¹https://github.com/gunturbudi/drawio_llm

4. Contributions to Privacy Requirements Engineering

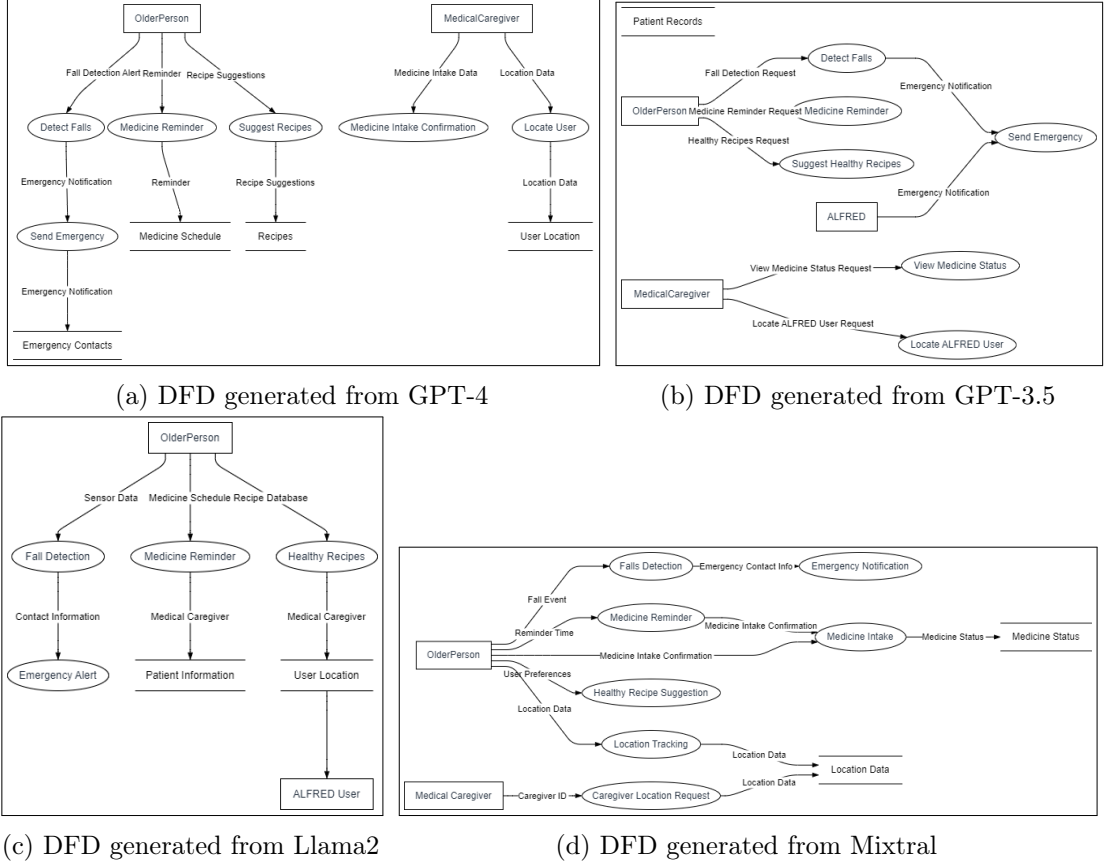


Figure 4.8.: DFDs Generated from Large Language Models

Answer to **RQ2**: GPT-4 outperformed other LLM for accurately representing system functionalities in DFDs, scoring a median of 4.0 on a 1 to 5 Likert scale.

Note that there are no special syntax rules in the prompt. This demonstrates the LLM’s internal knowledge of standard DFD conventions. Even if it is not a perfect score in terms of completeness and correctness, including human involvement in the further processing of the DFD could improve the applicability of the generated DFDs. In addition, the chat modality of these models offers the potential for iterative refinement of DFDs. Future studies could explore how iterative prompting could guide LLMs in adapting and improving DFDs. The threshold between what humans would accept the prompt and later edit in editable diagramming tools opens up an interesting empirical observation.

Table 4.8 presents the correlation coefficients among pairs of evaluators, aiming to assess the level of concordance between them to ensure reliability in their assessments. To achieve uniform coding, scores are converted into rankings among LLMs, even if an evaluator assigns identical scores to multiple LLMs within a project. The ranking is determined based on the lowest rank within each group, using a dense ranking method

4.4. Evaluating the Feasibility of the Developed NLP-based Approach

Table 4.7.: The Median Scores of Completeness (Cm) and Correctness (Cr)

Method	Cm (5)	Cr (5)	Cm (17)	Cr (17)
GPT-3.5	2.0	1.5	2.5	2.0
GPT-4	4.0	4.0	4.0	4.0
Llama2	1.0	1.0	2.0	1.0
Mixtral	3.5	3.0	3.5	3.0

Table 4.8.: Correlation Coefficients between Evaluator

Evaluator Pair	Spearman	Kendall
1 & 2	.594 (<.001)	.532 (<.001)
1 & 3	.666 (<.001)	.549 (<.001)
2 & 3	.594 (<.001)	.505 (<.001)
Average	0.618	0.529

that increments by one across different groups for consistency and clarity.

All pairs of evaluators demonstrate statistically significant positive correlations in their scoring, with Spearman’s correlation coefficients ranging from moderate to strong (0.618 on average). While slightly lower, Kendall’s tau values also indicate positive relationships and are significant (0.529 in average). These findings suggest a consistent agreement in the scoring among the evaluators, with higher scoring by one evaluator associated with higher scoring by another. The statistical significance of these correlations (p-values at <0.001) strongly supports the reliability of these observed associations, indicating that such agreement in scoring is not due to random chance but reflects a genuine pattern of concordance among the evaluators.

Discussion

Completeness scores are typically higher than correctness scores for all models, indicating that while LLMs can accurately identify necessary elements of DFDs with the correct syntax, they struggle to grasp the underlying semantics or logical connections between processes. This finding is consistent with previous research [25] and highlights the importance of a collaborative approach between humans and AI in the early stages of system design, particularly in diagram modeling. Nevertheless, the ability of LLMs to understand the syntax of DFD without explicitly mentioning the syntax and convention is a potential way to adapt to other open and widely used models, such as Business Process Model and Notation (BPMN) and UML.

The evaluator also found that data flow in the DFD tends to move exclusively to the left in horizontal and downward in vertical orientations, as shown in Figure 4.8. This observation suggests that data typically flows from external entities to processes and from processes to data stores, but rarely in the opposite direction. This limitation is likely due to the examples used in the prompts, which demonstrated data flow in only one direction,

4. Contributions to Privacy Requirements Engineering

highlighting the critical impact that the choice of examples in few-shot prompts can have on the model's performance.

This research employed a single prompt, which hinders the LLM potential of prompt engineering. For example, explicitly specifying the desired criteria for the DFD within the prompt might have been more advantageous than relying on the inherent knowledge of the LLM. Additionally, exploring the possibility of incorporating various aspects of DFD into a few-shot prompt represents another viable strategy. Despite LLM's understanding of the DFD syntax, it's critical to refine the few-shot examples to ensure they can accurately represent all potential directions of data flow.

Finally, it's important to recognize the inherent risks and limitations of relying solely on AI-generated models, such as the potential for confusing them with reality, losing information, or applying models that are inappropriate or incomplete [99].

4.5. Discussion

In this section, we revisit several critical issues that we have addressed in this chapter. First, we briefly discuss the challenges faced by our approach in closing the gap through adopting PRE in agile software development (ASD). Next, we discuss in more detail the potential limitations that could result from our decision to rely only on user stories as input to PRE. We then examine the performance of the core NLP-based algorithms. Finally, we discuss the implications of our PRE approach.

Adoption of PRE in ASD: Integrating privacy engineering with agile methodology has proven to be a challenge, as reported in many studies [62, 93]. This challenge affects a wide range of activities throughout the development life cycle. One specific challenge is that the requirements for a system may change during the development process, which can impact the characterization of the system. As a result, the privacy requirements must be revisited whenever the assets that need to be protected or the type of processing change. To address this challenge, we propose using automation to review the requirements. Since the requirements are typically expressed in natural language text, we have used natural language processing (NLP) techniques to automate this process.

User stories as input for PRE: Our proposed PRE approach focuses on user stories, a widely-used method for agile teams to create functional requirements. This strategy has already shown its value and has led to significant changes in how user requirements are captured and identified. A well-known example of this approach is Peter Checkland's Soft Systems Methodology, which fully utilizes rich pictures to facilitate the requirement-gathering process [30]. NLP has been shown to be effective in a wide variety of different application scenarios. Therefore, we have based our methodology on NLP to provide a solution that enables analysts to analyze user stories and identify potential privacy issues. However, the sole use of user stories can limit the trade-off between speed and change that agile methods require. Regarding generating privacy requirements, the completeness of requirements derived from user stories can become an issue. Other PRE approaches, such as ProPAN, require extensive user input to characterize the system [111]. We provide a much leaner approach by performing an automated PRE in each user story. Such a

lean approach allows the team to have an early awareness of privacy requirements. In the future, conducting a specific study on whether a user story is sufficient to cover all privacy needs that arise in a given context may be possible.

The use of NLP in PRE: Natural Language Processing (NLP) for Requirements Engineering (NLP4RE) has been extensively studied and is effective in helping software developers with hairy tasks [19]. However, privacy engineering, which heavily relies on the requirements phase, remains a challenge for NLP4RE. This study uses established NLP models, particularly in text classification for disclosure detection, named entity detection for asset identification, and dependency analysis for data flow diagramming. This automation of PRE tasks has the potential to lead to unwanted consequences such as misinterpretation and bias. Despite using state-of-the-art deep learning models to train our model, we recognize that its performance still has room for improvement. The results of our feasibility analysis show that no model achieves perfect precision and recall, which is common for machine learning models. Research indicates that higher recall is desirable for tasks with significant requirements. In the case of the PRE process, we assume that a higher recall is preferable to minimize undetected privacy risks.

Envisaged Implications: The main foreseeable impact of our contributions is to increase the level of automation in dealing with privacy requirements in agile development projects. We see our model and prototype as enablers for a more efficient and, thus, more feasible analysis of privacy requirements. Beyond the impact of these focus contributions, the developed model can be adapted to other types of requirements. As NLP and especially NER have proven their value in our research context, this chapter aims to motivate a much wider use of these technologies to support requirements engineering. While none of the existing solutions can provide a complete set of privacy requirements, a high level of automation is an important step in freeing privacy professionals from being drowned in routine work and allowing them to focus on the requirements that require thorough legal review. On a more strategic level, our approach has demonstrated that fundamental gaps between different paradigms, such as agile approaches and the classic waterfall model, can be bridged. Moreover, the solution proposed in this chapter could be seen as a template for dealing with legal issues that would otherwise significantly slow down agile development processes if not addressed.

4.6. Threats to Validity

There are several threats to the validity of our research. Following the recommendation by Wohlin et al. [170], we divide the threat types into construct validity, internal validity, external validity, and conclusion validity.

Construct validity refers to the generalization of our experiment to the PRE process. The fact that we only use user stories as input to the PRE model may threaten the construct validity. In order to mitigate this potential threat to the validity of our results, we have considered the well-known PRE pipeline, such as LINDDUN [40] and ProPAN [111], and have applied our proposed NLP model within the pipeline of this framework. In addition, our NLP pipeline includes two external works on user story

4. Contributions to Privacy Requirements Engineering

quality checking [97] and generating robustness diagram [54]. Limitations present in both studies may impact the construct validity of our studies. To minimize this, we have incorporated a human-in-the-loop interface that enables the review and modification of the quality check results and generated DFDs, mitigating the potential impact of these limitations. Nevertheless, subjective evaluations may not accurately capture the constructs of completeness and correctness due to evaluators’ variability, oversight of diagram details, and biases.

Internal validity of our research refers to whether the pilot example of our model and feasibility evaluation of the NLP model makes a difference in the validity. To minimize the internal validity, we took several approaches. Since our approach focuses on the privacy-sensitive project, we chose to use CamperPlus as the pilot example of our methodology. We took samples of a story to show the generated privacy requirements. Another user story in CamperPlus might generate different requirements that might be invalid, thus subject to manual validation. Nevertheless, our method can be used not only for privacy-sensitive projects. Thus, the feasibility of the NLP model was evaluated not only on the CamperPlus but also in the public dataset with 22 projects from different domains [39].

External validity refers to generalizing our findings to another environment. The evaluation approach with 22 different projects was also chosen to mitigate threats to external validity. Nevertheless, our approach requires a well-formed user story, which might be difficult to implement in practice. We mitigate this by applying automatic quality checking. Since the model quality check relies on AQUA, it relies on the completeness and validity of the AQUA model. In the feasibility evaluation, we only show an overall result of the performance of each NLP model without specifying the individual projects to save space due to the number of our experiments. However, in generating DFD with LLMs, the potential bias introduced by selecting a limited, possibly simpler set of user stories might skew the results towards more favorable outcomes by not fully representing the complexity and diversity necessary for a comprehensive assessment of LLM performance in generating DFDs.

Lastly, threats to the *conclusion validity* refer to problems that affect the ability to draw the correct conclusion about the relationship between treatment and outcome. To mitigate this threat, we conducted two evaluation treatments, which have been discussed concerning internal validity. The first evaluation focused on the feasibility of our NLP approach through statistical analysis based on ground truth labeled by human annotators. The statistical measures use F1-measure, which measures the harmonic mean between precision and recall. Since there is no benchmark to measure the significance of our NLP approach, we only compare it within the context of our internal experiments. In privacy disclosure detection, where we find the F1-measure unsatisfactory, we seek a method with the best recall measure, which is preferable in the hairy requirement task [19]. The second evaluation is based on a pilot example with CamperPlus, which follows our approach’s step-by-step process that shows our model’s real output.

4.7. Chapter Summary

This chapter presents an agile approach to Privacy Requirements Engineering (PRE) that leverages advances in machine learning (ML)-based Natural Language Processing (NLP) and large language models (LLMs). Many PRE tasks, such as system characterization, asset identification, and diagram modeling, are often complex and time-consuming. Automating these tasks can enhance their thoroughness and help identify potential privacy issues.

User stories were selected as the main input for our approach because they are fundamental artifacts in agile methodologies like Scrum. Their semi-structured nature makes them suitable for exploration with NLP techniques. Our feasibility evaluation demonstrates that our NLP approach can effectively manage PRE tasks, including disclosure detection, entity detection, and data flow diagram (DFD) creation. We also explored the capability of LLMs to generate DFDs, ensuring the accuracy and completeness of the derived privacy requirements.

The primary contributions of this research include automating the end-to-end PRE process, taking functional user stories as input and producing privacy requirements user stories as output.. Additionally, this research introduces guided threat modeling by contextualizing threat questions within the privacy entities of the user story and generating privacy requirements based on these threats within that same context. This integration of user stories simplifies the adoption of our PRE model by agile teams. While our NLP solution does not introduce a new technique, it is specifically tailored to our PRE approach. We believe that integrating NLP solutions into requirements engineering will promote further advancements in the field of NLP4RE (Natural Language Processing for Requirements Engineering) research. We have developed a GUI-based tool¹² to support the workflow integration of the proposed approach on this chapter. This will be elaborated more in detail in Chapter 6.

¹²<https://zenodo.org/doi/10.5281/zenodo.7598314>

5. Contributions to Privacy Design Engineering

This chapter presents an intelligent system designed to streamline the process of deriving privacy design solutions from privacy requirements. This work was published in [74] and presented at the conference.

5.1. Introduction

The enforcement of the General Data Protection Regulation (GDPR) has increased the focus on principles such as privacy by design and privacy by default, requiring developers to embed privacy considerations throughout the development process [29]. However, there's a gap in understanding how to technically implement these principles, leading to the emergence of privacy engineering [156].

During the requirements phase, developers should identify the privacy needs of the system. These needs come from various sources, such as risks associated with the use of personal data [40, 111], privacy principles [137], and legal mandates [15]. As legal regulations and cybersecurity complexities escalate, developers are faced with an increasing number of requirements to manage. For example, the ProPAN methodology lists 13 potential categories of privacy requirements [111]. Addressing these manually is tedious, and although there is a mapping table between privacy requirements and privacy design solutions, it can be a problem for developers to choose the exact solution. [4]. In addition, developers may not have the breadth of knowledge necessary to fully understand privacy implications or may overlook critical privacy requirements amidst a variety of development tasks [146]. These factors underscore the need for an automated approach to help developers navigate privacy requirements and select appropriate privacy design solutions.

Research in Privacy Enhancing Technologies (PET) and Privacy Design Patterns (PrDPs) aims to address privacy requirements (PRs) [36]. PrDPs provide documented answers to recurring problems and are often structured in catalogs [36, 46]. However, existing catalogs do not provide an agile way for developers to identify the most relevant privacy patterns for their projects. Moreover, they do not consistently incorporate the latest developments in PETs.

In this chapter, we propose a novel recommendation system that assists in the selection of PrDPs by analyzing the textual content of PRs. We recognize that text, while simple in form, contains rich and detailed meanings specifically relevant to PrDPs. No current solutions provide this linkage, and our research aims to fill this gap by leveraging the

capabilities of natural language understanding to semantically match PRs with PrDPs. In addition, our method simplifies the integration of current privacy solutions, even without structured patterns, facilitating the adoption of privacy design methodologies. By automating the correlation of PRs with PrDPs using advanced machine learning techniques, we take a significant step toward implementing agile privacy-by-design practices [1].

5.2. Background and Related Work

Privacy requirements engineering (PRE) is a process of identifying the privacy needs of a system to be built [111]. Prior research has explored the transformation of these principles into specific requirements [111, 137]. Meis and Heisel [111], in addition to Pfitzmann and Hansen [137] outlined 13 main requirements, including Data Confidentiality, Undetectability, and Data Unlinkability. Additionally, requirements can arise from system-specific risks, using a risk-based approach. For instance, the LINDDUN method [40] employs threat modeling to define requirements. A combination of risk-based and goal-based methods is optimal for creating PRs [121].

In the field of PRE, various solutions such as PETs, PrDPs, and privacy-aware architectures have been proposed to tackle PRs and associated threats. Deng et al. [40] suggested the use of PETs to counter threats identified by the LINDDUN framework. There are efforts to consolidate these PETs into standardized patterns [63, 114]. To aid in the engineering phase, it is recommended to categorize PETs using specific templates [36, 114]. The template by Meis and Heisel [114] emphasizes integration aspects, while Colesky's template [36] focuses on both soft and hard privacy goals, leading to what is known as PrDPs. Although there are around 70 cataloged patterns, selecting the most appropriate PrDPs and automating this selection remains a challenge [4, 24]. An example of a PrDP is illustrated in Table 5.1.

Considering the diverse range of PrDPs available, it becomes crucial to have an effective method for identifying the most suitable options for specific PRs. This selection process depends heavily on understanding the context and aligning it with the criteria of PrDPs. Previous efforts have provided decision support using rule-based systems [129], ontology databases [57, 155], and text relevance methods [80].

Rule-based systems require patterns with clearly defined properties [129], whereas ontology-based approaches organize patterns in an ontology format [57, 155]. Text-based methods are crucial, especially since pattern information is often stored in text form. For instance, the ArchReco system [155] employs the TF-IDF algorithm to assess the significance of words within a document collection. Hussain [80], on the other hand, uses a supervised machine learning technique called Learning to Rank (LeToR) for text analysis. LeToR specializes in sorting and prioritizing information based on relevance, using labeled training data that includes search queries, potential candidates, and their relevance ratings [95]. Our approach adopts LeToR for Privacy Design Patterns (PrDPs), favoring context-aware recommendations over traditional methods like TF-IDF [155]. This approach aims to streamline the privacy requirements engineering with privacy design engineering.

Table 5.1.: Privacy Pattern Example: Who’s Listening

Pattern Name Who’s Listening
Summary Inform users of content where other users or unauthenticated persons having accessed the same content are listed, and may access any further disclosures.
Context Users of a service regularly shares its usage with other users. Sometimes these are users they know personally, and sometimes these are anonymous, unauthenticated persons. This occurs particularly in shared or collaborative environments where content is generated. Knowledge of the contributions of other users contributes to additional or refined content in general.
Problem Users do not know if the content they are accessing or have disclosed has been accessed or modified by others, nor if it is someone they know.
Forces and Concerns - Users want to know who can access their disclosures and those of others - Users want to know that specific other users have accessed or modified content - Controllers do not want users to be unaware of who can see their disclosures - Controllers want to log access to prevent abuse
Solution Provided that users know their access is not private, inform them of other users, even unauthenticated, which are also accessing the content in question.

5.3. Recommending Privacy Design Patterns

Matching privacy requirements (PRs) with the appropriate privacy design patterns (PrDPs) presents a significant challenge due to the often indirect textual relationship between PRs and PrDPs. Basic approaches that rely solely on textual similarity may not effectively capture the nuanced context necessary for identifying the most suitable design pattern. To address this issue, we employ supervised learning techniques, leveraging their ability to learn from labeled data to more accurately predict the relevance of PrDPs to specific PRs. This method allows for a more nuanced understanding of the complex relationships between privacy requirements and design patterns, beyond mere word structure.

We have developed two key machine learning strategies to enhance the system’s recommendation accuracy: a classification model that categorizes privacy requirements into distinct groups, and a Learning-To-Rank (LeToR) algorithm that orders design patterns by their relevance to the given requirements. The distinction between hard and soft privacy requirements is a critical aspect of our classification model. Hard privacy requirements focus on technical aspects of data protection, such as anonymity and encryption, while soft privacy requirements relate to policy and user consent aspects [4]. This differentiation is crucial because it allows our system to tailor recommendations more effectively by understanding the type of privacy protection each requirement seeks. By classifying PRs into these categories, we provide the LeToR model with valuable input features that significantly refine the final pattern recommendations, ensuring that they align more closely with the specific needs of the privacy requirement in question.

5. Contributions to Privacy Design Engineering

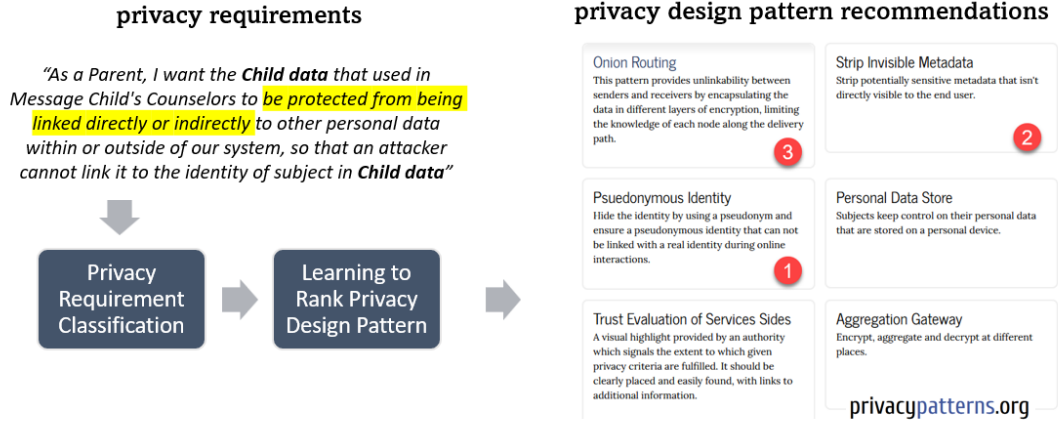


Figure 5.1.: Practical example showcasing the recommendation of PrDPs utilizing the learning-to-rank approach.

An illustrative example in Figure 5.1 visually explains the input and output process of our recommendation system, highlighting how these two stages classification and ranking work together to improve the system's ability to suggest context-appropriate PrDPs.

5.3.1. Datasets

Our approach mainly involves datasets containing Privacy Requirements (PRs) and Privacy Design Patterns (PrDPs). For the classification model, we utilize PRs tagged with their corresponding goals, either 'hard' or 'soft', to enable binary classification, along with a set of 13 distinct objectives for multiclass classification. In the context of the LeToR model, each pairing of a PR and a PrDP is assigned a specific relevance score, which indicates the level of relevance or suitability of the PrDP to the PR.

Privacy Requirements

We gather PRs data from three different PRE methods [72, 109, 118]. Examples of PRs from each method can be found in Table 5.2.

1. PrivacyStory [72] offers a tool for generating PRs. In total, we are able to generate 145 PRs based on two projects.
2. ProPAN [109] offers 16 requirements from a case study and additional text templates for each privacy requirement category.
3. Security Compass [118] presents 39 requirements in their paper.

Table 5.2.: Examples of privacy requirements for the three data sets: The first row is an example of a hard goal (unlinkability) and the second row is an example of a soft goal (transparency)

PrivacyStory [72]	ProPAN [111]	Security Compass [118]
As a Parent, I want the Child data that used in Enroll Children to be protected from being linked directly or indirectly to other personal data within or outside of our system, so that an attacker cannot link it to the identity of subject in Child data	For each pair of personal data links of the data Subject, counterstakeholders to whom the personal data personal Data are related to shall at most be able to link instances of the two elements of the pair to each other with linkability single	As a user, the goal is to be able to provide/withdraw consent to/-from the usage of data in identifiers that can be associated with individuals when combined with their personal data so that they cannot be identified indirectly by data in identifiers.
As a Parent, I want to be informed and consented that the Child data is used in Enroll Children, so that I can exercise my rights when it is used outside of this context.	The person concerned must be informed that his personal data is collected by the system executed by the controller.	As a user, the goal is to be able to easily access information about processing activities that involve individuals' personal data in clear and understandable language so they can exercise their right to view their processed personal data.

Privacy Design Patterns Catalogues

We found the most comprehensive catalog of PrDPs to be the one developed by Colesky et al. [35] which also available online¹. Although the patterns are not regularly updated, the total number of 71 PrDPs covering the objectives of PRs is sufficient for the proof of concept of our LeToR method.

Relevancy Score between Privacy Requirements and Design Patterns:

LeToR is designed to prioritize the most relevant Privacy Design Patterns (PrDPs) based on the requirements specified in the text. Since this process is supervised, it's essential to define a 'ground truth' that establishes how relevant each privacy requirement is to the PrDPs.

We draw on the initial mapping done by Al-Momani et al. [4] as a basis for evaluating the relevance of specific privacy objectives to privacy patterns. This mapping, found in Al-Momani et al.'s work [4], correlates eight privacy goals—namely anonymity, unlinkability, confidentiality, plausible deniability, undetectability, manageability, intervenability, and transparency—with PrDPs identified in the Colesky collection [36]. Although this provides a useful starting point for matching patterns to privacy goals, it doesn't always reflect the specific context of the requirements. For example, confidentiality-focused patterns such as 'Buddy List' and 'Encryption user-managed keys' may vary in their applicability depending on the scenario, which calls for additional context-specific adjustments. These adjustments are further refined with the help of human annotators.

¹<https://privacypatterns.org/>

5. Contributions to Privacy Design Engineering

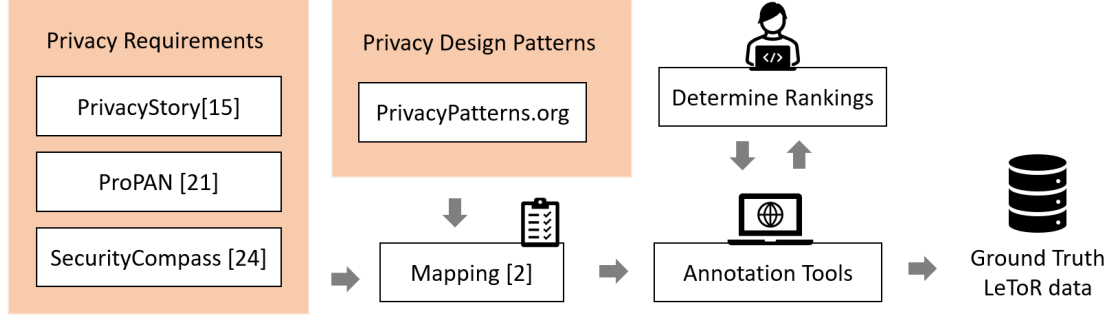


Figure 5.2.: Establishing LeToR ground truth data.

The refinement of these contextual adjustments and the evaluation of pattern relevance to privacy requirements are carried out by human annotators, notably the first author. Before the annotation process, the first author extensively reviewed all PrDPs to ensure a deep understanding of each pattern’s intended application and scope. This comprehensive review is critical for accurately assessing the relevance of PrDPs to the privacy requirements.

During the annotation phase, each privacy requirement’s relevance to the various PrDPs is scored on a scale from 1 to 5, where a score of 5 signifies the highest relevance and, consequently, the highest priority for recommendation. This grading system is facilitated through a custom-built annotation tool, designed to streamline the evaluation process and ensure consistent scoring. The outcomes of this detailed annotation exercise form the core of our training dataset, underpinning the supervised learning approach of our recommendation system.

To foster transparency and enable further research, we have made the dataset, annotation tools, and the recommendation system available in a public repository². The methodology for constructing the ground truth, from the initial review of PrDPs to the final annotation of privacy requirements, is depicted in Figure 5.2.

5.3.2. Privacy Requirements Classification

Privacy requirements are classified into 13 privacy goals [111,137], and PrDPs cover one or multiple goals. By identifying which privacy goals are linked to each requirement, we can suggest possible PrDPs. LINDDUN’s [40] concept of hard privacy comprises unlinkability, anonymity, pseudonymity, plausible deniability, undetectability, unobservability, and confidentiality. The soft privacy of LINDDUN incorporates awareness of user content, policy adherence, and consent compliance. ProPAN’s taxonomy, which is similar to LINDDUN’s soft privacy, emphasizes transparency and the ability to intervene [111]. Al-Momani maps the correlation between privacy goals and PrDPs, noting that a design pattern can address multiple goals, typically aligning with either hard or soft privacy goals [4]. Using this mapping, we classify PrDPs and PRs as either hard or soft privacy [4].

²<https://github.com/gunturbudi/pattern-recommender>

The classification workflow begins with preprocessing the requirement text, followed by tokenization and representation as a bag of words. Given the specificity of our dataset, we selected the Naive Bayes algorithm [106] for classification which is well-suited for small dataset. To improve the accuracy of our limited dataset, we integrated the EDA method [169].

During the prediction phase, users can input their PRs in textual format and avoid the restrictions of predefined templates [111] or initial mapping [4]. This input is subject to the same preprocessing steps, and the Naive Bayes model trained earlier categorizes the PRs into either hard or soft privacy goals. The result of this classification will serve as one of the features in identifying the relevance of PrDPs. Furthermore, the system uses the LeToR model to determine their priority.

5.3.3. Learning-to-Rank Privacy Design Patterns

To construct an LeToR model, two foundational steps must be followed: (1) selecting the appropriate features, also known as feature engineering, and (2) determining the relevance score between the query (PRs) and the documents (PrDPs). We can then apply the LeToR algorithms, which are explained in Section 5.4.

Feature engineering

is pivotal for the LeToR model's effective ranking. We adopt multiple features from the Microsoft LeToR dataset [64, 141] and interaction level of semantic features [58]. Specifically, we adopt three feature levels:

1. Query-level Features:

- **Number of Covered Query Terms:** The total number of terms from a PRs that are included in the PrDPs collection.
- **Ratio of Covered Query Terms:** This ratio represents the proportion of terms in a PRs that are covered by a PrDPs. It is calculated by dividing the number of terms in the PRs that are included by the PrDPs by the total number of terms in the PRs.
- **Character count** in the privacy requirement.
- **Query IDF (Inverse Document Frequency):** Measures a term's importance in a PRs, based on its rarity across PrDPs.
- **Query TF-IDF:** A metric that combines the frequency of a term in a PRs with its IDF.
- **Category** of PRs: Classifies PRs using binary (two-goals) or multi-class (multiple-goals) methods.

2. Document-level features:

- **Category** of PrDPs: Classifies PrDPs using binary (two-goals) or multi-class (multiple-goals) methods.

5. Contributions to Privacy Design Engineering

3. Interaction-level features:

- **Term frequency (TF)**: A measure of the count of each term in a PRs within the respective PrDPs.
- **Okapi BM-25**: Scores the relevance of a PRs to PrDPs based on TF and IDF.
- **Semantic similarity**: A cosine similarity of the embedding vectors between PRs and (1) PrDPs title, (2) PrDPs excerpt, (3) PrDPs content. Embedding vectors are generated using sentence transformers [147] that combine a general-purpose model³ with a domain-specific one designed for legal contexts⁴.
- **Hadamard Product of Embedding**: A vector produced by the element-wise multiplication of the embedding vectors from PRs and PrDPs. The production of these vectors employs the same sentence transformer models as those used for calculating semantic similarity.
- **Concatenation of Embedding**: A vector formed by the concatenation of the embedding vectors from PRs and PrDPs, maintaining the original vectors' structure. This method also uses the same embedding as the calculation of semantic similarity.

Our feature extraction method exclusively employs text-based attributes to ensure that both PRs and new PrDPs rely solely on textual data. This allows for the straightforward integration of PrDPs into the corpus, eliminating the need for manual curation of additional information, as is required for a PrDPs catalog [36].

5.4. Privacy Design Patterns: Results and Analysis

In this section, we present the results of our evaluation of the performance of our Learning-To-Rank (LeToR) model. Our analysis focuses on three main areas: evaluating the model's performance using standard metrics, investigating the importance of different features within the LeToR framework, and conducting a pilot example that applies our model to examples from the privacy requirements engineering method.

5.4.1. Experimental Setup

The goal of our experiment is to identify the performance of the classification model, as well as the optimal Learning-to-Rank (LeToR) algorithm and features for ranking PrDPs.

Classification Model

We are evaluating two classification models: (1) A binary classification model that classifies requirements as either "soft goals" or "hard goals". (2) A multi-class classification model that maps requirements to one of the privacy goals defined in [4]. This class will be

³<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁴https://huggingface.co/dean-ai/legal_heBERT_ft

used as a feature in LeToR, as mentioned in Section 5.3.3. Multinomial Naive Bayes was utilized as the classification algorithm for both cases. We assessed the classification model using 5-fold cross-validation.

Learning-to-Rank Model

We evaluate our LeToR model using the Normalized Discounted Cumulative Gain (NDCG). NDCG is calculated by first assigning a relevance score to each item in the ranked list, with higher relevance scores indicating more relevant items [31]. This initial score is called DCG. The formula can be seen in (1). Then, the relevance scores are discounted by the element's position in the ranking list, with higher-ranked elements receiving higher discounts. Finally, the discounted relevance scores are summed up and normalized by the maximum possible cumulative discounted score, to give the final NDCG score. The NDCG formula can be seen in (2). NDCG takes into account both relevance and rank position. NDCG aims to ensure that the ranked PrDPs are both relevant to the user and presented in an order that maximizes their usefulness. This can improve the user experience by providing them with PrDPs that are most relevant to the PRs. We use the "@k" value, which indicates the cutting threshold for the recommendation results. This means that only the top k items in the ranking will be considered when calculating the NDCG score. For NDCG, we use $k = \{1, 2, 3, 4, 5, 7, 10\}$.

$$DCG@k = \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)} \quad (5.1) \quad NDCG@k = \frac{DCG@k}{iDCG@k} \quad (5.2)$$

To enhance the model's generalization capability, we implement 5-fold stratified cross-validation on the training dataset, segmenting it into five equal parts of training and testing data. The stratification is conducted according to the binary classification outcomes of 'soft' and 'hard' goals in the context of privacy requirements, ensuring that each fold reflects a proportional distribution of these two categories. This approach guarantees that the evaluation of the model's performance is not skewed by an uneven distribution of the classification targets.

Our experimentation is structured into three distinct data sets. The first set, termed the baseline, includes features excluding the *hadamard* product and concatenation. This choice stems from the inherent sparsity of both *hadamard* product and concatenation features, which are derived from sparse sentence vector embeddings. The baseline set offers a clearer interpretation of each feature's individual importance. The second data set incorporates the *hadamard* product, while the third set combines elements from the baseline with both the *hadamard* product and concatenation features.

We utilize the LambdaRank algorithm for our LeToR experiment [23]. We have also experimented with other available algorithms, but LambdaRank has proven to be the best. Due to space constraints, we only show LambdaRank. LambdaRank is an algorithm derived from RankNet [23], which is itself a pairwise ranking algorithm using a neural network architecture. The lambda refers to the use of gradients of an approximation to the NDCG metric during training. This means that the model does not just learn to predict the correct ordering of documents, but it optimizes for the highest possible NDCG

5. Contributions to Privacy Design Engineering

Table 5.3.: Performance of Learning-to-Rank Algorithm

Data	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5	NDCG@7	NDCG@10
baseline	72,78%	70,44%	69,60%	69,93%	70,12%	72,17%	73,89%
+ hadamard	79,56%	79,15%	78,16%	79,48%	79,59%	80,79%	82,22%
+ hadamard, concat	80,64%	80,14%	79,54%	80,12%	80,62%	81,30%	83,05%

score, which makes it particularly suited for scenarios where the ranking quality at the top of the list is more important than elsewhere. We utilize LightGBM framework for the implementation of LambdaRank [92].

5.4.2. Model Performance

Classification Model Performance

The binary classification model achieved notably higher performance, with an accuracy of 96.52% and an F1-measure of 97.31%. The multi-class classification model achieved an accuracy of 83.77% and an F1-measure of 83.40%. Despite the lower performance of the multi-class classifier compared to the binary one, its results are still considered acceptable for incorporation into the LeToR model as a feature, given the potential for other features to compensate for inaccuracies. Moreover, certain misclassifications may be acceptable, as individual PrDPs can simultaneously address multiple goals. For instance, the "Protection against Tracking" pattern covers both unlinkability and anonymity, as shown by Al-Momani et al. [4].

LeToR Model Performance

The baseline model, excluding the *Hadamard* product and embedding concatenation features, achieves NDCG scores from 72.78% at NDCG@1 to 73.89% at NDCG@10. Adding the *Hadamard* product ('+ hadamard') enhances NDCG scores significantly, rising from 79.56% to 82.22%. The best results occur when combining the *Hadamard* product with concatenation ('+ hadamard, concat'), pushing NDCG scores from 80.64% to 83.05%, indicating that integrating the concatenation features refines the model's ranking performance. The detailed NDCG performance can be seen in Table 5.3.

5.4.3. Feature Importance of LeToR

In this section, we analyze the significance of features that enhance the performance of our LeToR model discussed in the previous section. Our first focus is on a third model that utilizes both 'hadamard products' and concatenation techniques to expand the baseline model's 36 features to a total of 9,252 features, mainly originating from the embedding vectors of PRs and PrDP. We assess the significance of each feature, categorizing them into three groups: the initial 36 features (baseline), features numbered 37 to 4,644 (hadamard), and features 4,645 to 9,252 (concatenation). We then calculated both the sum and the average importance of these features. The summed importance, depicted in Figure 5.3,

indicates that the '*hadamard*' features are the most significant overall. The importance of the baseline comes last due to its limited number of features. However, when we examined the average importance, we found that the baseline features are individually quite significant to the model's overall effectiveness. This significance is followed by the '*hadamard*' and then the '*concatenation*' features. The average importance was considered due to the sparse nature of the vector embeddings, which led to many features having zero importance.

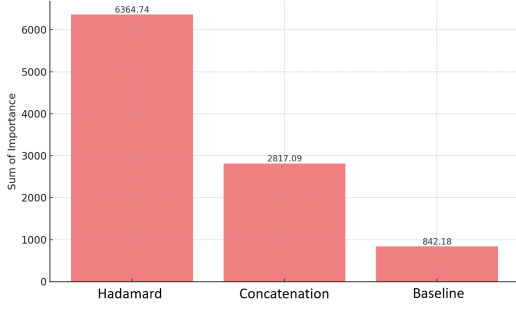


Figure 5.3.: Sum of Importance

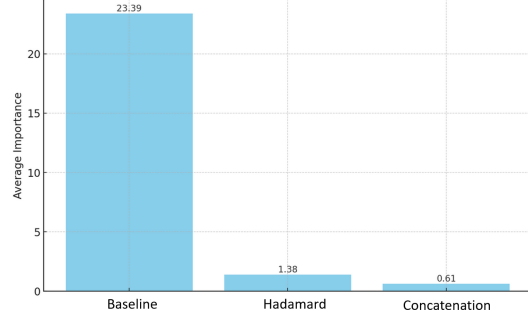


Figure 5.4.: Average of Importance

We examined the importance of individual features using SHAP values [101], which quantify a feature's contribution to the prediction's deviation from the dataset's average. The top features for our best model, as shown in Figure 5.5, are mainly semantic similarities between PRs and PrDP 'titles', 'content', or 'excerpts', with a strong emphasis on 'hadamard product' features. Notably, a single cosine similarity value between PRs and PrDP 'titles' was highly influential, suggesting that even simple measures can significantly impact the model's decisions alongside complex features like 'hadamard products'. It is noteworthy that the label "#1" following each feature implies the use of 'general' rather than 'legal' embeddings, suggesting that the latter do not significantly improve the performance of the model, which may warrant further analysis.

In the bee swarm plots, each point represents a SHAP value for a feature concerning a single data instance. The varied SHAP values (in Figure 5.6) for 'Hadamard Content #1 562' highlight its significant and diverse influence on model decisions, reflecting the feature's variability. In contrast, the concentration of high SHAP values for 'Title Similarity #1' on the positive end indicates a consistent and strong positive effect on rankings, with higher similarity scores correlating with higher ranks.

The baseline model is examined in greater detail as a result of its influence on the individual features. Similar to the previous model, as depicted in Figure 5.7, the cosine similarity between semantic embeddings between PRs and PrDPs 'title' is the most significant feature. It is also crucial to emphasize that the binary classification of 'hard' and 'soft' goals on both PRs and PrDPs influences the significance metrics employed in the model. Additionally, the beeswarm plot in Figure 5.8 reveals that features like 'Binary Pattern' and 'Content Similarity', with predominantly positive SHAP values, are generally indicative of higher relevance, which in turn, promotes a higher ranking of the

5. Contributions to Privacy Design Engineering

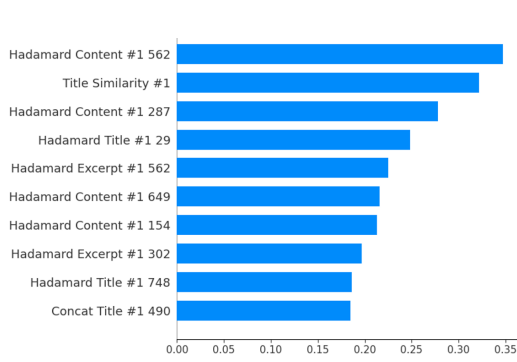


Figure 5.5.: Global Feature Importance (+hadamard, concat)

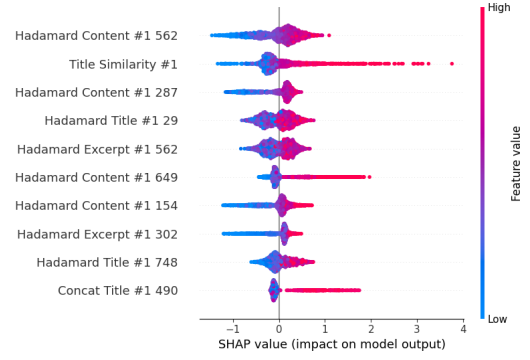


Figure 5.6.: Local Explanation Summary (+hadamard, concat)

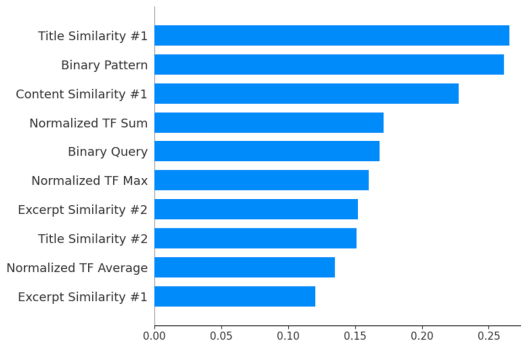


Figure 5.7.: Global Feature Importance (baseline)

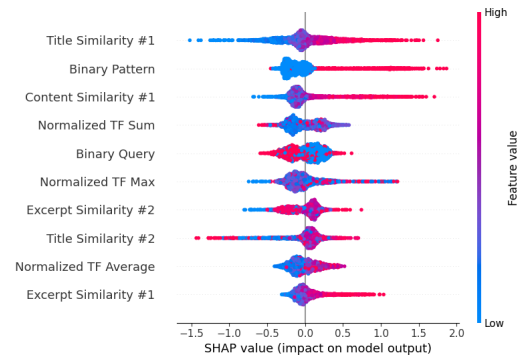


Figure 5.8.: Local Explanation Summary (baseline)

5.4. Privacy Design Patterns: Results and Analysis

Table 5.4.: Recommendation based on LINDDUN [40] Use Cases. The number after privacy requirements indicates the 'id' of the use case in the source.

#	Privacy Requirements	Privacy Design Pattern Recommendations
L	Limit the precision of stored location data to ensure individual user activities cannot be uniquely identified. (1.5)	(1) Attribute Based Credentials, (2) Pseudonymous Identity, (3) Location Granularity, (4) Decoupling [content] and location information visibility, (5) Onion Routing
I	Implement measures to prevent the combination of different properties from a user's browser to form a unique identifier. (2.7)	(1) Pseudonymous Identity, (2) Attribute-Based Credentials, (3) Obtaining Explicit Consent, (4) Onion Routing, (5) Protection against Tracking
N	Provide users with the option to communicate without revealing their corporate email address to avoid confirming their organizational affiliation. (3.5)	(1) [Support] Selective Disclosure, (2) Pseudonymous Messaging, (3) Pseudonymous Identity, (4) Attribute-Based Credentials, (5) Added-noise measurement obfuscation
D	Ensure that the operational characteristics of health devices do not inadvertently reveal the medical condition or patient status of the user. (4.2)	(1) Attribute-Based Credentials, (2) Active broadcast of presence, (3) Added-noise measurement obfuscation, (4) Awareness Feed, (5) Personal Data Store
DD	Strip any metadata that may contain personal data from documents upon processing. (5.5)	(1) Strip Invisible Metadata, (2) Added-noise measurement obfuscation, (3) Lawful Consent, (4) Obtaining Explicit Consent, (5) Attribute-Based Credentials
U	Generate notices for data subjects using clear and plain language, understandable to a general audience. (6.2)	(1) Active broadcast of presence, (2) Privacy dashboard, (3) Attribute-Based Credentials, (4) Abridged Terms and Conditions, (5) Dynamic Privacy Policy Display
Nc	Ensure System shall ensure that no viewing data is collected until lawful grounds for such collection are established. (7.5)	(1) Lawful Consent, (2) Attribute-Based Credentials, (3) Awareness Feed, (4) Selective access control, (5) Informed Implicit Consent

item. In contrast, a feature such as 'Binary Query' with mostly negative SHAP values suggests that its occurrence is often perceived by the model as a sign of lower relevance, thereby placing the item lower in the ranking.

5.4.4. Pilot Example

To evaluate our recommendation system, we applied our model to various test scenarios provided by LINDDUN [40] threat use cases⁵. For each type of threat, we selected a representative example and transformed the descriptions into specific privacy requirements. The outcomes of this process are presented in Table 5.4.

In this pilot example, we manually evaluated the ranking to determine the ideal order, aiming to calculate the NDCG for values of k ranging from 1 to 5. The results indicated NDCG scores of 90% at $k=1$, 88% at $k=2$, 81% at $k=3$, 87% at $k=4$, and 94% at $k=5$. Our thorough analysis revealed that the predictions for Data Disclosure (DD) were most accurately ordered, whereas the predictions for Linkability (L) were least accurate. To improve Linkability, we suggest prioritizing "Location Granularity" patterns in the first order as it involves location data directly. The presence of "Attribute-Based Credentials" in all predictions suggests a potential bias, particularly as it should not be included in predictions related to Unawareness (U) and Non-Compliance (Nc). Nevertheless, the recommendations consistently demonstrate desired PrDPs for each PR, indicating

⁵<https://linddun.org/cases/>

5. Contributions to Privacy Design Engineering

promising quality despite the challenges. Future strategies could improve performance by incorporating user feedback into the training process. This would require ongoing refinement of the model to more closely match its recommendations with user expectations.

5.5. Threats to Validity

Our study is susceptible to different types of validity issues, including construct, internal, external, and conclusion validity [170]. However, we have addressed and accounted for these concerns during the design and analysis stage. To ensure construct validity, we utilized well-established metrics for the LeToR model. We assessed our model using NDCG@k with varying '@k' values. Moreover, we incorporated PRs and PrDPs based on peer-reviewed research in our model. Internal validity, crucial for attributing outcomes to our methods and not to extraneous factors, presented a challenge due to the subjective nature of scoring privacy requirements' relevance to PrDPs. To overcome this, we based initial relevancy on Al-Momani's work [4] and refined it through careful human annotation. External validity pertains to the generalizability of our findings. To address potential sampling bias, we utilized data from three distinct methods of privacy requirements engineering. This approach facilitated the creation of a representative sample and minimized bias. We recognize the challenge of capturing the full spectrum of context-specific factors of design patterns—such as architectural considerations and the nuances of different application types—in our ground truth creation. We see it as an opportunity for refinement. Additionally, to ensure the reliability and soundness of our findings, we conducted multiple experiments and evaluated our model's strengths and limitations. Our datasets, annotations, and codes are publicly available for review and extension by the scientific community. We have taken steps to reduce the risks of internal and external validity, which bolsters the validity of our conclusions.

5.6. Chapter Summary

This chapter examines methods to streamline the selection process for the most appropriate privacy design patterns (PrDPs) to meet privacy requirements. To facilitate this process, we have developed a recommendation system that employs Learning-to-Rank (LeToR) in conjunction with semantic text characteristics to automatically match privacy requirements (PRs) with suitable PrDPs. Both the classification and LeToR models demonstrate satisfactory performance, confirming the potential of our model as a valuable tool in privacy engineering. The model's reliance on text features is particularly beneficial when other data types are not readily available. We also analyze the contribution of each feature to enhance the model's explainability in the recommendation process.

Our study currently uses a set of 70 PrDPs as a foundational baseline, which can be easily expanded. As the collection of PrDPs grows, our system is designed to seamlessly integrate additional patterns. This scalability ensures that our solution remains relevant and effective, addressing both current and future privacy design challenges.

Part III.

Implementation of the PrivacyStory Tool and Evaluation of the Approach

6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories

In this chapter, we introduce the PrivacyStory tool¹, a web-based application designed to integrate the intelligent systems developed in Chapters 4 and 5. This tool is essential for the empirical validation of our approach, as it allows users to directly interact with our proposed methods, providing feedback on their usefulness and potential adoption into the privacy-by-design workflow.

The PrivacyStory tool is built using the Flask framework, a web development framework written in Python. Utilizing Python enables us to incorporate open-source tools from previous studies, such as AQUASA [97] and Robustness Diagram Generation [55]. Although it heavily relies on machine learning, the tool allows users to intervene in several ways, such as editing data flow diagrams, modifying detected entities, and adjusting privacy requirement patterns for greater flexibility. Additionally, users can interact with predictions by performing group analysis and filtering user stories based on privacy-related entities, such as data subjects or personal data entities. The user interface of the tools can be seen in Figure 6.1. The video demonstration of the tools can be found at the following link². The tool is called PrivacyStory [73].

To elaborate in detail, the tool’s architecture will be explained in Section 6.1. The tool’s workflow will then be elaborated in Section 6.2 to illustrate how the intelligent system interacts with users to conduct privacy requirements and design engineering.

6.1. Architecture of the Tool

At the core of the system is the Web App based on python, which acts as the central controller, facilitating user interactions and coordinating the machine learning model and knowledge bases. For this prototype, the system relies on an SQLite database to store and manage its data, interfacing directly with the Web App. The component diagram of the tool can be seen in Figure 6.2.

The tool consists of several controllers, each responsible for distinct intelligent system functionalities. Many of these controllers are supported by machine learning models, including the AQUASA ML Model, Disclosure ML Model, Named Entity Recognition (NER) ML Model, Data Flow Diagram (DFD) ML Model, and a Learning to Rank Model,

¹<https://zenodo.org/doi/10.5281/zenodo.7598314>

²<https://bit.ly/privacystoryvideo>

6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories

User Stories : CamperPlus

Buttons: Add Story, Import Story From File, See Privacy Backlog, Check Quality, Recognize Entity & Disclosure, Generate Individual DFD, Generate Grouped DFD, See Grouped DFD, Filter

No	Story	Quality	Disclosure Potential	Action
1	As a camp administrator (SUBJECT), I want to be able to add (PROCESSING) campers (SUBJECT) so that I can keep track (PROCESSING) of each individual camper (SUBJECT).	Good	No	[Refresh] [Eye] [Trash]
2	As a camp administrator (SUBJECT), I want to be able to remove (PROCESSING) campers (SUBJECT) if they don't attend the camp anymore, so that I can keep the records organized.	Good	No	[Refresh] [Eye] [Trash]

DFD Triples

External Entity	Process	Data Store	
Parent	Track Child's Activity At Camp	Activity	[Trash]
Parent	Track Schedule	Schedule	[Trash]
Parent	Track Child's Activity At Camp	Camp	[Trash]
Parent	Track Child's Activity At Camp	Child	[Trash]
Parent	Track Child's Activity At Camp	Child	[Trash]

Regenerate Privacy Requirement

Privacy Requirements

As a **parent** (SUBJECT), I want to be able to **track** (PROCESSING) my **child's activity** (PII) and **schedule** (PII) at camp, so that I can have peace of mind.

Requirement For: **Parent** (SUBJECT) **Track Child's Activity At Camp** (PROCESSING) **Activity** (PII)

unlinkability

As a **Parent** (SUBJECT), I want the **Activity** (PII) data that used in **Track Child's Activity At Camp** (PROCESSING) to be protected from being linked directly or indirectly to other personal data within or outside of our system, so that an attacker cannot link it to the identity of **Parent** (SUBJECT).

Add to Privacy Backlog

Figure 6.1.: The three screenshots of the PrivacyStory tool. (1) The first screenshot on the top shows a project dashboard with several action buttons and the User Story table. (2) The second screenshot on the bottom left shows an example of an automatically generated DFD along with the DFD triples. (3) The last screenshot on the bottom right shows an example of the unlinkability requirements generated from the first triple.

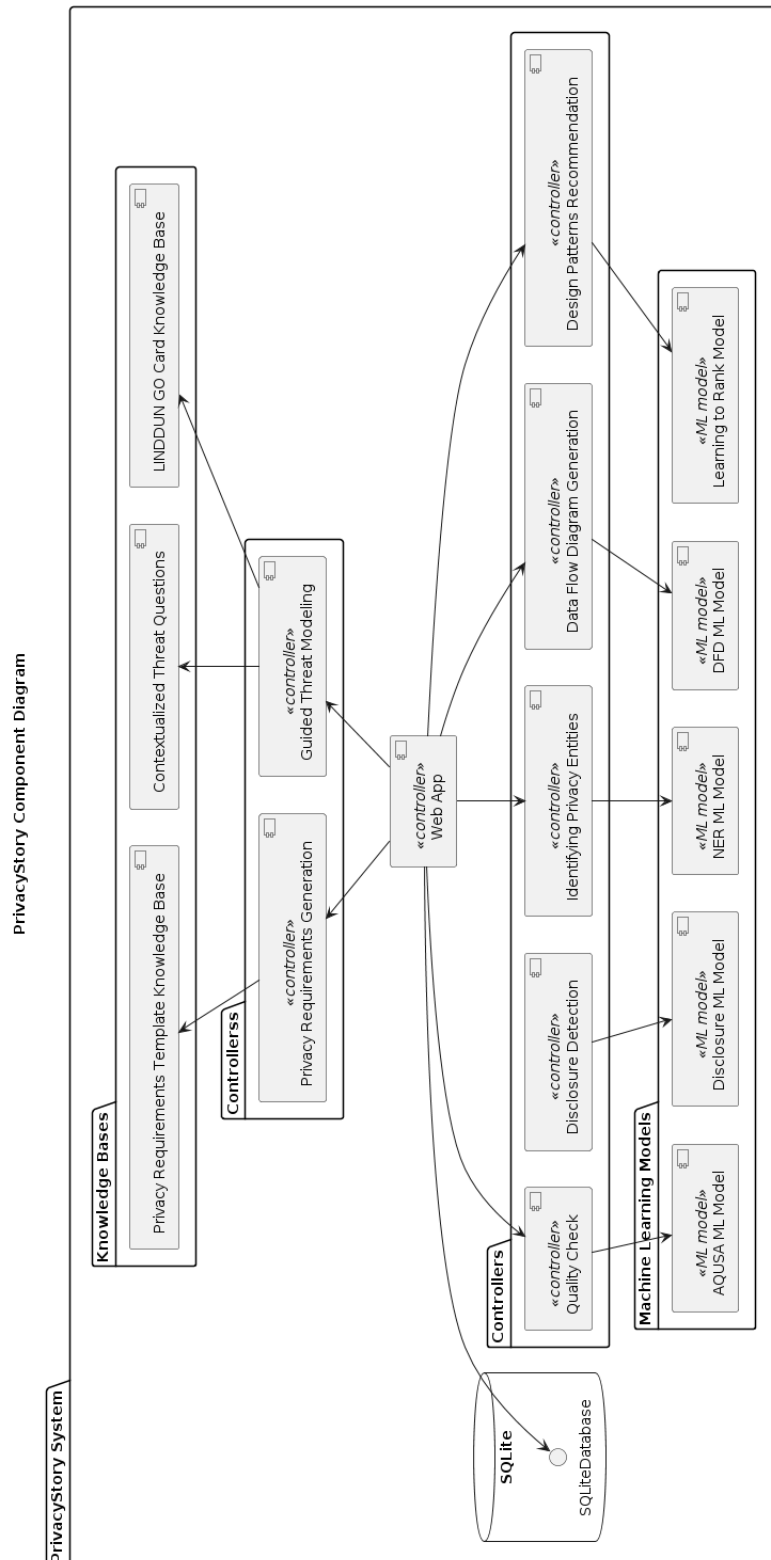


Figure 6.2.: Component Diagram of Privacy Story

6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories

each tailored to specific tasks. The system also utilizes various knowledge bases, such as the LINDDUN GO Card Knowledge Base [171], which is available online³, Contextualized Threat Question which inspired by LINDDUN, and the Privacy Requirements Template Knowledge Base presented in Chapter 4.

These functionalities are encapsulated in modules elaborated in the following sections.

6.1.1. Quality Check of User Stories

Ensuring that user stories adhere to syntactic guidelines is crucial for the accuracy of subsequent processes. PrivacyStory integrates AQUASA models [97] to provide users with a quality check. This allows users to identify problematic user stories and the specific quality aspects needing improvement. Users can then edit the stories to comply with QUS syntactic rules.

6.1.2. Disclosure Detection

Not all user stories are subject to privacy requirements. To identify those that might have privacy issues, we use a text classification model trained as described in Chapter 4. This model assesses the probability of a user story potentially disclosing sensitive information. The interface will show the classification result, enabling users to decide whether to proceed with stories that may be subject to disclosure concerns.

6.1.3. Detecting Privacy-Related Entities

Privacy-sensitive user stories often involve critical entities such as personal data, data subjects, and processing activities. PrivacyStory employs the Privacy Named Entity Recognition (NER) model to identify these entities. The NER model, trained as a supervised model as detailed in Herwanto [70] and in Chapter 4, effectively recognizes these important entities. Once the entities are detected, users can explore the user stories using filters, allowing them to select stories involving specific data subjects or those processing particular categories of personal data. This capability helps users understand the privacy-related entities in their user story backlog before proceeding to generate privacy requirements.

6.1.4. Generating Data Flow Diagram

PrivacyStory allows users to automatically generate a Data Flow Diagram (DFD) for individual user stories or a collection of user stories. This process uses the NLP model described in Herwanto [71] and in Chapter 4. The DFD can extract the relationship between data subjects, processing activities, and personal data. This relationship serves as the foundation for generating privacy requirements. Users can edit the diagram in draw.io by downloading the XML file and opening it in draw.io. Additionally, the extracted triples can be edited for increased flexibility.

³<https://downloads.linddun.org/linddun-go/default/latest/go.pdf>

6.1.5. Generating Privacy Requirements

Once the triples are extracted, they serve as inputs for generating privacy requirements. We use the requirements template described in Chapter 4, which includes (a) Unlinkability, (b) Undetectability, (c) Anonymity and Pseudonymity, (d) Transparency, (e) Intervenability, (f) Confidentiality, (g) Plausible Deniability, and (h) Content Awareness. The privacy requirements are modeled as user stories. Figure 6.1 demonstrates the application of the unlinkability pattern to one of the user stories. Users can select these patterns, count them as requirements, and add them to the privacy backlog.

6.1.6. Guided Threat Modeling

This module is largely inspired by LINDDUN [40,171]. However, we contextualize it using triples. The triples generated by the DFD form the foundation for contextualized, guided threat modeling. Figure 6.3 illustrates the detected triples from a user story. When these triples are selected, threat questions are displayed. Answering "yes" to these questions indicates potential threats, which are then added to the privacy backlog.

The interface displays a user story: "As a **Parent** (subject), I want to be able to track my **Child** (object)'s activity and schedule at camp, so that I can have peace of mind."

Detected Triple:

Parent (subject)	Track Child's Activity At Camp (process)	Child (object)
-------------------------	---	-----------------------

Contextualized Threat Questions:

Linkability Identifiability Non Repudiation Detectability Unawareness Non Compliance

NO TRANSPARENCY

Awareness: When the **Parent** (subject) interacts through the **Track Child's Activity At Camp** (process) process and stores data in **Child** (object) data, are they sufficiently informed about what personal data is being processed, for what purposes, and in what manner? Yes No

NO INFORMATION WHEN SHARING DATA OF OTHERS

NO PRIVACY CONTROLS

Preference Controls: In the system, when **Parent** (subject) interacts via the **Track Child's Activity At Camp** (process) process to **Child** (object) data, are they provided controls to manage what personal data is processed and for what purposes? Yes No

Preference Alteration: Once the **Parent** (subject) sets their preferences through the **Track Child's Activity At Camp** (process) process in relation to **Child** (object) data, can they alter them afterwards? Yes No

NO ACCESS

NO RECTIFICATION OR ERASURE

Figure 6.3.: Guided Threat Modeling

6.1.7. Privacy Design Patterns Recommendation

The privacy requirements in the backlog can be further analyzed to recommend suitable privacy design patterns. These recommendations are derived from the model described in

6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories

Chapter 5. The model evaluates the text of the privacy requirements and suggests the top 10 privacy design pattern solutions, ranked for users to select from. Users can also explore each recommendation in more detail by visiting the original source⁴.

The image shows a 'Privacy Backlog' table with two entries. The first entry, 'unlinkability', describes a requirement for data protection. The second entry, 'anonymity', describes a requirement for data anonymization. A blue arrow points from the 'anonymity' requirement to a 'Solution Recommendation' window. This window displays a list of solutions: 'Protection against Tracking', 'Anonymous Reputation-based Blacklisting', 'K-anonymity', and 'Data Masking'. A blue arrow points from 'Data Masking' to a webpage titled 'Anonymity Set' on 'privacypatterns.org'. The webpage provides a detailed summary, context, problem, and solution for the 'Anonymity Set' pattern.

No	Type	Requirement
1	unlinkability	As a Parent subject , I want the Activity data that used in Track Child's Activity At Camp processed to be protected from being linked directly or indirectly to other personal data within or outside of our system, so that an attacker cannot link it to the identity of subject in Activity data .
2	anonymity	As a Parent subject , I want that the Activity data to be anonymized when performing Track Child's Activity At Camp processed data, so that unwanted actors cannot directly or indirectly identify subject in Activity data .

Solution Recommendation

Anonymity Set

Summary

This pattern aggregates multiple entities into a set, such that they cannot be distinguished anymore.

Context

This pattern is applicable in a messaging scenario, where an attacker can track routing information. Another possible scenario would be the storage of personal information in a database.

Problem

In a system with different users we have the problem that we can often distinguish between them. This enables location tracking, analyzing the behaviour of the users or other privacy-infringing practices.

[Read More...](#)

- Protection against Tracking
- Anonymous Reputation-based Blacklisting
- K-anonymity
- Data Masking

privacypatterns.org

CATEGORIES: ☐ ANONYMITY • ☐ MIX-NETWORKS • ☐ OBFUSCATION • ☐ HIDE • ☐ MIX

Anonymity Set

Summary

This pattern aggregates multiple entities into a set, such that they cannot be distinguished anymore.

Context

This pattern is applicable in a messaging scenario, where an attacker can track routing information. Another possible scenario would be the storage of personal information in a database.

Problem

In a system with different users we have the problem that we can often distinguish between them. This enables location tracking, analyzing the behaviour of the users or other privacy-infringing practices.

Solution

There are multiple ways to apply this pattern. One possibility is, to strip away any distinguishing features from the entities. If we do not have enough entities, such that the anonymity set would be too small, then we could even insert fake identities.

The goal of this pattern is to aggregate different entities into a set, such that distinguishing between them becomes infeasible.

Figure 6.4.: Solution Recommendation

6.2. Workflow of the Tool

The PrivacyStory tool is designed to process user stories through several intelligent system functionalities, supported by various machine learning models and knowledge bases. The process begins when a user submits a user story to the User Story Input module. This story is then passed to the Quality Check module, where the AQUASA model performs inference to ensure the story adheres to syntactic guidelines. If the story meets the quality standards, it moves to the Disclosure Detection module. Here, a disclosure model assesses the likelihood of the story containing sensitive information. Detected disclosures are forwarded to the Identifying Privacy Entities module.

⁴<https://privacypatterns.org/>

In the Identifying Privacy Entities module, a Named Entity Recognition model identifies critical entities such as personal data, data subjects, and processing activities within the user story. These identified entities are then passed to the Data Flow Diagram Generation module. The Data Flow Diagram Generation module uses a specialized model to infer relationships between the identified entities, producing triples that represent these relationships. These triples are fundamental for generating a Data Flow Diagram, which visually depicts how data flows within the system, helping users understand the privacy implications of their user stories. This first part of the processes is depicted in the sequence diagram in Figure 6.5.

After presenting the triples to the user, they have the option to generate privacy requirements or engage in guided threat modeling. If the user opts for Privacy Requirements Generation, the system uses a template to generate privacy requirements, which the user can review and add to a privacy backlog. Alternatively, the user can choose Guided Threat Modeling, where the system poses threat-related questions based on the generated triples. Positive responses to these questions result in the identification of related requirements, which are then added to the backlog.

The final stage involves the Privacy Backlog interacting with the Privacy Design Patterns Recommendation module. The Privacy Design Patterns Recommendation module evaluates the privacy requirements in the backlog and generates recommendations for suitable privacy design patterns. These recommendations are ranked and presented to the user, who can then explore each suggestion in more detail. This comprehensive process ensures that user stories are systematically analyzed, privacy issues are detected, and appropriate privacy measures are recommended. The second part of these processes is depicted in the sequence diagram in Figure 6.6.

6.3. Chapter Summary

This chapter introduces a tool called PrivacyStory to assist agile development teams in conducting privacy requirements engineering. The core module of this tool heavily relies on automation provided by state-of-the-art NLP models. Despite the automation, our evaluation of the NLP model indicates that human involvement is still necessary to validate the final privacy recommendations.

The first version of the tool was presented at the requirements engineering conference, where we showcased a video demo, accessible via this link⁵. Based on ongoing developments, the new version of the tool includes additional features such as guided threat modeling and solution recommendations. A video demonstration of this updated version is available at this link⁶.

In the next chapter, Chapter 7, we will demonstrate the implementation of the latest version of this tool. In Chapter 8, we will conduct an empirical evaluation focused on the usefulness of PrivacyStory.

⁵<https://bit.ly/3K24oLm>

⁶<https://bit.ly/3K2TGnP>

6. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories

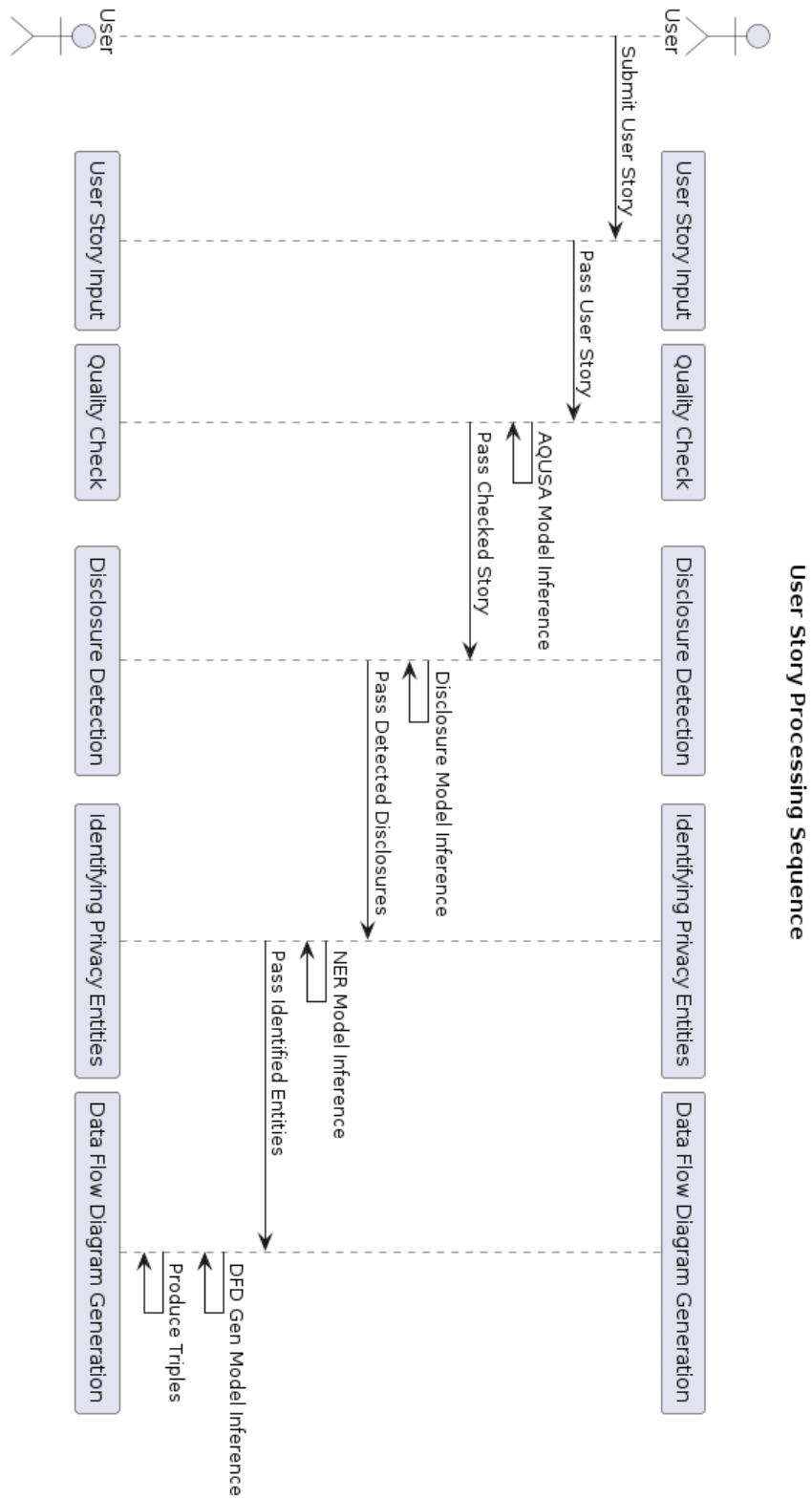


Figure 6.5.: Sequence Diagram of Privacy Story from User Story to Triples

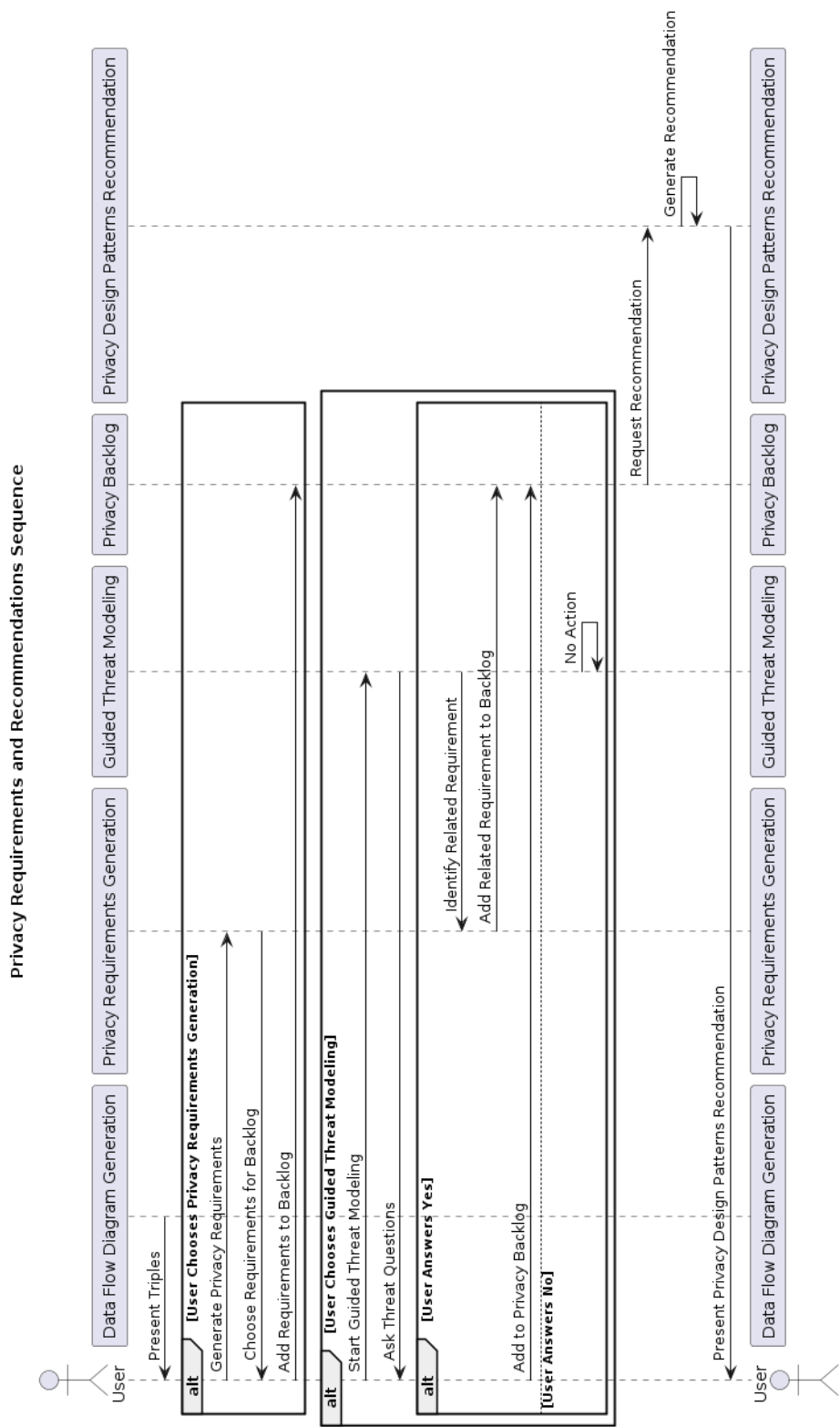


Figure 6.6.: Sequence Diagram of Privacy Story from Triples to Privacy Design Patterns

7. Application of the Intelligent System

This chapter demonstrates the implementation of our model from the collection of user stories [39].

7.1. The CamperPlus Project

We used the open-source project CamperPlus (*Camper+*)¹ as the pilot example for our approach. CamperPlus is a web-based application designed for camp administrators and parents responsible for managing camps and supervising their children. We chose CamperPlus because, based on prior research findings, it was deemed a privacy-sensitive project [70]. The user stories are accessible to the public via their repository page. The project includes 54 user stories that feature four actors:

- **Camp administrators** are responsible for the enrollment of campers, scheduling activities for the various camp groups, uploading consent forms, and many more.
- **Parents** can use the platform to register their children for camp, monitor the activities of their children while at camp, make a payment, and communicate with camp administrators and counselors.
- **Camp workers** are responsible for managing field work of camps such as reporting a necessary repairs, reporting camp supplies, and being in charge of camp and campers.
- **Camp counselors** are responsible for taking attendance of the children.

The camp administrator was mentioned most often with 35 stories, followed by the parents with 13 stories, the camp worker with 6 stories, and the camp counselor with one story. In addition to the actors mentioned above, there are a number of data subjects that are not mentioned in the actor part of the user story, but in the main functionality of the user story. For example, children as the campers is frequently mentioned but since they do not have the main functionality in the system, it is not mentioned in the actor part of a user story. Nevertheless, they are also important data subject to be considered in the privacy requirement. In GDPR Article 8.1 it stated that *"Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child"*. Thus the user story will have an additional detail when child is involved in the functional requirement. Therefore, it is necessary to extract these data subjects and the actions they perform.

¹<https://github.com/Notabela/Camper-Plus>

7.2. Automated Quality Check of User Stories

The AQUUSA was utilized to conduct a quality analysis of the 54 individual user stories belonging to CamperPlus. The results of this analysis indicated that the AQUUSA was able to identify a total of 7 syntactic and 5 pragmatic criteria problems within the user stories.

Among the syntactic criteria, one user story was deemed unwell-formed, four user stories were non-atomic, and one user story was not minimal. Non-atomic user stories are typically identified by the presence of conjunctions and may be broken down into two or more individual user stories.

The pragmatic criteria identified by the AQUUSA included 1 instance of duplication and 4 user stories that were deemed non-uniform. It should be noted that the lack of uniformity was attributed to several user stories lacking the mean part, potentially resulting in confusion for the NLP model.

Additionally, there was a repetition of certain indicators identified, which could also lead to potential confusion for the NLP model. Table 7.1 provides an illustrative example of the user story quality check. These user stories can then be edited to meet the quality criteria, making the analysis more accurate as the process continues.

Table 7.1.: Example of User Story Quality Check

ID	User Story	Defect kind	User Story Modification
12	As a camp administrator, I want to be able to create, modify rules that campers and camp workers has to follow.	Atomic (Conjunctions)	(1) As a camp administrator, I want to be able to create rules that campers and camp workers has to follow. (2) As a camp administrator, I want to be able to modify rules that campers and camp workers has to follow.
14	As a camp administrator, I want to create an avenue so parents can submit feedback and general concerns, so that I can keep improving the services I provide.	Minimal (Indicator Repetition)	As a camp administrator, I want to create an avenue for parents can submit feedback and general concerns, so that I can keep improving the services I provide.
50	As a camp worker, I would be able to submit a completion report for the tasks which was assigned to me.	Well formed (No Means) & Uniform (As a, I want to, So that)	As a camp worker, I want to be able to submit a completion report for the tasks which was assigned to me, so that I can keep update my progress.

7.3. Detecting Privacy Disclosures

We apply the model to determine the disclosure probability of 54 individual user stories from Camperplus. We applied the NER-based model based on Section 4.4.1, and we found that all of the user stories have a disclosure potential. We decided to rely on the NER-based model since the time to vet out the false positive would not be a burden [19], especially with the tool that we will introduce in Chapter 6.

One of the user stories that have disclosure potential is shown in Figure 7.1. An example of a user story that has no disclosure potential is, *"As a camp administrator, I want to be able to create, modify rules that campers and camp workers has to follow."*. Each of the user stories that have disclosure potential will undergo the next process.

7.4. Identification of Privacy Entities

The user stories that have been identified to have a disclosure potential will be going through NER prediction. The example of a user story to which the NER was applied is shown in Figure 7.1. The complete entities that are detected from the NER model can be seen in Table 7.2. Due to the nature of user stories, several identified data subjects have the same meaning such as child, kids, and children. This also happens in personal data entities such as parent information, parents, and guardian’s information. The advantage of NER is, it can identify the roles not only in the *role* part of user story, but also in *mean* and *end*. Examples of these data subjects are manager, child, and staff member, which appear in the *means* of user stories rather than the *role*.

Table 7.2.: Entities extracted from Camperplus Project

Data Subject	Processing	Personal Data
camp administrator, campers, individual camper, parents, camper, staff members, camp worker, camp workers, parent, child, staff, kids, enrolled campers, camp management, counselors, children, manager, camp counselor	add, keep track, remove, keep, upload, schedule, automatically create, suspend, set a, assign, organize, warn, create, modify, submit, store, call, provide, make, see, notify, log, track, delete, read, sign up, assigned, message, voice, check, sign, deal, share, connect, enroll, edit, know, report, take	camper records, consent forms, camper submitted, forms, activities, who is where, nametags, who had behavioral problems, reminders, tasks, positions, responsibilities, rules, usage of internal camp facilities, feedback, general concerns, emergency information, immediate, guardian’s information, registration form, information, medical forms, attendance, groups, camp groups, events scheduled, camp group, events, parents information, parents, groups information, child’s activity, schedule, account, kids, concerns, payments, photos, children, enrolled children, completion report, list of supplies, necessary repairs, assigned kids

Here are two user stories involving data subjects who were identified as a child: (1)

7. Application of the Intelligent System

"As a camp administrator , I want to provide bi-weekly feedback to camper's parents, so that they can be aware of their child's behavior and performance at camp.". (2) "As a parent, I want to be able to track my child's activity and schedule at camp, so that I can have peace of mind". As can be seen, the primary *role* in the first user story is the camp administrator, and the second user story is the parent. However, from the privacy perspective, the child is the data subject that the personal data is compromised. Our NER model is able to capture these entities. The example of NER detection from the first user story can be seen in Figure 7.1.

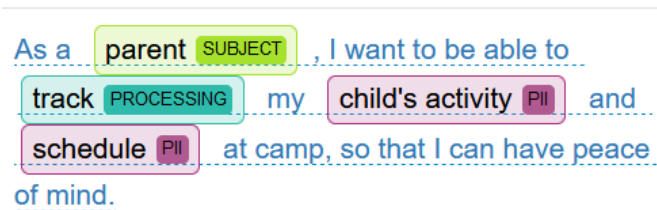


Figure 7.1.: The Example of Named Entity Recognition on User Stories.

7.5. Data Flow Diagram Generation

The generation of DFD can be conducted in two ways. The first way is to generate DFD for each of the individual user stories. An example of the generation of DFD for a single user story can be seen in Figure 7.2.

The second way is to combine user stories to become one DFD. The combination can be based on particular filters such as the same data subject, the same personal data, or other filters that the analysts might be interested in. For example, consider user stories that have "child" as data subject below:

- As a camp administrator, I want to provide bi-weekly feedback to camper's parents, so that they can be aware of their child's behavior and performance at camp.
- As a parent, I want to be able to message my child's counselors, so that I can voice my concerns or check on my child's progress.
- As a parent, I want to be able to share any photos the camp has taken of my child.

Our automatic generation can combine the three user stories into one DFD which can be seen in Figure 8.3. Based on the DFD, we can identify eight triples. However, we only consider the six triples that process personal data, which, based on the red highlight in the data store element. The processing that leads to the "Counselor" and "Camp" data is not counted as a triple that is considered when creating the privacy requirements. These six triples are further processed when creating the privacy requirement.

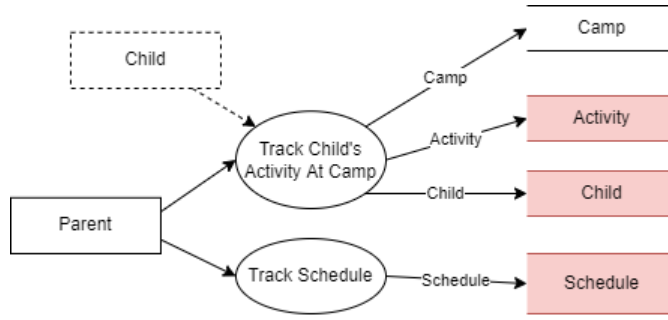


Figure 7.2.: The Data Flow Diagram for user story: "As a parent, I want to be able to track my child's activity and schedule at camp, so that I can have peace of mind."

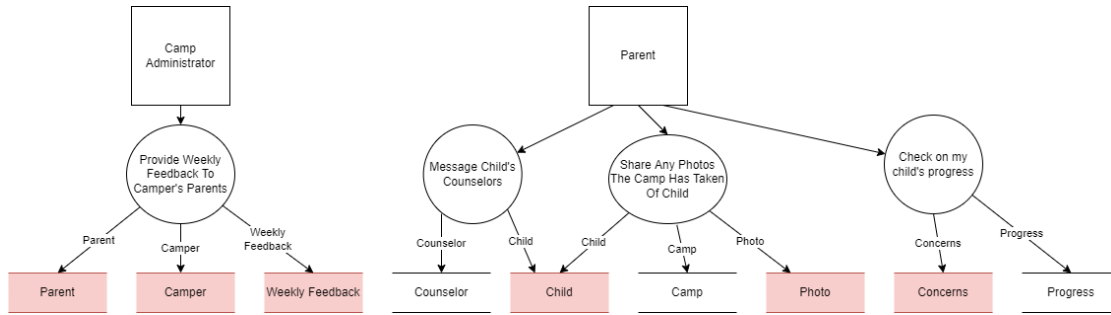


Figure 7.3.: The Data Flow Diagram that generated from three user stories that has "child" as data subject

7.6. Privacy Requirements Generation

Based on the model that been explained in Section 4.3.5 and DFD in Figure 8.3, it is known that the DFD triples are:

- $t_1 = (s_1, p_1, d_1)$
- $t_2 = (s_1, p_1, d_2)$
- $t_3 = (s_1, p_1, d_3)$
- $t_4 = (s_2, p_2, d_4)$
- $t_5 = (s_2, p_2, d_5)$
- $t_6 = (s_2, p_3, d_6)$

where $s_1 = \text{"Camp Administrator"}$, $s_2 = \text{"Parent"}$; $p_1 = \text{"Provide Weekly Feedback to Camper's Parents"}$, $p_2 = \text{"Share Any Photos the Camp has Taken of Child"}$, $p_3 = \text{"Check on my child's progress"}$; $d_1 = \text{"Parent"}$, $d_2 = \text{"Camper"}$, $d_3 = \text{"Weekly Feedback"}$, $d_4 = \text{"Child"}$, $d_5 = \text{"Photo"}$ and $d_6 = \text{"Concerns"}$.

Our approach will generate all requirements that are stated in Section 4.3.5 The example of the generated privacy requirement for t_4 can be seen in Table 7.3. It is based on a user story: "I want to be able to message my child's counselors, so that I can voice my concerns or check on my child's progress."

7. Application of the Intelligent System

Table 7.3.: Privacy Requirement Generation Result

Requirement	User Story
Confidentiality	As a parent , I want the child data that processed in message child's counselor to be kept confidential, so that unwanted actors are unable to access it.
Integrity and Availability	As a parent , I want the child data that processed in message child's counselor to be prevented from faults or unwanted actors, so that the consistency, correctness, and availability of the data is not compromised.
Unlinkability	As a parent , I want the child data that used in message child's counselor to be protected from being linked directly or indirectly to other personal data within or outside of our system, so that an attacker cannot link it to the identity of subject in child data.
Anonymity or Pseudonymity	As a parent , I want that the child data to be anonymized (or pseudonymized) when performing message child's counselor , so that unwanted actors cannot directly or indirectly identify subject in child data.
Undetectability	As a parent , I want unwanted actors to be unable to sufficiently distinguish whether or not child data is present, so that I can safely perform message child's counselor .
Transparency	(1) As a parent , I want to be informed and consented that the child data is used in message child's counselor , so that I can exercise my rights when it is used outside of this context. (2) As a parent , I want to download a copy of child data that used in message child's counselor data at camp, so that I can check their correctness.
Intervenability	(1) As a parent , I want to be able to modify the child data that have been processed at message child's counselor without undue delay, so that I can prevent the inaccuracy of data. (2) As a parent , I want to be able to delete the child data that have been processed at message child's counselor without undue delay, so that I can exercise my right. (3) As a parent , I want to withdraw my consent on the processing of message child's counselor on the child data, so that I can exercise my right.
Plausible Deniability	As a parent , I want to have the ability to deny performing message child's counselor on child data, so that unwanted actors unable to accuse me of doing such a thing.
Content Awareness	As a parent , I want to be informed that I should not share the child data outside of the platform so that my privacy or data subject in child data is not compromised.

7.7. Prioritization of Privacy Requirements

The privacy requirement listed in Table 7.3 was created for a messaging action between a parent and a child's counselor to check the child's progress. Due to the sensitive nature of the child's data, strong privacy measures are needed. Firstly, the RBAC system should only allow the parent and counselor to access the message data to ensure confidentiality. The messaging process must also be accurate and timely, requiring specifications for integrity and availability. Maintaining the unlinkability of the child's data within the message is essential, meaning it should not be linked to any other personal data, such as weekly feedback or a photo of the child. End-to-end encryption of the message could achieve this. If the unlinkability requirement is enforced, it could override other requirements. Anonymization or pseudonymization of the child's identity is unnecessary since the unlinkability requirement already targets the message containing the data. The unlinkability requirement also overrides undetectability, as it should be difficult to determine if personal data is present in the message. Transparency and intervenability are critical to ensure the parent understand the purpose and limitations of the processing, even with end-to-end encryption in place. The parent must also be able to change their consent or the message itself. This helps the team comply with GDPR requirements, and plausible deniability would not be possible with these measures in place. Finally, content awareness could be added to prevent the sharing of the message by the parent as a preventive measure.

All of these prioritized requirements will be systematically added to the privacy backlog to ensure that each is appropriately addressed in the development cycle. This methodical approach facilitates a comprehensive and agile response to privacy concerns.

7.8. Recommendation of Privacy Design Patterns

To prioritize the specific requirements for this user story, we consider Confidentiality, Pseudonymity, Transparency, Intervenability, Plausible Deniability, and Content Awareness. Based on the Privacy Requirements from Table 7.3, we can generate solution recommendations, which are summarized in 7.4.

For instance, ensuring confidentiality in messaging between parents and counselors can be achieved using privacy patterns like pseudonymous messaging, encryption with user-managed keys, and onion routing. These patterns are also effective in scenarios requiring anonymity or pseudonymity, as they prevent unwanted actors from identifying or accessing sensitive information. Additionally, transparency is improved by obtaining lawful consent and providing meaningful information and feedback, ensuring that parents are well-informed and can exercise their rights over their child's data. Patterns such as pseudonymous identity and onion routing are also recommended for plausible deniability, highlighting their broad applicability.

7. Application of the Intelligent System

Table 7.4.: User Stories and Corresponding Privacy Patterns

User Story	Privacy Patterns
<i>Confidentiality:</i> As a parent, I want the child data that is processed in message child's counselor to be kept confidential, so that unwanted actors are unable to access it.	(1) Attribute Based Credentials, (2) Pseudonymous Messaging, (3) Pseudonymous Identity, (4) Encryption with user-managed keys, (5) Onion Routing
<i>Anonymity or Pseudonymity:</i> As a parent, I want the child data to be anonymized (or pseudonymized) when performing message child's counselor, so that unwanted actors cannot directly or indirectly identify the subject in child data.	(1) Pseudonymous Messaging, (2) Attribute Based Credentials, (3) Anonymity Set, (4) Pseudonymous Identity, (5) Onion Routing
<i>Transparency:</i> As a parent, I want to be informed and consent that the child data is used in message child's counselor, so that I can exercise my rights when it is used outside of this context.	(1) Lawful Consent, (2) Abridged Terms and Conditions, (3) Obtaining Explicit Consent, (4) Impactful Information and Feedback, (5) Privacy Awareness Panel
<i>Intervenability:</i> As a parent, I want to withdraw my consent to the processing of message child's counselor on the child data, so that I can exercise my right.	(1) Lawful Consent, (2) Obtaining Explicit Consent, (3) Informed Implicit Consent, (4) Abridged Terms and Conditions, (5) Informed Consent for Web-based Transactions, (6) Sticky Policies
<i>Plausible Deniability:</i> As a parent, I want to have the ability to deny performing message child's counselor on child data, so that unwanted actors are unable to accuse me of doing such a thing.	(1) Pseudonymous Identity, (2) Onion Routing, (3) Attribute Based Credentials, (4) Anonymity Set, (5) Pseudonymous Messaging, (6) Added-noise measurement obfuscation
<i>Content Awareness:</i> As a parent, I want to be informed that I should not share the child data outside of the platform so that my privacy or the data subject in child data is not compromised.	(1) Privacy Awareness Panel, (2) Platform for Privacy Preferences, (3) Appropriate Privacy Feedback, (4) Impactful Information and Feedback, (5) Awareness Feed, (6) Active broadcast of presence, (7) Privacy dashboard

8. Empirical Evaluation

This chapter aims to evaluate the acceptance and willingness of software practitioners to adopt the PrivacyStory tool. The sections of this chapter are structured according to the empirical software engineering principles laid out by Wohlin [170].

8.1. Experiment Planning

This section outlines the plan for evaluating the effectiveness, efficiency, and user acceptance of the PrivacyStory tool in conducting privacy threat modeling. The experiment follows a mixed-methods approach, utilizing both quantitative and qualitative data collection techniques to gain a comprehensive understanding of the tool's performance and participants' experiences.

The study involves a controlled experiment with participants, consisting of both students and practitioners, who will perform privacy threat modeling tasks using the PrivacyStory tool and a manual approach. The experiment is designed to assess the relative advantages and disadvantages of each method while minimizing the impact of individual differences among participants.

The following subsections provide details on the goals, participants, experimental materials, tasks, and the experimental design used in this study.

8.1.1. Goals

The main goal of this evaluation is to assess the acceptance and willingness of developers to adopt our proposed method, which is encapsulated in the PrivacyStory tool. This goal can be formally expressed using the Goal/Question Metrics (GQM) paradigm [16]:

Goal: Analyze the PrivacyStory tool suite for the purpose of evaluation with respect to its acceptance and potential adoption from the point of view of software developers in the context of privacy requirements engineering.

To achieve this goal, we will focus on the following subgoals, each addressing specific aspects of the Technology Acceptance Model (TAM):

- Subgoal 1 (Perceived Usefulness): Assess the degree to which developers believe that using PrivacyStory will enhance their performance in conducting privacy threat modeling.
- Subgoal 2 (Perceived Ease of Use): Assess the degree to which developers expect PrivacyStory to be free of effort in learning, understanding, and using it.

8. Empirical Evaluation

- Subgoal 3 (Attitude Towards Using): Assess developers' overall affective reaction to using the PrivacyStory framework.
- Subgoal 4 (Behavioral Intention to Use): Assess the likelihood that developers will use the PrivacyStory tool in the future.

To operationalize these subgoals, we will refine them into specific questions and metrics, following the GQM paradigm [16]. These questions and metrics will be explained in the experimental design section. These questions and metrics will guide our data collection and analysis efforts, allowing us to draw meaningful conclusions about the acceptance and potential adoption of the PrivacyStory tool among software developers.

8.1.2. Participants

To address our research goals, which center on the acceptance and adoption potential of the PrivacyStory tool among software developers, we will involve two participant groups:

1. **Computer Science Students:** Students from an Indonesian university who have successfully completed a software engineering course will offer a perspective shaped by recent exposure to software development principles.
2. **Software Practitioners:** A diverse group of practitioners from Indonesia, representing various software development domains, will provide real-world insights into how PrivacyStory might integrate into existing workflows.

The selection process for both groups will prioritize participants with a fundamental understanding of software engineering concepts. While prior experience with privacy threat modeling is beneficial, we anticipate that most participants will have limited exposure to this area. This characteristic aligns with our goal of assessing the tool's usability and learnability for developers who may be new to privacy-focused activities.

8.1.3. Experimental Materials

The materials for this include:

- *Web Interface:* A user-friendly web interface that hosts all experimental materials and tools. This interface should allow participants to pause and resume their session to enhance participation flexibility.
- *Consent Forms:* Ensure ethical considerations are met with clear, understandable consent forms.
- *Demographic Questionnaire:* Collect background information on participants without compromising anonymity.
- *Video Courses on Privacy by Design:* Six videos in Bahasa Indonesia, each followed by a quiz to test understanding. These are essential for participants unfamiliar with privacy requirements engineering.

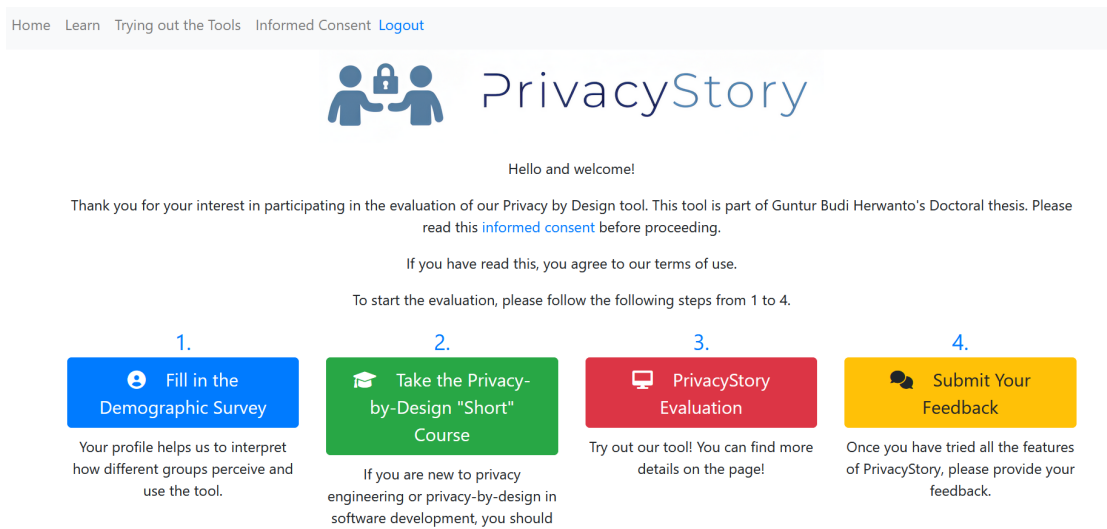


Figure 8.1.: Web interface that participants can view to understand the evaluation steps they need to go through.

- *PrivacyStory Tool Tutorial*: A video tutorial explaining how to use the PrivacyStory tool. The impact of this tutorial on the results should be monitored.
- *Software Project Description and User Stories*: Materials for participants to apply the PrivacyStory framework in a controlled setting.
- *Technology Acceptance Model (TAM) Questionnaire*: Adapted to specifically evaluate the PrivacyStory framework.

Privacy-by-Design Course

The "Privacy by Design" course is designed to equip participants with a comprehensive understanding of data privacy and its implementation in software development. The course comprises five modules, focusing on principles of "Privacy by Design" and the LINDDUN method for threat modeling.

Module 1: Introduction

- Discusses data privacy, defining it as the right to be left alone and control personal information and relationships.
- Highlights the secure, transparent, and lawful treatment of personal data.
- Emphasizes the responsibility of software developers in protecting user data, with Apple's approach as an example.

Module 2: Personal Data

8. Empirical Evaluation

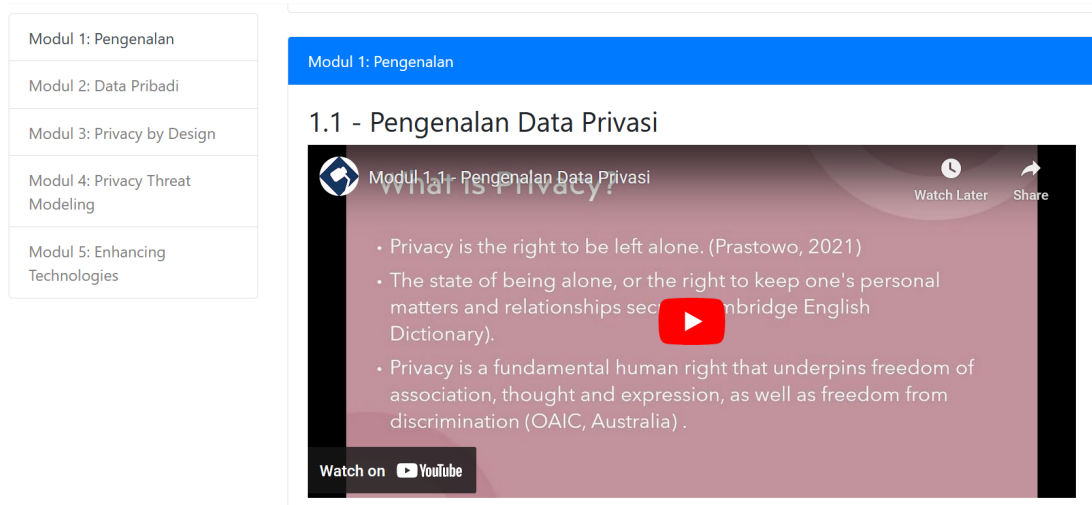


Figure 8.2.: The Online Course of Privacy by Design in Indonesian Language (Bahasa).

Modul 1 - Konsep Privasi Data

Apa definisi dari privasi data?

- ☐ Hak eksklusif untuk hidup tanpa gangguan.
- ☐ Hak untuk hidup tanpa gangguan, bebas dari intrusi, dengan kontrol atas informasi pribadi.
- ☐ Hak dasar manusia untuk kebebasan berbicara.
- ☐ Hak eksklusif untuk merahasiakan informasi pribadi.

Dalam konteks rekayasa perangkat lunak, privasi data berarti...

- ☐ Sebuah konsep teoretis tanpa aplikasi praktis.
- ☐ Kewajiban untuk melindungi dan menghormati data pribadi pengguna.
- ☐ Aspek yang terbatas pada fungsionalitas teknis sistem.
- ☐ Tugas yang hanya terkait dengan regulasi dan hukum.

Apa perbedaan kunci antara privasi dan keamanan data?

- ☐ Privasi berkaitan dengan penggunaan yang tepat dari data, sementara keamanan menyangkut perlindungannya.
- ☐ Keduanya adalah istilah yang saling bertukar dengan arti yang sama.
- ☐ Privasi hanya berlaku untuk data pribadi, sedangkan keamanan mencakup semua jenis data.
- ☐ Keamanan adalah masalah hukum, sementara privasi bukan.

Hak mana yang biasanya tidak termasuk dalam hak-hak yang diberikan oleh undang-undang?

Figure 8.3.: The example of a quiz that must be attempted to get to the next module.

- Covers the concept of personal data, including various categories like internal, historical, financial, external, social, and tracking data.
- Discusses the rights of data subjects under data privacy laws, like the right to information, access, rectification, erasure, restriction of processing, data portability, and objecting to processing.

- Explains data processing stages: collection, use, storage, sharing, and deletion.

Module 3: Privacy by Design

- Introduces the concept of integrating privacy considerations into every phase of software development.
- Details seven privacy principles, such as proactive privacy, default privacy settings, integrated privacy in design, full functionality, security throughout the software lifecycle, transparency, and respecting user privacy.

Module 4: Privacy Threat Modeling

- Describes threat modeling as a systematic method for identifying, assessing, and addressing privacy risks.
- Discusses the LINDDUN method, which categorizes threats into linkability, identifiability, non-repudiation, detectability, information disclosure, unawareness, and non-compliance.
- Outlines steps in threat modeling, including system modeling, identifying threats, and managing threats with privacy-enhancing technologies.

Module 5: Enhancing Technologies

- Focuses on Privacy Enhancing Technologies (PET), which embody data protection principles.
- PETs include technologies like Homomorphic Encryption, AI-Generated Synthetic Data, Secure Multiparty Computation, Federated Learning, and Differential Privacy.
- Discusses privacy patterns as documented solutions for recurring design issues, providing structured solutions like location granularity and various privacy design patterns.

The course includes supplementary materials such as reading material, case studies, and guides for protecting personal data and understanding the evolution of data privacy.

8.1.4. Tasks

The following sequence of tasks is designed to address our research goals regarding the acceptance and adoption potential of PrivacyStory for the privacy threat modeling and requirements engineering process:

Learning Phase (Addressing Baseline Knowledge)

To establish a common baseline understanding of privacy requirements engineering, participants will: (1) Watch curated video courses covering fundamental concepts that were explained in previous section. (2) Complete quizzes to reinforce learning and assess comprehension. This phase ensures that variations in prior experience with privacy engineering do not unduly influence the subsequent evaluation of the PrivacyStory tool.

8. Empirical Evaluation

Application Phase - Manual (Providing a Comparison Point)

Participants will gain hands-on experience with LINDDUN by: (1) Manually creating Data Flow Diagrams (DFDs). (2) Conducting privacy threat modeling based on the DFDs. This manual experience provides a direct comparison point for evaluating the effectiveness and usability of the PrivacyStory tool in the next phase.

Application Phase - PrivacyStory Tool (Evaluating Tool Effectiveness)

Participants will use the PrivacyStory tool to apply their knowledge and perform privacy threat modeling on a specific software project. This involves several detailed steps to guide the privacy threat modeling and requirements engineering process:

- *Identifying Privacy Entities*: Determine key individuals and systems involved in data handling within the project.
- *Creating Data Flow Diagrams*: Visualize data movements across the system to identify potential privacy risk points.
- *Identifying Threats*: Analyze the data flow diagrams to pinpoint privacy vulnerabilities.
- *Generating Privacy Requirements*: Based on identified threats, outline necessary measures to mitigate privacy risks.
- *Finding Solutions*: Propose practical solutions for the identified privacy requirements, aiming to address and mitigate risks effectively.

This phase allows us to gather data on the tool's ability to support the threat modeling process, its usability, and the perceived effort required.

Evaluation Phase (Assessing Perceptions and Attitudes)

Participants will complete a structured questionnaire based on the Technology Acceptance Model (TAM) metrics to assess their perception of the PrivacyStory tool's usefulness, ease of use, and attitudes toward its future adoption.

8.1.5. Experimental Design

We employ a sequential design where each participant performs privacy threat modeling manually using LINDDUN and then using the PrivacyStory tool. This sequence establishes a common baseline for privacy threat modeling understanding before participants interact with PrivacyStory, mitigating potential bias introduced by variations in prior experience.

We opted for this sequential design for several reasons. First, establishing a common baseline addresses potential bias. Participants will likely have varying levels of familiarity with privacy threat modeling concepts. The manual phase ensures everyone establishes a foundational understanding of the core concepts and the general process, regardless of

their prior experience. This mitigates the risk that pre-existing knowledge influences their perceptions of PrivacyStory, leading to a fairer comparison.

Second, the sequential approach allows us to more clearly isolate the specific effects of using PrivacyStory, targeting specific goals. By comparing the threat models created manually vs. with PrivacyStory, we can directly assess if the tool leads to more accurate, complete, or efficient threat modeling (relating to our goal of assessing the tool's *usefulness*). Changes in participants' perceptions, as measured by the TAM questionnaire, can be more confidently attributed to their experience with PrivacyStory, rather than to their general understanding of threat modeling.

Finally, the sequential design simulates real-world adoption, enhancing ecological validity. In a real-world setting, developers might first encounter the need for privacy threat modeling and then explore tools like PrivacyStory to support their work. The sequential design mirrors this potential adoption process, enhancing the ecological validity of our findings and increasing confidence that the results will generalize to real-world usage scenarios.

To gain a comprehensive understanding of developers' experiences with the PrivacyStory tool, we will collect both quantitative and qualitative data. The quantitative evaluation will use the Technology Acceptance Model (TAM) metrics, focusing on perceived usefulness, ease of use, attitudes, and behavioral intentions. Participants will rate various aspects of the tool on a Likert scale. The specific TAM questions used in the evaluation are detailed in the analysis section.

In addition to the quantitative TAM data, we will also gather qualitative insights through open-ended feedback. Participants will have a dedicated space to provide detailed comments about their experiences using PrivacyStory, challenges they encountered, and suggestions for improving the tool.

8.2. Execution

The execution phase of the study involved careful preparation to ensure participants could effectively engage with the materials and tools in a remote setting. A combination of synchronous and asynchronous activities were planned to accommodate the geographical distribution of participants. Web-based interfaces were set up to provide access to the video course, quizzes, and the PrivacyStory tool. The procedure was designed with three key checkpoints to guide participants through the learning process, hands-on threat modeling exercise, and evaluation of the PrivacyStory tool. Some deviations from the original plan were necessary to adapt to the time constraints and availability of the expert participants. The following subsections provide more details on the preparation, procedure, and deviations during the execution of the study.

8.2.1. Preparation

The experiment was organized to be in combination of synchronous and asynchronous, and everything was to be remotely accessible, accommodating the geographical distribution

8. Empirical Evaluation

of the participants in relation to the authors. To facilitate this, individual user logins were created for each participant. These logins provided access to a web interface, enabling participants to progress through a video course at their convenience. In addition, a separate set of login credentials was provided for access to the PrivacyStory tool. Participants were given the necessary links and could experiment using their own devices.

8.2.2. Procedure

The study design includes three checkpoints: (1) introduction and preparation, (2) threat modeling exercise, and (3) PrivacyStory tool try-out.

Checkpoint 1: Introduction and Preparation At the beginning of the study, participants are provided with an introduction to the tasks they will be undertaking. They are granted access to a range of relevant materials, including a video course that covers the principles of privacy by design and guides them through the process of privacy threat modeling using the LINDDUN methodology. To ensure that participants have ample time to familiarize themselves with the content and complete the necessary prerequisites, a one-week deadline is set for them to submit their demographic forms, watch the video course in its entirety, and complete the quiz embedded within each module. This quiz serves as a means to assess their understanding and reinforce the key concepts presented in the course. In addition to these preparatory activities, a mutually agreed-upon meeting time is established for the upcoming synchronous privacy threat modeling exercise, which constitutes the second checkpoint of the study.

Checkpoint 2: Threat Modeling Exercise In the second checkpoint, participants engage in a synchronous threat modeling exercise focused on a healthcare scenario. They are provided with a specific user story to guide their threat modeling efforts. During this exercise, participants are tasked with creating a Data Flow Diagram (DFD) and identifying data flows that potentially contain threats. As they work through the process, participants are encouraged to engage in a think-aloud scenario, verbally describing their experiences, thought processes, and any challenges or hurdles they encounter along the way. The exercise is time-boxed to one hour, but participants have the flexibility to complete it sooner if possible. Upon completion of the exercise, participants are given a break, and a meeting time is established for the third checkpoint. To ensure that participants' threat modeling experiences remain fresh in their minds, the gap between the second and third checkpoints should not exceed four days. In preparation for the third checkpoint, participants are required to watch a video tutorial that introduces them to the PrivacyStory tool and its functionalities.

Checkpoint 3: PrivacyStory Tool Try-out The third checkpoint of the study is the participants apply the PrivacyStory tool to conduct privacy threat modeling for the same healthcare scenario they worked on in the previous checkpoint. Participants are provided with a set of tasks to guide their exploration and utilization of the PrivacyStory tool throughout the privacy threat modeling process. This hands-on session is designed to last no longer than 45 minutes, allowing participants to gain practical experience with the tool while working within a focused timeframe. Upon completion of the PrivacyStory tool try-out, participants are asked to fill out an evaluation form, which aims to capture

their feedback, perceptions, and experiences using the tool. To gain deeper insights into their thoughts and opinions, the evaluation form is also designed to have open-ended feedback where they can elaborate on their evaluation responses and share any additional reflections on the PrivacyStory tool's impact on their privacy requirements engineering efforts.

8.2.3. Deviations from the plan

Due to the time constraints of some participants and the timezone differences, the threat modeling exercises are conducted asynchronously. This means the participant can complete the exercises at their convenience without my supervision. We consider this approach acceptable, as the primary goal is for them to gain hands-on experience with the manual process of conducting privacy threat modeling. To support them, they have access to the necessary resources, including a video tutorial in the course and the LINDDUN website.

8.2.4. Analysis

The analysis phase focused on examining the quantitative and qualitative data collected from participants. The data set was prepared, and the analysis procedure was designed to handle feedback from students and practitioners separately. Descriptive statistics and thematic analysis were used to gain insights from the data. The following subsections detail the data set preparation and analysis procedure.

Data set preparation

The data set for this study consisted of responses from two groups of participants: 13 students and 10 privacy engineering experts/practitioners. To cater to the distinct demographics of these groups, four different questionnaires were designed, with two specifically tailored for students and two for practitioners.

The questionnaires collected both demographic information and feedback on the privacy engineering tool/method being evaluated. Demographic data included age, gender, educational background, professional experience (for practitioners only), and familiarity with privacy engineering principles. Feedback data, on the other hand, comprised quantitative Technology Acceptance Model (TAM) Likert scale ratings and qualitative open-ended responses providing insights and suggestions. Although the feedback questions were mostly similar across the student and practitioner questionnaires, they were kept separate to facilitate analysis.

The demographic information and feedback responses were organized into separate CSV files to prepare the data set for efficient analysis and comparison between the two groups.

8. Empirical Evaluation

Analysis procedure

The analysis procedure was designed to examine the feedback from student and expert participants separately, considering their different backgrounds and levels of expertise in software development and industry experience.

To analyze the quantitative data, we began by calculating descriptive statistics for each of the Technology Acceptance Model (TAM) variables. This provided an initial overview of the data, including the mean, median, and standard deviation for each variable.

To assess the internal consistency reliability of our TAM measures, we calculated Cronbach's Alpha for the items within each construct. Cronbach's Alpha is a widely used metric that evaluates how closely related a set of items are as a group, thereby measuring the reliability of the scale.

The qualitative feedback from both student and expert participants was analyzed using thematic analysis. The open-ended responses were carefully reviewed to identify recurring themes, patterns, and insights. The analysis focused on understanding the participants' experiences, challenges, and suggestions for improving the PrivacyStory tool. The identified themes were organized and presented to highlight the key findings and areas for enhancement.

It is important to interpret these results with caution. The small sample size limits the generalizability of the quantitative findings and may not be representative of the larger population of software developers and privacy experts. The quantitative results should be considered exploratory and indicative rather than conclusive.

8.3. Analysis: Student Feedback

8.3.1. Descriptive Statistics

Table 8.1 presents the quantitative analysis of the student feedback data from 13 students, including average ratings, percentages, and counts of various aspects of the PrivacyStory tool. The analysis focuses on three key aspects of the Technology Acceptance Model (TAM): Perceived Usefulness (PU), Perceived Ease of Use (PEOU), and Attitude Towards Using (ATU). Behavioral Intention to Use (BI) is not included in this analysis, as it is more relevant for practitioners to evaluate.

As the respondents mentioned, the most useful features of the privacy tool are shown in Table 8.2.

8.3.2. Qualitative Feedback

The Student Feedback survey revealed several key themes regarding the tool's effectiveness and user experience. Many respondents emphasized the tool's ability to save time and improve efficiency in the privacy by design process. As one respondent noted, "*These functions are very useful because they can simplify and speed up activities to identify privacy threats.*" Another stated, "*Generating DFDS, require a lot of time to do manually. This tool facilitates that process.*"

Table 8.1.: Technology Acceptance Model (TAM) Analysis Results

Component / Item	Mean $\pm$ Std Dev
Perceived Usefulness (PU)	
Does the "Generate Entity" feature result in entities that you can use to identify privacy risks or requirements?	4.38 $\pm$ 0.49
Do you think automatic entity recognition saves you time in identifying privacy-related entities?	4.62 $\pm$ 0.62
Do you think the automatic generation of the DFD will save you time when modeling data protection threats?	100% Yes
How comprehensive was the threat modeling process with the use of triples in capturing potential privacy threats?	4.46 $\pm$ 0.50
How relevant were the recommended privacy design patterns to your project's needs?	4.15 $\pm$ 0.66
Perceived Ease of Use (PEOU)	
How easy was it to understand and interpret the generated Data Flow Diagrams?	4.00 $\pm$ 0.55
Did the threat identification process using the "threat question" help identify potential threats that might otherwise have been overlooked?	4.23 $\pm$ 0.70
How actionable were the design pattern recommendations?	4.08 $\pm$ 0.62
Attitude Toward Using (ATU)	
Do you think this tool is necessary or beneficial in the process of Privacy by Design?	92.31% Yes

8. Empirical Evaluation

Table 8.2.: Most Useful Features of the Privacy Tool

Feature	Number of Mentions
Guided Threat Modeling	9
Generate Entities	7
Privacy Requirements Template	7
Project Creation and User Stories	6
Generate Individual DFD	6

Respondents also appreciated the tool’s comprehensiveness in threat modeling and privacy requirements. One user commented, "*The threat modeling & requirement make me choose what’s relevant for my project. Sometimes not all the requirement is required, this template automate all possiblity and i just need to select things that relevant.*"

The Data Flow Diagrams (DFDs) generated by the tool were generally well-received, although some users suggested improvements. One respondent mentioned, "*in data flow diagrams, the confusing thing is the unfamiliarity with diagrams like this. I understand more with simpler diagrams such as context diagrams.*"

Users also provided several suggestions for enhancing the tool’s functionality, such as "*a direct editing feature would be very useful, or a feature to be able to directly delete parts that are not used*" and "*It could be more effective if it can allow users to use the tool even when they’re not connected to the internet and allow multiple people to work on a project together, which is handy for teams.*"

Most respondents found the tool beneficial for implementing Privacy by Design principles, with one stating, "*because with this tool, we can implement Privacy By Design according to the standards that should be implemented. And this tool makes these analysis activities faster and easier.*"

Some responses indicated a learning curve associated with using the tool, particularly with understanding the DFDs. One user mentioned, "*As a user who understands more about context diagrams. I would rather have a context diagram feature to see the data flow. Because context diagrams look more concise and easier to understand.*"

Respondents also highlighted specific features they found particularly useful, with one commenting, "*Overall these functionalities collectively contribute to a more organized, secure, and privacy-conscious project development process. They assist in setting clear objectives, building a strong foundation, addressing security concerns, and ensuring compliance with privacy standards.*"

These themes reflect the overall positive reception of the tool, its perceived benefits in streamlining the Privacy by Design process, and areas where users see potential for further improvement.

8.4. Analysis: Practitioners Feedback

8.4.1. Demographics of the Practitioners

To provide context for the evaluation, we first present the demographics of the practitioners who participated in this study. Table 8.3 summarizes key characteristics of the 10 professionals involved.

Table 8.3.: Demographics of Participating Practitioners

Characteristic	Distribution
Software Engineering Experience	
Average years	5.9
3-5 years	60%
6-10 years	40%
Privacy Engineering Experience	
Average years	1.7
Less than 1 year	70%
3-5 years	30%
Current Roles	
Developers	50%
System Analysts	20%
Project Managers	10%
Other roles	20%
Industry Sectors	
Technology/Software	40%
Government	20%
Education	20%
Others	20%
Agile Methodology Experience	
With experience	80%
1-3 years	30%
3-5 years	30%
6-10 years	20%
Privacy Engineering Perceptions	
Importance in software development (avg. out of 5)	4.7
Confidence in implementation (avg. out of 5)	3.4

The practitioners in our study represented a diverse group of software professionals with varying levels of experience. On average, they had 5.9 years of experience in software engineering, with the majority (60%) having 3-5 years of experience and the remainder (40%) having 6-10 years.

In terms of privacy engineering, the participants had less experience, averaging 1.7 years. A significant majority (70%) had less than one year of experience in this specific

8. Empirical Evaluation

domain, while 30% had 3-5 years of experience.

The participants held various roles within their organizations, with half (50%) being developers, followed by system analysts (20%), project managers (10%), and other roles (20%). They worked across different sectors, predominantly in technology/software (40%), with equal representation (20% each) from government, education, and other sectors.

Most of the practitioners (80%) reported experience with agile methodologies, particularly in practices such as daily stand-ups, sprint planning, and retrospectives. Their experience with agile development was fairly evenly distributed, with 30% each having 1-3 years and 3-5 years of experience, and 20% having 6-10 years.

Notably, the practitioners generally viewed privacy engineering as highly important in software development, rating its importance at an average of 4.7 out of 5. However, their confidence in implementing privacy principles in real-world projects was more moderate, with an average rating of 3.4 out of 5.

This demographic profile provides important context for interpreting the practitioners' feedback on the PrivacyStory tool, representing a group with substantial software engineering experience but varying levels of privacy engineering expertise.

8.4.2. Descriptive Statistics

Table 8.4 shows the overall scores for all TAM components. The overall scores are notably high, ranging from 4.46 to 4.67 on a 5-point scale, indicating a strongly positive perception of the PrivacyStory tool. The Attitude Toward Using (ATU) component shows the highest score (4.67 ± 0.42), suggesting a very favorable attitude towards the tool with the most consistent responses. Perceived Usefulness (PU) follows closely (4.60 ± 0.54), indicating that participants find the tool highly useful for their work.

Reliability measures (Cronbach's Alpha) for PU, PEOU, and BI are all above 0.8, indicating high internal consistency. The ATU component shows a lower reliability (0.69), which, while acceptable, suggests some variability in responses for this component.

Standard deviations across all components are relatively low (0.42 to 0.60), indicating general agreement among participants. The PEOU component shows the highest standard deviation (0.60), suggesting slightly more diverse opinions on the tool's ease of use compared to other aspects.

Table 8.4.: Overall TAM Component Scores, Reliability, and Standard Deviation

Component	Mean $\pm$ Std Dev	Cronbach's Alpha
Perceived Usefulness (PU)	4.60 ± 0.54	0.86
Perceived Ease of Use (PEOU)	4.46 ± 0.60	0.84
Attitude Toward Using (ATU)	4.67 ± 0.42	0.69
Behavioral Intention to Use (BI)	4.50 ± 0.56	0.88

Table 8.5 presents a detailed breakdown of the Perceived Usefulness (PU) items. The practitioners' ratings reveal that they strongly recognize the importance of the "Generate Data Flow Diagram" feature for privacy-by-design processes. This highlights the tool's

8.4. Analysis: Practitioners Feedback

ability to effectively support the creation of data flow diagrams, which are crucial for identifying and analyzing privacy risks. The "Generate Entity" feature, guided threat modeling, threat modeling comprehensiveness, and the relevance of recommended privacy design patterns also received high scores, indicating their significant contributions to the overall usefulness of the tool in privacy-by-design practices.

Table 8.5.: Perceived Usefulness (PU) Items

Item	Mean $\pm$ Std Dev
I believe that the "Generate Entity" feature is important for the privacy-by-design process.	4.4 $\pm$ 1.07
I believe that the "Generate Data Flow Diagram" feature is important for the privacy-by-design process.	4.9 $\pm$ 0.32
Do you think the guided threat modeling will save you time and effort when conducting threat modeling?	4.7 $\pm$ 0.48
How comprehensive was the threat modeling process with the use of triples in capturing potential privacy threats?	4.5 $\pm$ 0.53
How relevant were the recommended privacy design patterns to your project's needs?	4.5 $\pm$ 0.71

Moving on to the Perceived Ease of Use (PEOU) items in Table 8.6, we observe that the practitioners found the "Generate Entity" and "Guided Threat Modeling" features to be the most user-friendly and intuitive. This suggests that these features have been designed and implemented in a way that allows practitioners to easily understand and utilize them in their privacy-by-design workflows. The "Generate Data Flow Diagram," diagram interpretation, and "Solution Recommendation" features

Table 8.6.: Perceived Ease of Use (PEOU) Items

Item	Mean $\pm$ Std Dev
How easily can you understand and use the functions and features of "Generate Entity"	4.7 $\pm$ 0.67
How easily can you understand and use the functions and features of "Generate Data Flow Diagram"	4.4 $\pm$ 0.84
How easy was it to understand and interpret the generated Data Flow Diagrams?	4.4 $\pm$ 0.97
How easily can you understand and use the functions and features of "Guided Threat Modeling"	4.6 $\pm$ 0.52
How easily can you understand and use the functions and features of "Solution Recommendation"	4.2 $\pm$ 0.79

Continuing with the Attitude Toward Using (ATU) items in Table 8.7, we see that the practitioners expressed a highly positive attitude towards the user story template. They strongly agreed that the template was a useful tool for generating privacy requirements

8. Empirical Evaluation

and that it made the process more efficient. This indicates that the user story template effectively streamlines the elicitation and documentation of privacy requirements, saving practitioners time and effort. The actionability of the design pattern recommendations was also well-received, suggesting that the tool provides practical and applicable guidance for implementing privacy solutions.

Table 8.7.: Attitude Toward Using (ATU) Items

Item	Mean $\pm$ Std Dev
I found the user story template to be a useful tool for generating privacy requirements.	4.8 $\pm$ 0.42
The user story template made the process of generating privacy requirements more efficient.	4.8 $\pm$ 0.42
How actionable were the design pattern recommendations?	4.4 $\pm$ 0.70

Finally, the Behavioral Intention to Use (BI) items in Table 8.8 demonstrate the practitioners' willingness to adopt and utilize the various features of the PrivacyStory tool in their privacy-by-design processes. The high mean scores for the intention to use the "Generate Entity," "Generate Data Flow Diagram," "Guided Threat Modeling," "Privacy Requirements Template," and "Solution Recommendation" features indicate that practitioners perceive these functionalities as valuable and are likely to incorporate them into their workflows. Notably, the practitioners also expressed a high likelihood of adopting the tool to elicit privacy requirements in their own projects, suggesting that the PrivacyStory tool has the potential to be widely embraced and applied in real-world software development contexts.

Table 8.8.: Behavioral Intention to Use (BI) Items

Item	Mean $\pm$ Std Dev
Assuming I have access to the "Generate Entity" feature, I intend to use it for identifying privacy-related entities.	4.5 $\pm$ 0.71
Given that I have access to the "Generate Data Flow Diagram" feature, I predict that I would use it in the privacy-by-design process.	4.4 $\pm$ 0.84
Given that I have access to the "Guided Threat Modeling" feature, I predict that I would use it in the privacy-by-design process.	4.5 $\pm$ 0.71
Given that I have access to the "Privacy Requirements Template", I predict that I would use it in the privacy-by-design process.	4.6 $\pm$ 0.52
Given that I have access to the "Solution Recommendation" feature, I predict that I would use it in the privacy-by-design process.	4.4 $\pm$ 0.84
I would adopt this tool to help me elicit privacy requirements in my own project.	4.6 $\pm$ 0.52

When asked to identify the most useful features of the PrivacyStory tool, practitioners

highlighted several key functionalities shown in Table 8.9:

Table 8.9.: Most Useful Features of the PrivacyStory Tool According to Practitioners

Feature	Number of Mentions
Guided Threat Modeling	9
Solution Recommendation	9
Generate Entities	8
Generate Individual DFD	8
Privacy Requirements Template	8
Generate Grouped DFD	7
Project Creation and User Stories	3

The most frequently mentioned features were Guided Threat Modeling and Solution Recommendation, both cited by 9 practitioners. This suggests practitioners value structured guidance through the threat identification process and actionable suggestions for addressing identified privacy concerns.

Generate Entities, Generate Individual DFD, and Privacy Requirements Template were each mentioned by 8 practitioners. These features likely streamline the initial stages of privacy analysis by automating the identification of key entities and visualizing data flows while also providing a framework for articulating specific privacy requirements.

The Generate Grouped DFD feature, mentioned by 7 practitioners. This functionality appears to be particularly useful for understanding complex interactions and identifying potential privacy issues at a broader system level.

Interestingly, while still considered useful, the Project Creation and User Stories feature was mentioned less frequently (3 times). This could suggest that while the feature provides a good starting point, practitioners find the more specialized privacy analysis tools to be of greater direct value in their work.

8.4.3. Qualitative Feedback

This section presents the results of the feedback stage, analyzing the responses from the 10 practitioners (P1 to P10) across various industries and with different levels of software engineering experience. The profile of each practitioners can be seen in Table 8.10.

Usefulness of the PrivacyStory Tool

The majority of participants found the PrivacyStory tool to be useful in supporting privacy-aware software development activities. P4, a mobile developer with 1-3 years of experience in the technology sector, highlighted the usefulness of the **entity generation feature**:

“Generate Entities: It’s a great functionality due to it being able to highlight important aspects of the user story and also enabling filter functionality” (P4)

8. Empirical Evaluation

Table 8.10.: Profile of the participants

Code	Role	Experience	Industry
P1	Developer	6-10 years	Manufacture
P2	Fullstack Software Engineer	1-3 years	Technology/Software
P3	System Analyst	11-20 years	Education
P4	Mobile Developer	1-3 years	Technology/Software
P5	Developer	3-5 years	Education
P6	Project Manager	11-20 years	Education
P7	Back end Developer	6-10 years	Government
P8	Developer	3-5 years	Technology/Software
P9	System Analyst	6-10 years	Government
P10	Developer	6-10 years	Technology/Software

P9, a system analyst with 6-10 years of experience in the government sector, emphasized the value of the Data Flow Diagram (DFD) generation:

“Being able to generate the DFD straight away and also presenting it with a table is really helpful since it gives a great visualization for that story point”
(P9)

P1, a developer with 6-10 years of experience in manufacturing, appreciated the **guided threat modeling** feature:

“Great feature because it gives options between giving requirements right off the bat and also giving flexibility and option of asking users questions regarding potential threats” (P1)

P6, a project manager with 11-20 years of experience in education, highlighted the tool’s comprehensive approach to **privacy requirements template** and **solution recommendation**:

“Tool dilengkapi dengan fitur untuk mengidentifikasi privacy requirements dan memberikan rekomendasi solusi untuk mendesign privacy yang baik sesuai dengan kebutuhan sistem, serta dilengkapi dengan guide dan detail penjelasan terkait threat yang mungkin muncul.” (P6)

(Translation: *“The tool is equipped with features to identify privacy requirements and provide solution recommendations for good privacy design according to system needs, as well as guides and detailed explanations regarding potential threats.”*)

Suggestion for Improvement

While the overall reception was positive, participants also identified areas for improvement and potential challenges in adopting the tool. P4 noted **inconsistencies in entity recognition**, particularly for complex subjects:

8.4. Analysis: Practitioners Feedback

“In some instances using the TOMO user story, the subject TB MDR Patient is often inconsistently classified, sometimes not being classified as a subject at all, sometimes only identifying "patient" as a subject, and at other times even classifying "TB" as a subject.” (P4)

P9, drawing from their experience in the government sector, suggested the need for better alignment with agile development practices:

“It’s good for privacy engineering in software development stage. One thing to be concerned about is that sometimes people need straight points direction in overcoming security and privacy concerns. Therefore, providing some priority or severity level will be helpful for development strategies.” (P9)

Participants also expressed a desire for more flexibility in modifying the automatically generated Data Flow Diagrams (DFDs). For instance, P2, a fullstack software engineer in the technology sector, proposed:

“It would be great if we could modify the data flow diagrams directly within the tool. This would allow us to fine-tune the diagrams to reflect our specific system architecture better.” (P2)

To aid in decision-making and resource allocation, P9 recommended:

“For real business case, too many insights may lead to confusion. It would be better to provide priority rank or severity level of privacy requirement. Therefore, it will boost strategic consideration to implement most critical concerns on the first stage. While other privacy concerns can be implemented gradually at the next stage of system enhancement.” (P9)

Several participants, such as P8, a developer in the technology sector, suggested improvements to the tool’s user interface and user experience to facilitate better understanding and smoother navigation:

“Applying some detail UI UX to help user understand the flow.” (P8)

P5, a developer in the education sector, proposed adding offline functionality to the tool:

“Having the ability to use the tool offline would be a great addition. It would provide more flexibility and allow me to work on privacy analysis even when I don’t have a stable internet connection.” (P5)

To streamline the requirements gathering process and capture user requirements more effectively, P9 suggested integrating the tool with familiar formats such as Excel:

“Excel file for requirement input will provide additional value and expected to capture user requirement details.” (P9)

8. Empirical Evaluation

These suggestions highlight the participants' desire for a more flexible, user-friendly, and integrated tool that can adapt to their specific needs and workflows. By addressing these areas of improvement, the PrivacyStory tool can become an even more powerful and indispensable asset for incorporating privacy considerations into the software development process.

8.5. Results and Discussion

This section begins by presenting the results of our goal evaluation. It then moves on to discuss the findings and their implications for further research. Finally, we address the threats to validity in our empirical evaluation.

8.5.1. Goal Evaluation

Table 8.11.: TAM Component Scores (Students and Practitioners)

Component	Students	Practitioners
Perceived Usefulness (PU)	4.40	4.60
Perceived Ease of Use (PEOU)	4.10	4.46
Attitude Toward Using (ATU)	4.62	4.67
Behavioral Intention to Use (BI)	-	4.50

Based on the TAM scores from student and practitioner responses in Table 8.11, the evaluation of each subgoal is as follows:

Subgoal 1 (Perceived Usefulness)

For students, the score for Perceived Usefulness (PU) is 4.40 out of 5. This high score indicates that students believe the PrivacyStory tool will significantly enhance their performance in privacy threat modeling. Practitioners rated the PU at 4.60 out of 5, suggesting they also see substantial benefits in using PrivacyStory for privacy requirements engineering tasks. Both groups show strong agreement with the tool's usefulness.

Subgoal 2 (Perceived Ease of Use)

Students rated the Perceived Ease of Use (PEOU) at 4.10 out of 5. While this score is slightly lower than the PU score, it still reflects a positive perception. Practitioners rated PEOU at 4.46 out of 5, indicating they find PrivacyStory relatively easy to learn, understand, and use. The tool is generally perceived as user-friendly by both students and practitioners, which is crucial for its adoption and continued use.

Subgoal 3 (Attitude Towards Using)

The score for Attitude Toward Using (ATU) is 4.62 out of 5 among students, and 4.67 out of 5 among practitioners, making it the highest score among all components for both groups. This exceptionally high score indicates a very positive overall affective reaction to using the PrivacyStory framework. Both students and practitioners have a highly favorable attitude towards the tool, which is promising for its acceptance and integration into their workflow.

Subgoal 4 (Behavioral Intention to Use)

The Behavioral Intention to Use (BI) score is available only from practitioners, who rated it at 4.50 out of 5. This high score indicates a strong likelihood that these professionals intend to use the PrivacyStory tool in the future, suggesting good potential for adoption among experienced developers. The absence of student data for this component does not detract from the positive outlook provided by the practitioner data.

The consistently high scores across all TAM components (all above 4.0 out of 5) suggest that the PrivacyStory tool has been well received by both students and practitioners. The tool is perceived as useful, relatively easy to use, and has generated a positive attitude among potential users. The high behavioral intention score from practitioners is particularly encouraging for future adoption.

These results indicate that the main goal of assessing the acceptance and willingness of developers to adopt the PrivacyStory tool has been met with very positive outcomes. The tool shows strong potential for adoption in privacy requirements engineering, with users recognizing its value in enhancing their performance in privacy threat modeling tasks.

However, it is important to note that these results are based on initial perceptions and intentions. Long-term studies on actual usage and integration into real-world projects would provide more definitive evidence of the tool's success and impact on privacy requirements engineering practices.

8.5.2. Findings and Implications

The evaluation of the PrivacyStory tool by both student and practitioner participants has yielded several noteworthy findings regarding its effectiveness and potential impact on privacy engineering practices. These findings provide validation for the tool's utility and highlight its broader implications for the field of privacy requirements engineering.

A significant finding is the tool's potential to address the knowledge gap in privacy engineering. With a substantial proportion of practitioners having limited experience in this domain, the tool's guided approach and educational value are particularly relevant. PrivacyStory could contribute to making privacy engineering practices more accessible to developers with varying levels of expertise. By providing structured guidance and automation, the tool reduces the barrier to entry for implementing privacy-by-design principles, potentially facilitating the adoption of robust privacy practices across the industry.

8. Empirical Evaluation

The positive feedback on time-saving features, particularly in entity recognition and DFD generation, indicates that PrivacyStory enhances the efficiency of privacy analysis. This efficiency gain is especially relevant in agile methodologies, where rapid iterations are essential. The tool's ability to quickly generate privacy requirements aligns well with sprint planning and iterative development processes.

The positive feedback on the guided threat modeling and the comprehensiveness of the analysis suggests that PrivacyStory enables a more thorough consideration of privacy issues. This is important given the complexity of privacy concerns and the potential for oversight in manual processes. By automating and structuring the threat identification process, the tool helps ensure that a wide range of potential privacy risks are considered, potentially leading to more robust and secure software systems.

The high ratings across all TAM components suggest that PrivacyStory has the potential to foster a more privacy-aware development culture. By making privacy analysis more accessible and integrated into the development process, the tool could help shift the perception of privacy from a compliance burden to an integral part of software quality. This cultural shift could have implications for how organizations approach privacy, potentially leading to more privacy-respecting products and services.

However, it is noteworthy that PrivacyStory does not currently offer automation or assistance with prioritizing privacy requirements, which reveals a gap in current privacy engineering practices. While identifying privacy concerns is crucial, the challenge lies in determining which issues to address first, especially given limited resources.

Furthermore, requests for more customization, particularly in the diagramming tool, highlight a tension in expert-domain tools. This suggests that as tools become more sophisticated, they must also become more flexible. The challenge lies in striking a balance between automation and user control, especially in a field as nuanced as privacy engineering.

Lastly, requests for integration with existing tools and workflows indicate a shift towards viewing PrivacyStory not as a standalone tool, but as a service and integral part of the broader software development ecosystem.

8.5.3. Threats to Validity

While this study provides valuable insights into the effectiveness and acceptance of the PrivacyStory tool, it is important to acknowledge and address potential threats to the validity of our findings. These threats can be categorized into four main areas: construct validity, internal validity, external validity, and conclusion validity. By examining these threats, we can better understand the limitations of our study and provide context for interpreting the results.

Construct Validity: The study's reliance on Technology Acceptance Model (TAM) metrics, while useful, may not fully capture all aspects of tool acceptance and usefulness. Additional constructs beyond Perceived Usefulness and Perceived Ease of Use could provide a more comprehensive evaluation. Furthermore, the dependence on self-reported data through questionnaires and interviews may not accurately reflect actual tool usage

and effectiveness. Incorporating observational data or analysis of tool outputs could provide more objective measures and strengthen the construct validity of the study.

Internal Validity: Several factors may have influenced the internal validity of the study. The novelty effect could have positively biased participants' perceptions of the tool, as initial impressions may differ from long-term usage patterns. The training provided on privacy concepts before tool use may have influenced participants' ability to use the tool effectively, making it challenging to isolate the tool's inherent usability. Additionally, the asynchronous nature of some sessions due to time constraints and timezone differences may have introduced inconsistencies in how participants engaged with the tool, potentially affecting the consistency of results.

External Validity: The generalizability of the study's findings is limited by several factors. The small sample size of 13 students and 10 practitioners restricts the ability to extrapolate results to broader populations of software developers. The use of controlled scenarios may not fully represent the complexity and variety of real-world software projects, potentially limiting the applicability of findings to diverse development contexts. Moreover, participants were primarily those with limited prior experience in privacy threat modeling, which may not reflect how experts in the field would perceive and use the tool.

Conclusion Validity: The study's conclusion validity is impacted by several limitations. The relatively small sample size may lead to low statistical power, potentially affecting the reliability of quantitative conclusions drawn from the data. The lack of a control group or comparative analysis with other privacy engineering tools limits the ability to draw strong conclusions about the relative effectiveness of PrivacyStory. Finally, the short-term nature of the study does not allow for conclusions about the long-term impact and sustained usefulness of the tool in real development environments, which would be crucial for understanding its true value in practical applications.

8.6. Chapter Summary

In this chapter, we conducted an empirical evaluation to assess the acceptance and willingness of software developers to adopt the PrivacyStory tool. The study involved two groups of participants: computer science students and software practitioners from various domains. The evaluation was structured around the Technology Acceptance Model (TAM) metrics, focusing on Perceived Usefulness (PU), Perceived Ease of Use (PEOU), Attitude Toward Using (ATU), and Behavioral Intention to Use (BI).

The study was executed using a combination of synchronous and asynchronous activities, with participants accessing materials and tools remotely. The procedure included three checkpoints: introduction and preparation, a threat modeling exercise, and a PrivacyStory tool try-out. Some deviations from the original plan were necessary to accommodate participants' time constraints and availability.

The analysis phase focused on examining the quantitative and qualitative data collected from student and expert participants. Descriptive statistics and thematic analysis were used to gain insights from the data. The feedback from students indicated a positive reception of the PrivacyStory tool, with high ratings for its usefulness, time-saving

8. *Empirical Evaluation*

capabilities, and comprehensiveness in addressing privacy considerations. The qualitative feedback highlighted the tool's ability to streamline the privacy analysis process, provide structured guidance, and offer actionable privacy design pattern recommendations.

The expert feedback analysis revealed similar positive sentiments, with practitioners appreciating the tool's efficiency, comprehensiveness, and potential for integrating privacy considerations into the software development process. The discussion of results synthesized the findings from both groups, highlighting the implications for the PrivacyStory tool and its adoption in real-world scenarios.

The chapter also acknowledged the limitations of the study and identified potential areas for future research. Overall, the empirical evaluation demonstrated the promising potential of the PrivacyStory tool in supporting software developers in conducting privacy threat modeling and incorporating privacy by design principles into their projects.

9. Conclusion

This chapter concludes the thesis. It begins with a summary of each part of the research. Following the summary, the main research question is addressed, and potential directions for future work are discussed.

9.1. Summary

This thesis presents an intelligent system based on artificial intelligence and machine learning to address privacy requirements and design engineering. The primary motivation for this research is to tackle the challenges that agile teams face when integrating privacy engineering into their requirements and design processes, which demand quick iterations and adaptability to changes. Although intelligent software engineering has the potential to bridge this gap, there has been limited exploration in the field of privacy engineering. To gain a comprehensive understanding of the current state of privacy requirements engineering methods, a systematic literature review (SLR) was conducted in Chapter 3. The review reveals that agile and intelligent systems are underutilized despite their necessity. Common workflows and sequences in privacy requirements and design engineering were identified, pinpointing four key areas where intelligent systems can enhance these processes.

Three of these areas are elaborated in detail in Chapter 4: (1). Identifying privacy-related entities such as personal data, processing activities, and data subjects using Named Entity Recognition (NER) techniques. (2). Connecting these entities and visualizing their relationships through Data Flow Diagrams (DFDs) using Natural Language Processing (NLP) techniques and Large Language Models (LLMs). These models were evaluated using standard machine learning metrics as well as human evaluation, demonstrating their feasibility for use in the privacy requirements engineering pipeline. (3). Applying the knowledge of entities and their connections to generate privacy requirements using privacy requirements patterns in user stories. These patterns can be directly applied based on the identified entities or through a guided threat modeling process that questions potential threats in the context of the DFD.

The fourth intelligent system focuses on privacy design engineering, elaborated in Chapter 5. This involves selecting suitable design patterns using an Information Retrieval (IR) technique called Learning to Rank. The evaluation of this system was conducted using information retrieval metrics, and further analysis identified the most useful features for generating effective recommendations.

All these individual intelligent system models are combined into an easily accessible tool called PrivacyStory, introduced in Chapter 6. The approach is demonstrated through

9. Conclusion

an example implementation in a camp management project, showcasing the output of each step to enhance understanding of the methodology, elaborated in Chapter 7.

Finally, the intelligent system approach was tested with both students and practitioners, as detailed in Chapter 8. A series of evaluations, including learning phases, experiments, and interviews, were conducted to assess the tool's usefulness and the users' willingness to adopt it for privacy requirements engineering. Feedback indicated that this approach is a useful, time-saving tool for privacy requirements and design engineering. However, more mature methods, such as LINDDUN, must continue to serve as a reference method for privacy threat modeling.

9.2. Answers to the Research Questions

To what extent can an intelligent system improve the process of privacy requirements and the design engineering process for agile requirements?

An intelligent system can improve the process of privacy requirements and design engineering for agile requirements by assisting agile developers in several key areas. While agile developers often prioritize functional requirements, an intelligent system can help maintain this focus while simultaneously identifying privacy-related entities within functional user stories. By establishing relationships between these entities in a data flow diagram, the intelligent system can aid in the elicitation of privacy requirements. Furthermore, the system can guide developers in selecting the most appropriate solutions by recommending relevant privacy design patterns. (The question of how the benefits will be proven and the willingness of practitioners to accept this has not yet been answered)

RQ1: What is the current state of privacy engineering research?

Answer to RQ1: The current state of privacy engineering research, based on the systematic literature review conducted in Chapter 3, reveals that Privacy Requirements Elicitation (PR) plays a central role in the overall process. PR serves as both a starting point and a destination for various other processes, with most transitions occurring after the identification of privacy threats. The most common sequence in the privacy engineering process involves moving from PR to the Selection of Privacy Patterns and Privacy-Enhancing Technologies (PETs). This suggests that the standard approach is to implement standardized solutions and technologies immediately after defining privacy requirements. The second most frequent sequence involves transitioning from Model-Based Characterization to Privacy Threat Analysis, indicating that after characterizing a system using models, the recommended next step is to conduct a privacy threat analysis on that model.

However, the study also highlights that the methods and adaptations in privacy engineering research are lagging behind in keeping pace with agile methodologies. Only a small subset of studies focuses on integrating privacy into agile processes. Additionally,

while there is an inclination to use computer-aided tools in the field of privacy engineering, they are currently underutilized.

RQ2: How can an intelligent system help to specify the privacy requirements from an agile requirements artifact?

Answer to RQ2: Intelligent systems can assist agile teams in defining privacy requirements from agile requirements artifacts through a three-step process. First, Named Entity Recognition (NER) techniques can be employed to identify privacy-related entities within the artifacts. Second, Natural Language Processing (NLP) techniques and Large Language Models (LLMs) can be utilized to automatically generate Data Flow Diagrams (DFDs) that visualize the interactions between these privacy entities. The generated DFDs can also serve as a basis for conducting privacy threat modeling, which helps to identify potential privacy risks and vulnerabilities. Finally, the intelligent system can generate more granular and contextualized privacy requirements using predefined requirements patterns based on the identified entities, their relationships in the DFDs, and the results of the privacy threat modeling.

RQ3: How intelligent system can help to streamline the process from the privacy requirement into a suitable privacy design pattern?

Answer to RQ3: An intelligent system can streamline the process of mapping privacy requirements to suitable privacy design patterns (PrDPs) by employing a recommendation system. This system utilizes Learning-to-Rank (LeToR) algorithms alongside semantic text characteristics to automatically match privacy requirements with the most appropriate PrDPs. The recommendation system relies on two main components: a classification model and a LeToR model, both of which have demonstrated satisfactory performance in our study. The use of text features proves to be particularly advantageous when other, more detailed properties are not readily available.

RQ4: To what extent are practitioners willing to use intelligent systems in the privacy requirements engineering process?

Answer to RQ4: Based on the empirical evaluation conducted with practitioners, there is strong evidence that they are highly willing to use intelligent systems in the privacy requirements engineering process. This conclusion is supported by the high overall Technology Acceptance Model (TAM) scores, particularly for Perceived Usefulness (4.60/5), Attitude Toward Using (4.67/5), and Behavioral Intention to Use (4.50/5). The detailed analysis of individual TAM items and qualitative feedback further reinforces practitioners' willingness to adopt tools like PrivacyStory, citing its efficiency gains, comprehensive privacy analysis, and potential to make privacy-by-design principles more actionable. However, practitioners' willingness is contingent on factors such as accurate results, customization options, seamless integration with existing tools, and actionable recommendations, which need to be addressed to ensure widespread acceptance and long-term use in real-world software development contexts.

9.3. Comparison to the State of the Art

This research builds on previous work in privacy requirements engineering (PRE) and intelligent systems used primarily for requirements engineering. In this section, we focus on comparing our approach with existing PRE methods, as this is the core focus of this thesis. We selected three closely related methods in terms of workflow, which have inspired various aspects of our approach: (1) LINDDUN, (2) ProPAN, and (3) PCM. Figure 9.1, also referenced in Chapter 3, illustrates a comparison of these PRE processes. Our method is similar in that it starts directly from system characterization, rather than from Goals (such as regulations, standards, or organizational goals), and follows a nearly identical sequence. Clarifying our state of the art is important here, so we elaborate our contributions from each method.

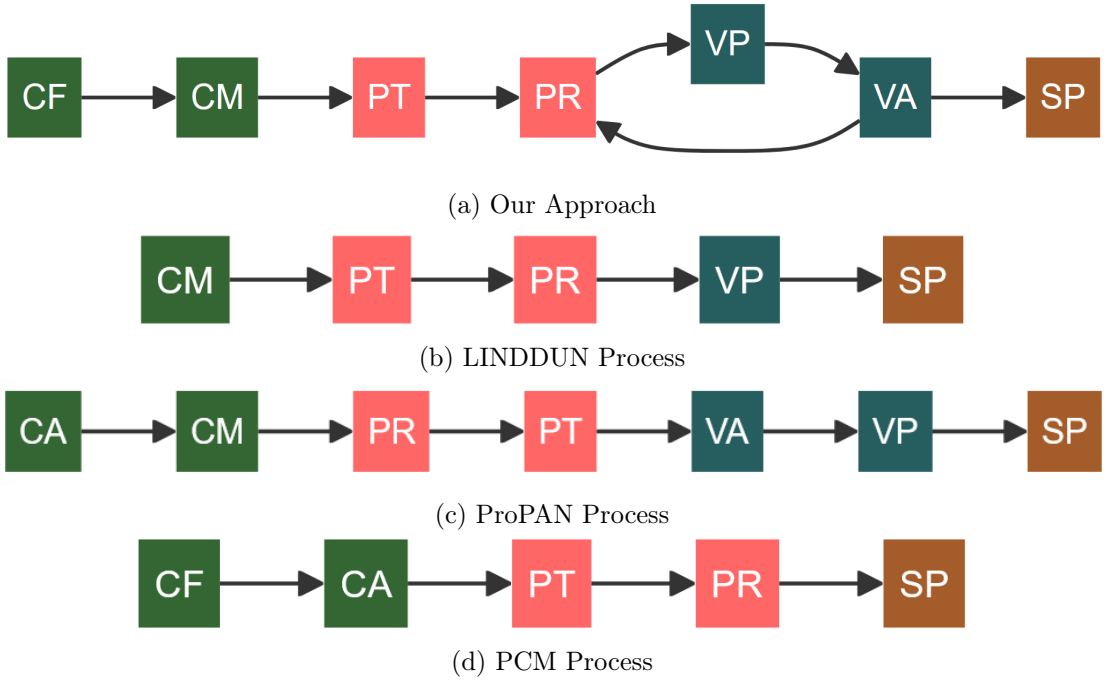


Figure 9.1.: Comparison of Processes among the State of the art. For reference on the codification, refer to Table 3.6

LINDDUN [40] and its streamlined version, LINDDUN GO [171], have significantly influenced this research, particularly through their data flow diagram (DFD)-based threat modeling introduced in the original method [40]. Our Guided Threat Modeling builds on the LINDDUN knowledge base, utilizing the content of LINDDUN GO Cards [171] as the foundation for our threat questions template. This research aims to address the challenges of manual DFD diagramming identified in previous studies [171]. We introduce the concept of triples for interpreting the DFD, serving as the basis for threat modeling instead of examining each interaction within every DFD element as LINDDUN does. This

approach is designed to be less exhaustive and more suitable for agile teams. Our method bridges the gap by automating DFD creation, guiding threat modeling, and enabling seamless integration into agile workflows.

ProPAN [112] significantly influenced this study, especially regarding the privacy requirements template used in computer-aided tools [112] and the main workflow of the privacy requirements engineering (PRE) process, particularly the requirements validation and prioritization steps. However, our work differs from ProPAN in that we do not use problem frame diagrams, which ProPAN employs. As mentioned in Meis's dissertation [109], these problem frame diagrams might hinder the adaptation of their approach, as users need to familiarize themselves with the notation. Additionally, Meis noted that ProPAN might face challenges in compatibility with agile software development processes [109]. In contrast, our research employs minimal modeling to facilitate easy adaptation and streamline integration with agile processes.

The PCM method [134] shares a similar objective of addressing agile privacy requirements, beginning with user stories. However, our approach differs as PCM utilizes questionnaires targeting specific privacy criteria templates, whereas our method reduces manual input by automatically detecting privacy artifacts such as actors, personal data, and processing directly from a collection of textual functional requirements, particularly user stories. Additionally, we enhance this by incorporating a threat modeling process and aligning it with privacy design patterns.

Overall, our work is unique in integrating agile requirements from the outset, using this as the foundation for engineering privacy requirements. This approach leverages intelligent systems while adhering to the comprehensive workflows of LINDDUN and ProPAN, traditionally suited for waterfall methodologies, but adapted here for a more agile context. This adaptation highlights our contribution to the field.

9.4. Reflection on Contribution and Implication

Requirements engineering, which focuses on defining, documenting, and maintaining software requirements, has garnered significant attention for its potential in automation. This discipline is crucial for the success of software development projects. Intelligent Systems, particularly Natural Language Processing (NLP), have become increasingly important for requirements engineering. NLP methods are now used to automatically extract and analyze requirements from textual data. The relevance of this development is highlighted by dedicated workshops like NLP4RE (Natural Language Processing for Requirements Engineering), which focus specifically on the intersection of NLP and requirements engineering.

Despite these advancements in requirements engineering automation, privacy requirements engineering has not received the same level of attention. Privacy requirements are essential to ensure that software systems comply with data protection regulations and protect user privacy. The lack of automated tools and processes in this field makes it challenging to guarantee privacy compliance. This thesis addresses this gap by focusing on privacy requirements engineering and exploring methods to streamline the identification,

9. Conclusion

analysis, and implementation of privacy requirements through automation. By doing so, it aims to bridge the divide between the advancements in general requirements engineering automation and the specific needs of privacy requirements engineering.

An intelligent system can greatly improve the process of privacy requirements and design engineering for agile needs. While agile developers often prioritize functional requirements, an intelligent system can help maintain this focus while also identifying privacy-related entities within functional user stories. The intelligent system supports the creation of privacy requirements by establishing connections between these entities in a data flow diagram and recommends relevant privacy design patterns, helping developers choose the most suitable solutions. Practitioners emphasized the tool's potential to improve efficiency in the privacy-by-design process and save time. However, it also highlighted the importance of striking a balance between traditional human expertise and automated systems. Although the tool can streamline processes, the need for human judgment in privacy matters became even more evident.

This thesis is particularly relevant to the privacy engineering and requirements engineering communities, which are gaining increasing attention from industry. As organizations and companies recognize the importance of integrating privacy considerations into their software development processes, the insights and tools provided by this thesis will be highly valuable. This work will further encourage the adoption of privacy by design, ensuring that privacy concerns are incorporated into the software development process from the outset.

In the broader context of the advancing era of artificial intelligence, particularly with the emergence of decoder-only large language models (LLMs), the intelligent system presented in this thesis may be overshadowed by the potential for LLM development, despite being a natural progression of technology. These models can handle a wide range of tasks and have a significant impact on various domains, including business process optimization, with the help of a simple text prompt. Essentially, the integration of the intelligent system discussed in this thesis addresses multiple aspects, highlighting the areas where LLMs can be particularly useful.

This thesis not only highlights the growing importance of automation in privacy engineering but also addresses a significant gap in agile privacy requirements engineering. This work makes the concept of privacy by design more tangible and achievable, potentially having a significant impact on both academic research and industry practices.

9.5. Basis for Future Work

During the writing of this thesis, numerous potential extensions to this work have emerged that could not be addressed within the study's timeframe. The emergence of LLMs during this research has also opened many possibilities for intelligent systems in privacy engineering. Several potential future research directions have been identified that would be worthwhile to explore and would likely interest other researchers in this field.

9.5.1. Expanding Intelligent Systems in the Holistic Process of Privacy Engineering

In Chapter 3, the holistic process of privacy requirements was introduced, laying the groundwork for a comprehensive and integrated approach to managing privacy in system design and development. Building on this foundation, there is significant potential to tailor intelligent systems for each component of this process to enhance efficiency and effectiveness.

Regulations and Standards Integration: The area of goal-based requirements derived from regulations, standards, or organizational policies is a promising field that has not been fully explored. Future work can focus on developing intelligent systems that refine these goals. For instance, leveraging technologies like knowledge graphs can systematically map regulatory requirements to specific system functionalities, ensuring comprehensive compliance. This approach can also facilitate the automatic update of requirements in response to regulatory changes.

Intelligent Systems for Goal Refinement: Developing systems that can automatically refine and interpret high-level privacy goals into actionable requirements is crucial. This could involve natural language processing (NLP) to understand and translate legal texts into technical specifications or using machine learning to predict the impact of new regulations on existing systems.

Managing Numerous Privacy Requirements: The holistic process often results in a large number of privacy requirements, which can be challenging to manage manually. Intelligent systems can assist in the automatic validation of these requirements, ensuring they are correctly implemented and maintained throughout the system's lifecycle. This involves developing algorithms that can continuously monitor system compliance with privacy requirements and flag deviations in real-time.

Effort Estimation and Prioritization: Prioritizing privacy requirements based on impact and likelihood is crucial. Future research can build on existing work in effort estimation to create models that predict the resources needed to implement specific privacy measures. These models can help organizations allocate resources more effectively and focus on the most critical privacy requirements first.

9.5.2. Exploring Agent-Based Large Language Models for Privacy Engineering

The emergence of Large Language Models (LLMs) during this research highlights the transformative potential of advanced AI systems in privacy engineering. These models, capable of performing tasks with minimal specific training (zero-shot learning), offer promising applications across various domains, including privacy requirements engineering. Although this research did not allow for a comprehensive exploration of LLMs, we identified

9. Conclusion

and leveraged one potential application in generating data flow diagrams. This preliminary application showcases the broader potential of LLMs in orchestrating intelligent systems for privacy engineering.

Generating Data Flow Diagrams (DFDs) LLMs can be used to automatically generate data flow diagrams based on system descriptions. This capability simplifies the process of visualizing data movement within a system, ensuring that privacy requirements are adequately addressed at each stage. Additionally, LLMs can facilitate the dynamic updating of DFDs in response to system changes, providing real-time analysis and ensuring continuous compliance with privacy standards.

Orchestrating LLMs as Agents in Holistic Modules LLMs can act as intelligent agents within each component of the holistic privacy requirements process. For instance, they can assist in eliciting goal-based requirements by interpreting regulatory texts and translating them into specific system requirements. We can achieve more thorough and dynamic assessments by integrating LLMs into Privacy Impact Assessments (PIAs). LLMs can analyze text data, identify potential privacy risks, and suggest mitigation strategies, enhancing the overall robustness of privacy impact assessments.

LLMs as Social Agents Privacy is not solely a technical issue but involves ethical and social dimensions. LLMs can be trained to understand and incorporate ethical considerations and social norms into privacy requirements, aligning with Nissenbaum's theory of contextual integrity [69, 120]. This involves assessing how information flows in different contexts and ensuring that these flows respect the expectations and norms of those contexts. Additionally, LLMs can serve as interactive agents that facilitate communication between technical and non-technical stakeholders. They can translate complex technical language into more accessible terms, ensuring all stakeholders understand the privacy implications of their decisions and actions.

Advanced Privacy Threat Analysis LLMs can enhance privacy threat analysis by predicting potential threats based on historical data and emerging trends. This predictive capability allows for implementing proactive measures, reducing the likelihood of privacy breaches. Furthermore, LLMs can thoroughly assess system vulnerabilities, identifying weaknesses that may not be immediately apparent to human analysts. This helps in creating more secure and privacy-compliant systems.

9.5.3. A More Robust Empirical Analysis with Practitioners

In Chapter 8, an empirical analysis was conducted with both students and practitioners. However, limitations on the practitioners' side are acknowledged. Addressing these limitations will be a focus for future research.

Effort Estimation with NASA-TLX The current study evaluated the intelligent system’s usefulness in accomplishing privacy requirements and design engineering tasks. However, an equally important aspect is the reduction of effort the intelligent system facilitates. Using the NASA Task Load Index (NASA-TLX), future studies should assess the perceived effort reduction to provide a more comprehensive understanding of the system’s impact on practitioners’ workload.

Actual Use (TAM) in the Work While the study utilized constructs from the Technology Acceptance Model (TAM), such as perceived usefulness and ease of use, it did not measure intention to use or actual use. Future research should incorporate these additional TAM constructs to better understand how practitioners adopt and integrate the intelligent system into their workflows. Monitoring and measuring intention to use and actual use will yield valuable insights into the system’s acceptance and long-term utilization.

Lastly, the quantity of practitioners involved in this experiment was limited, and the quality of their expertise varied. This limitation is acknowledged and suggests the need for a larger, more diverse sample of practitioners in future studies. Enhancing practitioner participants’ quantity and quality will strengthen the empirical analysis and provide more robust and generalizable results.

9.6. Concluding Remarks

The intention of this thesis is to make developer tasks easier when dealing with privacy regulations and principles that may appear difficult to understand at first glance. Current literature indicates that intelligent software development represents the future of the field. Although the rapid emergence of AI-assisted tools like Copilot [175] was not predicted within the next decade, their arrival has made coding easier and faster for developers. This development underscores the thesis’s intention to explore the potential of intelligent systems in the future, which also applies to privacy engineering.

This thesis approached the topic by examining privacy requirements and design considerations. Initially, the now-classic natural language processing and machine learning techniques were employed. Given the rapid advances, particularly with the advent of large language models and increasingly intelligent systems, the possibility of applying them to privacy engineering is growing. The primary goal is to not only use intelligent systems but also to understand their role in the development pipeline. The key contributions of this thesis have been to pave the way for more careful automation of complex tasks, such as detecting entities in requirements, creating data flow diagrams, and eliciting privacy requirements, with humans remaining in the loop. The positive reception of this research approach among practitioners will hopefully encourage more research attention and practical application.

Bibliography

- [1] ABERKANE, A.-J., POELS, G., AND BROUCKE, S. V. Exploring automated gdpr-compliance in requirements engineering: A systematic mapping study. *Ieee Access* 9 (2021), 66542–66559.
- [2] ABUALHAIJA, S., VOGELSANG, A., AND DESHPANDE, G. Preface: 6th workshop on natural language processing for requirements engineering (nlp4re’23).
- [3] AHMED, M., KHAN, S. U. R., AND ALAM, K. A. An nlp-based quality attributes extraction and prioritization framework in agile-driven software development. *Automated Software Engineering* 30, 1 (2023), 7.
- [4] AL-MOMANI, A., WUYTS, K., SION, L., KARGL, F., JOOSEN, W., ERB, B., AND BÖSCH, C. Land of the lost: Privacy patterns’ forgotten properties: Enhancing selection-support for privacy patterns. *Proceedings of the ACM Symposium on Applied Computing* (2021), 1217–1225.
- [5] ALKHARIJI, L., ALHIRABI, N., ALRAJA, M. N., BARHAMGI, M., RANA, O., AND PERERA, C. Synthesising Privacy by Design Knowledge Towards Explainable Internet of Things Application Designing in Healthcare.
- [6] ALKHARIJI, L., DE, S., RANA, O., AND PERERA, C. Semantics-based privacy by design for Internet of Things applications. *Future Generation Computer Systems* 138 (January 2023), 280–295.
- [7] ALSAQAF, W., DANEVA, M., AND WIERINGA, R. Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and Software Technology* 110 (2019), 39–55.
- [8] ALSHAREEF, H., STUCKI, S., AND SCHNEIDER, G. Transforming Data Flow Diagrams for Privacy Compliance (Long Version). *arXiv preprint arXiv:2011.12028* (2020).
- [9] AMBLER, S. W. The object primer: Agile model-driven development with uml 2.0, 2004.
- [10] ANSARI, M. T. J., BAZ, A., ALHAKAMI, H., ALHAKAMI, W., KUMAR, R., AND KHAN, R. A. P-STORE: Extension of STORE Methodology to Elicit Privacy Requirements. *Arabian Journal for Science and Engineering* 46, 9 (2021), 8287–8310.

Bibliography

- [11] ANSARI, M. T. J., BAZ, A., ALHAKAMI, H., ALHAKAMI, W., KUMAR, R., AND KHAN, R. A. P-store: Extension of store methodology to elicit privacy requirements. *Arabian Journal for Science and Engineering* 46, 9 (2021), 8287–8310.
- [12] ARFAOUI, S., MEZRIOUI, A., AND BELMEKKI, A. A Methodology for Assuring Privacy by Design in Information Systems. *International Journal of Communication Networks and Information Security (IJCNIS)* 12, 3 (April 2022).
- [13] AYALA-RIVERA, V., AND PASQUALE, L. The Grace Period Has Ended: An Approach to Operationalize GDPR Requirements. In *2018 IEEE 26th International Requirements Engineering Conference (RE)* (Banff, AB, August 2018), IEEE, pp. 136–146.
- [14] BAAS, J., SCHOTTEN, M., PLUME, A., CÔTÉ, G., AND KARIMI, R. Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies. *Quantitative science studies* 1, 1 (2020), 377–386.
- [15] BARTOLINI, C., DAOUDAGH, S., LENZINI, G., AND MARCHETTI, E. GDPR-Based User Stories in the Access Control Perspective. In *Quality of Information and Communications Technology* (Cham, 2019), M. Piattini, P. da Cunha, I. de Guzmán, and R. Pérez-Castillo, Eds., Springer International Publishing, pp. 3–17.
- [16] BASILI, V. R., AND ROMBACH, H. D. The tame project: Towards improvement-oriented software environments. *IEEE Transactions on software engineering* 14, 6 (1988), 758–773.
- [17] BECKERS, K., FASSBENDER, S., HEISEL, M., AND MEIS, R. A problem-based approach for computer-aided privacy threat identification. In *Privacy Technologies and Policy: First Annual Privacy Forum, APF 2012, Limassol, Cyprus, October 10-11, 2012, Revised Selected Papers 1* (2014), Springer, pp. 1–16.
- [18] BERRY, D. M. Evaluation of Tools for Hairy Requirements and Software Engineering Tasks. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)* (2017), pp. 284–291.
- [19] BERRY, D. M. Empirical evaluation of tools for hairy requirements engineering tasks. *Empirical Software Engineering* 26, 6 (2021), 1–77.
- [20] BLANCO-LAINÉ, G., SOTTET, J.-S., AND DUPUY-CHESSA, S. Using an Enterprise Architecture Model for GDPR Compliance Principles. In *The Practice of Enterprise Modeling*, J. Gordijn, W. Guédria, and H. A. Proper, Eds., vol. 369. Springer International Publishing, Cham, 2019, pp. 199–214. Series Title: Lecture Notes in Business Information Processing.
- [21] BOZDAG, E. Privacy at speed: Privacy by design for agile development at uber.

- [22] BREAU, T., AND ANTON, A. Analyzing Regulatory Rules for Privacy and Security Requirements. *IEEE Transactions on Software Engineering* 34, 1 (January 2008), 5–20.
- [23] BURGESS, C. J. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.
- [24] CAIZA, J. C., MARTÍN, Y. S., GUAMÁN, D. S., DEL ALAMO, J. M., AND YELMO, J. C. Reusable Elements for the Systematic Design of Privacy-Friendly Information Systems: A Mapping Study. *IEEE Access* 7 (2019), 66512–66535.
- [25] CÁMARA, J., TROYA, J., BURGUEÑO, L., AND VALLECILLO, A. On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml. *Software and Systems Modeling* (2023), 1–13.
- [26] CANEDO, E. D., CALAZANS, A. T. S., CERQUEIRA, A. J., COSTA, P. H. T., AND MASSON, E. T. S. Agile teams’ perception in privacy requirements elicitation: Lgpd’s compliance in brazil. In *2021 IEEE 29th International Requirements Engineering Conference (RE)* (2021), IEEE, pp. 58–69.
- [27] CASILLO, F., DEUFEMIA, V., AND GRAVINO, C. Detecting privacy requirements from User Stories with NLP transfer learning models. *Information and Software Technology* (2022), 106853.
- [28] CAVOUKIAN, A. Privacy by Design - The 7 foundational principles - Implementation and mapping of fair information practices. *Information and Privacy Commissioner of Ontario, Canada* (2009), 5.
- [29] CAVOUKIAN, A., ET AL. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada* 5 (2009), 12.
- [30] CHECKLAND, P., AND POULTER, J. *Soft Systems Methodology*. Springer London, London, 2020, pp. 201–253.
- [31] CHEN, W., LIU, T.-Y., LAN, Y., MA, Z.-M., AND LI, H. Ranking measures and loss functions in learning to rank. In *Advances in Neural Information Processing Systems* (2009), Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22, Curran Associates, Inc.
- [32] CHENG, B. H., AND ATLEE, J. M. Research directions in requirements engineering. *Future of software engineering (FOSE’07)* (2007), 285–303.
- [33] CNIL. CNIL PIA Software, 2019.
- [34] COHN, M. User stories applied: For agile software development, 2004.
- [35] COLES, J., FAILY, S., AND KI-ARIES, D. Tool-supporting data protection impact assessments with cairis. In *2018 IEEE 5th International Workshop on Evolving Security & Privacy Requirements Engineering (ESPREE)* (2018), IEEE, pp. 21–27.

Bibliography

- [36] COLESKY, M., AND CAIZA, J. C. A system of privacy patterns for informing users: Creating a pattern system. *ACM International Conference Proceeding Series* (2018).
- [37] COLESKY, M., DEMETZOU, K., FRITSCH, L., AND HEROLD, S. Helping Software Architects Familiarize with the General Data Protection Regulation. *Proceedings - 2019 IEEE International Conference on Software Architecture - Companion, ICSA-C 2019* (2019), 226–229.
- [38] DAI, X., AND ADEL, H. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020* (2020), D. Scott, N. Bel, and C. Zong, Eds., International Committee on Computational Linguistics, pp. 3861–3867.
- [39] DALPIAZ, F. Requirements data sets (user stories). *Mendeley Data, V1* (2018).
- [40] DENG, M., WUYTS, K., SCANDARIATO, R., PRENEEL, B., AND JOOSEN, W. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering* 16, 1 (2011), 3–32.
- [41] DENNEDY, M., FOX, J., AND FINNERAN, T. The privacy engineer’s manifesto: getting from policy to code to qa to value, 2014.
- [42] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [43] DIAMANTOPOULOU, V., ARGYROPOULOS, N., KALLONIATIS, C., AND GRITZALIS, S. Supporting the design of privacy-aware business processes via privacy process patterns. In *2017 11th International Conference on Research Challenges in Information Science (RCIS)* (Brighton, United Kingdom, May 2017), IEEE, pp. 187–198.
- [44] DIAMANTOPOULOU, V., AND KARYDA, M. Integrating Privacy-By-Design with Business Process Redesign. In *Computer Security. ESORICS 2021 International Workshops*, S. Katsikas, C. Lambrinoudakis, N. Cuppens, J. Mylopoulos, C. Kalloniatis, W. Meng, S. Furnell, F. Pallas, J. Pohle, M. A. Sasse, H. Abie, S. Ranise, L. Verderame, E. Cambiaso, J. Maestre Vidal, and M. A. Sotelo Monge, Eds., vol. 13106. Springer International Publishing, Cham, 2022, pp. 127–137. Series Title: Lecture Notes in Computer Science.
- [45] DIAS, E., IAN, C., BANDEIRA, N., TOFFANO, A., CALAZANS, S., HENRIQUE, P., COSTA, T., CATARINE, E., CANÇADO, R., AND BONIFÁCIO, R. Privacy requirements elicitation : a systematic literature review and perception analysis of IT practitioners. *Requirements Engineering*, 0123456789 (2022), 12–14.

- [46] DROZD, O. Privacy pattern catalogue: A tool for integrating privacy principles of iso/iec 29100 into the software development process. *Privacy and Identity Management. Time for a Revolution? 10th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2. 2 International Summer School, Edinburgh, UK, August 16-21, 2015, Revised Selected Papers 10* (2016), 129–140.
- [47] EHÉCATL MORALES-TRUJILLO, M., GARCÍA-MIRELES, G. A., MATLA-CRUZ, E. O., AND PIATTINI, M. A systematic mapping study on privacy by design in software engineering.
- [48] FERNÁNDEZ, D. M., WAGNER, S., KALINOWSKI, M., SCHEKELMANN, A., TUZCU, A., CONTE, T., SPINOLA, R., AND PRIKLADNICKI, R. Naming the pain in requirements engineering: comparing practices in brazil and germany. *IEEE Software* 32, 5 (2015), 16–23.
- [49] FERREYRA, N. E. D., TESSIER, P., PEDROZA, G., AND HEISEL, M. PDP-ReqLite: A Lightweight Approach for the Elicitation of Privacy and Data Protection Requirements. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, G. Navarro-Arribas, and J. Herrera-Joancomarti, Eds., vol. 12484. Springer International Publishing, Cham, 2020, pp. 161–177. Series Title: Lecture Notes in Computer Science.
- [50] GALSTER, M., GILSON, F., AND GEORIS, F. What Quality Attributes Can We Find in Product Backlogs? A Machine Learning Perspective. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11681 LNCS. 2019, pp. 88–96.
- [51] GALVEZ, R., AND GURSES, S. The Odyssey: Modeling Privacy Threats in a Brave New World. *Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018* (2018), 87–94.
- [52] GHARIB, M., GIORGINI, P., SALNITRI, M., PAJA, E., MOURATIDIS, H., PAVLIDIS, M., AND RUIZ, J. F. A holistic approach for privacy requirements analysis: An industrial case study. *Visual Privacy Management: Design and Applications of a Privacy-Enabling Platform* (2020), 22–53.
- [53] GHARIB, M., SALNITRI, M., PAJA, E., GIORGINI, P., MOURATIDIS, H., PAVLIDIS, M., RUIZ, J. F., FERNANDEZ, S., AND SIRIA, A. D. Privacy Requirements: Findings and Lessons Learned in Developing a Privacy Platform. In *2016 IEEE 24th International Requirements Engineering Conference (RE)* (Beijing, China, September 2016), IEEE, pp. 256–265.
- [54] GILSON, F., GALSTER, M., AND GEORIS, F. Generating Use Case Scenarios from User Stories. In *Proceedings of the International Conference on Software and System Processes* (New York, NY, USA, 2020), ICSSP '20, Association for Computing Machinery, pp. 31–40.

Bibliography

- [55] GILSON, F., AND IRWIN, C. From User Stories to Use Case Scenarios towards a Generative Approach. In *2018 25th Australasian Software Engineering Conference (ASWEC)* (2018), IEEE, pp. 61–65.
- [56] GORMLEY, K. One hundred years of privacy. *Wis. L. Rev.* (1992), 1335.
- [57] GUAN, H., YANG, H., AND WANG, J. An ontology-based approach to security pattern selection. *International Journal of Automation and Computing* 13, 2 (apr 2016), 168–182.
- [58] GUO, W., LIU, X., WANG, S., GAO, H., SANKAR, A., YANG, Z., GUO, Q., ZHANG, L., LONG, B., CHEN, B.-C., ET AL. Detext: A deep text ranking framework with bert. In *Proceedings of the 29th ACM international conference on information & knowledge management* (2020), pp. 2509–2516.
- [59] GUPTA, A., POELS, G., AND BERA, P. Creation of multiple conceptual models from user stories—a natural language processing approach. In *Advances in Conceptual Modeling: ER 2019 Workshops FAIR, MREBA, EmpER, MoBiD, OntoCom, and ER Doctoral Symposium Papers, Salvador, Brazil, November 4–7, 2019, Proceedings* (2019), pp. 47–57.
- [60] GÜRSSES, S., AND DEL ALAMO, J. M. Privacy Engineering: Shaping an Emerging Field of Research and Practice. *IEEE Security and Privacy* 14, 2 (2016), 40–46.
- [61] GÜRSSES, S., AND DEL ALAMO, J. M. Privacy engineering: Shaping an emerging field of research and practice. *IEEE Security & Privacy* 14, 2 (2016), 40–46.
- [62] GÜRSSES, S., AND VAN HOBOKEN, J. Privacy after the Agile Turn. *The Cambridge Handbook of Consumer Privacy*, Ili (2018), 579–601.
- [63] HAFIZ, M. A pattern language for developing privacy enhancing technologies. *Software: Practice and Experience* 43, 7 (2013), 769–787.
- [64] HAN, X., AND LEI, S. Feature selection and model comparison on microsoft learning-to-rank data sets. *arXiv preprint arXiv:1803.05127* (2018).
- [65] HANSEN, M., JENSEN, M., AND ROST, M. Protection Goals for Privacy Engineering. In *2015 IEEE Security and Privacy Workshops* (2015), pp. 159–166.
- [66] HAREL, D., AND RUMPE, B. Meaningful modeling: what’s the semantics of "semantics"? *Computer* 37, 10 (2004), 64–72.
- [67] HERWANTO, G. B. Automating data flow diagram generation from user stories using large language models. In *7th Workshop on Natural Language Processing for Requirements Engineering* (2024).
- [68] HERWANTO, G. B., EKAPUTRA, F. J., QUIRCHMAYR, G., AND TJOA, A. M. Towards a holistic privacy requirements engineering process: Insights from a systematic literature review. *IEEE Access* (2024).

- [69] HERWANTO, G. B., PUTRI, D. U. K., NINGTYAS, A. M., FUAD, A., QUIRCHMAYR, G., AND TJOA, A. M. Integrating contextual integrity in privacy requirements engineering: A study case in personal e-health applications. In *International Conference on Innovations for Community Services* (2024), Springer, pp. 237–256.
- [70] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. A named entity recognition based approach for privacy requirements engineering. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)* (2021), IEEE, pp. 406–411.
- [71] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. From user stories to data flow diagrams for privacy awareness: A research preview. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (2022), Springer, pp. 148–155.
- [72] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. PrivacyStory: Tool Support for Extracting Privacy Requirements from User Stories. In *2022 IEEE 30th International Requirements Engineering Conference (RE)* (Melbourne, Australia, August 2022), IEEE, pp. 264–265.
- [73] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. Privacystory: Tool support for extracting privacy requirements from user stories. In *30th IEEE international requirements engineering conference, RE 2022* (2022), Proceedings of the 30th IEEE international requirements engineering conference, RE 2022.
- [74] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. Learning to rank privacy design patterns: A semantic approach to meeting privacy requirements. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (2024), Springer, pp. 57–73.
- [75] HERWANTO, G. B., QUIRCHMAYR, G., AND TJOA, A. M. Leveraging nlp techniques for privacy requirements engineering in user stories. *IEEE Access* (2024).
- [76] HEVNER, A. R., MARCH, S. T., PARK, J., AND RAM, S. Design science in information systems research. *MIS quarterly* (2004), 75–105.
- [77] HILL, K. How target figured out a teen girl was pregnant before her father did. *Forbes, Inc* 7 (2012), 4–1.
- [78] HOEPMAN, J.-H. Privacy design strategies. In *ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings 29* (2014), Springer, pp. 446–459.
- [79] HUANG, Z., XU, W., AND YU, K. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

Bibliography

- [80] HUSSAIN, S., KEUNG, J., SOHAIL, M. K., KHAN, A. A., ILAHI, M., AHMAD, G., MUFTI, M. R., AND NOOR, M. A. A methodology to rank the design patterns on the base of text relevancy. *Soft Computing* 23, 24 (2019), 13433–13448.
- [81] HUSSAIN, W., SHAHIN, M., HODA, R., WHITTLE, J., PERERA, H., NURWIDYANTORO, A., SHAMS, R. A., AND OLIVER, G. How Can Human Values Be Addressed in Agile Methods? A Case Study on SAFe. 1–18.
- [82] INAYAT, I., SALIM, S. S., MARCZAK, S., DANEVA, M., AND SHAMSHIRBAND, S. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior* 51 (2015), 915–929.
- [83] ISLAM, M. B., WATSON, J., IANNELLA, R., AND GEVA, S. A greater understanding of social networks privacy requirements: The user perspective. *Journal of Information Security and Applications* 33 (April 2017), 30–44.
- [84] IWAYA, L. H., BABAR, M. A., AND RASHID, A. Privacy engineering in the wild: Understanding the practitioners’ mindset, organisational culture, and current practices. *arXiv preprint arXiv:2211.08916* (2022).
- [85] JARZĘBOWICZ, A., AND WEICHBROTH, P. A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices. In *Lean and Agile Software Development* (Cham, 2021), A. Przybyłek, J. Miler, A. Poth, and A. Riel, Eds., Springer International Publishing, pp. 91–110.
- [86] JENSEN, C., TULLIO, J., POTTS, C., AND MYNATT, E. D. Strap: a structured analysis framework for privacy. Tech. rep., Georgia Institute of Technology, 2005.
- [87] JOYEE DE, S., AND LE MÉTAYER, D. PRIAM: A privacy risk analysis methodology. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9963 LNCS (2016), 221–229.
- [88] KALLONIATIS, C., KAVAKLI, E., AND GRITZALIS, S. Dealing with privacy issues during the system design process. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.* (2005), IEEE, pp. 546–551.
- [89] KALLONIATIS, C., KAVAKLI, E., AND GRITZALIS, S. Using Privacy Process Patterns for Incorporating Privacy Requirements into the System Design Process. In *The Second International Conference on Availability, Reliability and Security (ARES’07)* (Vienna, April 2007), IEEE, pp. 1009–1017.
- [90] KALLONIATIS, C., KAVAKLI, E., AND GRITZALIS, S. Addressing privacy requirements in system design: the PriS method. *Requirements Engineering* 13, 3 (September 2008), 241–255.

- [91] KAVAKLI, E., GRITZALIS, S., AND CHRISTOS, K. Protecting privacy in system design: the electronic voting case. *Transforming Government: People, Process and Policy* 1, 4 (2007), 307–332.
- [92] KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., YE, Q., AND LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
- [93] KOSTOVA, B., GÜRSSES, S., AND TRONCOSO, C. Privacy engineering meets software engineering. on the challenges of engineering privacy bydesign.
- [94] LE, D., INRIA, M., TRILATERAL, I. K., AND MARÍA, J. PRIPARE Privacy- and Security-by-Design Methodology Handbook.
- [95] LIU, T.-Y., ET AL. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.
- [96] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTLEMOYER, L., AND STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [97] LUCASSEN, G., DALPIAZ, F., VAN DER WERF, J. M. E., AND BRINKKEMPER, S. Visualizing user story requirements at multiple granularity levels via semantic relatedness. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2016), 463–478.
- [98] LUCASSEN, G., DALPIAZ, F., WERF, J. M. E. V. D., AND BRINKKEMPER, S. The use and effectiveness of user stories in practice. In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14-17, 2016, Proceedings 22* (2016), Springer, pp. 205–222.
- [99] LUDEWIG, J. Models in software engineering—an introduction. *Software and Systems Modeling* 2 (2003), 5–14.
- [100] LUNA, J., SURI, N., AND Krontiris, I. Privacy-by-design based on quantitative threat modeling. In *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)* (Cork, Ireland, October 2012), IEEE, pp. 1–8.
- [101] LUNDBERG, S. M., AND LEE, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.
- [102] MAI, P. X., GOKNIL, A., SHAR, L. K., PASTORE, F., BRIAND, L. C., AND SHAAME, S. Modeling Security and Privacy Requirements: a Use Case-Driven Approach. *Information and Software Technology* 100 (August 2018), 165–182.

Bibliography

- [103] MARTIN, Y.-S., AND KUNG, A. Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (London, United Kingdom, April 2018), IEEE, pp. 108–111.
- [104] MARTÍN GARCÍA, Y. S., AND ÁLAMO RAMIRO, J. M. D. A metamodel for privacy engineering methods. *CEUR Workshop Proceedings*.
- [105] MATHARU, G. S., MISHRA, A., SINGH, H., AND UPADHYAY, P. Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes* 40, 1 (2015), 1–6.
- [106] MCCALLUM, A., NIGAM, K., ET AL. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (1998), vol. 752, Madison, WI, pp. 41–48.
- [107] MEAD, N. R., MIYAZAKI, S., AND ZHAN, J. Integrating privacy requirements considerations into a security requirements engineering method and tool. *International Journal of Information Privacy, Security and Integrity* 1, 1 (2011), 106–126.
- [108] MEHDY, N., KENNINGTON, C., AND MEHRPOUYAN, H. Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks. In *Security and Privacy in New Computing Environments* (Cham, 2019), J. Li, Z. Liu, and H. Peng, Eds., Springer International Publishing, pp. 152–177.
- [109] MEIS, R. *Problem-Based Privacy Analysis (ProPan): A Computer-aided Privacy Requirements Engineering Method*. Universitaet Duisburg-Essen (Germany), 2018.
- [110] MEIS, R., AND HEISEL, M. Systematic Identification of Information Flows from Requirements to Support Privacy Impact Assessments:. In *Proceedings of the 10th International Conference on Software Paradigm Trends* (Colmar, Alsace, France, 2015), SCITEPRESS - Science and Technology Publications, pp. 43–52.
- [111] MEIS, R., AND HEISEL, M. Computer-aided identification and validation of privacy requirements. *Information (Switzerland)* 7, 2 (2016), 28.
- [112] MEIS, R., AND HEISEL, M. Computer-Aided Identification and Validation of Privacy Requirements. *Information* 7, 2 (May 2016), 28.
- [113] MEIS, R., AND HEISEL, M. Understanding the privacy goal intervenability. In *Trust, Privacy and Security in Digital Business* (Cham, 2016), S. Katsikas, C. Lambrinoudakis, and S. Furnell, Eds., Springer International Publishing, pp. 79–94.
- [114] MEIS, R., AND HEISEL, M. Pattern-based representation of privacy enhancing technologies as early aspects. In *International Conference on Trust and Privacy in Digital Business* (2017), Springer, pp. 49–65.

- [115] MEIS, R., WIRTZ, R., AND HEISEL, M. A taxonomy of requirements for the privacy goal transparency. In *Trust, Privacy and Security in Digital Business: 12th International Conference, TrustBus 2015, Valencia, Spain, September 1-2, 2015, Proceedings 12* (2015), Springer, pp. 195–209.
- [116] MIKHEEV, A., MOENS, M., AND GROVER, C. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics* (1999).
- [117] MIR, D. J., SHVARTZSHNAIDER, Y., AND LATONERO, M. It Takes a Village: A Community Based Participatory Framework for Privacy Design. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (London, April 2018), IEEE, pp. 112–115.
- [118] MIRI, M., FOOMANY, F. H., AND MOHAMMED, N. Complying with gdpr: An agile case study. *ISACA Journal 2* (2018).
- [119] MIRI, M., H FOOMANY, F., AND MOHAMMED, N. Isaca: Complying with gdpr, an agile case study. *ISACA Journal 2* (04 2018).
- [120] NISSENBAUM, H. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press, 2020.
- [121] NOTARIO, N., CRESPO, A., MARTÍN, Y.-S., DEL ALAMO, J. M., LE MÉTAYER, D., ANTIGNAC, T., KUNG, A., KROENER, I., AND WRIGHT, D. Pripare: integrating privacy best practices into a privacy engineering methodology. In *2015 IEEE Security and Privacy Workshops* (2015), IEEE, pp. 151–158.
- [122] NOTARIO, N., CRESPO, A., MARTIN, Y. S., DEL ALAMO, J. M., METAYER, D. L., ANTIGNAC, T., KUNG, A., KROENER, I., AND WRIGHT, D. PRIPARE: Integrating privacy best practices into a privacy engineering methodology. *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015* (2015), 151–158.
- [123] OETZEL, M. C., AND SPIEKERMANN, S. A systematic methodology for privacy impact assessments: A design science approach. *European Journal of Information Systems 23*, 2 (2014), 126–150.
- [124] OLUKOYA, O. Assessing frameworks for eliciting privacy & security requirements from laws and regulations. *Computers & Security 117* (June 2022), 102697.
- [125] ORIOL, M., SEPPÄNEN, P., BEHUTIYE, W., FARRÉ, C., KOZIK, R., MARTÍNEZ-FERNÁNDEZ, S., RODRÍGUEZ, P., FRANCH, X., AARAMAA, S., ABHERVÉ, A., ET AL. Data-driven elicitation of quality requirements in agile companies. In *Quality of Information and Communications Technology: 12th International Conference, QUATIC 2019, Ciudad Real, Spain, September 11–13, 2019, Proceedings* (2019), Springer, pp. 49–63.

Bibliography

- [126] PAGE, M. J., MCKENZIE, J. E., BOSSUYT, P. M., BOUTRON, I., HOFFMANN, T. C., MULROW, C. D., SHAMSEER, L., TETZLAFF, J. M., AKL, E. A., BRENNAN, S. E., ET AL. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *International journal of surgery* 88 (2021), 105906.
- [127] PANDIT, H. J., POLLERES, A., BOS, B., BRENNAN, R., BRUEGGER, B., EKAPUTRA, F. J., FERNÁNDEZ, J. D., HAMED, R. G., KIESLING, E., LIZAR, M., SCHLEHAHN, E., STEYSKAL, S., AND WENNING, R. Creating a vocabulary for data privacy. In *On the Move to Meaningful Internet Systems: OTM 2019 Conferences* (Cham, 2019), H. Panetto, C. Debruyne, M. Hepp, D. Lewis, C. A. Ardagna, and R. Meersman, Eds., Springer International Publishing, pp. 714–730.
- [128] PATRICK, A. S., AND KENNY, S. From Privacy Legislation to Interface Design: Implementing Information Privacy in Human-Computer Interactions. In *Privacy Enhancing Technologies*, G. Goos, J. Hartmanis, J. Van Leeuwen, and R. Dingleline, Eds., vol. 2760. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 107–124. Series Title: Lecture Notes in Computer Science.
- [129] PEARSON, S., AND SHEN, Y. Context-aware privacy design pattern selection. In *International Conference on Trust, Privacy and Security in Digital Business* (2010), Springer, pp. 69–80.
- [130] PEDROZA, G., MUNTES-MULERO, V., MARTIN, Y. S., AND MOCKLY, G. A Model-based Approach to Realize Privacy and Data Protection by Design. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (Vienna, Austria, September 2021), IEEE, pp. 332–339.
- [131] PEIXOTO, M., SILVA, C., ARAÚJO, J., GORSCHKE, T., VASCONCELOS, A., AND VILELA, J. Evaluating a privacy requirements specification method by using a mixed-method approach: results and lessons learned. *Requirements Engineering* (September 2022).
- [132] PEIXOTO, M., SILVA, C., LIMA, R., ARAÚJO, J., GORSCHKE, T., AND SILVA, J. Pcm tool: Privacy requirements specification in agile software development. In *Anais Estendidos do X Congresso Brasileiro de Software: Teoria e Prática* (2019), SBC, pp. 108–113.
- [133] PEIXOTO, M. M. Privacy requirements engineering in agile software development: A specification method. *CEUR Workshop Proceedings* 2584, March (2020).
- [134] PEIXOTO, M. M., AND SILVA, C. Specifying privacy requirements with goal-oriented modeling languages. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (Sao Carlos Brazil, September 2018), ACM, pp. 112–121.
- [135] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.

- [136] PERKUSICH, M., CHAVES E SILVA, L., COSTA, A., RAMOS, F., SARAIVA, R., FREIRE, A., DILORENZO, E., DANTAS, E., SANTOS, D., GORGÔNIO, K., ALMEIDA, H., AND PERKUSICH, A. Intelligent software engineering in the context of agile software development: A systematic literature review. *Information and Software Technology* 119, February 2019 (2020), 106241.
- [137] PFITZMANN, A., AND HANSEN, M. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010.
- [138] PIRAS, L., AL-OBEIDALLAH, M. G., PAVLIDIS, M., MOURATIDIS, H., TSOHOU, A., MAGKOS, E., PRAITANO, A., IODICE, A., AND CRESPO, B. G.-N. DEFEND DSM: A Data Scope Management Service for Model-Based Privacy by Design GDPR Compliance. In *Trust, Privacy and Security in Digital Business*, S. Gritzalis, E. R. Weippl, G. Kotsis, A. M. Tjoa, and I. Khalil, Eds., vol. 12395. Springer International Publishing, Cham, 2020, pp. 186–201. Series Title: Lecture Notes in Computer Science.
- [139] PIRAS, L., AL-OBEIDALLAH, M. G., PRAITANO, A., TSOHOU, A., MOURATIDIS, H., GALLEGU-NICASIO CRESPO, B., BERNARD, J. B., FIORANI, M., MAGKOS, E., SANZ, A. C., ET AL. Defend architecture: a privacy by design platform for gdpr compliance. In *Trust, Privacy and Security in Digital Business: 16th International Conference, TrustBus 2019, Linz, Austria, August 26–29, 2019, Proceedings 16* (2019), Springer, pp. 78–93.
- [140] PLAPPERT, C., STANCKE, J., AND JAGER, L. Towards a Privacy-Aware Electric Vehicle Architecture. In *2022 30th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (Valladolid, Spain, March 2022), IEEE, pp. 265–274.
- [141] QIN, T., AND LIU, T. Introducing LETOR 4.0 datasets. *CoRR abs/1306.2597* (2013).
- [142] RADICS, P. J., GRACANIN, D., AND KAFURA, D. Preprocess before You Build: Introducing a Framework for Privacy Requirements Engineering. In *2013 International Conference on Social Computing* (Alexandria, VA, USA, September 2013), IEEE, pp. 564–569.
- [143] RAHARJANA, I. K., SIAHAAN, D., AND FATICHAH, C. User Stories and Natural Language Processing: A Systematic Literature Review. *IEEE Access* 9 (2021), 53811–53826.
- [144] RAMESH, B., CAO, L., AND BASKERVILLE, R. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal* 20, 5 (2010), 449–480.

Bibliography

- [145] RAMOS, F. B. A., COSTA, A. A. M., PERKUSICH, M. B., ALMEIDA, H. O., AND PERKUSICH, A. A non-functional requirements recommendation system for scrum-based projects. In *SEKE* (2018), pp. 149–148.
- [146] RAUF, I., PETRE, M., TUN, T., LOPEZ, T., LUNN, P., VAN DER LINDEN, D., TOWSE, J., SHARP, H., LEVINE, M., RASHID, A., ET AL. The case for adaptive security interventions. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 1 (2021), 1–52.
- [147] REIMERS, N., AND GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (11 2019), Association for Computational Linguistics.
- [148] RINGMANN, S. D., LANGWEG, H., AND WALDVOGEL, M. Requirements for Legally Compliant Software Based on the GDPR. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, H. Panetto, C. Debruyne, H. A. Proper, C. A. Ardagna, D. Roman, and R. Meersman, Eds., vol. 11230. Springer International Publishing, Cham, 2018, pp. 258–276. Series Title: Lecture Notes in Computer Science.
- [149] ROSENBERG, D., AND SCOTT, K. Use case driven object modeling with uml, 1999.
- [150] RYGGE, H., AND JØSANG, A. Threat Poker: Solving Security and Privacy Threats in Agile Software Development. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11252 LNCS, November (2018), 468–483.
- [151] SALNITRI, M., ANGELOPOULOS, K., PAVLIDIS, M., DIAMANTOPOULOU, V., MOURATIDIS, H., AND GIORGINI, P. Modelling the interplay of security, privacy and trust in sociotechnical systems: a computer-aided design approach. *Software and Systems Modeling* 19, 2 (March 2020), 467–491.
- [152] SAMONAS, S., AND COSS, D. The cia strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security* 10, 3 (2014).
- [153] SCHÖN, E.-M., WINTER, D., ESCALONA, M. J., AND THOMASCHEWSKI, J. Key challenges in agile requirements engineering. In *Agile Processes in Software Engineering and Extreme Programming: 18th International Conference, XP 2017, Cologne, Germany, May 22-26, 2017, Proceedings* 18 (2017), Springer International Publishing, pp. 37–51.
- [154] SHOSTACK, A. Experiences threat modeling at Microsoft. *CEUR Workshop Proceedings* 413 (2008), 1–11.
- [155] SIELIS, G. A., TZANAVARI, A., AND PAPADOPOULOS, G. A. Archreco: a software tool to assist software design based on context aware recommendations of design

- patterns. *Journal of Software Engineering Research and Development* 5 (2017), 1–36.
- [156] SPIEKERMANN, S., AND CRANOR, L. Engineering Privacy. *IEEE Transactions on Software Engineering* 35, 1 (January 2009), 67–82.
- [157] STACH, C., AND STEIMLE, F. Recommender-based privacy requirements elicitation - EPICUREAN: an approach to simplify privacy settings in IoT applications with respect to the GDPR. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (Limassol Cyprus, April 2019), ACM, pp. 1500–1507.
- [158] STEFANOVA, E., AND DIMOV, A. Privacy Enabled Software Architecture. In *Business Modeling and Software Design*, B. Shishkov, Ed., vol. 422. Springer International Publishing, Cham, 2021, pp. 190–206. Series Title: Lecture Notes in Business Information Processing.
- [159] SWIDERSKI, F., AND SNYDER, W. *Threat modeling*. Microsoft Press, 2004.
- [160] TAHAEI, M., FRIK, A., AND VANIEA, K. Privacy champions in software teams: Understanding their motivations, strategies, and challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–15.
- [161] TAKEDA, H., VEERKAMP, P., AND YOSHIKAWA, H. Modeling design process. *AI magazine* 11, 4 (1990), 37–37.
- [162] TEIXEIRA, C., VASCONCELOS, A., SOUSA, P., AND MARQUES, M. Enterprise Architecture Patterns for GDPR Compliance:. In *Proceedings of the 23rd International Conference on Enterprise Information Systems* (Online Streaming, — Select a Country —, 2021), SCITEPRESS - Science and Technology Publications, pp. 715–725.
- [163] TOVAL, A., OLMOS, A., AND PIATTINI, M. Legal requirements reuse: a critical success factor for requirements quality and personal data protection. In *Proceedings IEEE Joint International Conference on Requirements Engineering* (Essen, Germany, 2002), IEEE Comput. Soc, pp. 95–103.
- [164] TSOHOU, A., MAGKOS, M., MOURATIDIS, H., CHRYSOLORAS, G., PIRAS, L., PAVLIDIS, M., DEBUSSCHE, J., ROTOLONI, M., AND CRESPO, B. G.-N. Privacy, Security, Legal and Technology Acceptance Requirements for a GDPR Compliance Platform. In *Computer Security*, S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, F. Pallas, J. Pohle, A. Sasse, W. Meng, S. Furnell, and J. Garcia-Alfaro, Eds., vol. 11980. Springer International Publishing, Cham, 2020, pp. 204–223. Series Title: Lecture Notes in Computer Science.
- [165] VAISHNAVI, V. K. *Design science research methods and patterns: innovating information and communication technology*. Auerbach Publications, 2007.

Bibliography

- [166] VAN LANDUYT, D., PASQUALE, L., SION, L., AND JOOSEN, W. Threat modeling at run time: The case for reflective and adaptive threat management (NIER track). *Proceedings - 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2021* (2021), 203–209.
- [167] VICINI, S., ALBERTI, F., NOTARIO, N., CRESPO, A., PASTORIZA, J. R. T., AND SANNA, A. Co-creating Security-and-Privacy-by-Design Systems. In *2016 11th International Conference on Availability, Reliability and Security (ARES)* (Salzburg, Austria, August 2016), IEEE, pp. 768–775.
- [168] WANG, X., ZHAO, L., WANG, Y., AND SUN, J. The Role of Requirements Engineering Practices in Agile Development: An Empirical Study. In *Requirements Engineering* (Berlin, Heidelberg, 2014), D. Zowghi and Z. Jin, Eds., Springer Berlin Heidelberg, pp. 195–209.
- [169] WEI, J., AND ZOU, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, November 2019), Association for Computational Linguistics, pp. 6382–6388.
- [170] WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M. C., REGNELL, B., AND WESSLÉN, A. Experimentation in software engineering, 2012.
- [171] WUYTS, K., SION, L., AND JOOSEN, W. LINDDUN GO: A Lightweight Approach to Privacy Threat Modeling. *Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020* (2020), 302–309.
- [172] XIE, T. The synergy of human and artificial intelligence in software engineering. In *2013 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)* (2013), IEEE, pp. 4–6.
- [173] XIE, T. Intelligent Software Engineering: Synergy Between AI and Software Engineering. In *Dependable Software Engineering. Theories, Tools, and Applications* (Cham, 2018), X. Feng, M. Müller-Olm, and Z. Yang, Eds., Springer International Publishing, pp. 3–7.
- [174] YE, X., ZHU, Z., PENG, Y., AND XIE, F. Privacy Aware Engineering: A Case Study. *JOURNAL OF SOFTWARE* 4, 3 (2009).
- [175] ZHANG, B., LIANG, P., ZHOU, X., AHMAD, A., AND WASEEM, M. Practices and challenges of using github copilot: An empirical study. *arXiv preprint arXiv:2303.08733* (2023).
- [176] ZHAO, L., ALHOSHAN, W., FERRARI, A., LETSHOLO, K. J., AJAGBE, M. A., CHIOASCA, E. V., AND BATISTA-NAVARRO, R. T. Natural Language Processing for Requirements Engineering. *ACM Computing Surveys* 54, 3 (2021), 1–41.

- [177] ZIBUSCHKA, J. Analysis of automation potentials in privacy impact assessment processes. In *Computer Security: ESORICS 2019 International Workshops, Cyber-ICPS, SECPRE, SPOSE, and ADIoT, Luxembourg City, Luxembourg, September 26–27, 2019 Revised Selected Papers 5* (2020), Springer, pp. 279–286.
- [178] ZIBUSCHKA, J., AND ZIMMERMANN, C. Lean privacy by design. *SICHERHEIT 2020* (2020).
- [179] ZIMMERMANN, C. Automation potentials in privacy engineering.

Acronyms

- ASD** Agile Software Development. 4, 16, 61, 62, 64, 65, 67, 77, 94
- BERT** Bidirectional Encoder Representations from Transformers. 70, 86–88
- CIA** Confidentiality, Integrity, Availability. 14
- DFD** Data Flow Diagram. ix, 88
- DPV** Data Privacy Vocabulary. 69
- GDPR** General Data Protection Regulation. 21, 24, 25, 31, 32, 53–57, 61, 68, 69, 80, 99, 125, 131
- IS** Intelligent System. 4–7, 62
- ISE** Intelligent Software Engineering. 4
- LLM** Large Language Model. ix, xvi, 75, 76, 90–94, 96, 97, 162
- NER** Named Entity Recognition. 69, 70, 73–75, 84, 86–90, 95, 115, 118, 127, 128, 157, 159
- NLP** Natural Language Processing. 161
- NLP4RE** Natural Language Processing for Requirements Engineering. 161
- OECD** Organisation for Economic Co-operation and Development. 32, 33
- PET** Privacy Enhancing Technologies. 39–43, 45, 99, 100
- PIA** Privacy Impact Assessment. 15, 35, 36, 52, 61, 64
- PrDP** Privacy Design Pattern. xvi, 99–109, 111, 112
- PRE** Privacy Requirements Engineering. viii, 5, 14–17, 61–67, 69–71, 73, 75, 77, 79, 81, 83, 84, 94, 95, 97, 100, 102, 160, 161
- RoBERTa** Robustly Optimized BERT Pretraining Approach. 86–88

A. Appendix

A.1. Practitioners Feedback Form

1. How easily can you understand and use the functions and features of "Generate Entity"? (Likert scale)
2. In your opinion, how much effort is required to use "Generate Entity" feature effectively? (Likert scale)
3. I believe that the "Generate Entity" feature is important for the privacy-by-design process. (Likert scale)
4. Do you think automatic entity recognition saves you time in identifying privacy-related entities (compared to doing it manually)? (Likert scale)
5. Does the "Generate Entity" feature result in entities that you can use to identify privacy risks or requirements? (Likert scale)
6. Assuming I have access to the "Generate Entity" feature, I intend to use it for identifying privacy-related entities. (Likert scale)
7. Were there any entities that the system missed or misclassified? (Yes/No with follow-up)
8. What are the entities you think are being misclassified? (Open-ended)
9. Explain any areas of the "Generate Entity" that were unclear or confusing. (If any) (Open-ended)
10. How easily can you understand and use the functions and features of "Generate Data Flow Diagram"? (Likert scale)
11. In your opinion, how much effort is required to use "Generate Data Flow Diagram" feature effectively? (Likert scale)
12. I believe that the "Generate Data Flow Diagram" feature is important for the privacy-by-design process. (Likert scale)
13. How easy was it to understand and interpret the generated Data Flow Diagrams? (Likert scale)
14. Do you think the automatic generation of the DFD will save you time when conducting threat modeling (compared to manually drawing it)? (Likert scale)

A. Appendix

15. Explain any areas of the Data Flow Diagram that were unclear or confusing. (If any) (Open-ended)
16. Given that I have access to the "Generate Data Flow Diagram" feature, I predict that I would use it in the privacy-by-design process. (Likert scale)
17. Are there any additional features or details you would like to see in the Data Flow Diagram to assist with understanding or analysis? (Open-ended)
18. How easily can you understand and use the functions and features of "Guided Threat Modeling"? (Likert scale)
19. In your opinion, how much effort is required to use "Guided Threat Modeling" feature effectively? (Likert scale)
20. Do you think the guided threat modeling will save you time and effort when conducting threat modeling? (Likert scale)
21. Did the threat identification process using the "threat question" help identify potential threats that might otherwise have been overlooked? (Likert scale)
22. How comprehensive was the threat modeling process with the use of triples in capturing potential privacy threats? (Likert scale)
23. Given that I have access to the "Guided Threat Modeling" feature, I predict that I would use it in the privacy-by-design process. (Likert scale)
24. The user story template made the process of generating privacy requirements more efficient. (Likert scale)
25. I found the user story template to be a useful tool for generating privacy requirements. (Likert scale)
26. Given that I have access to the "Privacy Requirements Template", I predict that I would use it in the privacy-by-design process. (Likert scale)
27. What aspects of the user story template did you find most helpful for generating privacy requirements? (Open-ended)
28. Were there any parts of the user story template that you found confusing or less useful? If so, please explain. (Open-ended)
29. How easily can you understand and use the functions and features of "Solution Recommendation"? (Likert scale)
30. In your opinion, how much effort is required to use "Solution Recommendation" feature effectively? (Likert scale)
31. How relevant were the recommended privacy design patterns to your project's needs? (Likert scale)

A.1. Practitioners Feedback Form

32. How actionable were the design pattern recommendations? (Likert scale)
33. Given that I have access to the "Solution Recommendation" feature, I predict that I would use it in the privacy-by-design process. (Likert scale)
34. Were there any design patterns you expected to see but were not recommended? Please explain. (Open-ended)
35. From the functionalities you've experienced, which feature or aspect of the tool did you find most useful? (Multiple choice)
36. Can you provide a brief explanation regarding why you found this functionality particularly beneficial? (Open-ended)
37. To what extent do you feel the need for extensive training to use this tool effectively? (Multiple choice)
38. To what extent do you believe this tool could be used in agile software development? (Likert scale)
39. Please specify your reason. (Open-ended)
40. I would adopt this tool to help me elicit privacy requirements in my own project. (Likert scale)
41. Are there any enhancements or changes you'd suggest to make it even more effective for your needs? (Open-ended)