

# A Framework Managing Conflicts between Security and Privacy Requirements

Duaa Alkubaisy

A thesis submitted in partial fulfilment  
of the requirements of the University of Brighton  
for the degree of Doctor of Philosophy

March 2021

# ABSTRACT

Conflicting requirements are the key reasons for inconsistencies in software development. Privacy and security requirements, and their potential conflicts, are increasingly becoming more important to software development. Over the last few years, this has become formalised and required by law. A relevant example is the case of the General Data Protection Regulation (GDPR), which requires organisations and their software engineers to enforce and guarantee privacy-by-design to make their platforms compliant.

A thorough literature review revealed that there does not exist a comprehensive requirement engineering-oriented tool for supporting users in identifying conflicts between privacy and security requirements. To fill this gap, this research aims to address the problem of identifying and mitigating conflicts between security and privacy requirements. The research designs ConfIS; a three-phrase semi-automated framework which identifies, analyses and resolves conflict between security and privacy requirements. The proposed framework is implemented using Secure Tropos, a CASE Tool for Modelling Security in Requirements Engineering.

To achieve a comprehensive evaluation, we designed a focus group session, including participants who are both experts and researchers. They applied ConfIS framework to a realistic example from DEFEND, an EU project aiming at supporting organisations in achieving GDPR compliance. Findings revealed that over 80% found the framework to be very supportive; 87% agreed that mapping between security and privacy for identifying conflict was clear and easy to follow very detailed steps. Additionally, 86% agreed that the framework adequately identified conflicts between requirements, and 77% agreed that the framework supported in understanding conflict resolutions' patterns and its supporting tools.

Through the use of this framework, conflicts can be identified at an early stage of the development process and remedied, thereby reducing development costs. Therefore, this framework builds on existing research by identifying the relevant resolution tools to identify and mitigate conflicts between security and privacy requirements.

# TABLE OF CONTENTS

<b>ABSTRACT</b> .....	<b>i</b>
<b>LIST OF FIGURES</b> .....	<b>VII</b>
<b>LIST OF TABLES</b> .....	<b>IX</b>
<b>DEDICATION</b> .....	<b>X</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>xi</b>
<b>DECLARATION</b> .....	<b>xii</b>
<b>ACRONYMS</b> .....	<b>xiii</b>
<b>CHAPTER 1</b> .....	<b>1</b>
<b>INTRODUCTION</b> .....	<b>1</b>
<b>1.1 Introduction</b> .....	<b>1</b>
<b>1.2 Motivation</b> .....	<b>2</b>
<b>1.3 Aims and Objectives</b> .....	<b>4</b>
<b>1.4 Research Questions</b> .....	<b>6</b>
<b>1.5 Contribution to Knowledge</b> .....	<b>9</b>
<b>1.6 Overall Thesis Structure</b> .....	<b>10</b>
<b>1.7 Published Work</b> .....	<b>11</b>
<b>CHAPTER 2</b> .....	<b>13</b>
<b>2.1. Introduction</b> .....	<b>13</b>
<b>2.2 Background on Software and Requirements Engineering</b> .....	<b>14</b>
<b>2.3. Security engineering framework</b> .....	<b>16</b>
<b>2.3-1. OBJECT-LEVEL MODELLING</b> .....	<b>19</b>
<b>2.3-2. META-LEVEL MODELLING</b> .....	<b>19</b>
<b>2.3-3. LIMITATIONS OF EARLIER SECURITY METHODS</b> .....	<b>26</b>
<b>2.4. Modelling Language</b> .....	<b>28</b>
<b>2.4-1. GOAL MODEL (GM)</b> .....	<b>28</b>
<b>2.4-2. I*</b> .....	<b>30</b>
<b>2.4-3. SECURE TROPOS</b> .....	<b>31</b>
<b>2.4.3.1 SECURE TROPOS METHODOLOGY</b> .....	<b>32</b>
<b>2.4-4. BUSINESS PROCESS MODEL AND NOTATION (BPMN)</b> .....	<b>37</b>
<b>2.5. The Importance of User Privacy</b> .....	<b>37</b>
<b>2.5-1. GENERAL DATA PROTECTION REGULATION (GDPR)</b> .....	<b>40</b>
<b>2.5-2. SECURITY AND PRIVACY REQUIREMENTS IN CLOUD COMPUTING</b> .....	<b>41</b>
<b>2.5-3. ESSENTIAL PRINCIPLES OF PRIVACY</b> .....	<b>45</b>

2.5-4.	<b>ROLE OF PRIVACY BY DESIGN PRINCIPLES.....</b>	<b>46</b>
2.5-5.	<b>PRIVACY FRAMEWORKS .....</b>	<b>48</b>
2.5-6.	<b>LIMITATIONS OF PREVIOUS PRIVACY FRAMEWORKS.....</b>	<b>53</b>
2.6	<b>Conflict.....</b>	<b>54</b>
2.6-1.	<b>DEFINITION .....</b>	<b>54</b>
2.6-2.	<b>CONFLICTING REQUIREMENTS .....</b>	<b>55</b>
2.6-3.	<b>CAUSES OF CONFLICTING REQUIREMENTS.....</b>	<b>57</b>
2.6-4.	<b>CONFLICT IDENTIFICATION, ANALYSIS AND MITIGATE APPROACHES.....</b>	<b>60</b>
2.6-5.	<b>COMPARISON BETWEEN EXISTING WORKS IN CONFLICT REQUIREMENTS .....</b>	<b>63</b>
2.6-6.	<b>DEALING WITH CONFLICTS BETWEEN REQUIREMENTS.....</b>	<b>72</b>
2.6-7.	<b>REQUIREMENT NEGOTIATION .....</b>	<b>73</b>
2.6-8.	<b>REQUIREMENT PRIORITISATION .....</b>	<b>76</b>
2.6-8-1	<b>REQUIREMENTS PRIORITISATION TECHNIQUES.....</b>	<b>77</b>
2.7	<b>Chapter Summary .....</b>	<b>79</b>
<b>CHAPTER 3 .....</b>		<b>82</b>
3.1	<b>Overview of Research Methods .....</b>	<b>82</b>
3.2	<b>Research Approaches .....</b>	<b>84</b>
3.2-1	<b>STRATEGIES RELATED WITH THE QUALITATIVE APPROACH.....</b>	<b>85</b>
3.2-1-1	<b>CASE STUDY AS A RESEARCH METHOD.....</b>	<b>86</b>
3.2-1-2	<b>CATEGORIES OF A CASE STUDY .....</b>	<b>88</b>
3.2-1-3	<b>LIMITATIONS OF A CASE STUDY.....</b>	<b>89</b>
3.3	<b>Validity of the Framework.....</b>	<b>89</b>
3.4	<b>Research Process .....</b>	<b>90</b>
I.	<b>FORMULATING THE RESEARCH PROBLEM.....</b>	<b>91</b>
II.	<b>REVIEWING THE LITERATURE .....</b>	<b>91</b>
III.	<b>FORMULATING THE RESEARCH QUESTION(S).....</b>	<b>92</b>
IV.	<b>PREPARING THE FRAMEWORK DESIGN.....</b>	<b>92</b>
V.	<b>FRAMEWORK PROCESSES .....</b>	<b>93</b>
VI.	<b>EVALUATING THE FRAMEWORK .....</b>	<b>94</b>
a.	<b>Focus Groups in person.....</b>	<b>94</b>
b.	<b>Online focus groups .....</b>	<b>95</b>
c.	<b>Steps in Focus Group Research .....</b>	<b>96</b>
3.5	<b>Defining the Research Problem .....</b>	<b>96</b>
3.6	<b>Planning the focus group session .....</b>	<b>97</b>
3.7	<b>Selecting participants.....</b>	<b>97</b>
3.8	<b>Conducting the focus group session .....</b>	<b>98</b>

3.9	Analysing and Interpreting Data.....	98
3.10	Report Results .....	101
3.11	Chapter Summary .....	102
<b>CHAPTER 4.....</b>		<b>104</b>
4.1.	Introduction.....	104
4.2	Theoretical Framework Phases .....	105
4.2-1	HOW THE FRAMEWORK WORKS: .....	107
4.2-1-1	PHASE 1: IDENTIFY REQUIREMENTS: (SECURITY AND PRIVACY REQUIREMENTS) .....	107
4.2-1-2	PHASE 2: IDENTIFY CONFLICTS BETWEEN REQUIREMENTS AND CONFLICT DECISIONS.....	113
4.2-1-3	PHASE 3: ANALYSIS CONFLICTS BASED ON SUPPORT TECHNIQUES.....	115
4.3	Chapter Summary .....	126
<b>CHAPTER 5.....</b>		<b>127</b>
5.1	Introduction.....	127
5.2	Method supported by the model.....	127
5.3	Conflict Resolution Model.....	146
5.4	Chapter Summary .....	153
<b>CHAPTER 6.....</b>		<b>154</b>
6.1	Introduction.....	154
6.2	Secure Tropos Framework.....	155
6.3	E-health Scenario .....	157
6.4	Applying ConfIS framework to E-health scenario .....	158
6.4.1	PHASE 1: IDENTIFY REQUIREMENTS: (SECURITY AND PRIVACY REQUIREMENTS).....	158
6.4.2	PHASE 2: IDENTIFY CONFLICTS BETWEEN REQUIREMENTS:.....	161
6.4.3	PHASE 3: RESOLVE CONFLICTS BASED ON SUPPORT TECHNIQUES: .....	166
6.5	Chapter Summary .....	172
<b>CHAPTER 7.....</b>		<b>173</b>
7.1	Introduction.....	173
7.2	Data: Privacy and Security within DEFEND.....	174
7.3	DEFEND Project.....	174
7.3.1	THEMES AND SERVICES:.....	177
7.4	Data Scope Management.....	179
7.4-1	ACTIVITIES AND STRATEGIES (AS) FOR PBD .....	179
7.4-2	MODEL DSM THEMES WITH RELATED REQUIREMENTS.....	181
7.5	Data Scope Management (DSM) storyline .....	181

7.5-1	LINKING SCENARIOS WITH ASSOCIATED REQUIREMENTS.....	183
7.5-2	ASSIGNING APPROPRIATE TOOLS TO PRIVACY AND SECURITY REQUIREMENTS IN THE.....	187
7.5-3	IDENTIFYING CONFLICTS BETWEEN REQUIREMENTS.....	192
7.6	Applying the ConfIS framework to case study example .....	199
7.6-1	PHASE 2: IDENTIFY CONFLICTS BETWEEN REQUIREMENTS AND CONFLICT DECISIONS....	202
7.6-2	PHASE 3: CONFLICT RESOLUTION.....	205
7.6-3	DISCUSSION.....	208
7.7	Benefits of applying the ConfIS framework within DEFEND boundaries .....	211
7.8	Chapter Summary .....	212
<b>CHAPTER 8 .....</b>		<b>213</b>
8.1	INTRODUCTION .....	213
8.2	ETHICS IN EVALUATION .....	214
8.2.1	ETHICS REVIEW AND DATA MANAGEMENT PLAN .....	215
8.2.2	CONSENT DOCUMENTATION .....	216
8.3	PRELIMINARY EVALUATION.....	216
8.4	ACTUAL EVALUATION .....	218
8.4.1	EVALUATION STRATEGY.....	219
8.4.2	EVALUATION RESULTS .....	220
8.5	EVALUATION METHODS.....	233
8.6	THEMATIC ANALYSIS .....	234
8.6.1	THEMATIC ANALYSIS STEPS .....	239
8.6.1-1	FAMILIARISATION .....	239
8.6.1-2	CODING.....	239
8.6.1-3	GENERATING AND REVIEWING THEMES .....	250
8.6.1-4	SYNOPSIS.....	264
8.7	APPLICATION TO CONFIS FRAMEWORK.....	269
8.8	SUMMARY.....	271
<b>CHAPTER 9 .....</b>		<b>272</b>
9.1	INTRODUCTION .....	272
9.2	RESEARCH AIMS REVISITED .....	272
9.3	ISSUES ARISING FROM CONDUCTING THE RESEARCH.....	273
9.4	LIMITATIONS.....	275
9.4.1	LIST OF REQUIREMENTS NOT EXHAUSTIVE.....	275
9.4.2	SIDE- EFFECTS OF DEPLOYING FRAMEWORK AND RESOLUTION STRATEGIES .....	276
9.5	CONTRIBUTION .....	276

<b>9.6</b>	<b>FUTURE WORK .....</b>	<b>279</b>
	<b>REFERENCES.....</b>	<b>283</b>
	<b>APPENDICES .....</b>	<b>302</b>
	<b>APPENDIX A: LIST OF REQUIREMENTS FOR E-HEALTH SCENARIO .....</b>	<b>302</b>
	<b>APPENDIX B: ETHICAL APPROVAL.....</b>	<b>305</b>
	<b>APPENDIX C: EVALUATION FORM .....</b>	<b>306</b>
	<b>APPENDIX D: TOOLKIT FOR THE FOCUS GROUP SESSION.....</b>	<b>309</b>
	<b>PHASE 1: MAPPING SECURITY AND PRIVACY REQUIREMENTS .....</b>	<b>312</b>
	<b>PHASE 2: IDENTIFY CONFLICTS BETWEEN REQUIREMENTS AND CONFLICT DECISIONS:.....</b>	<b>314</b>
	<b>PHASE 3: CONFLICT RESOLUTION PATTERNS.....</b>	<b>321</b>

# LIST OF FIGURES

Figure 1.1 Thesis Structure .....	12
Figure 2.1 Literature Review Map- Security and Privacy Issues .....	14
Figure 2.2 Conflict approaches dealing with Non-functional requirements.....	70
Figure 2.3 Two-Dimensional Model of Conflict .....	74
Figure 2.4 Requirements Prioritization Techniques .....	79
Figure 3.1 Research Process .....	91
Figure 4.1 Phases of the Theoretical Framework .....	105
Figure 4.2 Detecting conflicts between Security and Privacy Requirements Venn Diagram	113
Figure 5.1 EPOS Night Club .....	133
Figure 5.2 Health Insurance.....	135
Figure 5.3 Greek National Gazette .....	137
Figure 5.4 Customer Relationship Management .....	139
Figure 5.5 UoB Records Management system .....	141
Figure 5.6 Healthcare Management.....	143
Figure 5.7 Conflict Resolution Model .....	147
Figure 5.8 Confidentiality conflicts with privacy requirements .....	149
Figure 5.9 Integrity conflicts with privacy requirements.....	150
Figure 5.10 Binding of duties conflicts with privacy requirements.....	150
Figure 5.11 Availability conflicts with privacy requirements .....	150
Figure 5.12 Non-repudiation conflicts with privacy requirements .....	151
Figure 5.13 Accountability conflicts with privacy requirements.....	151
Figure 5.14 Authentication conflicts with privacy requirements.....	152
Figure 6.1 Organisational View .....	159
Figure 6.2 Security Requirements View .....	160
Figure 6.3 Patient access to measure vital .....	161
Figure 6.4 Tele-medicine Sending data to system portal.....	162
Figure 6.5 Completing the evaluation form.....	163
Figure 6.6 Submitting the evaluation form .....	164
Figure 6.7 Sending the submitted evaluation form.....	165
Figure 7.1 Organisational View of Managing Patient Records .....	201
Figure 7.2 Privacy by Design View of Managing Patient Record.....	204
Figure 7.3 Accountability conflicts anonymity .....	205
Figure 7.4 Adding the Supporting Tool in Privacy Pattern Library .....	209
Figure 7.5 Integrating Conflict Resolution in Privacy-by-Design view .....	210
Figure 8.1 Survey Respondents .....	221
Figure 8.2 Research Design Questions .....	222
Figure 8.3 Research Design Per Respondent Group.....	224
Figure 8.4 General Framework .....	225
Figure 8.5 General Framework Per Respondent Group.....	226
Figure 8.6 ConfIS Framework Phases and Survey Questions .....	228
Figure 8.7 ConfIS Framework Phase 1 Per Respondent Group .....	229
Figure 8.8 ConfIS Framework Phase 2 Per Respondent Group .....	230
Figure 8.9 ConfIS Framework Phase 3 Per Respondent Group .....	230



Figure 8.10 ConfIS Framework and Focus Group Response using Ranking Evaluation	
Method .....	232
Figure 9.1 Conflict notation in Privacy by Design View .....	278

## LIST OF TABLES

Table 2.1 Concept Types on Secure Tropos methodology .....	33
Table 2.2 Relationship Types on Secure Tropos methodology .....	35
Table 2.3 Comparison between conflict analysis approach.....	69
Table 4.1 Most Frequent Security and Privacy Requirements being in Conflict .....	109
Table 4.2 Mapping conflicts between Security and Privacy Requirements .....	112
Table 4.3 Supporting Tools.....	116
Table 4.4 Conflict Cases and Likelihood of Tools .....	124
Table 4.5 Techniques suitable for both Security and Privacy Requirements .....	124
Table 4.6 Techniques suitable for Privacy Requirements .....	125
Table 4.7 Techniques suitable for Security Requirements .....	125
Table 5.1 Security Requirements Conflicts with some Privacy Requirements .....	129
Table 5.2 Literature Review – Conflict Requirements.....	130
Table 5.3 Security and Privacy requirements for EPOS Night Club.....	132
Table 5.4 Security and Privacy requirements for Health Insurance.....	134
Table 5.5 Security and Privacy requirements for Greek National Gazette .....	136
Table 5.6 Security and Privacy requirements for Customer Relationship Management .....	138
Table 5.7 Security and Privacy requirements for UoB Records Management system .....	140
Table 5.8 Security and Privacy requirements for Healthcare Management .....	143
Table 5.9 Security and Privacy requirements for ‘Supporting the design of privacy-aware. ....	144
Table 5.10 Conflicts between intrusion detection and privacy mechanisms .....	146
Table 6.1 Conflicting Requirements (security/privacy).....	165
Table 6.2 Unlinkability vs BOD .....	167
Table 6.3 Anonymity vs Accountability .....	168
Table 6.4 Anonymity vs Non-repudiation .....	169
Table 6.5 Confidentiality vs Unobservability .....	170
Table 6.6 Confidentiality vs Undetectability .....	171
Table 7.1 GDPR 12 Theme Obligations .....	178
Table 7.2 GDPR Service Description and Related Themes .....	178
Table 7.3 Identifying Privacy/Security Requirements in Medical Scenario.....	183
Table 7.4 Identify Requirements and Tools to Mitigate Conflict.....	190
Table 7.5 Privacy/Security Requirements and Supporting Tools to Mitigate Conflict .....	195
Table 7.6 Example- Phase 1: Mapping Security and Privacy Requirements .....	199
Table 7.7 Identifying Requirements for each Scenario .....	200
Table 8.1 ConfIS Framework Phases and Survey Responses.....	228
Table 8.2 Advantages of Thematic Analysis .....	235
Table 8.3 Coding Participant Responses – Evaluating ConfIS Framework.....	240
Table 8.4 Interview Extracts – Evaluating <i>ConfIS</i> Framework.....	241
Table 8.5 Turning Codes into Themes.....	250
Table 8.6 Interview Extracts: Turning codes into Themes .....	260
Table 8.7 Ranking Themes: Pre-Framework .....	267
Table 8.8 Ranking Themes: Post-Framework .....	269
Table 8.9 Generic List of Common Security and Privacy Requirements.....	270

## DEDICATION

*" To my father  
to whom I promised to dedicate  
this thesis before he left this world"*

## ACKNOWLEDGEMENTS

I would like to thank everyone who provided me with support, guidance and encouragement throughout my PhD journey.

First of all, I would like to thank my supervisors, Dr Karl Cox and Prof. Haralambos Mouratidis, for their continued guidance. Their insights and expertise were invaluable in shaping this thesis.

Co-supervisor Dr Luca Piras also played an important role in my research work, providing further insights and supporting me during my journey. For his helpful ideas and motivating suggestions, I am deeply grateful.

To Christos Kalloniatis, for believing in me, supporting me even in my weakest time. He inspired me and gave me confidence to believe in myself.

Furthermore, I am thankful to The Ministry of Higher Education in Saudi Arabia, and Emam bin Abdulrahman University for their financial sponsorship of my PhD journey in the United Kingdom.

I wish to thank my parents who have provided me with the ideal environment to grow up in, and from whom I have inherited my determination and strength.

All my gratitude to my Mom Fawziyah and Sister Dina; the support they have provided me over the years was the greatest gift anyone has ever given me. Without their love, patience, support and understanding, all the way from Saudi Arabia, accomplishing my PhD would have not been possible today.

To all my friends, thank you for your understanding and encouragement in my many, many moments of crisis. Your friendship makes my life a wonderful experience. I cannot list all your names here, but you are always on my mind.

Last but not least, I would like to thank all the participants who contributed to my research. For their invaluable effort, time, and precious input, which helped me to accomplish my objectives. I dedicate this PhD to my beloved son Hamad, who I know will achieve valuable goals in his life and make me proud, as I trust I have made him proud.

Finally, to my Dad, this is dedicated to you. You have been a tremendous support from the very beginning; and though you have not seen the end of my thesis journey, every page is written with you in my heart.

# DECLARATION

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

Signed



Dated

March 2021

# ACRONYMS

**BPMN** Business Process Modeling Notation

**Confab** privacy-sensitive ubiquitous computing applications

**CPS** Cyber-Physical Systems

**DPbDD** Data Protection by Design and by Default

**DPIA** Data Protection Impact Assessments

**EDPB** European Data Protection Board

**EKD** Enterprise Knowledge Development

**eSAP** electronic Single Assessment Process

**GDPR** General Data Protection Regulation

**GORE** Goal-oriented Requirements Engineering

**IOI** Item of Interest

**ICT** Information and Communication Technology

**IoT** Internet of Things

**IS** Information System

**ISSRM** Information Systems Security Risks Management

**ISO** International Organisation for Standardisation

**KAOS** Keep All Objectives Satisfied

**NFR** Non-Functional Requirements

**RAVE** Ravenscroft Audio Video Environment

**RBAC** Role-Based Access Control

**RE** Requirements Engineering

**PARCTab** Partnership Against Cancer

**PbD** Privacy by Design

**PET** Privacy Enhancing Technologies

**PriS** Incorporating Privacy Requirements into the System Design Process

**SecTro** Security Requirements Engineering CASE tool

**SDLC** Software Development Life Cycle

**STRAP** Structured Requirements Analysis Planning

**SQUARE** Security Quality Requirements Engineering method

**UML** Unified Modeling Language



# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Requirements Engineering (RE) is the process of eliciting, evaluating, specifying, consolidating and changing the objectives, functionalities, qualities and constraints to be achieved by a software-intensive system (van Lamsweerde, 2009). Over the years, RE research has been driven by the urgent need for methodical RE in large software projects. Still, the environment in which RE is practised has changed dramatically (Jarke *et al.*, 2011) over the past 20 years as a result of various almost simultaneous reasons: delivery platforms are changing (mobile, cloud, social); communication and collaboration channels are being renovated (Internet, mobile, social); the consumer world of technology is driving innovation; and data is opening up and overflowing out of the growing apps, devices and sensors deployed by or connected with organisations (Sherief, 2017).

The field's focus and scope has moved from the engineering of individual systems and components towards the generation and adaptation of software intensive ecosystems. This shift has created a strong need to understand more deeply the issues that underlie current RE, and reconsider RE practices and methods to meet these new challenges. Currently, requirements engineering is one of the most challenging fields in software development, has the most impact on project success, and is a major issue for decision-makers in enterprises (Jarke *et al.*, 2011). Developing complex software systems raises a large number of needs, wishes and requirements that are – due to differing viewpoints and stakeholders – often in conflict with each other (Siegemund, 2015).



Furthermore, the security policies made by different developer may disagree or even conflict with one another. Such disagreements or conflicts could introduce threats to tenants' needs, interests or concerns (Liu *et al.*, 2015). This implies a correct specification of both functional and non-functional requirements in order to avoid conflicts (Martinez *et al.*, 2019). Requirements uncertainty and conflict have become the two pervasive phenomena that are currently receiving more attention in the development of information system (IS) projects (Shameem *et al.*, 2018).

## **1.2 Motivation**

Nowadays, most software systems come under attack and have their performance threatened due to issues of dependability (Noll, 2014). This means that the availability and reliability of the system cannot be trusted. For instance, systems increasingly hold more sensitive and personal information of their users and therefore there is a need to secure the systems in order keep the information private. The analysis of security and privacy, not just as a technical aspect but also from the early stages of development (i.e. security and privacy requirements) therefore become highly important (Dubois and Mouratidis, 2010; Mouratidis *et al.*, 2012).

Security and privacy are important aspects of non-functional requirements (NFRs), but conflict between software requirements is impossible to avoid. There are several reasons why it is vital to deal with conflict between NFRs. Several studies assert that failure in understanding and managing requirements in general, and requirement conflicts in particular, are one of the main problems of exceeding a project's cost and allocated time which in turn can result in project failure (Aldekhail *et al.*, 2016; Jannat, 2019). Conflicting requirements represent the major reasons for inconsistencies in software development. A two-year study conducted by Egyed and Boehm reported that between 40% and 60% of requirements involved in any software

system are in conflict with at least one other requirement. The study also found out that among all software conflicts, NFRs represented the highest number of conflicts, comprising nearly half the instances (Egyed & Boehm, 1998 a; 1998b).

For optimal success in software implementation, inconsistencies diagnosed during the development process should be resolved immediately through the engineering process (Van *et al.*,1998). Organisations perform security measures to protect their sensitive assets and confidential data, and there is a lot of work on identity security as a part of security by design, about how to apply usability design approaches the design of security mechanisms (Faily *et al.*, 2015). Before we can ascertain whether an organisation is secure, we have to determine its security requirements. Clearly, specific requirements depend on the kind of system and data that need to be managed. However, organisations need to ensure that their software systems, storage and sharing of data are compliant with privacy laws and regulations.

Conflicts between privacy and security requirements are likely to occur in every business sector. These sectors are required to ensure users' privacy whilst also maintaining system security and invulnerability. For instance, in the case of a university system, we have many actors, goals and the associated relationships between them. If a student's goal is to submit assignments through the system, the teacher should receive these assignments while the identity of the student remains anonymous. Hence, the teacher needs to have non-repudiation as a security requirement. However, the two requirements of anonymity and non-repudiation lie in conflict, since both are naturally inconsistent with each other, and one cannot be achieved without compromising the other.

Moreover, security and privacy requirements are measures of the capabilities and functions that a system should achieve for eliminating security and privacy vulnerabilities (Boote, 2019). When these requirements are satisfied, conflicts are minimised and the system complies successfully with the imperative private and secure targets, as well as relevant regulatory guidance (TM Corporation, 2019; Yahuza et al., 2020). Security and privacy issues are among the most significant challenges affecting the acceptance of a new system. As such, studying ways of mitigating such problems is of paramount importance.

### **1.3 Aims and Objectives**

The aim of this thesis is to provide analysts with a framework to help them identify conflicts between security and privacy requirements, and to recommend tools that could help mitigate those conflicts. The proposed framework is implemented using SecTro (Secure Tropos) and is a CASE Tool for Modelling Security in Requirements Engineering. Secure Tropos is a software methodology which ensures that software is developed according to the user's needs as well as security requirements (Mouratidis & Giorgini, 2007). While all systems are prone to cyber-attacks, it is key that a system is able to defend itself effectively. In summary the objectives of this research are:

- To provide a framework that will clearly define and separate security and privacy at the requirements level. This will enable software engineers to analyse each one of these dimensions more in detail and also understand the relationship between them.
- To enable software engineers to understand how security requirements (which mostly arise due to the organisation's security policy) and privacy requirements (which mostly arise from data privacy laws) can co-exist in a system's design while that system remains functional. Therefore, any issues that need addressing (from potential conflicts) can be identified at an early stage of the development process.

- To allow semi-automated detection of security and privacy requirements conflicts which assist software engineers and analysts in their decision-making.
- To identify, characterise and define similar resolution strategies that consider security and privacy under one approach. This is important to overcome the limitations and issues discussed above.

Research Objective – The steps taken to achieve the objectives of this research thesis include:

- **Background Review.** A critical analysis and discussion of conflicts between security and privacy requirements is presented, with particular attention given to critical success factors which make a framework fit for purpose.
- **Literature Review and Identification of Gaps.** Findings from the existing literature are explored to identify both limitations and strategies that can be applied to resolve the present study's research problem.
- **Technique Development.** To address conflicts between security and privacy requirements identified in the literature review, a framework is formulated to determine and map conflicts, having an automated approach to detecting conflicts between requirements.
- **Technique Method.** A framework consisting of three phases: Phase 1 identifies and maps the most common privacy and security requirements; Phase 2 identifies conflicts between these requirements; and lastly Phase 3 presents a supporting toolkit with recommended tools, to reduce and solve conflicts between requirements. Different requirements will necessitate the use of different techniques, so strategies must be

presented to help the analyst prioritise requirements. These can be tailored to suit stakeholder or end user needs, depending on the intended use of the system.

- **Technique Validation.** Analysis is integrated into the standard requirements process so that security and privacy are taken into account from the very beginning. Situational implementation, a hypothetical case study and expert consultation are employed to improve and validate the proposed method prior to publication.

#### **1.4 Research Questions**

In order to satisfy the research aims and objectives, this research work proposes to answer the following research questions:

**RQ 1 – How to classify conflicting security and privacy requirements?**

**RQ 2 – What are the main characteristics of a requirements-based framework to support security and privacy requirements conflict resolution?**

**RQ 3 – What tool support is useful for the requirements analyst in identifying and solving conflicts between security and privacy?**

**RQ 4 – Does the proposed solution mitigate conflicts?**

There is an abundance of studies that have focused on identifying conflicts between requirements (Diamantopoulou *et al.*, 2017; Kalloniatis *et al.*, 2013; Matyás & Kur, 2013; Mellado *et al.*, 2014; Mouratidis *et al.*, 2013; Ramadan *et al.*, 2018; Shei *et al.*, 2015). However, most of these studies identify conflicts between requirements in general without focusing on privacy or security aspects in particular. Seeking to identify and advance knowledge in this area, it is important to protect security and privacy requirements, while reinforcing the credibility of the user's information and protection of the systems.

Recognising privacy and security requirements is an important step in identifying potential conflicts. This is important so that conflicts can be addressed and dealt with. For RQ1, based on literature reviews and innovative contribution, the researcher will incorporate steps that are semi-automated with regards to the supporting tools in the privacy pattern library, By the developer added all conceivable supporting tools into the privacy pattern library to make it easier for the analyst to insert the supporting tool to solve any issue that might arise.

as well as manual steps. First, conflicts between requirements are identified, derived from literature review searches, putting together an exhaustive list of the key requirements. The researcher then incorporates a matrix depicting a pictorial representation showing the possible relationships between and within requirements that are likely to present conflicts.

To answer RQ 1, *How to classify conflicting security and privacy requirements?* the conflict detection process has been represented as a matrix. A list of security and privacy requirements will be created based on most frequent requirements based on the literature reviews, which most systems should have in place. Each requirement's meaning will then be described. The mapping between privacy and security requirements has been undertaken based on the studies in the literature which are presented in further depth in the following chapters. In addition, we detect the key conflicting requirements, in order to determine the most vulnerable requirements, and to find a way to analyse such issues. Moreover, we model those requirements to have a better understanding of the conflicts between them. Our goal in this stage is to detect conflicts between requirements and find a way to mitigate them.

A prioritisation method is employed to sort the importance of each requirement (based on participant responses), as a way to find resolution. Moreover, we apply tools to support the

requirements. Each requirement has a corresponding tool which is most efficient at supporting it.

For RQ 2, *What are the main characteristics of a requirements-based framework to support security and privacy requirements conflict resolution?* This research will design a framework that can support the analyst to analyse conflicts. The existing literature shows no framework that incorporates requirements, possible conflicts arising and supporting tools for mitigating these in one tool. This is presented using three phases: Phase 1 Mapping Security and Privacy Requirements using a mapping matrix, this helps to identify requirements; Phase 2 Identify Conflicts between Requirements and Conflict Decisions via Phase 1 and introducing supporting tools; and Phase 3 Conflict Resolution by adding the supporting tool, in the privacy pattern library which is a component of the SecTro tool.

Moreover, to evaluate, the proposed framework is applied to the DEFEND Project is a European partnership that will afford a platform to permit organisations in different regions to consider and comply with the European Union's General Data Protection Regulation (GDPR). This will provide an opportunity to validate the framework within a real case study and enables us to detect conflicts at the early phase. Subsequently, we have pilot users from CSIUS (the Centre for Secure, Intelligent and Usable Systems) from the University of Brighton, presenting the framework and seeing how it works. These participants then evaluate the framework by completing questionnaires, which help us to ascertain how effective the proposed framework is and whether there are valuable recommendations to build from thereafter.

For RQ 3, *What tool support is useful for the requirements analyst in identifying and solving conflicts between security and privacy?* Phase 3 of the framework will offer a conflict

resolution table with support in terms of how we address the conflict concept and detect conflicts with supporting tools to analyse the conflict. This will be evaluated by participants of the pilot study and focus group. After the participants have grasped the full idea of the framework, and learned how to use it, they will apply the framework to a task scenario. Participants' responses will be measured and assessed to determine the framework's suitability for identifying and mitigating conflict between security and privacy requirements.

Furthermore, we approach RQ 4, *Does the proposed solution mitigate conflicts?* The focus group will evaluate the framework to determine whether or not it does mitigate conflicts. Results will be presented, reporting first from the pilot study and then the focus group. This group of participants are knowledgeable in the field of software engineering and privacy and security requirements. They are research fellows and doctoral students. Their expert advice will therefore bring a wealth of contribution and knowledge to the research.

## **1.5 Contribution to Knowledge**

This thesis aims to address the problem of identifying and analysing conflicts between security and privacy requirements. The contribution of the thesis is to present a three-phase framework to identify and analyse conflicts between security and privacy requirements. A semi-automated process is introduced employing a mapping matrix to identify and analyse conflicts, and appropriate tools specific to each conflict, are introduced to resolve conflicts, all integrated in one place, for the user of the system. The proposed framework is implemented using SecTro, a CASE Tool for Modelling Security in Requirements Engineering using Secure Tropos, a software system which ensures that software is developed according to the user's needs as well as security.



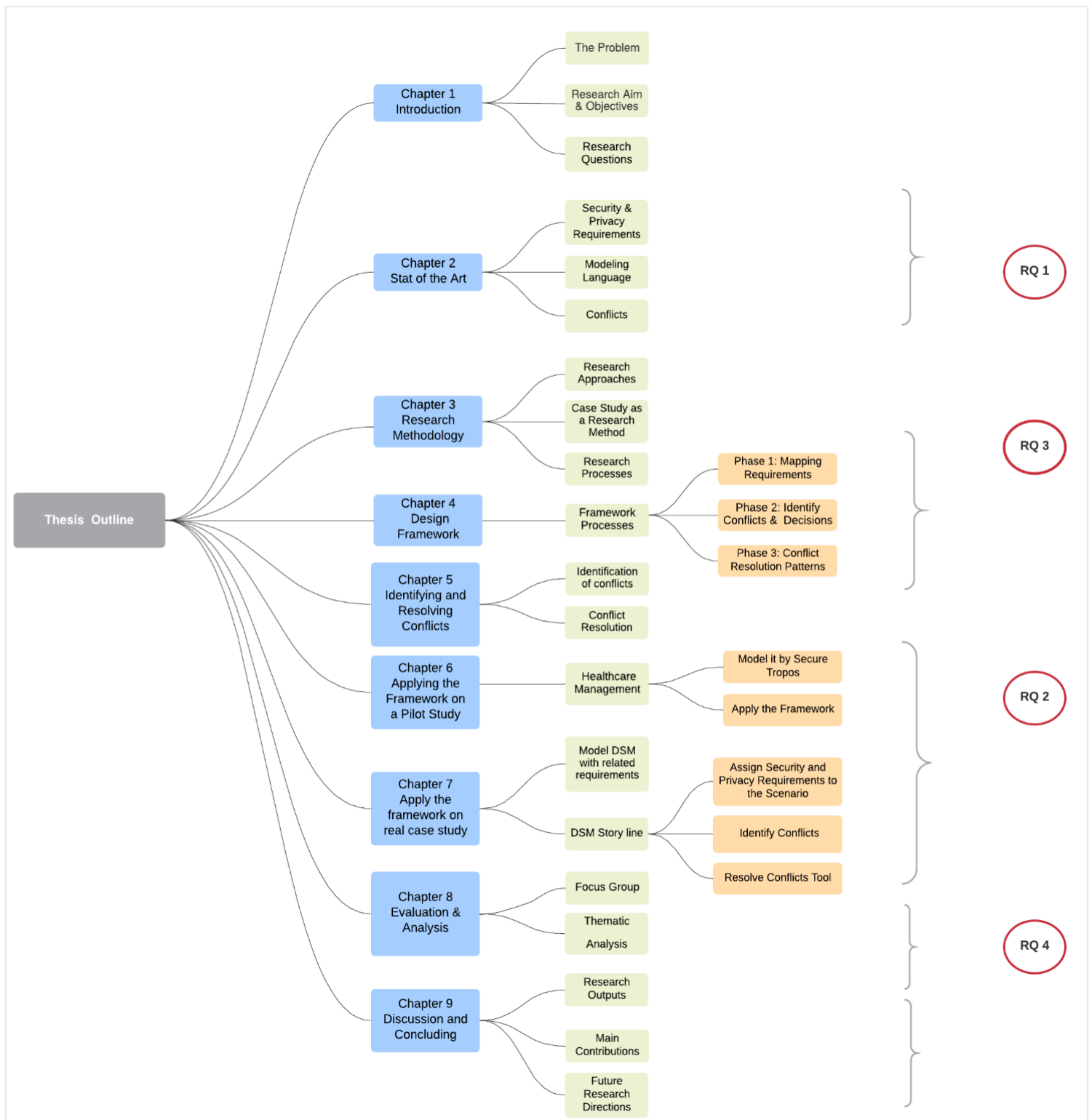
## 1.6 Overall Thesis Structure

This thesis is categorised into nine chapters (see Figure 1). **Chapter 1** is dedicated to introducing the research problem: the introduction, motivation to focus on conflicts between security and privacy requirements, and research gaps. These guide the aims, objectives and research questions of the thesis to fulfil those gaps. Here, the contribution to research in software engineering is specified. Thereafter, **Chapter 2** provides a comprehensive review of related works in security and privacy frameworks, and conflict between requirements. This guides the research in pinpointing the overall research gaps and limitations. Also, in order to mitigate conflicts, prioritisation methods are examined, as well as tool techniques to support requirements and to analysis conflicts. **Chapter 3** will discuss the methodology in general and present the justification as to which method the thesis will follow. **Chapter 4** presents an overview vision to model the framework. We extract the framework into three steps: identifying requirements, mapping between conflicts requirements based on our findings from the literature and determining which of the supporting tools are best suited to mitigate each conflict. In **Chapter 5**, we look at the types of conflicts in more detail and present the framework by introducing conflicts into the model, to sort types of conflicts and them identify and analysis those conflicts. This shows the importance of detecting conflicts between security and privacy requirements. Next, **Chapter 6** validates the framework by applying it to a pilot study in the field of healthcare management. Firstly, we will present an example about the E-Health system to point out conflicts between requirements, and then will apply the proposed framework to mitigate conflicts. The second case study will apply to the DEFEND project, to evaluate the framework and support the framework's validity in terms of analysing conflicts and increasing the efficiency and effectiveness of the overall system.

**Chapter 7** evaluates the framework in two phases. Phase 1 via DEFEND partners and Phase 2 presenting our work to the CSIUS group; this second phase involves receiving the group's feedback about the framework and how it can be improved. **Chapter 8** discusses the evaluation of the framework and integrating analysing conflicts in SecTro, digging further into thematic analysis and its application to the research in answering its research question. We present a discussion to outline the main contributions of this work, and the continuing phases as part of future work. In addition, we will summarise the outcomes of the research and areas for possible future work. **Chapter 9** will conclude by addressing the major topics addressed in this research, including the research stages employed throughout to answer the research questions. A summary of contributions will finally be discussed, including threats to validity and the opportunity for future research arising from the current work.

### **1.7 Published Work**

- Alkubaisy, Duaa. (2017). A framework managing conflicts between security and privacy requirements. 427-432. 10.1109/RCIS.2017.7956571.
- Alkubaisy, Duaa & Cox, Karl & Mouratidis, Haris. (2019). Towards Detecting and Mitigating Conflicts for Privacy and Security Requirements. 10.1109/RCIS.2019.8876999.
- [Accepted paper] Alkubaisy D, Piras L, Al-Obeidallah MG, Cox K and Mouratidis H, "Conflicts: A Tool for Privacy and Security Conflict Resolution for Supporting GDPR Compliance through Privacy-by-Design". 16th international conference on Evaluation of Novel Approaches to Software Engineering -ENASE 2021, 26-27, April 2021.



**Figure 1.1 Thesis Structure**

# CHAPTER 2

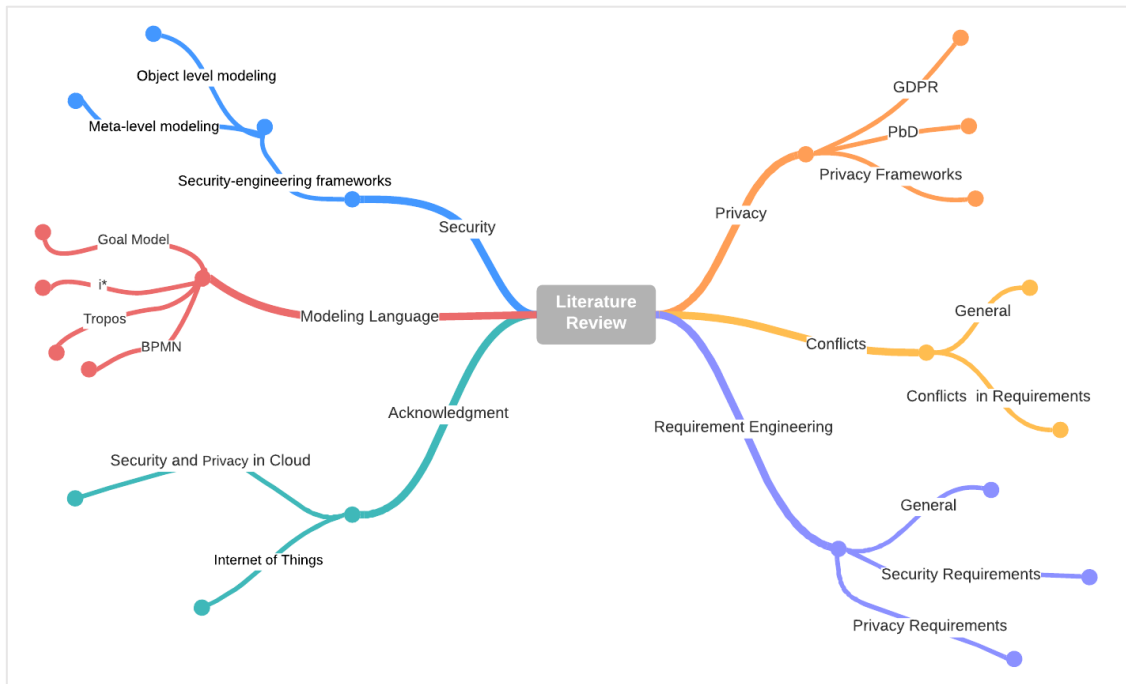
## LITERATURE REVIEW

### 2.1. Introduction

This chapter provides an overview of previous work. It presents recent privacy and security frameworks and the limitations of those frameworks, after which we will review the challenges to managing security and privacy requirements.

The main purpose of the literature review (see Figure 2.1) is to survey previous studies on security and privacy issues. This is done to reveal the gaps in recent studies and analyse conflicts between requirements and their place in the research.

The structure of this chapter will be as follows. Firstly, we will present the background of security requirements in engineering, and the importance of privacy and security requirements for both users and organisations. Next, we will highlight the various modelling languages with regards to *i\**, Tropos, and Business Process Model and Notation (BPMN). Thirdly, we will outline the importance of user privacy, with GDPR interventions, and limitations of recent frameworks, and present studies to address conflicts between requirements in more general terms. This will comprise an investigation of how these requirements would conflict and in how such conflict determines the effects on the systems involved. This chapter will guide us into the problem statement, having looked at the limitations of previous studies and how this research will fulfil this gap.



**Figure 2.1 Literature Review Map- Security and Privacy Issues**

## 2.2 Background on Software and Requirements Engineering

According to Pamela Zave (1997) Requirements Engineering is a branch of software engineering concerned with real-world goals for the function of, and constraints on, software systems. It is also concerned with the relationship of these factors to precise specification of software behaviour, and for their evolution over time and across software families (Zave, 1997). It can be further inferred that requirements engineering is the process of establishing the services that a customer requires from a system and the constraints under which these services are developed. The requirements are conclusively the description of the system's services and constraints that are generated during the requirement engineering process.

In software engineering, a requirement can be roughly defined as the documented physical or functional specification of a procedural or product necessity that aims to achieve a specific

objective. The requirement may range in complexity from a high-level abstract statement of a service or a system constraint, to a detailed mathematical functional specification. Requirements always initially come in the form of business goals which are represented as the incomplete and imprecise wishes and needs of stakeholders and potential end users of the software. It is during the requirement engineering process that these goals are translated into a complete and precise specification which we call ‘requirements’. Let us take for example this goal statement: Provide security for the properties of assets that are of importance and interest to the stakeholders.

Breaking down this requirement, we can infer more granular goals, such as:

Security Goals – confidentiality, integrity, availability, accountability, non-repudiation, authentication, authorisation, identification, and data protection; and

Privacy Goals – anonymity, pseudonymity, unlinkability, undetectability and unobservability.

A sample requirement statement that could be developed from the specific goals enumerated above would be: ‘Access to the system should be controlled by an Access Control List (ACL) administered through Active Directory (AD)’. Albeit an overly simplistic declaration, this is complete as it defines the action and the required behaviour (access to the system) and is complete in terms of the technological specification (ACL and AD).

We use concretisation to describe a refinement step in which a concept becomes more detailed. A similar, but restricted, sense is sometimes used in requirement engineering for the term ‘operationalisation’. In contrast, another established meaning of operationalisation expresses the transformation of non-functional requirements into functional ones. Typically, in both cases, the operationalisation of requirements generates a specification.

Requirement engineering, with specific requirements being the by-product, is the most important phase in the design of any framework. The specification that is born out of this exercise acts as a binding contract between the stakeholders and end users and the developers. This contract ensures that the software development effort is effective and that software errors are reduced at an early stage of the cycle. Since the requirements have a critical role in the determination of actions during the various stages of the Software Development Life Cycle (SDLC), considering them during the development of the framework is crucial.

### **2.3. Security engineering framework**

A security requirements engineering framework is needed to facilitate the production of security requirements that satisfy the Security Requirements, Incorporation of Assumptions about Behaviour and Satisfaction of Security Requirements (Haley *et al.*, 2008). In this framework:

- Security goals and security requirements aim to protect assets from harm. Primary security goals are operationalised into primary security requirements, which take the form of constraints on functional requirements and are sufficient to protect assets from identified harms. Primary security requirements are, consequently, preventative (Haley *et al.*, 2008).
- Feasibility, trade-off and conflict analyses may lead to the addition of secondary security goals, which result in additional functional and/or secondary security requirements. Secondary security goals and requirements may call for detective or preventative measures, a possibility further discussed below.

- Security satisfaction arguments show that the system can respect security requirements. These arguments expose and take trust assumptions into account.

Such a framework assists with understanding the place of security requirements within the development of an individual application, as well as the relationship between security requirements and other artefacts produced during development (Haley *et al.*, 2008).

Security requirements are constraints on a system's functions, which operationalise one or more security goals as follows:

- (i) They are constraints on a system's functional requirements, rather than being functional requirements themselves.
- (ii) They express the system's security goals in operational terms precisely enough to be given to a designer/architect. Security requirements, like functional requirements, are prescriptive, providing a specification (behaviour in terms of phenomena) to achieve a desired effect.

Security engineering is about building systems that will remain dependable in the face of malice, error or mischance. As a discipline, it focuses on the tools, processes and methods needed to design, implement and test complete systems, and to adapt existing systems as their environment evolves (Alberts, Woody & Dorofee, 2014). It requires cross-disciplinary expertise, ranging from cryptography and computer security to hardware tamper-resistance with formal methods, knowledge of economics, applied psychology, organisations and law.

System engineering skills are required at this stage, from business process analysis and software engineering to evaluation and testing, but these are not sufficient in themselves, as they only deal with error and mischance, rather than malice. Many security systems also have



critical-assurance requirements whose failure may endanger human life and the environment (as with nuclear safety and control systems) or do serious damage to major economic infrastructure (cash machines and other bank systems). Furthermore, malicious, or accidental failure may endanger personal privacy (medical records can undermine the viability of whole business sectors) and facilitate crime (for example the failure of burglar and car alarms). Even the perception that a system is more vulnerable than it really is (paying with a credit card over the internet, for example) can significantly hinder economic development.

Good security engineering requires four things to come together: (i) policy: what you are supposed to achieve; (ii) mechanism: the ciphers, access controls, hardware tamper-resistance and other machinery that you assemble in order to implement the policy; (iii) assurance: the amount of reliance you can place on each particular mechanism; and (iv) incentive: the motive that people guarding and maintaining the system have to do their jobs properly, as well as an understanding of the motive that attackers have to defeat your policy.

According to Nuseibeh and Easterbrook (2000, p. 37), software system requirements engineering is the process of discovering the purpose for which a software system is intended, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication and subsequent implementation.

Security requirements engineering can be classified in two ways (Giorgini, Massacci & Zannone, 2005): object-level and meta-level modelling. We will explore these two forms in the following sections.

### **2.3-1. Object-level modelling**

Object-level modelling uses requirement framework such as KAOS, i\* and Tropos to model a number of security requirements (further elaborated in section 2.4). The analysis features of these frameworks are used to draw conclusions about security modelling or to derive some guidance for the implementation of such models. The benefits of the object-level approach are that it is virtually cost-free from the viewpoint of the user and there is no new language to learn. Also, the framework is equipped with formal semantics and reasoning procedures. In the formal framework, security notions are indistinguishable from other objects or requirements. This is a major disadvantage which means that the link between the security and functional requirements is lost and must be presented via ad-hoc predicates or relationships by the designer. This makes the modelling of general relationships or rules particularly difficult; for example, the rule that the processing of personal data should be authorised by the person whose data is being processed.

### **2.3-2. Meta-level modelling**

Meta-level modelling utilises an off-the-shelf requirements framework (in the same way as object-level-modelling) but enhances it by using linguistic constructs that capture security requirements. The analysis features and implementation guidance related to the framework must be revised after security requirements are captured to allow new features to be incorporated. Meta-level models trade readiness for expressivity and compactness; the addition of linguistic constructs usually makes the model more compact and more intuitive. This advantage is coupled with the ability to design analysis features tailored to the security domain. However, the addition of new features is carefully planned, which requires the definition of semantics and analysis as well as reasoning procedures. To minimise this problem, most researchers try to design the framework in such a way that if the new features are not used, then the capabilities of the original framework can be inherited.

Previous studies of security engineering frameworks include the work of Antón (1996), who uses a Goal-Based Requirement Analysis Method (GBRAM) as a logical mechanism for identifying, organising and justifying software requirements. Goals can be discussed in terms of two themes: goal analysis and goal evolution. This method is useful in the management of goals for requirements specifications. In the future, it may also be used for electronic commerce. The researcher hopes to test and develop strategies for identifying and constructing goals further in the future.

Van Lamsweerde and Letier (2000) introduce systematic techniques for idealising goals, assumptions and requirements, integrating such techniques via a goal-oriented requirement elaboration method. They use Knowledge Acquisition in automated Specification (KAOS) in order to obtain complete and realistic requirements from which robust systems can be built. The techniques utilised are based on the temporal-logic formalisation of goals and domain properties; they are integrated into an existing method for goal-oriented requirements elaboration, with the aim of deriving more realistic, complete, and robust requirements specifications. The key purpose of this study is to allow exceptions to be handled during requirements engineering at the goal level so that there is more freedom to resolve them in a satisfactory way.

Butler (2002) aims to help information-system stakeholders to decide whether their security investments are consistent with the expected risks. Butler proposes a cost-benefits-analysis method using the Security Attribute Evaluation Method (SAEM). In addition to offering better security estimates, this method determines the level of benefit of the security technology, assuming that no other such technology is present. This method shows that it is worth investing in the development of better benefit estimates. These estimates are used to support security

managers in making estimations when they lack expertise. Finally, Butler compares the proposed method with the security technologies through the organisation under study to ascertain whether it offers a more cost-effective solution.

Moffett and Nuseibeh (2003) elaborate a framework which unites the two disciplines of requirements engineering and security engineering. From requirements engineering, they take the concept of functional goals and operationalise these goals into functional requirements (with appropriate constraints). From security engineering, they utilise the concept of assets (and threats of harm to those assets). Furthermore, they evaluate the relationship between software performance and security requirements using Jackson's Problem Frames. However, the analyst failed to construct a convincing satisfaction argument as there was not enough data available to justify the usage of trust assumption.

Kalloniatis et. al, (2004) describe the use of several well-known requirements engineering frameworks (NFR, i\* Tropos, KAOS, the M-N Framework, GBRAM and RBAC) for the provision and management of security requirements. Their work also presents a comparative analysis of existing frameworks, offering a number of viewpoints. The results of the analysis indicate that there are some unresolved issues that need to be addressed through further research in the security requirements field. The study concludes that the full range of potential security issues has not yet been encountered in the system design. Most of the methodologies presented in the paper do not reach the system policies level, but remain at the organisational requirements level, which cannot prove very helpful for the developer during the implementation phase.

Mead and Stehney (2005) use a Security Quality Requirements Engineering (SQUARE) methodology. This aims to elicit and prioritise the security requirements within software-development projects set up via the Software Engineering Institute's Networked Systems Survivability (NSS) Program. The methodology is applied to a number of recent case studies. The SQUARE method has proven effective in helping organisations to understand their security positions and to generate products with verifiable security requirements. In addition, the NSS is currently developing a web-based CASE tool to support this approach. The tool will assist requirements engineering teams with each step of the SQUARE process by automating documentation and streamlining communication with stakeholders. The researchers state that this prototype will be implemented as a standard model for NSS in the future.

Chung (1993) offers a process-oriented approach, which understands security requirements as non-functional requirements within the information system design process. Through the implementation of a prototype design tool and experimentation with a credit card system, the study demonstrates that parts of the design process can be automated, such as: (i) the display of method hierarchies and instantiation of the method selected; and (ii) the display of correlation rule tables and the use of correlation rules to detect potentially conflicting or harmonious goal interactions. Such automation can prevent certain actions that might jeopardise certain non-functional requirements (NFRs). Furthermore, Chung evaluates the effects of various design decisions using labelling procedures and maintaining goal graph structures.

Mouratidis and Giorgini (2007) propose extensions to the Tropos methodology to enable it to model security concerns throughout the development process, utilising a case study from the health and social care sector. Their objective is to provide a development methodology that

allows developers to integrate security-related analysis into a system, enabling them to identify desirable security aspects and apply reasoning methods to them, thus creating a secure system. Tropos adopts the *i\** modelling framework (Yu, 1997), which uses the concepts of actors, goals and social dependencies to define the obligations of certain actors towards other actors. Here, a multi-agent system and its environment are viewed as one set of actors, which depends on other actors to fulfil its goals. The study identifies the need for further research regarding the integration of security and functional requirements into the development stages of multi-agent systems. The aim is to make this approach applicable even to developers with little knowledge of security.

Massacci *et al.* (2008) provide a framework for security requirements elicitation and analysis. Their framework is based on constructing a context for the system, representing security requirements as constraints and developing satisfaction arguments for these security requirements. The system context is described using a problem-oriented notation which is then validated against the security requirements through the construction of a satisfaction argument. The satisfaction argument consists of two parts: a formal argument that the system can meet its security requirements and a structured informal argument supporting the assumptions expressed in the formal argument. One potential problem with this approach is that the construction of the satisfaction argument may fail, revealing either that the security requirement cannot be satisfied in the given context or that the context does not contain sufficient information to develop the argument. In this case, designers and architects should be asked to provide additional design information to resolve the problems.

Pandey, Suman & Ramani (2011) concentrate on the avoidance of anti-requirements and risk management. They recognise crucial activity in the requirements engineering community as

well as handling security aspects. They propose a framework that involves a risk management approach with a security requirement. This method expands the iterative security engineering activity at the earliest stages of development.

Moreover, Mayer *et al.* (2008) discuss the problems relating to the language that supports the agent-based Information Security (IS) development system Secure Tropos. They focus on the early phase (early and late requirements) of IS development. After analysing existing models for IS, they use security-risk management to suggest improvements. This study is located in the healthcare domain and uses the electronic Single Assessment Process (eSAP). Secure Tropos could be improved through the addition of extra constructs; the semantics of individual modelling constructs should be adapted so that they represent the Information Systems Security Risks Management (ISSRM) concept adequately. Mayer *et al.* appraise the Secure Tropos metamodeling, clarifying the unclear use of language constructs, before discussing the secure tropos and ISSRM alignment. In addition to the secure tropos, the study also investigates the possibility of extending KAOS for use in relation to security and cases of misuse.

Smith, Beaulieu & Phillips' (2011) modelling approach uses the Unified Modelling Language (UML) 2 without extensions to support the design, composition and verification of security protocols. The approach assumes a strong threat model, in which an attacker can intercept, modify and spoof all communication, with the exception of those protected by known strong encryption. Using a series of models of extensively studied protocols, the researchers demonstrate that the approach allows protocol properties to be accurately represented and protocols to be automatically tested to detect potential security flaws. The approach benefits from the existing strong tool support for UML 2, allowing automatic generation of protocol implementations from the models. The findings show that UML 2 can be used to model simple

protocols, as well as Needham-Schroeder and Yahalom protocols, without requiring any extensions to the language. The approach of building the framework by using increasingly more complex protocols allows the researchers to demonstrate that UML 2 could support all of the protocols modelled.

Bryl *et al.* (2006) utilise secure tropos to create a design that consists of a network of actors with delegation or permission dependencies among them. A planner is used, which inputs a set of actors and goals, and generates alternative multi-agent plans to fulfil the given goals. Bryl *et al.* show that it is possible to use an off-the-shelf planner to generate possible designs for significant security requirements, further noting that the designer does remain in the design loop. Therefore, the designs generated by the planner are suggestions to be refined, amended and approved by the designer. In other words, a planner is a support tool intended to facilitate the design process. Possible future work includes extending the application of this idea to other phases of the design process and to progressively larger industrial case studies to see to what extent it can operate without specialised solvers being employed.

Pan (2012) compares i\*-based and Use Case-based security-modelling initiatives. Secure Tropos and Misuse Cases are utilised within this empirical investigation (Sindre and Opdahl, 2005). The participants' perception of the two modelling approaches was sought by asking them to estimate the usage of modelling diagrams, textual description of cases and memory in the experiment. The results indicate that there is no significant difference between the two modelling techniques in terms of identifying threats. However, their ability to identify mitigations was markedly different. The participants were complementary regarding the goal-based modelling approach to security issues; hence, secure tropos was perceived as the



preferred approach. The investigation shows that most of the expected advantages of the two modelling approaches were confirmed.

### **2.3-3. Limitations of earlier security methods**

Different requirements in engineering methodologies have been proposed for managing security issues during system design. This includes non-functional requirements (NFR) (Chung, 1993; Mylopoulos, Chung & Nixon, 1992), Tropos (Liu, Yu & Mylopoulos, 2003; Mouratidis, Giorgini & Manson, 2003a, 2003b), KAOS (Van Lamsweerde & Letier, 2000), i\* (Liu, Yu & Mylopoulos, 2002), role-based access control- RBAC (He & Antón, 2003), the M-N framework (Moffett & Nuseibeh, 2003) and GBRAM (Antón & Earp, 2000; Bellotti & Sellen, 1993).

The above methodologies do not address privacy specifically but treat it as part of a system security. Privacy is inherently linked to security; the more security in place means loss of privacy. Loss of privacy means loss of liberty unfortunately. As such, they do not offer specific techniques for identifying privacy issues. Furthermore, the majority of the proposed methodologies (with the exception of GBRAM) focus on the elicitation of security requirements from business goals, but they neither handle how these requirements are translated into system components nor suggest sufficient implementation techniques. RBAC is the only method that can generate system policies based on the security requirements elicited. However, this method does not provide a systematic way of eliciting and managing these requirements.

In 2004, a study by Kalloniatis, Kavakli & Gritzalis, (2011) compared a range of security requirements frameworks applied in e-government. They concluded that the methodologies

studied did not cover all the necessary components for a requirement framework and suggested that a combination of different methodologies might be used to create a strong security requirements framework. In a recently updated version of this study, the authors investigated privacy in relation to these methodologies, and concluded that only the secure i\* framework (Elahi & Yu, 2007) and secure tropos (Kalloniatis *et al.*, 2004; Massacci, Prest & Zannone, 2005) consider privacy goals to be soft goals for the actors in their frameworks (Kalloniatis, Kavakli & Gritzalis, 2011).

Previous studies in software engineering have been based on Unified Modeling Language (UML) (Basin, Doser & Lodderstedt, 2006; Doan *et al.*, 2004; Jurjens, 2004; France *et al.*, 2004; Ray, France & Kim, 2004; Sindre & Opdahl, 2005). Such models have been used to address security concerns related to IT systems (Van Lamsweerde, 2004; OASIS, 2005; Yu & Cysneiros, 2002; Yu & Liu, 2001; Fabian, B. *et al.* 2010). Furthermore, these approaches also support attackers modelling, along with the objectives and representations of decisions that contribute to security goals. However, they lack fundamental concepts such as ownership and trust, which are the foundations of security. Overall, the UML methodology fails to describe the firm's operational procedures and their security policies in terms of model-driven development. Evidence given by the ACFE (2006) emphasises the importance of modelling an organisation and of the social relationship among all actors included in a system. Yet such an issue has only been addressed in part by earlier approaches (Van Lamsweerde, 2004; Liu, Yu & Mylopoulos, 2002).

## 2.4. Modelling Language

### 2.4-1. Goal Model (GM)

Horkoff *et al.* (2019) examined the 246 top-cited papers over the past 20 years, using Scopus. They make several observations about the goal-oriented requirements engineering (GORE) field, where goals are used as a useful conceptualisation to elicit, model and analyse requirements, capturing alternatives and conflicts. GORE is an effective way of capturing high level organisational strategy via the use of actors, goals, resources and dependencies. Since this work focuses on the aspects of security and privacy, we also need to be able to include such concerns at the highest level of analysis. This is why Secure Tropos (Mouratidis and Giorgini, 2007; Mouratidis *et al.*, 2016) concepts were selected for the current study, as they allow us to perform security and privacy analysis from an organisational perspective (Argyropoulos *et al.*, 2017).

Despite extensive efforts in this field, the requirements engineering community lacks a recent, general systematic literature review of the area. In (Argyropoulos *et al.*, 2017) work, they utilised a literature map, evaluating trends over time. Findings reveal that interest in topics such as scenarios, business modelling and intelligence (BI), and (model driven) MD seems to have dropped in recent years, while research into topics such as early requirements engineering, conflicts, patterns, security, privacy and risk, and architecture appear to hold steady.

Goal-Models (GM) are well-established requirements engineering tools to depict and break down systems using socio-technical concepts (Mylopoulos, 1998). In other words, it provides the goals for which the system should be designed and the various possible ways to reach those goals.

The variability of goal achievement strategies is the baseline for an actor to adapt by deciding which alternative to adopt as a response to certain triggers or adaptation drivers, e.g., faults, errors, availability of computational resources and newly available services and packages. The dynamic environment in which the system operates, i.e., its context, could also be an adaptation driver. The Contextual Goal Model (CGM) (Ali *et al.*, 2010) extends the traditional goal model (Bresciani *et al.*, 2004; Castro, Kolp & Mylopoulos, 2002; Yu 2011) with the notion of context. Context may be an activator of goals, a precondition on the applicability of certain alternatives to reach a goal and a factor to consider when evaluating the quality provided by each of these alternatives.

Goal models have been used as an effective means for capturing the interactions and trade-offs between requirements, but they have also been applied more broadly to advance the state of software adaption, security, legal compliance and business intelligence, among other areas (Horkoff *et al.*, 2019).

Nevertheless, aligning software to meet privacy requirements is a challenging task, because there is still no unified vision of privacy in RE. Privacy is a multifaceted concept, as well as it can often be vague and elusive, comes in many forms, relating to what one wishes to keep private (Kalloniatis *et al.*, 2008; Gharib, 2017). This has resulted in much confusion among software designers and stakeholders, and, in turn, has led to wrong design decisions (Gharib, 2017). An issue GDPR is yet to clarify.

Therefore, providing a conceptual foundation and a conceptual model of privacy may help software engineers to meet users' privacy needs. Privacy requirements can be specified through models (Kalloniatis *et al.*, 2009). Models are graphical or visual representations that describe

the problem to be solved and the system to be developed (Sommerville, 2011). Requirements models are described in a specific modelling language. In addition to creating models, modelling languages can be used to support requirements analysis, which assists in the understanding of the various solution possibilities for a problem and the implications of each alternative solution considered (Yu, 1997).

Modelling languages and framework - i\*, Secure Tropos, and the Business Process Modelling Notation (BPMN), are investigated moving forward. These are modelling languages that have been applied in previous research, which will provide more context about their uses and applications. Elaborating on these also provides justification for the choice of modelling language that will be employed in this research, as it offers a holistic approach to acknowledging the different modelling languages.

#### **2.4-2. i\***

Liu, Yu & Mylopoulos (2002) analyse and model the security concerns in a Peer to Peer (P2P) setting using the i\* modelling framework. This i\* approach inspires as well as simplifies the analysis of security-related issues within the full operational and social context of relevant actors. This model also encompasses the potential attacks, normal-case operational procedures, and other indirectly related factors and countermeasures against perceived threats. Furthermore, they examine security in the P2P domain, which it is not a hard-wired concept in the i\* framework. However, it is flexible enough to handle the different security concerns that may apply to a certain context or problem domain.

According to Dalpiaz *et al.* (2016), the i\* 2.0 is a goal- and actor-oriented modelling and reasoning framework. In this language, there is an actor representation who can be specialised

in a role or agent (using the —is all or —participates-inll links) and have dependency links with other actors. In i\* 2.0, there are also AND/OR refinements and contribution links (make, help, hurt and break), as well as four intentional elements (Dalpiaz *et al.*, 2016) as follows:

- Goal: a state of affairs that the actor wants to achieve and that have clear-cut criteria of achievement;
- Quality: an attribute for which an actor desires some level of achievement;
- Task: actions that an actor wants to be executed, usually with the purpose of achieving some goal;
- Resource: A physical or informational entity that the actor requires in order to perform a task.

### **2.4-3. Secure Tropos**

The framework provided uses the Goal-Oriented Requirements Engineering (GORE) approach. Amid the various GORE methodologies presented in the literature such as KAOS, we embraced secure Tropos as our framework baseline. Tropos offers primitives for modelling of the system alongside their goals, entitlements and abilities. Objectives are used to denote the actors' interests strategically and can be polished by the disintegration of a root objective into sub-goals. Moreover, resources represent both informational and physical units that are required, and created, by the accomplishment of the goals. On the other hand, secure Tropos offers the concept of assignment to model shifting roles between actors within the system.

Mouratidis & Giorgini (2007) propose an extension of the Tropos approach in order to enable modelling of security concerns all through the development process. Their paper uses a case study from health care and social sector. Their objective is to offer a development policy that enables developers to assimilate security connected analysis into the system, allowing them to

recognise desired security features that apply a rational approach and develop a secure system. Tropos assumes an i\* framework concepts modelling background, which uses the notions of actors' social and goals reliance for the definition of the responsibilities of some actors on behalf of others. The multi-agent system and its surroundings are seen as one set of actors who rely on their fellow actors to accomplish their goals.

Mouratidis & Giorgini ascertains the need for future study to be carried out regarding the incorporation of security and practical necessities into the development phases of multi-agent systems. The purpose is to make this method relevant even to developers with minimal knowledge of security issues.

#### **2.4.3.1 Secure Tropos Methodology**

Secure Troops Methodology consists of an engineering approach for security and privacy requirements, starting from early stage requirements of the IS (Information System) development process. Secure Tropos must be specified in the early phases of an IS development process, as it is an organised approach for goal-oriented security and privacy requirements modelling.

Besides, the structures of a Secure Tropos are enhanced with security and privacy concepts (see table 2.1, 2.2). Moreover, Secure Tropos support designing and analysing activities in software development processes, considering the relationship between security devices and security requirements.

Overall, Secure Tropos methodology supports modelling language, security aware process and an automated process. By explaining how the secure tropos methodology will enhance our framework, we will translate conflicts between requirements in a goal model using a sector tool.




## Secure Tropos model views

The Secure Tropos method presents models that contain security and privacy requirements analysis, moreover it supports the corresponding tool, namely SecTro (Pavlidis and Islam, 2011), the information listed based on three perspectives (views):


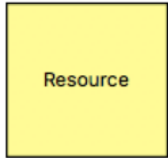
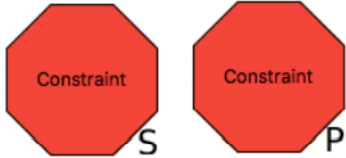
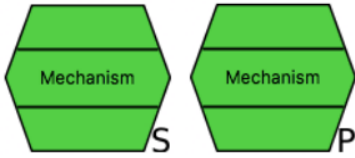
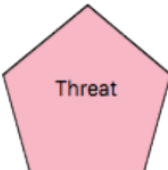
- (i) **The organisational view,**
- (ii) **The requirements view,**
- (iii) **The attacks view.**

These modelling views are used to facilitate system design, security and privacy requirements elicitation. Therefore, each view arranges for focus of the system under analysis.

**Table 2.1 Concept Types on Secure Tropos methodology**



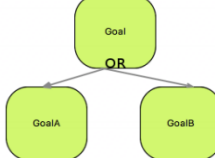
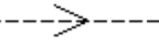
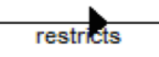
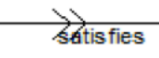
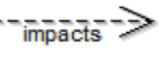
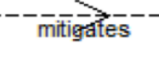
Concept	Description	Notation
Actor	Active entities that carry out actions to achieve goals by exercising its know-how. It refers generically to any unit to which intentional dependencies can be ascribed. Actors depend on each other to achieve goals, perform tasks, and furnish resources. While each actor has strategic goals to pursue, they are achieved through a network of intentional dependencies	
Hard Goal	A condition or state of affairs to be achieved. An actor can choose freely among different ways to achieve a goal. Thus, a goal represents an intentional desire of an actor. The specifics of <i>how</i> the goal is to be satisfied is not described by the goal but through task decomposition.	
Soft Goal	A goal that does not have clear-cut definition or criteria on whether it has been achieved. It represents quality attributes for which there are no a priori, clear criteria for satisfaction, but actors have to fulfil. Soft goals are typically used to model non-functional requirements.	



Plan	Represents a way of doing something. The fulfilment of a plan can be a means of satisfying a goal. As such, different alternative plans that actors might employ to achieve their goals, are modelled to enable software engineers to reason about the different ways that actors can achieve their goals, and decide upon the optimal way.	
Resource	Represents a physical or informational entity that an actor requires. The main concern is whether the resource is available and who is responsible for its delivery.	
Constraint	A restriction on an actor's function. There are two types of Constraints, namely Security and Privacy. Additionally, a Constraint is related to an objective that needs to be fulfilled, which is expressed through the constraint, such as Confidentiality, Integrity, Authentication, etc.	
Mechanism	Represents a system mechanism that supports the satisfaction of a security or privacy constraint. It can be any of two types, Security or Privacy.	
Threat	Represents a circumstance that has the potential to cause damage to the system.	

(Diamantopoulou *et al.*, 2018)

**Table 2.2 Relationship Types on Secure Tropos methodology**

Relation class	Description	Notation
Dependency	The depender depends on the dependee to bring about a certain state of affairs in the world. The dependum is expressed as an assertion statement. The dependee is free to and is expected to make whatever decisions are necessary to achieve the goal (namely, the dependum). The depender does not care how the dependee goes about achieving the goal.	
And	Allows the decomposition of an element to more fine grained elements. All the sub-elements need to be fulfilled in order the parent element to be fulfilled as well. The elements that can be decomposed are a goal, a plan, a resource, a mechanism, an attack method.	
Or	Allows the decomposition of an element to more fine grained elements. The difference with the 'And' relationship is that only one element is needed for the fulfilment of the parent element.	
Contribution	Shows a contribution toward satisfying a soft goal, typically from a task or another soft goal. Any of these Contribution links can be used to link any of the elements to a soft goal to model the way any of these Elements contributes to the satisfaction or fulfilment of the soft goal.	
Restricts	Shows the goal that is restricted by a Security Constraint.	
Satisfies	Shows the security or privacy Constraint that a mechanism satisfies.	
Impacts	Shows the Goal that is affected by a Threat	
Mitigates	Shows the Threat that is mitigated by a Security Mechanism.	

### **The working of Secure Tropos:**

According to Mouratidis (2004), there are organisational, security and privacy goals; these goals introduce security and privacy constraints. A constraint is used to represent a set of restrictions that do not permit specific actions to be taken, restrict the way that actions can be taken or prevent certain system objectives from being achieved (Mouratidis, 2004). Security and privacy constraints are clearly defined as separate concepts to support a clear and well-structured elicitation and analysis of security and privacy requirements. When a constraint is introduced, further analysis is required to establish if and how that constraint can be satisfied. Within the context of our metamodel, a constraint is satisfied by a measure. This measure represents a generic, implementation-independent form of control that indicates how a constraint will be achieved, which are operationalised by relevant plans (Mouratidis, 2004).

Plans are implemented by relevant mechanisms; a plan defines a specific way of operationalising a measure, i.e. the details and conditions under which a specific measure is operationalised, while a mechanism is a technical solution that realises one or more plans. Furthermore, mechanisms are software products which are developed or customised software tools for realising plans for the specific organisation. These mechanisms require resources, and they are supported by capabilities. It is worth mentioning that the types of measures, plans and mechanisms follow the type of constraint that the measure satisfies. For example, a security constraint is satisfied by a security measure, which is operationalised by a security plan, which is implemented by a security mechanism (Mouratidis *et al.*, 2013).

#### **2.4-4. Business Process Model and Notation (BPMN)**

BPMN<sup>1</sup> is a standard for business process modelling that provides graphical notation for specifying business processes in a Business Process Diagram (BPD)<sup>2</sup> based on traditional flowcharting techniques. The objective of BPMN is to support business process modelling for both technical users and business users, by providing notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN 2.0 specification also provides execution semantics as well as mapping between the graphics of the notation and other execution languages, particularly Business Process Execution Language (BPEL).<sup>3</sup> BPMN is designed to be readily understandable by all business stakeholders, yet its technicalities make it much more understandable by engineers who programme processes. These include the business analysts who create and refine the processes, the technical developers responsible for implementing them, and the business managers who monitor and manage them. Consequently, BPMN serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation (Von Rosing, et al., 2015).

#### **BPMN-based data-protection engineering**

Modelling data protection requirements during the design phase of the business processes models is a promising research direction in the field of data protection engineering. The key idea is to extend graphical business process modelling languages such as BPMN to support the modelling and analysis of data protection requirements (Ramadan., 2018).

### **2.5. The Importance of User Privacy**

Privacy is becoming increasingly important as more and more information is digitised, facilitating quick and easy access to data. Digital privacy is key to preventing personal

---

<sup>1</sup> <https://www.bpmn.org/>

information from being revealed to unauthorised subjects. Privacy-related issues are many and varied, as privacy itself is a multifaceted concept that is dependent on what one wishes to keep private.

The term ‘privacy protection’ is based on privacy requirements such as authorisation, identification, anonymity, authentication, data protection, unlink-ability, pseudonymity and unobservability. The term authorisation is the procedure through which a framework finds what level of access a specifically confirmed client must need to secure assets controlled by the framework. Anonymity is characterised as the capacity of a client to utilise an asset or administration without unveiling their personality. Authentication is the instrument whereby frameworks recognise their clients in a safe way. Unlink-ability communicates powerlessness to interface related data. Pseudonymity is the client’s capacity to use an asset or administration by acting under one or numerous *noms de plume*. Lastly, unobservability preserves a client from being tracked or observed while browsing the Internet or accessing a service (Nithya & Subha, 2013).

The announcement of regulations such as GDPR has increased awareness and legal obligations for organisations in terms of how to process and protect personally identifiable information. Designing and building a privacy-preserving system is challenging since these systems have to address conflicting security properties and system requirements to avoid any security vs privacy trade-off. If security and privacy are addressed together as a unified project, the resulting system will have security and privacy built-in rather than employing a bolt-on approach (Ganji, 2019).

The protection of client privacy is classified into two types, namely:

- security-oriented requirements engineering methodologies; and
- privacy-enhancing technologies (PET).

First, the security-oriented requirements engineering method does not associate the requirements recognised with implementation results. Additionally, the relationship between a client's needs and the capacities of supporting programming frameworks is an important significance. Second, the privacy enhancing technique concentrates on programming implementation alone, independent of the hierarchical setting in which a framework will be consolidated. PET devices can allow propelled data trade where security limitations exist (Nithya & Subha, 2013).

Recently, a report was published by the European Data Protection Board (EDPB) (2019). This body has implemented early deliberation of Data Protection by Design and by Default (DPbDD), through developing a new processing operation. Data protection by design is required to be implemented together at the time of determining the means of processing and at the time of processing itself. With a view to ensuring effective data protection at the time of administration, the controller should consistently review the effectiveness of the chosen measures and safeguards. These are strong guidelines; however, they are not tool-supported in terms of analysing conflict. Therefore, we present an approach based on the guidelines, but we go further in mitigating conflict by adding tools to the privacy library, for mainly tool-supported conflict resolution.

### **2.5-1. General Data Protection Regulation (GDPR)**

In May 2018, the General Data Protection Regulation 2016/679 (GDPR) came into effect to replace the Data Protection Directive 95/46/EC (DPD95). The GDPR was designed to harmonise data privacy laws across Europe in order to give greater protection and capabilities to individuals for controlling their personal data in the face of new technological developments (EUGDPR, 2018). GDPR applies to all organisations that handle personal data about EU residents, regardless of their physical locations.

Infringements of GDPR can incur fines up to 20 million Euros or 4% of an organisation's global turnover. Therefore, most organisations have implemented measures to comply with the GDPR. However, organisations do face several obstacles in their journey towards GDPR compliance. Some organisations are not aware of or do not understand the changes that GDPR will bring to their businesses (Rivera *et al.*, 2018). For example, a survey conducted between July and August 2017 by the Institute of Directors among 869 of its members in the UK revealed that 30% of company directors had not heard of GDPR, while 40% were still unsure about whether their company would be affected by it (Rivera *et al.*, 2018).

Other surveys expose similar problems such as a lack of preparation for meeting the GDPR legal obligations and a lack of awareness about the consequences of noncompliance (Rivera *et al.*, 2018). Most of these problems are rooted in the vague, ambiguous and verbose nature of regulations, which individuals – those who do not possess legal expertise – often find difficult to understand. Likewise, understanding legal requirements is generally time-consuming and cumbersome, thus complicating their operationalisation. These problems can jeopardise compliance with GDPR, especially when this process is not assisted by data protection law experts.

Mapping legal obligations into software functionality is also non-trivial (Colesky *et al.*, 2016; Gjermundrød *et al.*, 2016). As legal requirements are oftentimes too abstract, they may leave space for multiple interpretations. For example, GDPR states that companies must provide a reasonable level of protection of personal data, without clarifying exactly what ‘reasonable’ means (Brooks *et al.*, 2017). Similarly, GDPR promotes ‘privacy by design’, without detailing how this should be achieved (Koops & Leenes, 2014). Therefore, it is often the case that those in charge of implementing software changes are also responsible for ensuring compliance with GDPR and understanding which requirements should be operationalised and implemented in their organisation’s software system (Breux *et al.*, 2009; Dittel, 2016).

Organisations are currently implementing various measures to ensure that their software systems fulfil GDPR obligations such as identifying a legal basis for data processing or enforcing data anonymisation. In this context, this research aims to develop a framework for identifying and analysing discrepancies between security and privacy requirements and resolve such conflicts in the context of system development and design. The framework links GDPR obligations and related business requirements to privacy controls necessary to satisfy them. Privacy controls are also contextualised, depending on the stakeholder scenario and the data processing activity to which they should be applied. Effective implementation of such a framework would increase confidence in the effectiveness of privacy controls.

### **2.5-2. Security and Privacy Requirements in Cloud Computing**

Cloud computing is an evolving paradigm that is radically changing the way humans store, share and access their digital files. Despite the many benefits, such as the introduction of a rapid elastic resource pool and on-demand service, the paradigm also creates challenges for both users and providers. In particular, there are issues related to security and privacy, such as



unauthorised access, loss of privacy, data replication and regulatory violation that require adequate attention (Mouratidis *et al.*, 2013). Given the relevance of user privacy today, its growing demand, and significance in privacy and security risks, which cannot be ignored, iCloud and Internet of Things (IoT), are examined briefly.

### **iCloud**

Cloud computing supports software systems' infrastructure where the availability of resources, computational or otherwise, used in the specific model is dynamic; meaning that the hardware and software are dealt with as services offered to the users of the cloud every time, needed for effective use of the cloud. iCloud- due to the relevance of user privacy and its growing trend today. By virtually grouping hardware and software and providing it efficiently, cloud users are able to achieve great economical savings both in the functional and the administrative costs of their specific ICT infrastructures.

However, the storage of personal and sensitive information in the cloud raises concerns about the security and privacy of such information and to what extent the cloud can be trusted. Security and privacy in this context require solutions very different to those provided by current research efforts and industrial practices. Solutions must not only try to guarantee security and/or privacy from a technical point of view, but also provide clear understanding of the social aspects of security and privacy and take into account, for example, organisational structures, privacy needs and appropriate laws and regulations.

As the concept of cloud computing is relatively new, many organisations and individuals are still avoiding cloud services because they are not sure whether the services provided by different providers are suitable for their security and privacy requirements. This is especially

true since organisations and individuals would have to hand over their personal and organisational data to service providers over which they have no control. This introduces an extra layer of complexity on top of the expected security and privacy issues that are present in any type of software systems and services, whether on the cloud or not. These concerns make risky a transition to cloud computing or integration of a cloud solution to an existing IT infrastructure (Mouratidis *et al.*, 2013).

### **Internet of Things**

The Internet of Things (IoT) is composed of physical objects embedded with electronics, software and sensors, which allows objects to be sensed and controlled remotely across an existing network infrastructure, facilitating direct integration between the physical world and computer communication networks. IoT has been widely applied in various applications such as environment monitoring, energy management, medical healthcare systems, building automation and transportation. Therefore, IoT is exposed to significant privacy and security risks which cannot be ignored in this research.

Unfortunately, due to the resource constraints of IoT devices, they always delegate highly complex computation to the energy abundant cloud. However, the inputs, outputs and function of the underlying computation may be closely related to the privacy of IoT users, which cannot be exposed to collusion between malicious cloud servers and malicious IoT users.

Therefore, how to design new efficient privacy-preserving solutions for next generation mobile technologies with IoT– cloud convergence is a crucial issue of great concern (Zhou et al., 2017). To consider the possibility of accomplishing it, is another topic for another day.

Cloud-based IoT can be categorised into static and mobile, the latter of which is more challenging in protocol design. Therefore, we mainly focus on the security and privacy issues and corresponding countermeasures in mobile cloud-based IoT. The development of next generation mobile technologies such as fifth generation (5G) on IoT–cloud convergence has cast light on a variety of types of security and privacy issues which have lain unaddressed for years. The characteristics of resource-constrained short-range communication and mobility result in the unique features of packet forwarding in cloud-based IoT (Zhou *et al.*, 2017).

IoT is exposed to significant privacy and security risks. It can be used to both protect and violate individual's privacy and security. First, Atzori *et al.* (2010) raise theft as one of the potential applications of IoT because the existence of IoT will make it possible to develop an application which sends out SMS messages immediately to users whenever their personal stuff (such as television or wallet) is moved from predefined locations without their permission (burglary/theft).

Second, the architectural nature of the IoT, where trillions of objects may interact with each other, will attract malicious attackers and eavesdroppers to collect data, thus breaking privacy and security rules (Roman *et al.*, 2013). Hence, maintaining secure and private connections and transmission of information, in addition to preventing data collection, are unquestionably crucial requirements for the development of IoT. The most complex challenge from the requirements engineering perspective is the difficulty of specifying requirements, and security and privacy requirements in particular, for a system with so many components that can be randomly integrated in various systems at various times and places. For the IoT, it is difficult even to envision what system an object will be a part of. As such, we must develop a

requirements framework that takes security and privacy into account both at the component level as well as a part of a system-of-systems (Alqassem, 2014).

### **2.5-3. Essential Principles of Privacy**

**Transparency:** Prior to the initial registration of data, the individual must be informed of the organisation's identity and the reason for processing their data, and consent to the processing of said data (Koorn *et al.*, 2004).

**Justification:** Personal data should only be processed if the purpose for which it was collected can be justified and if it will not be processed later in any manner that is incompatible with that purpose (Koorn *et al.*, 2004).

**Legitimate ground:** The Personal Data Protection Act restricts the instances in which personal data may be processed. The processing of data (religion, race, health, sex lives, trade union memberships, etc.) is unlawful unless specific conditions have been satisfied (Koorn *et al.*, 2004).

**Quality:** The personal data collected should be relevant to the purpose for which it is intended; it should be adequate and accurate, and it should not be kept longer than necessary (Koorn *et al.*, 2004).

**Rights of the individual:** The individual concerned (data subject) should have the right to access, rectify and erase his/her personal data, or to block/object to the processing of the data (Koorn *et al.*, 2004).

**Security:** The party responsible must take the necessary technical and organisational precautions for the safeguarding of personal data, thus preventing loss or any form of unlawful processing (Koorn *et al.*, 2004).

#### **2.5-4.Role of Privacy by Design Principles**

Privacy officials in the United States and Europe embrace privacy by design (Cavoukian 2012, European Commissioner, 2012). Such is the concept that in planning information and communication technologies, creating privacy from the outset accomplishes a better outcome than securing it at the end. For instance, Rubinstein claims that companies participating in privacy by design, when they advertise consumer privacy via organisations at every aspect of the development of their services and products, has two primary components – integrating four substantive protections of privacy into a company’s practices and upholding comprehensive procedures of data management throughout the life cycle of their services and products (Rubinstein *et al.*, 2012).

Preceding the specific dynamic privacy frameworks, design constraints acted as guiding principles to design and develop systems with privacy considerations as the core set of requirements. Privacy by design, being a mainly ethical guideline is one such system for example, as proposed by the Federal Trade Commission (FTC) in their recent set of guidelines for the design and development of privacy sensitive ubiquitous computing platforms (Federal Trade Commission, 2000). According to FTC guidelines, privacy by design is best enforced by following four guidelines:

- i) Notice/Awareness;
- ii) Choice/Consent;
- iii) Integrity/Security; and
- iv) Enforcement Redress.

These principles, or Privacy by Design Heuristics state:

1. Notice/awareness

Users of emerging or existing technological devices, services or products should be explicitly informed about the information being collected during the lifecycle of the usage of the service or product (Duncan *et al.*, 2001).

## 2. Choice/Consent

This means giving users the ability to opt out of information sharing specific formats or to explicitly allow them to approve any information before sharing to any third party for fair usage or for reasons which would allow optimised solutions, such as the targeted advertising used by various search engines. This often involves user acceptance by requiring the user to agree to the privacy statement of the product or service provider (Fienberg, 2005).

## 3. Integrity/Security

This requirement highlights the importance of the fact that user data and private information must be stored as it is, without tempering the information or breaching such information to the rest of the world (Nissenbaum, 2004).

## 4. Enforcement/Redress

This guideline states that the privacy and protection of user data is possible only if there is a mechanism or privacy framework to enforce such dynamic privacy requirements. This requires privacy aware systems to be designed from the ground-up using a privacy framework proposed for maintaining the integrity and privacy of user data (Van Lamsweerde & Letier, 2000). As discussed in following paragraphs, the work of Bellotti and Sellen (1993) propose using feedback and control over all phases of system development, from requirements gathering, requirements specification document development, system design, development phases, application of software engineering practices and during the deployment of a solution.

To overcome privacy concerns, several frameworks and privacy models as well as design principles have been proposed.

### **2.5-5. Privacy Frameworks**

Most research has focused on the design of privacy frameworks, such as: Bellotti and Sellen (1993); Dey, Abowd and Salber (2001); Jiang, Hong and Landay (2002); Hong and Landay (2004); Jensen *et al.* (2005); Kavakli *et al.* (2006); Kalloniatis, Kavakli and Gritzalis (2008); Kalloniatis, Kavakli and Kontelis (2010); Shapiro (2012); and Nithya and Subha (2013). By reviewing these studies, the present study identifies the gap that exists in such framework designs and proposes a new model. Several mechanisms have been suggested in relation to mitigating privacy threats in pervasive environments. Each seeks to fulfil certain privacy principles, as explained. These studies are discussed as follows.

Bellotti and Sellen (1993) describe a privacy framework design for the control of information captured by multimedia in ubiquitous computing environments. Their design seeks to maintain a balance between awareness and privacy and involves the analysis of privacy issues using RAVE and other similar systems. In spite of the limitations inherent in RAVE, it is widely accepted as a useful laboratory tool.

Dey, Abowd and Salber (2001) present a conceptual framework that uses a context-aware application and divides the acquisition and representation of context from delivery as well as a context reaction towards the application. The Context Toolkit is used to enable the rapid development of multiple context-aware applications. Firstly, the contexts are developed; secondly, categories of contextual information are detected; and thirdly, context-aware application behaviour is formulated. Although context-aware computing requires a very

detailed understanding of contexts, the types of context used in this study are those that can be detected automatically via sensors in the physical environment.

Jiang, Hong and Landay (2002) develop a framework to support socially compatible privacy objectives in ubiquitous computing settings. They implement the Principle of Minimum Asymmetry, which seeks to reduce the gulf between systems, data subjects and data users. The study puts forward the Approximate Information Flow (AIF) model, which manages the interactions between various actors and personal data. Their results show that AIF effectively supports varying degrees of asymmetry in Ubicomp systems. They suggest a new privacy-protection mechanism and propose further inspection of the privacy friendliness in Ubicomp systems.

Hong and Landay (2004) present an extensive analysis of end user needs and application developer needs for privacy-sensitive systems using Confab. The end user's requirements were gathered through scenario-based interviews and the use of location-enhanced applications. An analysis of surveys, research papers, message boards, proposed/existing privacy-protection laws and design guidelines for privacy-sensitive systems was also conducted. The developer requirements were determined through an analysis of research based and commercial Ubicomp applications. In the future, these researchers plan to build more Ubicomp applications using Confab.

Jensen *et al.* (2005) put forward the Structured Requirements Analysis Planning (STRAP) framework model, which is a heuristic-based framework that performs a goal-oriented analysis in order to identify relevant actors, goals and major system components, as well as privacy vulnerabilities. The model is similar to that proposed by Bellotti and Sellen (1993), except that



it borrows its methods from requirements engineering and goal-oriented analysis. However, Jensen *et al.*'s model fails to offer adequate implementation techniques for the elimination of vulnerabilities. The study uses the predictive group calendar system to provide an analytical structure for a privacy-aware design, as well as a method for deriving policy requirements from the analysis. The results show that the STRAP framework performs better than Bellotti and Sellen's heuristic model. The time on task is the same in both studies, yet Jensen *et al.*'s model detected more privacy-related vulnerabilities. In the present study, STRAP will be applied to a real software development cycle to see whether iteration and refinement are adequately supported by this tool.

Kavakli *et al.* (2006) present a methodology for integrating privacy requirements into the process of a system named PriS. Their methodology is requirements engineering based and uses the e-VOTE system. The study focuses on privacy issues and provides a set of concepts to model privacy requirements in the organisational domain and a systematic way to transform these requirements into system models. The Enterprise Knowledge Development (EKD) framework is used as a conceptual model here. Based on the analysis of a number of well-known privacy-enhancing technologies and security requirements engineering methodologies, this paper highlights the gap between system design methodologies and technological solutions. The PriS methodology has a high degree of applicability to Internet systems that seek to deliver services such as untraceable transactions and anonymous browsing, which ensure user privacy.

Kalloniatis, Kavakli and Gritzalis (2008) use PriS with an e-voting case study to integrate privacy requirements into the system development process. The transformation of an Internet-based electronic voting system to accommodate a new legal framework regarding privacy

protection is also discussed. The PriS methodology used here envisages the achievement of privacy requirements as the primary goal of an organisation. Privacy-process patterns are used to describe the effect of privacy requirements on business processes. This method facilitates the identification of suitable system architecture to support privacy requirements. It is shown that PriS can be used as a general method to create privacy-compliant IT systems. The paper identifies the need for improved automated tools to facilitate the application of PriS. Specifically, they are improved using fuzzy modelling in the selection of implementation technologies.

Kalloniatis, Kavakli and Kontelis (2010) offer a PriS-based conceptual framework and a case tool to support the PriS way of working. This PriS tool helps developers to design a goal process model for an organisation. This allows them to monitor the impact of privacy requirements on an organisation's goals and processes, and to suggest a set of approaches for the realisation of privacy-related processes, as well as guidance for their implementation.

Shapiro (2012) identifies that the privacy risk analysis of composite socio-technical systems suffers from the lack of an adequate risk model for the implementation of Fair Information Practice Principles (FIPPs). By interrelating an enhanced privacy risk design that moves away from FIPPs and an integrated anonymisation framework, the selection and implementation of anonymisation as a privacy risk control can be more systematically considered and carried out. The Science and Technology Directorate of the U.S. Department of Homeland Security has sponsored development of both an integrated anonymisation framework and an enhanced privacy risk model to support more effective privacy risk management (Shapiro, 2012). Both are described at a high level and their interoperability is illustrated by the Google Street View controversy.

Nithya and Subha (2013) use the PriS method to address privacy requirements. They observe privacy variables and map the degree of participation in each privacy variable relative to the interval. The Dempster-Shafer rule of combination is used to implement the privacy requirements. Their study suggests a modified combination rule based on an ambiguity measure.

Panusuwan, Batlagundu and Mead (2009) developed the Security Quality Requirements Engineering (SQUARE) method, addressing the privacy question. Their report examines privacy definitions, privacy regulations and risk assessment techniques for privacy. The researchers selected two risk-assessment methods – Privacy Risk Analysis for Ubiquitous Computing and STRAP – which were applied to two case studies. The results indicated that neither approach was ideal, suggesting that a different method is needed for maximum privacy benefits to be achieved.

Jensen and Potts (2007) used STRAP to address requirements issues and as an example of an augmentation method for NFRs that can be used by intelligent people with no prior training in goal-oriented analysis or privacy considerations. They conclude that it is quite possible that the strategies adopted in devising STRAP could be used for other NFRs and for the augmentation of methods other than those based on goal refinement. It could also be used for other goal-oriented methods and may be compatible with teleological actor frameworks such as  $i^*$ , as well as with basic structured analysis and top-down function decomposition.

He and Antón (2007) present a framework for modelling privacy requirements in the role engineering process. Role engineering entails defining roles and permissions, as well as

assigning permission to roles. This is the first step towards implementing a Role-Based Access Control (RBAC) system and is, essentially, a requirement engineering process. The framework given includes a data model and a goal-driven role engineering process. The study seeks to bridge the gap between high-level privacy requirements and low-level access control policies by modelling privacy requirements as the contexts and obligations of RBAC entities and relationships. A healthcare example is used to illustrate how the framework operates.

### **2.5-6.Limitations of previous privacy frameworks**

Existing tools such as the PARCTab system (Partnership Against Cancer, 2014), the Context Toolkit (Dey, Abowd & Salber, 2001) and iROS (Johanson, Fox & Winograd, 2002) provide assistance for the construction of Ubicomp applications but do not offer features for managing privacy. This makes the design and implementation of privacy sensitive Ubicomp applications difficult. Consequently, there is little guidance from system developers and a lack of programming support in constructing architecture and user interfaces that are efficient in helping end users manage privacy. The result is that privacy is handled in an *ad hoc* manner, often as an afterthought (if at all), leading to the creation of applications that end users may ultimately reject because they are uncomfortable using them or they find them intrusive. Kalloniatis, Kavakli and Gritzalis (2008) sought to detect the impact of privacy goals on the goal process structure automatically, using PriS, but further refinement of the study is required via the use of fuzzy modelling, as suggested above. In Jensen *et al.*'s (2005) method, STRAP needs to be applied further, on a real software development cycle, to see whether iteration and refinement will be adequately supported. Hong and Landay (2004) need to concentrate on building more applications using Confab.

## 2.6 Conflict

As described above, security and privacy requirements are important for every software system environment although conflicts among software requirements are unavoidable because they must be eliminated. Conflicting requirements form the key reasons for inconsistencies in software development.

A software system is considered successful in its development when the systems strictly follow complete, consistent and clear-cut requirements. However, even when meticulous requirements are set, conflicting requirements cannot be avoided. There are various definitions of the term ‘conflicting requirements.’ According to Schär (2015, p. 98-109), “Conflicting requirement is a problem that occurs when a requirement is inconsistent with another requirement”. Additionally, Kim *et al.* (2007, p. 417-432) further provide a useful definition of requirements conflict by stating that “The interactions and dependencies between requirements [that] can lead to negative or undesired operation of the system”. This literature review presents the wide variety of research which has taken place in relation to the term ‘conflict’ in software systems.

### 2.6-1. Definition

Conflict in this context can be broadly defined as a clash of interest from the security side against the privacy side. Specifically, conflict can consist of an agent being assigned to a task, but without permission to handle it. It can also consist of an agent assigned to do a task by a dependent agent who has no permission to assign it. Usually, conflict occurs at the stage of goal and requirements setting. We can analysis conflicts by introducing new goals and transforming goal specifications. Conflicts are often found among users who have different perspectives on solutions to a problem, and system design exercises arrive at too many goals.

In addition, there is no common agreement on what conflict among requirements really means.

It can mean detection of conflict among requirements. In addition, there are no common types of conflict among requirements, so we are unable to automatically detect and solve them. Most techniques for resolution only consider binary conflicts. Conflicts among three requirements have not yet been considered. Having no systematic support for detecting conflicts, it is difficult to identify conflicts at the level of goal or requirements setting. Also, the lack of systematic techniques for mitigating conflicts at the goal or requirements levels adds to the challenge. The only solution we have is to propose a set of procedures for restructuring objects involved in conflicting goals.

Generally, a conflict could occur at any level of system development. The levels at which we find conflicts are among goals at the goal level, among requirements at the requirements level, and between the goals and requirements levels. Conflicts could also be found at the technical level and implementation level. Therefore, if we suspect conflict, we must determine and create a model that would analyse conflicts between the requirements and suggest tools to mitigate these conflicts. Providing reasoning models is crucial to detect conflicts, and this would help system developers to realise conflicts and resolve them effectively.

### **2.6-2. Conflicting Requirements**

The need to account for multiple security and privacy requirements simultaneously has been known to result in conflicting requirements, because two or more goals may not be entirely compatible with one another (Salado and Roshanak, 2014). This is quite normal, as goal conflicts need to be resolved at the business level in order to ensure that the development of a supporting system is implemented in accordance with business need.

This in turn creates a need to prioritize requirements and to identify which can be discarded

based on the business goal, should the need arise. Alternatively, new goals can be added throughout the design process, even as existing goals can also be modified. Business goals can result in significant changes to the requirements set. Yet too many goal changes at the business level can result in a very unstable system or even a failed project.

According to Ramadan (2020), a few existing approaches are available to deal with different types of data protection requirements in the early stages of development. These approaches focus on the identification of security and privacy requirements in the elicitation phase without detecting conflicts between them. The output of these approaches is usually a set of textual requirements. Relying on textually specified data protection requirements to manually discover conflicts is a difficult and error-prone task for two main reasons.

First, conflicts between the data protection requirements depend on the context of how the technical and organisational components of the target system interact with each other. Specifically, conflicts not only result from trade-offs between requirements related to the same asset in the system (e.g., anonymous vs accountable execution of a task), but also from those related to different assets. For example, a task may be required to be executed anonymously while writing data to a secure data storage where the identity of the writer must be known for accountability reasons. The detection of such conflicts requires an understanding of the underlying business processes and their included interactions between security and data-minimisation requirements, which is a difficult task if the requirements are provided in a textual format and distributed through multiple documents (Ramadan, 2020).

On a more general level, conflict may also occur when users hold perspectives of a problem for which developers and/or stakeholders have failed to account. In order to detect, identify

and ultimately resolve these conflicts, most of which occur at the goal or requirement level, there is a need to create an appropriate model. For instance, certain conflicts can be anticipated and pre-empted by introducing new goals or altering existing goals and/or objects, as necessary (Ramadan, 2020).

Among the most common problems encountered in this regard are, first and foremost, a lack of agreement or consistency among requirements. Second, developers tend to focus excessively on binary conflicts, while paying little to no attention to conflicts among multiple simultaneous requirements. Another problem is the lack of systematic support for the detection of conflicts and issues at the goal or requirements level, as well as a lack of systematized procedures for resolving such conflicts. As such, there is a need to establish standard operating procedures for restructuring objects in such scenarios (Ramadan *et. al.*, 2018, Ramadan, 2020).

### **2.6-3. Causes of Conflicting Requirements**

Massive quantities of requirements can lead to conflicts. The number of conflicts increase exponentially with the number of requirements in a typical modern software development scenario. Changes in requirements during system development phases causes many conflicts if the requirements and features were systematically analysed and taken into consideration for laying the foundation framework of the software. These changes may occur after the addition of new requirements or the update of old ones. Complex system domains can lead to the misunderstanding of requirements, and therefore, cause conflicts between them. Social difficulties that lead to requirements conflicts are as follows:

- The system has different stakeholders with diverse interests which usually interact with each other and cause conflicts; and
- Changes in the system's stakeholders by adding new stakeholders with different needs or by changing the stakeholders' requests.



Therefore, there are different sources for inconsistencies between requirements and these may cause problems in the success of software development. Researchers have been working to find various solutions for this problem. Usually, conflicts come from: 1) misinterpretation, 2) conflict design, 3) conflict terminology, or 4) other sources of conflict. These are solution components (sometimes those components are not available or need to have permission or authorisation to access or use), or conflict could exist between constraints (state of constraints), resource usage and capability, evaluation of priority and perceived needs.

In this research, we will consider conflicts at the requirements level between security and privacy requirements since many conflicts occur at the requirements level. As they say, the devil is in the detail, and as the requirements become clearer, the potential for conflicts likewise multiply and actual conflicts materialise and become clearer. The impact of doing this work at the requirements level is huge, as it allows developers to clear away conflicts and propose suitable interventions for doing so.

The importance of dealing with conflicts between requirements for successful system development is widely recognised. Conflicting requirements are usually detected in a continuous manner during a system's development. In fact, this has led the development of design processes toward iterative approaches to achieve high levels of effectiveness (Buede, 2009; Walden, 2015). Different identification approaches are employed, however, during a system's development. While conflicts between requirements naturally emerge during detailed design and testing activities, they must be actively sought during the early phases. Since cost of repairing defects increases as a system's development matures, early identification of conflicting requirements is therefore of paramount importance for successfully developing a system (Boehm and Papaccio, 1988).

Several approaches and techniques to mitigate those conflicts have been proposed. However, literature on identifying such conflicts remains scarce and often vague, particularly in the field of systems engineering. Existing work primarily concerns software systems, but as will be discussed below, the results of such research are only partially applicable to systems engineering due to the focus on logical statements and their isolation from the laws of physics and social laws and regulations. The identification and resolution of conflicting requirements is not only of concern in large-scale systems but can also be found in various other domains, i.e. in software systems (Robertson and Robertson, 2012), embedded systems (Eisenring *et al.*, 2000), antenna systems (Skou, 2003; Chen *et al.*, 2012), financial systems or even personal decision systems (Vartiainen, 2008) or legislations (Domec *et al.*, 2008).

The absence of conflicts between requirements or, in a different terminology, the consistency of a set of requirements is therefore considered a to be quality of good sets of requirements in the existing literature (Carson *et al.*, 2004; Hood *et al.*, 2007; INCOSE, 2012; Kar & Bailey, 1996; Katasonov & Sakkinen, 2006). Conflicting requirements can be defined as those in which “the solution to one requirement prohibits implementing the other” (Robertson and Robertson, 2012). Furthermore, goals can be implemented even when in conflict; however, one will negatively affect the other. Some authors in the field of software systems have further granulated the meaning of conflicting requirements. For example, Liu and Yen (1996) propose that conflicting requirements do not necessarily imply they are mutually exclusive, but only that they reduce the available solutions to some degree. The extreme would be reflected by so-called mutually exclusive requirements, in which a solution would be certainly impossible. In fact, the concept of conflicting requirements has been also categorized in various ways in software systems. Though not used in the proposed framework in the following chapters, it can

be quite valuable to consider and cite- a comprehensive categorization proposed in Van Lamsweerde et al., (1998), is as follows:

- 1) process-level deviation, which indicates inconsistency between a process-level rule and a specific process state;
- 2) instance-level deviation, which indicates inconsistency between a product-level requirement and a specific state of the running system;
- 3) terminology clash, which indicates that a single real-world concept is given different syntactic names in the requirements specification;
- 4) designation clash, which indicates that a single syntactic name in the requirements specification designates different real-world concepts;
- 5) structure clash, which indicates that a single real-world concept is given different structures in the requirements specification;
- 6) conflict, which indicates that a set of two or more assertions cannot be fulfilled at the same time;
- 7) divergence, which indicates that a set of two or more assertions cannot be fulfilled simultaneously for a given scenario;
- 8) competition, which indicates divergence for a single requirement;
- 9) obstruction, which indicates divergence with only one assertion.

#### **2.6-4. Conflict Identification, Analysis and Mitigate Approaches**

Requirements engineering is an important part of software development which plays a significant role in the software project's success. However, providing incorrect requirements may cause detrimental effects, and this may lead to the project's failure. This, therefore, is a primary problem affecting software and the threat of conflicting requirements, which can also occur when a requirement is not inconsistent with another requirement. Consistency between

requirements necessitates that two or more requirements contradict each other. In requirements engineering, the term ‘conflict’ means that there is an interference, interdependency or inconsistency between conditions or that they simply are not in tandem (Mairiza, Zowghi & Nurmuliani, 2009). Kim *et al.* (2007, p.417-432) define conflict requirements as: “The interactions and dependencies between requirements that can lead to harmful or an undesired operation of the system”. Therefore, conflicting requirements occur when the process of documenting and maintaining records do not match the set requirements.

This section provides a literature review of existing research on requirements conflicts. This study will determine a research gap and offer suggestions for future research. A comprehensive analysis will also be conducted to examine the various evaluations applied in the research. Following the study, it was of utmost importance to decrease the risks and detect requirements conflicts. Some of the proposed methods used manual techniques which efficiently analyse system requirements. The drawback of manual methods is that they can result in human error as well as inflated costs; however, they are substantial and important for analysing system requirements.

Moreover, most of the proposed approaches were not evaluated to measure their efficiency. In the end, important issues were given as general recommendations when introducing requirement conflicts techniques. Requirements engineering mainly concerns resolving:

- i) Customer requirements (its conflicting viewpoints);
- ii) Conflicting functional and non-functional requirements (e.g. level of functionality vs delivery time);
- iii) Conflicting non-functional requirements (e.g. performance vs reliability).

There are three suggested techniques for classifying the requirements necessary to make an easier identification of conflicts and redundancies. These are: partitioning, abstraction and projection. Abstraction is about identifying generalisations, wherein the highest and lowest levels of abstraction can be posed when considering solutions to any problem. Projection is also known as estimation of the likelihood or probability of occurrence once the problem is computed. Partitioning is about identifying aggregations, dividing a problem into smaller parts to be easily understood, and can be accomplished by establishing interfaces among the smaller parts. Through the use of partitioning, the identification of conflicts and priorities will be divided into smaller parts and analysed.

Although NFRs (non-functional requirements) are more critical than FRs (Chung and Supakkul (2004)). Their study aims to use NFRs to determine the trends and updates in ever-changing organisational policies, increase in need for interoperability with other software or hardware systems, and external factors such as safety and privacy regulations. The study aims to focus more on security and privacy regulations to analyse the factors external to the system and its development process. By analysing the factors external to the system and its development process, it is more appropriate to employ partitioning wherein the main problem will be divided into smaller parts for easy analysis of the underlying causes. Furthermore, partitioning is also relevant in analysing the acceptability of the system to its users and the public.

Furthermore, Salnitri *et al.* (2020) propose a novel method named SePTA (Security, Privacy and Trust Approach). This method supports a unified specification of security, privacy and trust requirements, under one framework. Moreover, it enables software designers and security experts to enforce such requirements. SePTA is designed for sociotechnical systems, i.e. complex information systems such as those of public administrations and large companies,

where there is an interplay between people and autonomous technical components, that collaborate in order to achieve common objectives. They focus on how security, privacy and trust requirements can be specified in the early requirement phase, using a goal-based modelling language, and how such requirements can be correctly enforced in the late requirement phase, using goal-based modelling languages and a modelling language for business processes. A business process modelling language was adopted for the definition of the late requirements since it can be used as a specification of how goals can be achieved.

More specifically, their work integrated and extended work from security, privacy and trust modelling languages, providing a method that introduced the following original contributions:

- i) providing a holistic requirements modelling and analysis approach that also included security, privacy and trust requirements, supporting both early and late requirements elicitation;
- ii) providing a software tool that offers automated functionalities that reduce the effort of the designers, by not having to repeat modelling tasks in order to analyse a different aspect of the system;
- iii) facilitating the solution discovery in terms of security and privacy, by providing patterns that address common issues;
- iv) providing a method to enforce privacy and security requirements.

Finally, they illustrate the application of the proposed approach and its benefits through a real-world case study from the domain of e-government. While this research work is recent and provides rich information regarding security, privacy and trust in sociotechnical systems, they have not considered the conflicting issue between requirements.

#### **2.6-5. Comparison between existing works in conflict requirements**

Many studies have been conducted regarding the decision criteria to use in non-functional requirements conflicts (Mairiza, Zowghi & Nurmuliani, 2009). Also, studies have been undertaken to determine an active catalogue among non-functional requirements (Mairiza and Zowgi, 2010a). This research paper, therefore, seeks to determine the need for conflict identification, analysis and resolve approaches.

One of the major aims of requirements engineering is to improve systems modelling (Robinson 2004). One of the techniques used to identify non-functional requirements is the informal technique whereby experts are hired to check inconsistencies within the non-functional requirements. After that, using an automation process, experts use tools to analyse the requirements of the system and consequently identify conflicts or potential ones. After detecting the these, experts resolve them and prevent any future breakdowns.

The other technique used to identify conflicts is known as the negotiation method. Here, the stakeholders and software engineers discuss the project orally and analyse the requirements as well as the conflict the project would be facing (Aldekhail, Chikh & Ziani, 2016).

Mairiza and Zowghi (2010a, 2010b) suggest that to manage conflicts, some techniques can be used when viewing, interpreting and evaluating NFRs. It is important to assess NFRs while knowing their impact on and importance to the system. There is a significant relationship between non-functional requirements and solutions. The primary requirements for functionality of all businesses include authorisation, reliability and delivery time (Poort *et al.*, 2004). The researchers also present a non-functional decomposition (NFD) model that gives a new classification for requirements. Primary functional requirements and additional requirements are classified as secondary functional requirements, quality attribute requirements and implementation requirements.

Mairiza, Zowghi & Gervasi (2013) apply an experimental approach to design a framework that manages the relative conflicts among NFRs. This is a suitable extermine intended to use the metric and measure of the NFRs with the functionality of the system and how to implement the functionality (operationalisation). The result of the experiment is the satisfaction level of NFRs in the system. A two-dimensional conflict relationship graph is created to determine whether there is a dispute between the two NFRs and the severity of any existing conflicts. This means that whether a conflict is strong or weak will determine the shape of the graph.

Moreover, a recent study conducted by Ramadan *et al.* (2020) examines the issue of detecting conflicts between data-minimisation and security requirements. They investigate how conflicts between security and privacy requirements gather into the systems, in business process models.

Two categories of conflict are considered here: absolute conflicts and potential conflicts. While the focus of their research is on business processes models, Ramadan et al.'s 2020 work will explore the early requirement stage, as it will help in understanding conflicts between requirements. For instance, within the healthcare sector, patients may have doubts about the privacy of their information. Their concerns regarding the way in which, and for what purpose their health information is being retained, can obstruct an organisation's responsibility to maintain sufficient documentation, ensuring complete accountability.

Sadana and Liu (2007) have proposed a framework to analyse the conflicts among non-functional requirements using the integrated analysis of functional and non-functional requirements. Conflict detection is performed on high-level NFRs based on the relationship between quality attributes, constraints and functionality. FR and NFR hierarchies are built and integrated to produce a high-level NFR while the conflict detection in NFRs is based on the



relationship among ISO 9126 quality attributes. Two types of conflict in NFRs are defined, which are: mutually exclusive and partial conflict. If the conflicts are identified subjectively, there is a lack of conflict analysis.

Mairiza, Zowghi and Gervasi (2014) propose a novel idea of utilising TOPIS (Technique for Order of Preference by Similarity to Ideal Solution) to resolve non-functional requirements conflicts. TOPIS is a goal-based technique for finding the alternative that is nearest to the ideal solution. The dimensional graph is important in solving the conflict requirements, and it is efficient because it shows the relationship between two NFRs. A decision matrix is then constructed based on the graph. The technique calculates the distance from each alternative to the ideal solution and chooses the final solution based on the maximisation of both NFRs.

Egyed and Grünbacher (2004) use an automated traceability technique to eliminate false conflicts and cooperation. Analysing the requirements is the first step, after which an identification of the requirements is made based on their attributes, which are: cooperative or conflicting. The trace analyser then automatically detects the trace dependencies among the requirements. The system aids in determining the extent requirements that overlap by using trade dependencies knowledge. If two requirements overlap, then the two requirements are conflicts. However, if there is no overlap between them, conflicts cannot exist. The word 'automatic' reflects the intention to use tools to analyse and detect requirements conflicts rather than doing that manually. Traceability techniques must be automated to identify false conflicts.

Some works cannot be categorised as manual or automatic methods. Thus, they are only considered to be general frameworks that detect conflicts between requirements. For instance, Mairiza and Zowghi (2010a, 2010b) demonstrate the results of the investigation and research

on NFRs conflicts that led to a catalogue of conflicts among NFRs. The catalogue is a two-dimensional matrix that represents the interrelationships among 20 types of NFRs. The importance of the NFRs may vary according to the system being developed. It is, however, important for NFRs to be viewed, interpreted and evaluated by various people so that there can be positive relations.

It is evident that to solve conflict in non-functional requirements, there must be a semi-automatic tool and empirical evaluation. A semi-automatic tool aids software engineers to undertake conflict management in case of NFRs. Empirical evaluation is used to assure certainty in the project. For instance, when a framework receives a proposal, it must be experimented to ascertain before implementation. NFR characteristics are carefully evaluated so that they are proven to work efficiently.

It is important that system requirements assess current actions. This means that software engineers' access what they are working on and the prior specifications of the program. Afterward, if their results fall below their expectations, they must develop a decision criterion to solve the arising conflicts.

Therefore, we can conclude that any conflict affecting non-functional requirements (NFRs) should be addressed immediately in order to not impact the quality of the software. To address these conflicts, it is important to develop a conflict catalogue to deal with all of these factors. If all the methods discussed above are implemented, software quality can be a major success for many companies.

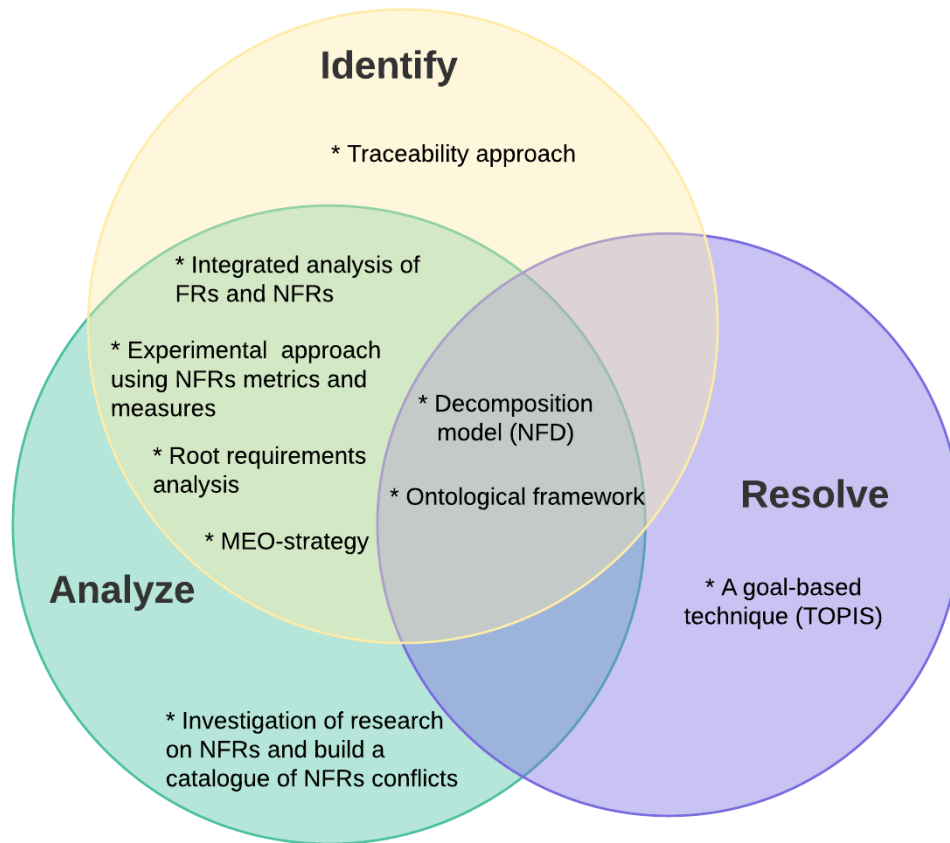
Aldekhail, Chikh and Ziani (2016) provide a comparative review on the conflict analysis approach, which was conducted with 20 studies from 2001 to 2014. As is shown in Table 2.3, we have two types of requirements as discussed earlier in this chapter. Moreover, some approaches focus on identify conflict, analysis and/or resolve (see Figure 2.2). For instance, for identifying conflicts in NFRs, the traceability approach can be used, while the goal-based technique (TOPIS) is applied for resolving conflict; moreover, for analysing conflict, an investigation of NFRs is utilised. Some are used interchangeably, for instance MEO-strategy and root requirements analysis can be applied for identifying and analysing conflict. On the other hand, for identifying, analysing and resolving conflicts, techniques such as the ontological framework and decomposition model (NFD) can be utilised.

As revealed from the table below, most functional requirements have been evaluated. Despite non-functional requirements, it is difficult to evaluate the approach as a result of conflict at NFR being complex and variable.

**Table 2.3 Comparison between conflict analysis approach**

conflict Analysis Approach	Type of Requirements		Scope of the Approach			Evaluation
	Functional	Nonfunctional	Identify	Analysis	Resolve	
Heuristic algorithm	X		X			
Root requirements analysis	X	X				X
Nonfunctional decomposition model(NFD)		X				
Integrated analysis of FRs and NFRs		X				
Model based in UML activity diagram	X					
Non-mathematical technique	X					
Ontological framework		X	X	X	X	
MEO-strategy	X	X	X	X		X
Experimental approach using NFRs metrics and measures as parameters		X	X	X		
A goal-based technique (TOPIS)		X			X	
Graphical method using problem diagram	X		X			X
Traceability approach	X	X	X			
Requirements partition in natural language	X		X	X		X
Tractability approach	X		X			X
Semantic based approach	X		X	X		X
Graphical method using NDT meta model	X		X	X		X
Graphical method using requirement goal graph	X		X	X		X
Validation rules	X		X	X	X	X
Three-level interaction detection framework	X		X			X
Investigation of research on NFRs and build a catalogue of NFRs conflicts		X		X		

(Aldekhail, Chikh and Ziani, 2016)



(Aldekhail, Chikh and Ziani, 2016)

**Figure 2.2 Conflict approaches dealing with Non-functional requirements**

Bhavsar *et al.* (2019) present a survey paper comparing recent studies of conflict between requirements in the early stage of development. In their survey, they summarise case studies related to different domains of software engineering with respect to requirement gathering techniques, and how conflicts can be resolved, which arise at the RE phase, using the Agile software development method. This model includes a continuous iteration of development and testing phases so that the product can be delivered in the early stage, meaning that Agile software development is used widely by companies around the world. While this is so, it also increases the complexity of the system.

The authors have also cited the work of Alkubaisy, Cox and Mouratidis (2019), who investigate conflicts between security and privacy requirements.

Maxwell, Antón & Swire (2011) also conduct a cross-reference approach for identifying conflicting software requirements. Their work reveals that rules and laws are easier to handle, and that the reputation of a company depends on the rules and regulations which are followed. On the other hand, this can lead to an increase in costs because system laws become overloaded. Furthermore, Schon, Thomaschewski and Eascalona (2017) investigate agile software development and discover that rapid change in requirements can be easy to handle, whilst on the other hand, more complexities arise when a hybrid development model is used.

Matsumoto, Shirai and Ohnishi (2017) explores verification of non-functional requirements. They suggest a refinement of the requirements, which simplifies makes the process. However, their approach is only applicable to common and basic non-functional requirements, rendering it incredulous to more critical systems. Similarly, Kaur and Sharma (2016) also examine non-functional requirements but implement extended use cases. Through use cases, the explanation of the system becomes more straightforward for the user, although creating use cases for NFRs can be complex compared to creating them for functional requirements.

Additionally, Sadana and Liu (2007) use integrated analysis of both functional and non-functional requirements. Their findings reveal that conflicts can be removed automatically, however, with greater complexity, performance can be degraded.

### **2.6-6. Dealing with Conflicts Between Requirements**

To provide a complete picture about how we could mitigate conflicts practically, different techniques are proposed by experts and software engineers. Described below are some techniques that are used to analysis conflicts between requirements. According to Aldekhail, and Ziani, D., 2017, there are a number of techniques which can be used. These techniques include: “rethinking the requirements; gathering the stakeholders and discussing and analysing the trade-offs of the conflicting requirements; and attempting to replace some of the conflicting requirements” (p. 91-95). On the other hand, Sepúlveda *et al.* (2014), propose the “use of group-techniques, a win-win model, the GORE and i\* diagrams” Espina and Scope (2016) propose that another solution could be found in “deploying a prioritization method that revealed a score based on the value, cost, and risk for the organization”.

Conflict resolution begins with conflict. The term points to “a situation between two parties that is characterized by perceived differences and that the parties evaluate as negative” (Katz and McNulty, 1994,). Additionally, different strategies enable the developer to resolve conflicts and assist users in having access to an optimum solution.

There are a wide variety of potential consequences of conflict. Whether it is neutral, positive or negative, the consequences involve different parties and a larger social system. A positive result of conflict can bring about pleasing results. “Conflict can bring opportunity, development, resulting in increased cohesion and trust” (Katz and McNulty, 1994). This can be the case when those involved in the conflict later realise that the conflict has led to an understanding of effective personal and organisational performance. Positive consequences for those involved in conflict can include reconciliation of the interests, interaction and clarification of the problem. Some conflicts can end with the notion and satisfaction that each

of the legitimate interests of all the parties involved were satisfied. Furthermore, with conflict, there is the promotion of interaction which can be useful to the requirements involved because interaction encourages the search for a solution and with communication, the real problem can also be identified. Negative results of conflict, on the other hand, are found when there are minor differences which can escalate into major conflicts, increasing the number of issues in the conflict.

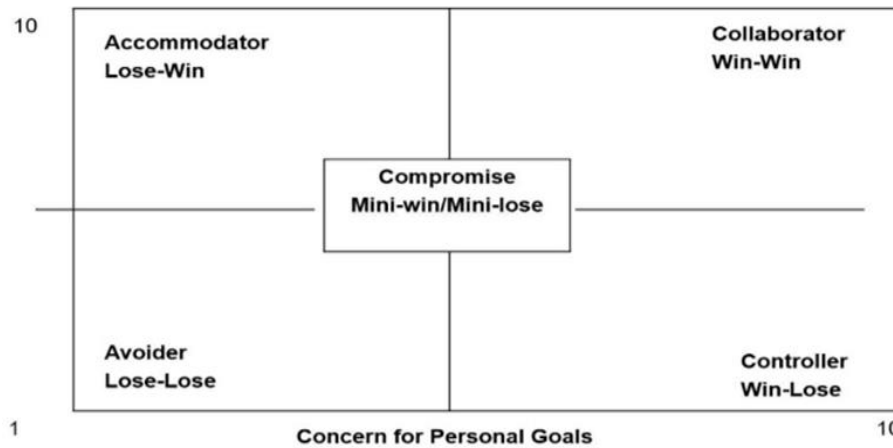
#### **2.6-7. Requirement Negotiation**

Conflict can also appear in some specific outcomes, labelled as dominance or imposition, withdrawal or avoidance, and compromise of resolution (Katz and McNulty, 1994). These outcomes are revealed depending on the approach used to deal with the conflict as well as the choice of alternative strategies. There are five basic approaches used to address the conflict situation in this type of circumstance. These are collaboration, compromise, accommodation, controlling and avoiding (Katz and McNulty, 1994), which will be discussed below.

One of the methods for addressing conflict is through the two-dimensional model for conflict as shown below. This model is based on the concerns around the task and relationship in the conflict situation. There are people who prefer to avoid conflict and others who confront conflict and try to find a solution, where each of the parties meet in the middle. Some people are concerned about compromising the relationship, so they abandon their interests in order to protect the relationship, while others attempt to take advantage of the relationship while protecting it at the same time. Still others are only concerned with gaining their own interest or advantage, and do not consider the damage they do to a relationship or other parties. How someone approaches conflict is determined by how he sees his relationship and the persons or



parties involved in the situation. There are five common responses to conflict as shown in the two-dimensional model below:



(Katz and McNulty, 1994)

**Figure 2.3 Two-Dimensional Model of Conflict**

Each window proposes an element of the two-dimensional model, which are explained as follows:

**Collaborating:** The purpose of the collaborative approach is to manage conflict by maintaining interpersonal relationships and enduring that all parties achieve their interests (Katz and McNulty, 1994). The act of collaborating lies in each requirement having a complementary role with the other requirements. In this model, both parties recognise that there is a conflict, and each utilises the appropriate problem-solving strategy to solve the problem. In this sense, a win-win solution is achieved when both requirements manage the conflict, and the solutions are favourable for both.

**Compromise:** The compromise approach to conflict assumes that “a win/win solution is not possible and adopt[s] a negotiating stance that involves a little bit of winning and a little bit of losing, with respect to both the interests and the relationships of the involved parties” (Katz

and McNulty, 1994,). The two styles of persuasion and manipulation are commonly used here. The objective of this approach is for both parties to come to a mutually acceptable solution which recognises that it does not completely satisfy all of the requirements of both parties involved in the problem.

**Accommodating:** In the accommodating approach, there is little concern paid to the interests of those involved with the requirement in the problem. Some of the ways in which accommodation is achieved are through one party giving in, appeasing or avoiding conflict which is done to protect the relationships between requirements. Thus, this approach is also called the yield-lose/win approach where a requirement would give way to another in order that there would be no conflict.

**Controlling:** In this approach to conflict, necessary steps are taken so that all of the interests of one party are met using whatever means is necessary to do so. Conflict in this approach is seen as a win or lose proposition. The idea of this approach is for one requirement to gain leverage over the other party and be regarded as the one that elevates their own status and competence. Hence, one party uses power to gain advantage or win over the other.

**Avoiding:** “The avoidance approach to conflict is to view it as something to be shunned at all costs” (Katz and McNulty, 1994). In this scenario, pressures of hopelessness and a high degree of frustration are expressed by all the parties involved. Each requirement has interests which are not met, and their interpersonal relationship is not nurtured by the experience. Using this approach, a requirement may avoid dealing with the issue by delaying the resolution of the issue or withdrawing from a threatening situation. For that reason, this is also called the leave-lose/win approach because one party decides to leave and so lose, allowing the other party to win.

Conflicts arise from many different situations and dealing with conflicts can be tackled with many approaches. Managing conflict requires the detection of discord between the different stakeholders or analysis. Negotiating these modifications must be done in order to find coherence and proper integration of an invention. In negotiation, many arguments and opinions propose modifications on the propositions. We can use negotiation to provide guides to assist in identifying conflicts and negotiate solutions. Analysis of conflict types as conditions of conflict methods, selection, and application, can thus help to create an index of its generic components.

#### **2.6-8. Requirement Prioritisation**

There are two meanings attached to requirements prioritisation. According to Sommerville (cited in Greer & Bustard, 1997), decision-makers acknowledge that requirements prioritisation is one of their most important tasks. Firesmith (2004) posits that this is a significant process in software engineering because it provides a perfect implementation for facilitating software versions and supplying reliable functionality, that defines the priority of the requirements to its stakeholders (Kousalya *et al.*, 2012). Requirements prioritisation is thus associated with prioritisation, depending on its importance or by implementation.

Requirements prioritisation techniques depend on the experts who rely on close communication with stakeholders and other requirements. In this sense, the task of proposing the right technique becomes more tedious and difficult, as well as that of making improvements in the techniques required and expected.

### **2.6-8-1 Requirements Prioritisation Techniques**

Three elements are needed to facilitate the need to prioritise stakeholders' requirements. These are software development or other projects that need such requirements, budgetary constraints and those that need to strictly follow its stakeholders' requirements. A point will also be reached where decisions may be needed relating to a specific set of requirements, those that need to be implemented first and those that could be delayed until needed.

There are different methods of discussing and presenting how requirements are developed. Some people can efficiently achieve the task when working on a small number of requirements, while others may prefer the challenge of working on complex projects that involve many decision-makers and other variables.

Thus, what is found in this study is the popular techniques used for requirements prioritisation. These techniques depend on the results of lower-level prioritisation activities compared to the computation needed to determine the requirements ordering (Vestola, 2010). Prioritisation techniques have three general scales that are used to present the results, as outlined below. Choosing which technique and method is most suitable for a given situation and then applying it to prioritise requirements is the objective of this research. This list of requirements prioritisation techniques provides an overview of common techniques that can be used for prioritising requirements (Figure 2.4) (Vestola, 2010).

#### **Ordinal Scale**

Ordinal scale prioritisation techniques produce ranked lists of requirements. Unlike ratio scale techniques, ordinal scale techniques cannot answer the question: 'How important is this one requirement when compared to another?'. In other words, these techniques can only tell us that

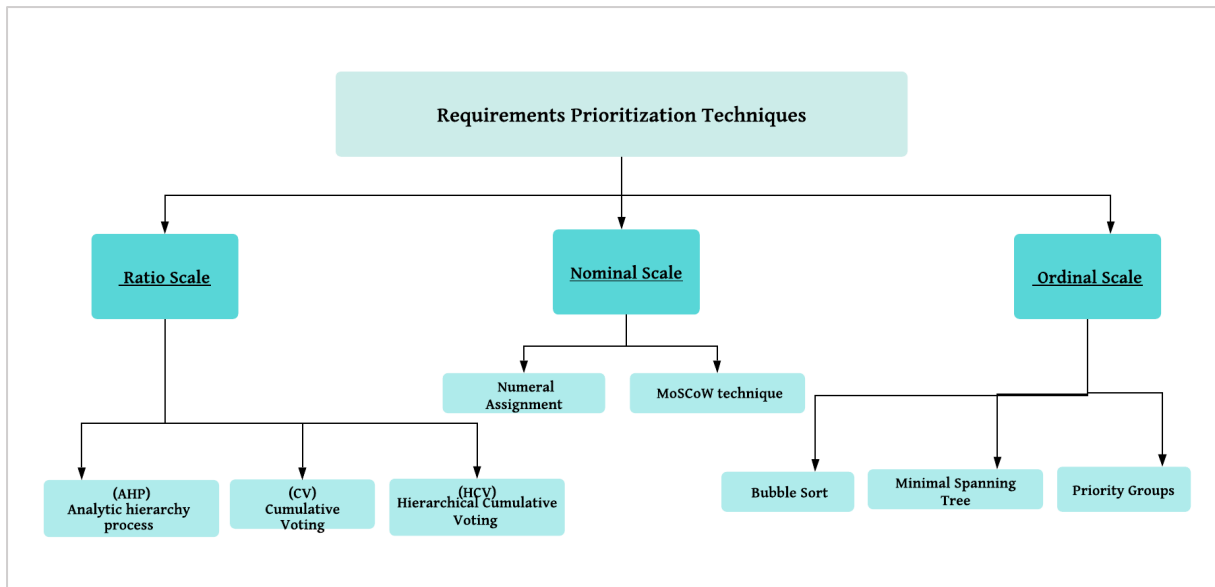
one requirement is more important than another but not to what extent. The following techniques are included in this category: priority groups, minimal spanning tree and bubble sort (Figure 2.4) (Vestola, 2010).

### **Nominal Scale**

Nominal scale prioritisation techniques produce lists of categories into which objects can be classified. In other words, requirements are categorised into groups based on their importance. All requirements in one priority group have equal priority. One cannot tell if a specific requirement is more or less important than another within the same priority group. Numeral assignment technique is the only technique included in this category. The MoSCoW technique is also included, however, it is basically a numeral assignment technique and thus is not included as a separate subsection (Figure 2.4) (Vestola, 2010).

### **Ratio Scale**

Ratio scale prioritisation techniques produce ranked lists of requirements. These techniques can answer the question: ‘How important is this one requirement when compared to another?’. In other words, these techniques can provide the relative difference between requirements. The following techniques are included in this category: analytic hierarchy process (AHP), hierarchy AHP, minimal spanning tree, cumulative voting (CV) and hierarchical cumulative voting (HCV) (Figure 2.4) (Vestola, 2010).



**Figure 2.4 Requirements Prioritization Techniques**

This research will apply one of those methods in the following chapter in a case study example, to give a practical application of our framework. After assigning requirements prioritization and having all requirements sorted, if there is a conflicting issue, we will investigate resolving techniques, which are tools to support the requirements the framework will support.

## 2.7 Chapter Summary

In this chapter, we have reviewed a number of areas including recent studies in relation to software and requirements engineering, security engineering frameworks and modelling languages; the importance of user privacy/security; and understanding resolutions to conflict.

Furthermore, we have undertaken a review of recent studies in relation to privacy and security, and the importance of protecting the confidentiality of users' information in terms of GDPR, cloud computing platforms and privacy principles, as mentioned at section 2.5. Although security has a lower importance than privacy, we reviewed recent security frameworks

regarding object and meta-level modelling (section 2.3), followed by the limitations of those frameworks.

Thereafter, we investigated the significance of user privacy in section 2.5, regarding its essential principles, role by design principles, previous frameworks and their limitations. Furthermore, we considered conflicts between requirements and how such conflict would affect the systems' security and protect personal data. As we have read in the news, problems like this could cost an organisation its reputation for reliability and safety.

With regard to conflict, we started with an overview (at section 2.6), after which we reviewed the causes of requirements conflict at section 2.6-3, considering recent studies which have explored conflict between requirements in general, identification of conflicts, analysis and resolution approaches. Most of the literature to date covers one or two of these factors as we mentioned at section 2.6-4 and we have explored which approach could be most helpful to our framework. As we discussed at section 2.6-5, a comparison was conducted between existing works in conflict requirements.

Having understood conflicts, to fulfil our framework we then considered mitigating conflicts at section 2.6-6, looking at requirement negotiation at section 2.6-7 and requirement prioritisation in section 2.6-8. This discussion would be very helpful to the analyst, enabling them to make a decision about which requirement to support. Finally, we have illustrated some supporting techniques to resolve this issue, as described at section 2.6.8-1.

The following chapter will introduce the research methodology, state of the art and types of methodology in existence within this field. Finally, we will discuss the selection of the most appropriate methodology for use in this research.



# CHAPTER 3

## RESEARCH METHODOLOGY

### 3.1 Overview of Research Methods

A research methodology is a way of solving a research problem (Kothari, 2004) systematically. In this sense, the research methodology is the science of systematic research. There are various steps involved in answering a research problem, including the logic required to undertake the research. A researcher must be adept not only at design but also at developing some aspects of the study such as tests and calculations, as well as using different research techniques. Additionally, a researcher must know how to determine which methods or techniques are most suited to his or her study. Furthermore, a researcher must understand the different assumptions involved in the various techniques and determine which procedures would be best applied to their problem.

Therefore, research has different areas and dimensions that form part of the research methodology. In research methodology, what is discussed is the objective reasoning behind the methods used in the research. One of the objectives in this research is that it will provide a new model that will clearly define and separate security and privacy. This will enable software engineers to analyse each of these dimensions in greater detail and understand the relationship between them. In addition, this enables software engineers to understand how security requirements and privacy requirements can co-exist in a system's design. Therefore, any issues that need addressing (in terms of potential conflicts) can be identified at an early stage of the development process.

Methodology also considers the reasons why certain methods or techniques are used and what makes such techniques appropriate compared to other methods.

According to Creswell, John and Creswell (2017), there is an established set of ideas which serves as groundwork for designing a research proposal. The researcher must consider four questions which would lead to a structured piece of research. These are to identify the “epistemology-theory of knowledge embedded in the theoretical perspective of the research, theoretical perspective-philosophical stance behind the methodology, methodology-strategy that links the choice and use of the methods, and the methods-techniques and procedures that are proposed to be used” (Crotty, 1998, p. 10). All the above-mentioned elements dictate the decisions that lead to the process of designing research.

Phillips and Burbules (2000) also provide key assumptions on research methodology and claim that evidence established in research is always imperfect and fallible, which suggests that there is no perfect research as there is a tendency for researchers to not disprove the hypothesis and show a failure to reject. Philipps and Burbules (2000) add that research makes claims with the idea of refining as well as abandoning some of these claims for a more preferred claim. Thirdly, data, evidence and rational considerations dictate knowledge. It is the researcher’s responsibility to collect information on instruments based on certain parameters determined by the researcher as completed by participants or by observations. Also, it is the researcher’s responsibility to expand his knowledge, as then and there he can identify the gap that this research will fulfil based on scientific knowledge. Fourth, the goal of research is to seek the relevant statements that can explain the situation or interest. Finally, research must be factual, which means exploring the essential aspect of competent inquiry and examining the different methods and conclusions for bias. This part of the research particularly validates and evaluates

the method that the researcher is working on, therefore, to provide a concrete contribution to knowledge.

### **3.2 Research Approaches**

There are two basic approaches to research – the *Quantitative Approach* and the *Qualitative Approach* although some research uses a mixture of these methods by combining quantitative and qualitative approaches. A quantitative approach is defined as one in which “the investigator primarily develops knowledge and employs strategies of inquiry such as experiments and surveys and collects data on predetermined instruments that yield statistical data” (Creswell, John & Creswell, 2017, p. 490-495). On the other hand, the qualitative approach is used by researchers who are concerned with “subjective assessment of attitudes, opinions and behaviour” .Such an approach to research results either in the non-quantitative form or in a form which is not subjected to rigorous quantitative analysis (Kothari, 2004, p. 5).

According to Dybå *et al.* (2011), numerous organisations in the software industry distinguish that software development presents a few unique management and organisational issues which need to be addressed and resolved for the field to progress.

Frequently software engineering research applies the qualitative approach, for complex software engineering issues can be difficult to study using a purely quantitative approach. However, qualitative studies can generate well-grounded hypotheses and findings that integrate the complexity of the phenomenon under study.

In addition, the qualitative method offers prosperous clarifications and new areas for future study. It is also suitable when variables are not defined or quantified and there is little prior

theoretical or empirical work. Finally, the main advantage of using qualitative methods is that they force the researcher to explore the complexity of a problem rather than abstracting it away, and the outcomes can therefore be more informative. For these reasons, the qualitative approach is most suitable for the field of software engineering and thus we use this approach within this piece of research.

### **3.2-1 Strategies related with the Qualitative Approach**

There are many strategies involved in qualitative research. These strategies include ethnography, grounded theory, case studies, phenomenological research and narrative research. Ethnography involves a systematic strategy for observing and studying people and their cultures. It aims to explore the culture in which the researcher makes his observations through the subject of the study. “Ethnographies, in which the researcher studies an intact cultural group in a natural setting over a prolonged period by collecting, primarily, observational data” (Creswell, 2017).

Secondly, grounded theory involves the idea of the researcher attempting to conjure a theory of a process, action or interaction from among the participants of a study. Thus, the study would involve different stages of data collection and careful adjustment and categorisation of information. The third strategy is case studies which involve having the researcher explore “in depth a program, event, activity, or a process of one or more individuals” (Creswell, 2017).

Fourthly, phenomenological research involves observing human experiences as the participants in the study describe it. The personal experiences determine the philosophy as well as the method and procedure used in observing a small number of participants. Finally, narrative research is a form of inquiry in which the researcher studies the lives of individuals and asks

one or more individuals to provide stories about their lives (Clandinin & Connelly, 2000). Hence, narratives contain the points of views of the participants in collaboration with the researcher's life.

### **3.2-1-1 Case Study as a Research Method**

It is believed that the employment of the case study method will best suit the needs of this current piece of research. Case studies can either be qualitative or quantitative and may involve fieldwork, looking through archival records, observations or any other combinations of these data gathering strategies. In the same sense, using any of these methods does not always lead to a case study. It could also result in ethnographic or observational research.

It is evident that a case study is similar to an experiment or a history study which can be linked to another particular data collection strategy. Thus, a case study attempts to examine a “real-life phenomenon where boundaries between the phenomenon and context have not been clearly defined” (Runeson and Höst, 2009, p. 131-164). An observation of a phenomenon is different from experiments because experiments steer away from phenomena. Instead, experiments deal with conditions and states of a context. Additionally, history is different because it is based only on the past and relevant information related to it. These differences are what lead to a case study.

Case studies have long been a reputable means of data collection. According to Zainal (2007), case study research, through reports of past studies, allows the exploration and understanding of complex issues. It can be considered a robust research method particularly when a holistic, in-depth investigation is required. This method is widely used in many studies including “social science studies where in-depth explanations of social behaviour are sought after” (Zainal,

2007). There are thus several aspects of case studies which bear exploration. Because case studies are used as an important tool in most social science studies, they can often be observed in topics such as education, poverty, sociology, community-based situations and illiteracy, among others. Case studies have become popular because researchers recognise that there are limitations to using quantitative methods in answering social and behavioural questions. A case study permits the researcher to work beyond numerical results and interpret behavioural conditions. In this sense, a case study combines both quantitative and qualitative data which supports the explanation of the process and phenomenon that are completed through “observation, reconstruction, and analysis of the cases that are studied” (Tellis, 1997, p. 1-19).

A case study allows the researcher to carefully examine the data within a context; case studies often use a small population as subjects. Therefore, case studies attempt to explore and investigate first-hand experiences through a detailed analysis of “limited number of events, conditions, and their relationships” (Zainal, 2007). Yin’s (as cited in Zainal, 2007) definition of the case study research method is that it is “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, when the boundaries between phenomenon and context are not evident”.

In addition, case studies are preferred when working in topics dealing with software engineering research because they focus on the idea of occurrence in its natural context. However, determining what comprises a case study still varies and thus the quality of the result could also be compromised. Case studies do not necessarily create the same results using different execution of experiments. However, case studies provide information that leads to a deeper understanding of a phenomenon. Many researchers claim different opinions on the issue of case studies. There are scholars who state that case studies have “less value, [are] impossible

to generalize from, and are results of the researchers being biased, etc.” (Runeson and Höst, 2009, p. 131-164). However, it is essential to note that case studies are primarily used for exploratory purposes, with some researchers explicitly declaring this as a limitation of their study (Zainal, 2007).

Nevertheless, case studies can still be used to describe a phenomenon, in situations which call for generalisation. Case studies may also be used to explain instances through testing of existing theories. As such, case studies in software engineering often echo an additional need to improve or act.

### **3.2-1-2 Categories of a Case Study**

There are three categories of a case study. According to Zainal (2007) these categories are exploratory, descriptive and explanatory. An explanatory case study is aimed at exploring a phenomenon which is set to be the researcher’s point of interest. In descriptive case studies, on the other hand, a natural phenomenon is observed and described within the data, which can also be presented in a narrative form. In explanatory case studies, the researcher “interprets the data by developing conceptual categories, supporting or challenging the assumptions made” (Zainal, 2007). A researcher must go further by including an interpretation of the phenomena observed in the study.

There are advantages to undertaking case studies. One key advantage is that according to Yin (1981, p. 97-114), “the examination of the data is most often conducted within the context of its use”, which means that the observation is taken within the situation and does not go beyond what the limitation calls for. Another key advantage of a case study is that it allows the “variations in terms of intrinsic, instrumental and collective approaches to case studies,

allowing for both quantitative and qualitative analyses of the data” (Yin, 1981, p. 97-114). Although this should not be confused with qualitative studies, both use qualitative results as the basis for their analysis. Thirdly, qualitative results from case studies provide researchers with results that are taken from real-life situations and which cannot be gathered from experimental or survey studies.

### **3.2-1-3 Limitations of a Case Study**

As mentioned earlier, no research methodology is perfect. However, there are appropriate conditions that will drive the researcher to lean on a particular methodology. Thus, there are some important limitations of a case study that should be considered. First, it should be noted that case situations are not always comparable, and thus the information gathered in case studies can also not be said to be identical (Kothari, 2004). Secondly, some authors feel that case studies present scientific data because they show objective knowledge that reflects “impersonal, universal, non-ethical, non-practical, repetitive aspects of phenomena” (Kothari, 2004). Thirdly, there is a danger with case studies of achieving a result that reflects false generalisations since there are not set rules to be followed in data collection. Fourth, case studies are time-consuming and cost. There is also a threat to the subjectivity of the researcher, which could be based on any number of assumptions, some or all of which might be unrealistic. Finally, case studies can only be used in a limited situation. It is not possible to expand the outcome of research by looking at a larger society or population. In addition to this, sampling is not possible in case studies.

### **3.3 Validity of the Framework**

As with any research, the researcher must seek validity for their study. Validity presents the accuracy, truthfulness and trustworthiness of the results of the study as well as the extent to

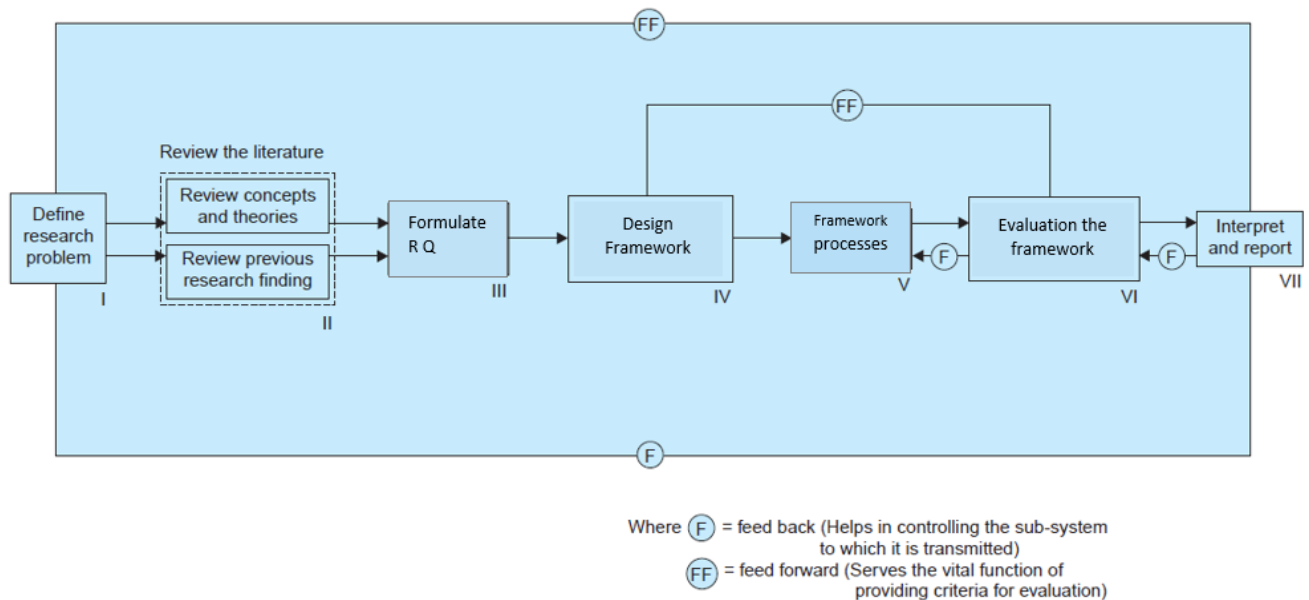


which the results can be deemed as true and unbiased by the researcher. Therefore, during the analysis of the results, the validity must already be considered and addressed. According to Runeson and Höst (2009), there are different classifications to the aspects of validity. These are construct validity, internal and external validity, and reliability. Construct validity shows the “extent [to which] the operational measures that are studied represent what the researcher has in mind and what is investigated according to the research questions” (Runeson and Höst (2009)). Internal validity, on the other hand, is concerned with causal relations. As such, causal relationships are examined as well as the factors affecting or relating to these relationships. The third kind of validity is external validity which deals with the extent of the generalisation of findings and that which concerns the interest of people in the case study, the participants. It is during analysis using this validity that the researcher attempts to analyse the result in relevance to other cases. Reliability is an aspect concerned with the extent of the data and the analysis which the researchers see as dependent on each other. Hence, if another researcher did the same study, they would expect to see a similar result. There are of course some threats to this study such as the clarity of the coding used in the collected data or the questionnaires and interviews (Runeson and Höst, 2009).

### **3.4 Research Process**

According to the research process flow chart shown in Figure 3.1, the activity levels I to VII are closely related, continuously overlapping due to the lack of a clearly defined sequence. In some instances, the first step significantly determines the nature of the last step under the condition that the subsequent procedures were not initially considered. As a result, there is a high likelihood of serious challenges that may end up interfering with the study’s completion. It is important to note that most stages of the research process are not distinct or mutually exclusive since there is no standard procedure that subsequent steps should follow. However,

the research flow requires a certain sequence of processes to coordinate and integrate the activities involved.



**Figure 3.1 Research Process**

## I. Formulating the research problem

The first step of a research process is the identification of the study problem depending on the choice of the issues or subject that interests the researcher. In the formulation of the research question, a generalised area of study can be considered as the researcher develops approaches to eliminate any possible ambiguities. The next step is to carry out a feasibility study to guide the formulation and quantification of the generalised study to narrow down the research field to a specific research problem based on an analytical perspective.

## II. Reviewing the literature

The next step after formulating the research problem is to summarise the topic. The researcher is required to write a research synopsis regarding the selected area of study and to submit the

research topic for approval by the relevant research committee. This is a critical step that requires the researcher to synthesise the available literature in the selected topic to connect past studies with the current research. The literature review stage involves abstracting and indexing all relevant resources ranging from peer-reviewed journals to publications to books and reports. The type of materials synthesised in this section depends on the nature of the research problem. Since most academic publications also contain related materials, finding relevant literature is not difficult, especially when the researcher is utilising a good library.

### **III. Formulating the research question(s)**

Formulation of the research question/s provides researchers with clarity of methodology and focus when carrying out a study. Findings from the existing literature are explored to identify both limitations and strategies that can be applied to resolve the present study's research problem. The specificity of the research question/s narrows the study, thus helping the researcher to avoid generalisation of the topic and aid the development of a clear, analytical and arguable thesis. Therefore, the research question/s should have a clear but complex hypothesis that can be critically argued and addressed either qualitatively or quantitatively. Put simply, a research question integrates a wide range of related processes that are carried out in organising the content and summarising the evidence to provide an answer.

### **IV. Preparing the framework design**

The establishment of a research framework design is required within a clear set of guidelines that explain the conceptual framework on which the research is based. This significantly increases the research efficiency and enables the writer to collect as much information as possible to address the research question. Moreover, research design plays an important role in reducing the financial and time costs of the study.

## **V. Framework processes**

Establishing a clearly defined framework process is another important step that a researcher applies in the SecTro Tropos based system model for conflict detection. This process is semi-automated and utilises special analytic tools to generate a list of standards to guide the resolution process, although the software engineer makes the optimal decision. The purpose of this framework is to identify the conflicting requirements and provide automated solutions based on predetermined conditions.

The functionality of the framework process is explained in more detail in this chapter. Once conflict requirements are identified, a list of practical techniques to support these requirements is generated. While this is not an exclusive list, it could be updated with the use of additional tools, serving as a guide to inform the analyst of the most up-to-date techniques and ways to choose the best method for their particular research problem. Different requirements will necessitate the use of different techniques, so strategies must be presented to help the analyst prioritise requirements. These can be tailored to suit stakeholder or end user needs, depending on the intended use of the system.

Moreover, the process of project execution greatly influences the entire research process based on the suitability in the application of the framework to several case studies. Framework validation is the next step after preparation of the framework design, especially in the field of healthcare management. Take, for instance, a case in which some patients are concerned about the level of confidentiality with which a healthcare organisation treats their personal information. The escalation of patients' concerns would ultimately influence the organisation's data management practices due to the need to reiterate its accountability for the privacy of patient information. In chapter four, the sequential process of applying framework design has

been explained based on the DEFEND<sup>2</sup> project commissioned by the European Union. Since the project is in its final stages, due to be completed March 2021, this requires complete refinement and application of the framework design in the process of conflict identification and the formulation of optimal improvement suggestions. This current research project first began in December 2018 and attempts to contribute to the DEFEND project in this way.

## **VI. Evaluating the framework**

This is an important step, which focuses on determining the project outcomes and providing important information to inform future initiatives based on the study results. The process of framework evaluation is implemented in the following two stages.

### **a. Focus Groups in person**

The Focus Group (FG) method is a specific qualitative research method. We supplement current research by providing guidelines for the method's use in software engineering research (Kontio *et al.*, 2003). FG is a form of qualitative research methodology used in the search for answers to questions. Its general characteristics include people's involvement (participants/moderator), a series of meetings, the homogeneity of participants (regarding their research interests), the generation of qualitative data and discussion focused on a topic, as determined by the purpose of the research. Generally, a FG's overall goal is to have the participants understand the topic of interest to the researcher, irrespective of its use, alone or together with other research methods so that it provides increased understanding or clarity on issues including previously obtained qualitative and quantitative data results (Freitas *et al.*,

---

<sup>2</sup> <https://www.defendproject.eu/>

1998). Focus groups are carefully planned discussions, designed to obtain the perceptions of the group's participants on a defined area of interest.

According to Langford and McDonough (2003), there are typically 3 to 12 participants involved in a focus group; the discussion is guided and facilitated by a moderator, who follows a predefined structure so that the discussion stays focused. Participants are selected based on their individual characteristics, which is related to the session's topic (this is called 'purposive sampling'). Furthermore, the group's setting enables participants to build on the responses and ideas of others, which increases the richness of the information gained (Langford and McDonough, 2003).

Unlike other methods, the FG offers an in-depth interview accomplished in a group, whose meetings present characteristics defined with respect to the proposal, size and composition of group and interview procedures. The focus or object of analysis is interaction inside the group, and participants influence each other through their answers to the ideas and contributions during the discussion. Furthermore, the moderator adds further richness to the research, as he stimulates discussion with comments or subjects. The success of focus groups is acutely attributed to gathering the right people into groups, creating environmental conditions for more spontaneous expression of each one, and facilitating the interaction of every participant. Eventually, the information gained from the group and the data produced by this technique are the transcripts of the group discussions and the moderator's reflections and annotations.

#### **b. Online focus groups**

For the benefits of having live focus groups, this study aimed to use that approach, but in light of the COVID-19 pandemic and lockdowns, the focus groups were run remotely. This does not

defeat the richness of the study but conveys the same or even added benefits to the richness of the research and its results.

Online focus groups are not a different type of focus group discussion, but one borne out of the introduction of the Internet as an adaptation of traditional methods. It is applied within the online environment, using conference calling, chat rooms or other online means (Kamberelis & Dimitriadis, 2005). Online focus groups boast an aura of dynamism, modernity and competitiveness that transcends classic problems with face-to-face focus group discussion (Edmunds, 1999). However, these discussion platforms are only accessible to participants with access to the Internet and are prone to technical problems such as poor or loss of connectivity and failure to capture non-verbal data (Dubrovsky, Kiesler & Sethna, 1991).

### **c. Steps in Focus Group Research**

Based on several academic sources on focus groups, we have summarised the main steps of focus group research into the following steps (Edmunds, 1991; King, 2004; Kontio *et al.*, 2004; Krueger and Casey, 2000; Langford and McDonough, 2003; Myers, 2004; Nyumba *et al.*, 2018; Stewart *et al.*, 2007; Tremblay *et al.*, 2010).

## **3.5 Defining the Research Problem**

The aim of the research problem is to provide a structure for the concepts, tools and methods that can be used to resolve an issue. It would not be wrong to see the research problem as the central focus of a thesis or dissertation, as the entire methodology, data collection, analysis and conclusion process depends on it.

The focus group method is best suited to obtaining initial feedback on new concepts, developing questionnaires, generating ideas, collecting, or prioritizing potential problems, obtaining feedback on how models are presented/documentated and discovering underlying motivations of participants. This method is not suitable for testing hypotheses, making final decisions, obtaining quantitative assessments (such as ‘how much’, ‘how many’), exploring issues with potential political/sensitive issues or studying complex issues that are difficult to grasp in a session, such as defining prices or cost preferences.

### **3.6 Planning the focus group session**

The focus group session usually lasts two to three hours and has a predefined schedule and structure. The content is determined and prepared beforehand in order to help participants and moderator/s to have a successful session. Moreover, the number of issues to be covered needs to be carefully planned so that enough time can be allocated for the participants to comprehend the issue and have a meaningful discussion and interaction about them. On the other hand, the limited time also creates a constraint on how complex issues can be addressed.

### **3.7 Selecting participants**

The value of the focus group method is overly sensitive to the experience and insight of the participants. Thus, the recruiting of representative, insightful and motivated participants is critical to the success of any focus group study. Depending on the type of research question, participants may be people that have much experience in the field, software engineering, requirements and modelling language. It may be useful to use pre-group questionnaires so that the session’s time is used most effectively for discussions.



### **3.8 Conducting the focus group session**

There must be careful management during the focus group session to ensure that all key contributions are made during the allocated time. The session should be initiated by an introduction in which the goals and ground rules are explained to the participants. The discussion and interaction in a focus group session can take many forms. It can be a structured discussion, where the moderator acts as a chair; it can involve brainstorming techniques, such as affinity grouping or teamwork methods; polling and voting using preference votes or the Delphi method; comparison games; or even role plays. Furthermore, each of the topics is usually presented one after another for better clarity. Langford and McDonagh (2003) present 38 different tools and techniques that can be used to supplement a traditional focus group discussion.

The role of the moderator is critical to the success of the focus group. The moderator is responsible for facilitating discussion, but they must do this while preventing their own opinions from influencing the discussion. The moderator's main task is to listen and probe deeper, when necessary, while grasping substance discussions quickly. It is often necessary to paraphrase participant's points to ensure that the contribution is correctly understood.

### **3.9 Analysing and Interpreting Data**

Methods used in qualitative data analysis can be used for the analysis of focus groups data, for example through content analysis, narrative analysis and grounded theory. Quantitative data, if gathered, can be analysed using descriptive statistics and other standard quantitative methods.

### **Developing and Pre-Testing a Questioning Route**

The questioning route is the agenda for the focus group. In the questioning route, the moderator is setting the direction for a group discussion which should closely align with the research objectives. There should be no more than twelve questions for a two-hour session (Krueger and Casey, 2000; Stewart *et al.*, 2007). Two general principals outlined by Stewart *et al.* (2007, p. 61) are to order the questions from the most general to the more specific and to order the topics by their relative importance to the research agenda. Thus, the topics to be discussed are ordered by importance, and within those topics, the questions are ordered from general to specific.

A promising evaluation approach in designing research focus groups is to create a manipulation within the focus group. Participants can be asked to collectively complete a task without an artefact and then again with the artefact. The ensuing discussion should revolve around how the artefact was used and how the completion of the task was altered by its use.

The two key research design goals for using focus groups are the incremental improvement of the design of the artefact and the demonstration of the utility of the design. For this reason, we have suggested two different focus group types, namely exploratory focus groups (EFGs), and confirmatory focus groups (CFGs). While the objectives of the two group types are very different, the methods of analyzing the focus group data from EFGs and CFGs can be similar. The interpretation of focus group discussions has many of the same challenges in demonstrating rigor that all qualitative research encounters share. Several techniques that are used for qualitative data analysis can be considered, carefully selecting those techniques that emphasize the reliability and replicability of the observations and results (Stewart *et al.*, 2007).

One possible approach is template analysis. Template analysis normally starts with at least a few predefined codes which help guide analysis. The first step in template analysis is to create

an initial template by exploring the focus group transcripts, academic literature, the researchers' own experiences, anecdotal and informal evidence, and other exploratory research (King, 1998). The contents of the discussions are also examined for their meanings and implications for the research questions.

Analysts will look for common themes and variations within the transcripts that would provide rich descriptions of the participants' reactions to design features. In template analysis, the initial template is applied to analyze the text but is revised between each EFG session. Once the final template is created after the final EFG, it is used to code the CFG sessions.

### **Thematic Analysis**

Thematic analysis is a widely used method in qualitative research. First named as an approach in the 1970s (Merton, 1975), it is as a method of identifying, analysing and reporting patterns (themes) within qualitative data (Braun & Clarke, 2006). It is used commonly because of the wide variety of research questions and topics that can be addressed, and through this flexibility, it allows for rich, detailed and complex description of the data.

A theme may be initially generated inductively from the raw data or generated deductively from theory and prior research (Boyatzis, 1998). With an inductive approach, the themes identified are strongly linked to the data themselves and may bear little relation to the specific questions that were asked of participants. Inductive analysis is a process of coding the data without trying to fit it into a pre-existing coding frame or the researcher's analytic preconceptions. In this sense, this form of thematic analysis is data-driven (Braun & Clarke, 2006). In contrast, deductive analysis is driven by the researcher's theoretical or analytic interest and may provide a more detailed analysis of some aspect of the data but tends to

produce a less rich description of the overall data (Braun & Clarke, 2006). Researchers must distinguish whether they are conducting an inductive or deductive thematic analysis as this will inform how themes are theorized (Braun and Clarke, 2006).

To produce a richer description of the overall data, and for a more data driven analysis, the inductive approach is used in this research. Here, themes identified are strongly linked to the data themselves, and data is coded without any pre-existing coding frame and/or analytic preconceptions.

Although there are several advantages to using this form of qualitative research method, we must not ignore its drawbacks. A simple thematic analysis is disadvantaged when compared to other methods, as it does not allow the researcher to make claims about language use (Braun & Clarke, 2006).

Furthermore, while thematic analysis is flexible, this flexibility can lead to inconsistency and a lack of coherence when developing themes derived from the research data (Holloway & Todres, 2003).

### **3.10 Report Results**

King (1998) suggests that qualitative results can be reported by creating an account structured around the main themes identified; drawing illustrative examples from each transcript as required. A similar approach can be taken when reporting focus group results. Short quotes are used to aid in the specific points of interpretation, while longer passages of quotation are used to give a flavor of the original discussions. Summary tables can be immensely helpful, displaying both evidence and counterevidence of the utility of the artefact by focus group. Rich descriptions can further corroborate results by using quotes from the focus group participants.

## **Preparation of the report or the thesis**

In conclusion, the researcher is required to prepare and table a project report that carefully addresses the following aspects in the body text.

(a) **Introduction:** This section should clearly state the project objectives and provide an outline of the applied research methodology. Also, important aspects such as the scope of the study and the challenges experienced in the research process should be indicated in this section.

(b) **Summary of findings:** In this part, the researcher presents a statement of the study results and actionable recommendations in the form of a summary.

(c) **Main report:** This is the major section of the report. It requires a clear and concise presentation of the entire study process following a differentiated logical sequence for easy identification and follow-up.

(d) **Conclusion:** This section finalises the study by providing the research results precisely and clearly while integrating the researcher's own thoughts about the project to summarise the study.

At the end of the report, the researcher should list appendices for all technical data that was collected during the study from materials such as books and journals. If the researcher has used a published report, an index should be provided.

The following chapter will introduce the framework and discuss how we can solve the problems we found in this chapter in order to gain a complete overview of the framework.

### **3.11 Chapter Summary**

This chapter seeks to present a general overview of the research methodology. The basic approaches to research – the quantitative and qualitative approaches – are examined, including

their strategies and their various pros and cons, these strategies being grounded theory, case studies and phenomenological research. Next, we look at the research processes and the use of flow charts, applying them to this research thesis. The relevant stages include formulating the research problem, literature review, formulating research questions, framework design, processes and evaluation. In evaluating the framework, the focus groups method – which plays a key role in this research – is examined in depth. Furthermore, thematic analysis, a widely used method in qualitative research, is introduced as it supports us in identifying, analysing and reporting patterns within the data. Lastly, a guide on how to write up and report results is shown.

# CHAPTER 4

## FRAMEWORK DESIGN

### 4.1. Introduction

When dealing with conflicts between security and privacy requirements, it is necessary to develop appropriate methods and techniques to ensure safety and confidentiality of user information and to make sure that neither are in conflict at any stage of the system so as to avoid potential risk to the system. Previous research shows that there is a significant conflict between security and privacy requirements. However, some studies have differed and stated that privacy is a part of security requirements, hence it would be appropriate to handle privacy as a separate requirement than security requirements. This approach has consequently failed to address the conflict between privacy and security requirements. It is important to analyse security and privacy under one framework (Islam *et al.*, 2012). This research thesis will therefore explore a modelling framework for security and privacy as part of the field of requirements engineering.

The proposed framework contains a formal representation and process that focuses on the requirements engineering stage. The language applies concepts from the requirements, security and privacy engineering domains, and is based on previous works of security requirements engineering – in particular Secure Tropos. The process follows the cycle of the requirements engineering process such as requirements elicitation and analysis.

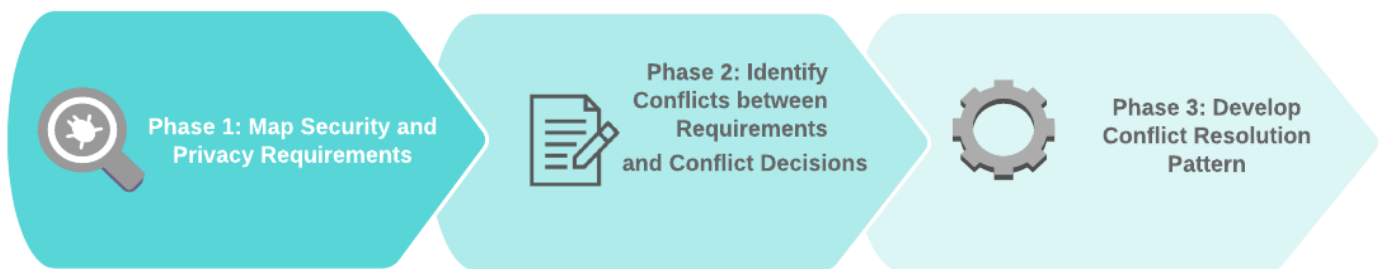
**The proposed framework determines the importance of:**

- Modelling the system using Secure Tropos tools.

- The semi-automated framework to support the analysis to identify conflicts between requirements by mapping between security and privacy requirements.
- Suggestions and strategies to analysis conflicts between security and privacy requirements.
- Having the suggested resolution automated. Since technology is developing, the possibility of finding a solution now is within our reach.

## 4.2 Theoretical Framework Phases

Our proposed framework has a sequence of phases to achieve conflict detection and resolution



**Figure 4.1 Phases of the Theoretical Framework**

As Figure 4.1 illustrates, some of the steps are semi-automated, while the others are manual steps, based on the analyst’s point of view. The stakeholders might have a special perspective on requirements. First, the conflicts between requirements are identified, based on a matrix built by previous studies, sorting requirements that could lead to potential conflicts. After identifying the requirements that are in conflict, the analyst must decide whether this kind of conflict would affect the system, based on the presented scenarios. Therefore, the first phase of the framework is performed manually by the software requirements analyst.



Phase Two identifies the potential conflicts between requirements that were detected in the previous phase. Additionally, the final phase proposes conflict resolution patterns by matching the problem to a resolution pattern for each conflict that the analyst might face. Those patterns act as a reference for the analyst to mitigate conflicts between requirements. This final phase of the framework process is to be semi-automated by supporting these techniques in an imported privacy pattern library. The developer inserts all possible supporting tools into the privacy pattern library to make it easier for the analyst to insert the supporting tool to solve any issue that might arise.

This research proposes a framework methodology which is used to organise and manage the study through a process. Eventually, as an outcome of the framework, we evaluate its effectiveness to determine whether there are any recommendations arising from the research to improve it.

At the beginning, we will use the requirements as an input. The first step is to model the system using Secure Tropos language, to articulate the problem. Secure Tropos has its own notation elaborated on in Chapter 2.4.3-1. After this, we reach the conflict detection phase, in which we can map requirements facing conflicts by using a matrix. Our output of this phase is a list of conflicts, as well to achieving the second contribution to mitigate conflicts between requirements. Here we can prioritise the requirements, in order to sort them out in numerical order, so that when an issue with conflicts between requirements arises, the analyst can make a decision based on prioritisation requirements.

#### 4.2-1 How the framework works:

In this section, we describe the steps of the framework through an explanation of each section, and how it contributes to our work.

##### 4.2-1-1 Phase 1: Identify Requirements: (Security and Privacy Requirements)

The first step of detecting conflicts is to review the literature to determine more about conflicting issues. This provides some examples to detect how conflict affects a system. Below are the most frequent requirements in the security and privacy aspects of software engineering.

*Security Requirements are:*

- **Authentication** is the process of determining whether an entity is in fact, what or who it is declared to be. This process involves validation of identity (Lopez, Oppliger & Pernul., 2004).
- **Authorisation** logically follows from the previous requirement. This is where the identified entity is provided permission to access data or functional resources based on set privileges (Lopez, Oppliger & Pernul, 2004).
- **Confidentiality** is the assurance of a capability to impose limitations of access to or exposure of a specific resource as mandated by a policy (Tange *et al.*, 2020).
- **Non-repudiation** is the facility that ensures accountability of actions. It is the association of actions or changes to a unique entity (Tange *et al.*, 2020).
- **Integrity** is the assurance and the maintenance of the accuracy and consistency of information over its life cycle; this requirement mandates that information should remain unadulterated (Tange *et al.*, 2020).
- **Availability** assures that data is always accessible and can easily be provided to an authorised entity. Denying information can cause both inconvenience and delays which may prove to be critical (Tange *et al.*, 2020).

- **Separation of Duties (SoD)** acts as a restricting agent for any individual to have too much or inappropriate control over the system (Ramadan, 2020).
- **Binding of Duties (BoD)** similarly ensures that two separate entities are needed to have sufficient control over the system (Ramadan, 2020).
- **Accountability** is the requirement that holds entities responsible for their actions or lack thereof (Ramadan, 2020).
- **Auditability** ensures that a trace can be done on an entity's activities within the system (Ramadan, 2020).

On the other hand, we have to consider *Privacy Requirements*. This is often in compliance with existing data laws or rules within a country. For a project to be compliant, they must be able to ensure privacy within the system. The relevant privacy requirements, according to Diamantopoulou (2017), include:

- **Anonymity** allows entities to use resources or services without having to reveal their identity.
- **Unlinkability** ensures that an entity can use a service without being associated with the service itself.
- **Pseudonymity** gives the users the freedom to work under an alias or aliases, without having to provide personal information sufficient to determine their identity.
- **Unobservability** denies any entity from knowing for sure that a user is accessing a service, as well as the inability to track a user's actions while using a service or resource.
- **Undetectability** ensures that an entity cannot identify which user among a user pool, is accessing the service.

We address these requirements based on their expected frequency within any system, as outlined in Table 4.1.

**Table 4.1 Most Frequent Security and Privacy Requirements being in Conflict**

<b>Security Requirements</b>	<b>Privacy Requirements</b>
Availability	Anonymity
Non-Repudiation	Unlinkability
Confidentiality	Pseudonymity
Integrity	Unobservability
Authentication	Undetectability
AuthoriSation	
Separation of duties (SoD)	
Binding of duties (BoD)	
Accountability	
Auditability	

Security and privacy requirements have a lot of potential for conflicts when considered together. For instance, **authentication** requirements warrant **disclosure**, but **anonymity requirements** are against **disclosure**. Furthermore, **integrity** requirements can conflict with **unobservability**. Digging further, one can see that regarding **integrity** requirements, there need to be mechanisms for tracking user behaviour across networks, for instance.

Additionally, when one demands requirements in security **authentication**, these requirements touch on revealing identity information and gathering as much information as possible relating to the authenticity of the identity of the user. However, this runs counter to the requirements for **anonymity** and **pseudonymity**, which require the disclosure of as little personal information as possible. Where requirements for security become dominant, what suffers are requirements relating to privacy. This conflict plays out in real life scenarios and therefore mimics life in

general. For instance, as far as the government are concerned, the focus is on making sure that it has as much information about citizens as it can. On the other hand, as far as private citizens are concerned, concerns relate to ensuring that the government does not encroach on the individual's personal life.

On the other hand, such conflicts with privacy requirements are tied to **unobservability** and **unlinkability**. Where privacy requirements dominate related to these latter two requirements, then the corresponding counter requirements in security are impacted adversely, and vice versa. These are potential and actual sources of conflict in requirements setting that need to be ironed out and resolved because giving leeway on one side compromises aspects of the other, and there are no easy set of answers that work in all scenarios. Examples of this situation are further illustrated in the following chapters.

**Authorisation**, a security requirement, also conflicts with the privacy requirement of **unobservability**, because the latter requires the preservation of the privacy of the party, while the authorisation requirement posits the proper identification of the party before being granted the go ahead, the authentication seal of approval.

It is therefore apparent that some aspects of security and privacy requirements are already in conflict with each other. Security requirements, such as **accountability**, **authenticity**, **auditability** and **non-repudiation**, require a log of movement and activity within the system. However, these lie directly in conflict with the privacy requirements of **anonymity** and **unobservability**, which should conceal the user's actions. On the other hand, **binding of duties** (BoD) and **separation of duties (SoD)** could conflict with **anonymity** and **unlinkability** as well, since the steps to be executed would have to verify their identity.

Other aspects may have conflicts which are not apparent at the requirements stage. For example, the security requirements **confidentiality**, **integrity** and **availability** depend on having proper authorisation to access or modify resources. Identification is not necessary to achieve these requirements, but it is a common approach which may be used by system developers. As such, it opens a potential conflict between the security requirements mentioned previously and data minimisation privacy requirements. Conflicts can also occur within each aspect, both in terms of security and privacy. These conflicts arise when more concrete requirements are specified. For example, if a user should be required to access a service using their alias, then it conflicts with the general concept of **anonymity**. However, if some aspects supplement or overlap concerning requirements, then they cannot be considered as conflicts. **Confidentiality**, **integrity** and **anonymity**, which are different aspects but ultimately strive towards the same goal of protecting data against unauthorised tampering, do not conflict with each other.

### **Mapping Between Security and Privacy Requirements**

The matrix maps conflicts between security requirements and privacy requirements. While there may indeed be conflicts among security requirements themselves, the matrix will focus on conflicts that cross the two aspects.

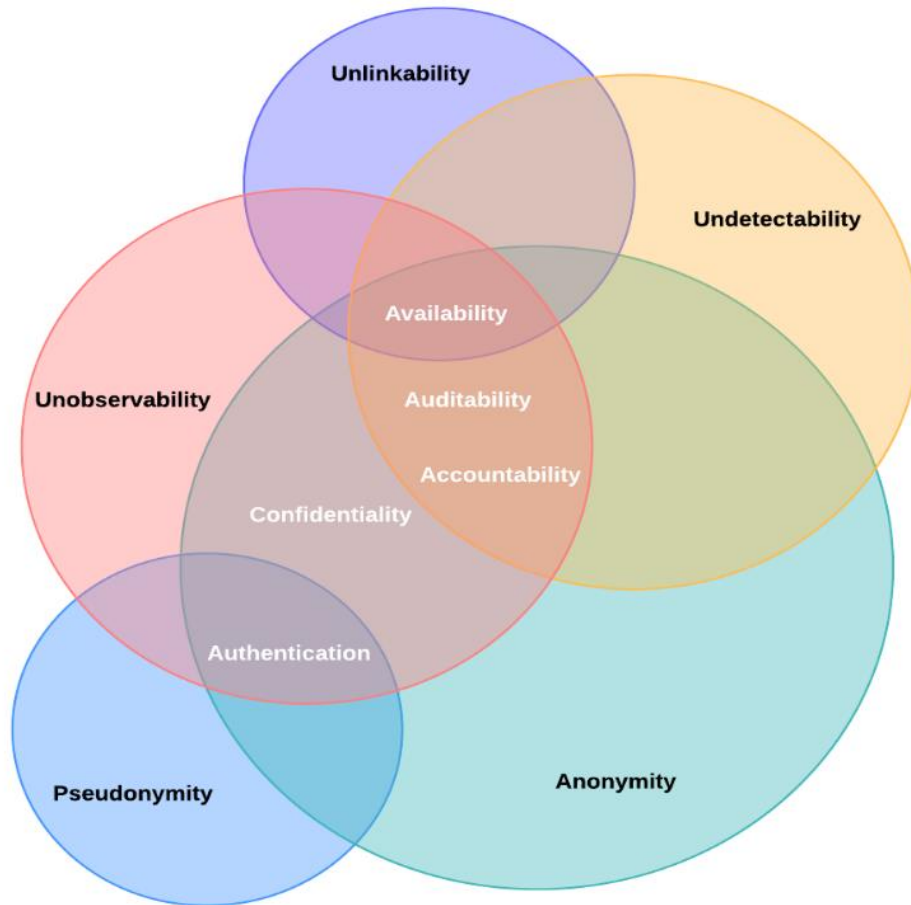
The matrix helps us to visualise the requirements with the most conflicts, which aids in identifying which deserve focus. From this matrix, **anonymity** and **unobservability** conflict the most with other security requirements (see Table 4.2). We will describe each requirement and the matrix in further detail in Chapter 5.

**Table 4.2 Mapping conflicts between Security and Privacy Requirements**

		Security Requirements:							Privacy Requirements:								
		Availability	Non repudiation	Confidentiality	Integrity	Authentication	Authorization	SOD	BOD	Accountability	Auditability	Anonymity	Unlinkability	Pseudonymity	Unobservability	Undetectability	
Security Requirements:	Availability							o				o	o		o	o	
	Non repudiation											o			o		
	Confidentiality											o			o		
	Integrity											o	o		o		
	Authentication						o					o		o	o		
	Authorization							o							o		
	SOD	o						o									
	BOD													o	o		
	Accountability												o			o	o
	Auditability												o			o	o
Privacy Requirements:	Anonymity	o	o	o	o	o			o	o	o	o	o	o	o		
	Unlinkability	o			o				o			o		o		o	
	Pseudonymity					o			o			o	o				
	Unobservability	o	o	o	o	o	o			o	o	o				o	
	Undetectability	o								o	o		o		o		

It is our purpose to generate more visual mapping to draw our attention to the most frequent requirements in conflict. Based on previous studies, those five security requirements which are likely in conflict with more than one privacy requirement, are depicted in Table 4.2 and Figure 4.2. This includes **availability**, as a security requirement is a more complicated requirement while it is involved with most privacy requirements, therefore it conflicts with four privacy requirements: **anonymity**, **unlinkability**, **unobservability** and **undetectability**. We next found that **confidentiality**, **accountability** and **auditability**, security requirements, could be in conflict with three of the privacy requirements. Finally, **authentication**, a security requirement, is always involved with two privacy requirements, **anonymity** and **pseudonymity** (see Figure 4.2). Furthermore, not all security requirements are addressed. Figure 4.2 shows the most/common requirements that are in conflict (to emphasise how this

conflict is complicated and related to privacy requirements). Moreover, some of the security requirements are not included because they have no conflict with privacy requirements.



**Figure 4.2 Detecting conflicts between Security and Privacy Requirements Venn Diagram**

#### **4.2-1-2 Phase 2: Identify Conflicts between Requirements and Conflict Decisions**

Conflicts arising between different requirements, such as privacy and security, are a common problem in engineering software systems. Conflicts in software requirements are inevitable because of the nature of software development for realistic systems. Every case of conflict based on requirements is surrounded by complex issues which should be taken into



consideration when mitigating conflicts. Security and privacy requirements conflict resolution should be considered essential for every software system.

Privacy has become a mainstream topic and is especially problematic for software development companies. Problems around misuse of presumed personal data by organisations, especially social media companies, has led to moves to ‘guarantee’ privacy at legislative levels, as envisioned in the EU’s General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017). However, from the developer’s point of view, certain issues crop up when adhering to security requirements, while others appear when adhering to privacy requirements. This can lead to conflict when trying to meet these requirements, and it is now necessary for developers to manage these conflicts in order to be compliant with GDPR.

For a brief example, to identify conflicts, we divide each scenario task to address the possible conflicts. For each case, we assign the involved requirements, as shown in Table 4.1. Based on the task scenario we would use, we will address the security and privacy requirements for each activity. For instance: a lab must perform a medical examination then send the results to the medical doctor (security requirements: confidentiality and integrity). These medical results will be sent to the medical doctor to update the patient’s medical record; this action must be compatible with the GDPR accountability principle. While the medical doctor is updating the patient’s medical record, this action should be done anonymously, but it is important to maintain accountability, for example when updating a medical record in case an audit is needed, because of an error in prescribing new drugs and the need for an investigation. This could therefore lead to conflicts between accountability and anonymity. To process the updated results, they should be verified by the supervisor; therefore, this requirement involves accountability as a security requirement. However, updating the patient medical record

involves anonymity to keep the patient's record private, according to Privacy-by-Design principles.

At this point we could have conflicts between anonymity as a privacy requirement and accountability as a security requirement. This task can require more than one requirement which leads to potential conflict between requirements, especially based on privacy and security requirements. It can be difficult to fulfil both requirements. Accountability is the requirement that holds each participant responsible for their actions; anonymity allows entities to use resources or services without having to reveal their identity. As discussed above, we have already identified a conflict between accountability related to the supervisor and anonymity related to the medical doctor. In this phase, we only highlight the conflict issue.

#### **4.2-1-3 Phase 3: Analysis conflicts based on Support Techniques**

Phase 3 offers a supporting tool that is suitable for security and privacy requirements. Here we list below (see Table 4.3) the most common tools and link them with the most suitable requirements. Research reveals that there are tools which can support both security and privacy requirements, while others support a privacy or security requirement. For instance, a supporting cryptographic tool is suitable for mitigating conflicts in security requirement *confidentiality*, but also for privacy requirement *anonymity*. Similarly, conflicts arising between both requirements can be mitigated by using cryptography. On the other hand, conflict arising between *audibility* and *undetectability* shows that only supporting tool steganographic technologies can be used. We will next look at each supporting tool in further depth.

**Table 4.3 Supporting Tools**

<b>SECURITY REQ.</b>	<b>Suitable Tool for this Requirement</b>
<b>Confidentiality</b>	Cryptographic, access control enforcement, Symmetric key and public key encryption, Steganographic technologies, Homomorphic encryption, Onion Routing, Searchable encryption
<b>Integrity</b>	Cryptographic, Accesses Control Enforcement, Message Authentication Codes (MAC) Redundancy and Comparison
<b>Accountability</b>	ADOPT, IDMEX
<b>Audibility</b>	Cryptographic, Steganographic Technologies, Onion Routing
<b>Non-repudiation</b>	Onion Routing, Dummy traffic
<b>Authorisation</b>	Accesses Control Enforcement
<b>Authentication</b>	Trusted third parties, Message Authentication Codes (MAC)
<b>Availability</b>	Redundancy
<b>PRIVACY REQ.</b>	<b>Suitable Tool for this Requirement</b>
<b>Anonymity</b>	Cryptographic, Steganographic Technologies, Onion Routing, Trusted Third Parties, Dummy traffic, K-anonymity, Zero-Knowledge Proofs of Knowledge (ZKPoKs)
<b>Unlinkability</b>	Cryptographic, Steganographic Technologies, Homomorphic encryption, Data Hiding, Onion Routing, K-anonymity, Trusted Third Parties, Dummy Traffic
<b>Unobservability</b>	Dummy Traffic
<b>Undetectability</b>	Dummy Traffic, Steganographic Technologies

**Support Requirement Tools**

**Cryptography**

According to Biswas, Das Gupta and Haque (2019), the encryption of data is a common way of security of implementing confidentiality. Standard examples include simple passwords, security tokens and two-factor authentication. Both symmetric and asymmetric algorithms may be used to provide encryption. Cryptographic protocols not only protect data from external threats and malicious attacks but should also capture threats arising from the execution

environment. This includes ‘bad interactions’ which may occur with other valid protocols running within the same system (Menezes, Van Oorschot & Vanstone, 2018).

### **Onion Routing**

According to Syverson *et al.*, (2001), Onion Routing builds anonymous connections within a network of Onion Routers, which work in, roughly, real time. Onion Routing’s anonymous connections are protocol independent and exist in three phases: connection setup, data movement and connection termination. Setup begins when the initiator creates an onion, which defines the path of the connection through the network. An onion is a (recursively) layered data structure that specifies properties of the connection at each point along the route, e.g., cryptographic control information such as the different symmetric cryptographic algorithms and keys used during the data movement phase. Each onion router along the route uses its private key to decrypt the entire onion that it receives. This operation exposes the cryptographic control information for this onion router, the identity of the next onion router in the path for this connection, and the embedded onion. The onion router pads the embedded onion to maintain a fixed size and sends it onward. Then all onion router in the path connects to a responder proxy, which will forward data to the remote application.

### **Steganographic technologies**

Fridrich (2013) classifies steganographic methods into three main categories: cover-selection, cover-synthesis and cover-modification. The cover-modification based steganography is the main body of steganographic techniques. The other two categories are concluded as non-modified steganography in this paper. According to information theory, there are artificially irreversible changes added into the cover in cover modification-based steganography, thus resulting in a distinct deviation when fitting the distributions of cover model and stego-cover

model under KL distance (Ke *et al.*, 2018). Thereafter, it continues to evolve to steganography by minimising additive distortion using Syndrome-Trellis codes. Steganography is based on non-modified steganographic methods, whose characteristic is that no modification occurs in the cover after embedding, it can effectively guarantee the indistinguishability, such as the coverless information hiding, and includes generative steganography based on GANs.

### **Homomorphic Encryption**

Homomorphic encryption is a form of encryption allowing one to perform calculations on encrypted data without decrypting it first. Homomorphic encryption can be used for privacy-preserving outsourced storage and computation. This allows data to be encrypted and outsourced to commercial cloud environments for processing, all while encrypted (Gentry, 2010). Semantic security of a homomorphic encryption scheme is defined in the same way as for an ordinary encryption scheme, without reference to the Evaluatee algorithm. If we manage to prove a reduction – i.e. that an attacker who breaks  $e$  can be used to solve a hard problem like factoring – then this reduction holds whether or not  $e$  has an Evaluatee algorithm that works for a large set of functions (Gentry, 2010).

### **Accesses Control Enforcement**

Traditionally, an access control mechanism is used to protect information stored in an information system on a host computer or server for security and/or privacy (Choy, 2000). It allows registered/recognised users (and applications/agents acting on their behalf) to access (read, append, update, delete, create, etc.) information stored in the system. For each user, access is restricted to only the information that they are authorised to access and only for the operation(s) that they are authorised to perform. Accesses are always initiated by a user or an

application program, and information protection ceases as soon as information leaves the information system.

### **Integrity: Message Authentication Codes (MAC)**

These are a fundamental technique to verify both the integrity and authenticity of transmitted data. Initially, the construction of most of these codes were based on pseudorandom functions which were generated either through fast block-cipher based algorithms or slower number-based theories. Each method has its own disadvantage: the cipher-based algorithms were found to have issues with efficient zero-knowledge proofs about authenticated data, while number-theoretic PRFs are comparably inefficient due to their dependency on number theory (Hayes & El-Khatib, 2013).

### **Availability: Redundancy to the system**

Redundancy, or the duplication of critical points of the system, ensures that an application is reliable and available for its intended users. Should a function or component fail, another instance would be ready to take its place so that the system can perform with little or no downtime (Leydesdorff, 2010). However, a disadvantage is that redundancy increases both cost and complexity of the system. An architecture that correctly models the system is vital to the success of high availability. While redundancy is not necessary in many applications, it is a critical component if system failure or downtime has severe consequences (Bhagwan, Savage & Voelker, 2003).

### **Pseudonymity: Public key**

Pseudonymity provides a consistent identity without having to tie it to a specific physical person or organisation. It allows for the advantages of having a known identity, such as

accountability, while still maintaining anonymity. One way to implement pseudonymity is through a public key that verifies digital signatures anonymously made by the holder of the corresponding private key. Users can create their own public keys for digital pseudonyms. Each key pair may be bound to an email address, self-certified and used thereafter (Deng & Kuzmanovic, 2011).

### **Undetectability: Steganography technologies**

Stenography is the art of invisible communication, a technique where data is transmitted in a way that conceals the existence of another message. Unlike cryptography, which only encrypts the message itself, stenography encrypts the message such that unauthorised parties would not be aware of a message at all. An example of stenography in images is the Least Significant Bit (LSB) method, which hides information within the least significant bit of each pixel. This method works well because a change in the least significant bit (0 to 1 or vice versa) does not drastically change the overall appearance of the image (Aos *et al.*, 2009).

### **Anonymity-Zero-knowledge Protocol**

This allows a party to prove that a statement is certainly true without revealing additional information. This protocol must have the following properties: Completeness – the honest verifier should be convinced by an equally honest prover; Soundness – the probability of satisfying a verifier that a false statement is true is minimal; and Zero-knowledge – that a cheating verifier can learn nothing from the statement but the truth. Usually, this method mechanism is used to maintain anonymity (Lam *et al.*, 2007).

## **IDEMIX**

According to Drijvers (2014), Identity Mixer (IDEMIX) is a solution for minimising the release of personal information and can be based on one of many proposed techniques for anonymising the transport medium employed between users and service providers. IDEMIX is an optimising cryptographic compiler that achieves an unprecedented level of assurance, without sacrificing the practicality for a comprehensive class of cryptographic protocols. This protocol satisfies the conditions for anonymous, authenticated and accountable transactions between users and service providers.

## **Trade-off Analysis**

This is a simple give-and-take wherein one quality, quantity, or property is lost or diminished to increase these in another aspect. Trade-offs are usually obtained through discussions and sharing of insights (Pasquale *et al.*, 2016; Regnell, Berntsson Svensson & Olsson, 2008).

## **Dummy Traffic**

A dummy message is a fake message introduced in a mix network in order to make it more difficult for an attacker to deploy passive and active attacks (Diaz & Preneel, 2004). Dummy messages are normally generated by the mixes (although users may also generate dummies, which increases the anonymity level of the mix network and prevents end-to-end intersection attacks (Berthold & Langos, 2002); they have as their destination another mix, instead of a real recipient. Dai proposed the Pipenet system (Diaz, 2004) a system in which the traffic is constant: the links between mixes are padded with dummy messages whenever the real traffic is not enough to fill them. This system provides not only anonymity, but also unobservability, since an observer of the network cannot tell whether or not the messages traveling in the network are real. Unfortunately, the system is not practical due to the enormous number of



resources it needs. The generation and transmission of dummy traffic has a cost, and it is therefore very important to find the right balance of the number of dummies that should be created in a mix network. The rest of this section studies the possible choices we can make when designing a dummy policy.

### **Trusted Third Party**

Several protocols for certified email have been developed recently and several systems for certified email are being deployed commercially (Abadi & Glew, 2002). Generally speaking, their main goal is to guarantee that the receipt of an email message produces a receipt certificate whether or not the receiver is honest and diligent. They sometimes have secondary goals, such as authenticity of sender and receiver, and message confidentiality. In order to achieve their goals, the protocols and systems often require some new assumptions and new software for email senders and receivers. Most also rely on some new infrastructure, in particular on a trusted third party of some sort that serves as a mediator. There are further protocols which do not use a trusted third party and operate by having the parties undertake a bit-by-bit exchange of each message against the corresponding receipt.

### **ADOPT**

According to Baldoni *et al.* (2018), accountability-driven organisation programming technique (ADOPT) is a protocol for creating and manipulating accountability relationships. Technically, the core of the proposal builds upon the notion of role and in the action of role adoption (or enactment), on one side, and on the concept of social commitment on the other. ADOPT allows the realisation of accountable MAS organisations. Agents and organisations will share relevant information by exchanging messages, whose structure follows the Foundation for Intelligent Physical Agents (FIPA), Agent Communication Language (ACL) specification.

### **4.3 Mitigating Conflicts**

To mitigate conflicts between requirements, we found in the literature some techniques to support security or privacy requirements. These are categorised based on:

- 1- Techniques being suitable for both security and privacy requirements, which means that one technique could support either security or privacy requirements.
- 2- Some techniques supporting privacy requirements only or security requirements only. This decision based on a previous step (requirement prioritisation) is necessary to ascertain which requirements the analyst will choose to support.

In Tables 4.4 to 4.7, we find approximately 14 cases of conflict with mitigating techniques for each. For instance, in Table 4.4a there is a common tool working with both privacy and security requirements, while in part 4.4b, there is no common tool, so the need for a trade-off is necessary. In addition, some cases have more than one mitigating technique. This revelation therefore assists the software engineering analyst to select the best or most appropriate techniques for the optimum solution. To mitigate a conflict via supported tools, we try to find a relevant tool that could satisfy both types of requirements (see Table 4.5) or suitable for privacy (see Table 4.6) and security requirements (see Table 4.7) alone. This well allocates the technical tool for each case of conflicts, supporting the requirements.

**Table 4.4 Conflict Cases and Likelihood of Tools**

<i>Security Requirement</i>	<i>Privacy Requirements</i>
<b>a. There is a common tool working with both requirements:</b>	
<i>Confidentiality</i>	Anonymity, Unlinkability, Undetectability, Pseudonymise
<i>Integrity</i>	Anonymity, Unlinkability, Unobservability
<i>Availability</i>	Anonymity, Unlinkability, Undetectability, Unobservability
<i>BOD</i>	Unlinkability
<i>Accountability</i>	Anonymity, Undetectability, Unobservability
<i>Non-repudiation</i>	Anonymity, Unobservability
<b>b. There is no common tool, so need to do trade off:</b>	
<i>Authentication</i>	Anonymity, Pseudonymise, Unobservability
<i>Authorisation</i>	Unobservability
<i>Audibility</i>	Anonymity, Undetectability, Unobservability
<i>BOD</i>	Pseudonymise
<i>Accountability</i>	Undetectability, Unobservability
<i>Non-repudiation</i>	Unobservability

**Table 4.5 Techniques suitable for both Security and Privacy Requirements**

<i>Security and Privacy Requirements</i>	<i>Tool to Support Requirement</i>
<i>Anonymity vs Confidentiality</i>	Cryptographic, Steganographic technologies, Onion Routing
<i>Unlinkability vs Confidentiality</i>	Cryptographic, Steganographic technologies, Homomorphic encryption, Onion Routing
<i>Unlinkability vs Integrity</i>	Cryptographic
<i>Pseudonymity vs Confidentiality</i>	Searchable encryption
<i>Undetectability vs Confidentiality</i>	Steganographic technologies

**Table 4.6 Techniques suitable for Privacy Requirements**

<i>Privacy Requirements</i>	<i>Tool to Support Requirement</i>
<i>Anonymity</i>	Cryptographic, Steganographic technologies, Onion Routing, trusted third parties, Dummy traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), K-anonymity
<i>Unlinkability</i>	Cryptographic, Steganographic technologies, Homomorphic encryption, Onion Routing, K-anonymity, data hiding, trusted third parties, dummy traffic
<i>Pseudonymity</i>	Searchable encryption, Public key
<i>Unobservability</i>	Dummy traffic
<i>Undetectability</i>	Dummy traffic, Steganographic technologies

**Table 4.7 Techniques suitable for Security Requirements**

<i>Security requirements</i>	<i>Tool to Support Requirement</i>
<i>Confidentiality</i>	Cryptographic, accesses control enforcement, Symmetric key and public key encryption, Steganographic technologies, Homomorphic encryption, Onion Routing, Searchable encryption
<i>Integrity</i>	Cryptographic, accesses control enforcement, message authentication codes (MAC), redundancy and comparison
<i>Availability</i>	Redundancy to the system
<i>Accountability</i>	ADOPT

### **4.3 Chapter Summary**

This chapter describes the framework process, as illustrated in Figure 4.1. We present a theoretical framework, explaining how it works with regards to the three phases. Firstly, we identify requirements as related to security and privacy in Section 4.2.1-1. Although the list is not exhaustive, these are mostly applied in organizations. Next, within the context of this list, we recognize that some of these requirements lie in conflict with each other. Using a matrix, this helps us to detect and visualise the requirements with the most conflicts, which aids us in identifying those most deserving focus. In identifying those conflicts in Section 4.2.1-2, we move towards mitigating these conflicts between requirements based on supporting techniques, or suitable for either requirement in Section 4.2.1-3. Whether we have decided to use requirement prioritization or requirement negotiation, we must illustrate the criteria to apply those methods based on the user's point of view. Furthermore, this section highlights each supporting tool, and outlines each with its workings.

In the following chapter, we will describe modelling conflicts in greater depth, including both identification of conflicts and their solutions. We will fully describe the problem and provide examples from the literature. Here, modelling the solution using a more graphic demonstration will be conducted.

# CHAPTER 5

## IDENTIFYING AND ANALYZING CONFLICTS

### 5.1 Introduction

In this chapter, we model conflicts between requirements in order to identify conflicts and determine a technical solution at the analysis stage. Next, we describe the approach to reduce conflicts. A key principle of this phase is to identify and resolve conflicts at the early requirements stage, so that it becomes more flexible in finding adequate ways to handle conflicts (Paja, Dalpiaz & Giorgini, 2013). This includes alternative supporting tools that fulfil certain types of conflicts between requirements. To be precise, the following phases will be followed in the development of the framework phases: firstly, the problem is presented per conflict, moving on to examples from the literature, next discussing as to how such conflict can be tackled, and lastly the suggestion of resolving tools is identified.

### 5.2 Method supported by the model

In order to develop an efficient and effective framework for managing conflict between security and privacy requirements, and to reduce risk impact in software systems when an organization's system can potentially be shut down/hacked with a huge impact on the business, the goals set by the stakeholders need to be ascertained. Thereafter, the modeling language, tools, implementation and validation procedures need to be determined accordingly. According to Jannat (2019), the various reasons for conflict between requirements include:

- a massive number of requirements to fulfil.
- changes in prerequisites during framework improvement stages, after the expansion of new prerequisites or the update of old ones.

- complex framework space can prompt misconception of necessities, and consequently, conflicts between them.

The social troubles which can lead to necessities clashing are as follows:

- the system has various stakeholders with assorted interests which associate with one another and cause conflicts.
- changes in the framework's stakeholders by including new stakeholders with various needs or by changing stakeholder's solicitations.

A model for contextual requirements has to represent conflicts between requirements as unsolved conflicts could lead to a malfunctioning final system. Each model identified below represents the interaction between requirements and identifies their conflict accordingly. A generic list of security and privacy requirements is identified in Table 5.1, expounded upon in Chapter 4.

Furthermore, Table 4.2 presents a matrix, mapping the privacy and security requirements. This matrix table reflects the findings of various types of conflicts between security and privacy requirements all supported by literature reviews and academic research. For the purposes of this research, the focus will primarily develop conflict between privacy and security requirements (CPS).

Additionally, this chapter will have two parts:

- A- Identification of the conflicts (based on previous studies)
- B- Resolving conflicts, by describing our model with both types of requirements and linking it with the supporting tool in order to resolve this conflict.

We will identify conflicts between security and privacy requirements in a matrix presented below, which will present each type of conflict between security and privacy requirements (see

Table 4.1). In Table 5.2 we list each conflict we found based on the specific papers in which they were identified, as explained in the following section.

**Table 5.1 Security Requirements Conflicts with some Privacy Requirements**

<b>Security requirement</b>	<b>Privacy requirements</b>
<b>Confidentiality</b>	Anonymity, Unlinkability, Undetectability, Pseudonymity
<b>Integrity</b>	Anonymity, Unlinkability, Unobservability
<b>Availability</b>	Anonymity, Unlinkability, Undetectability, Unobservability
<b>BOD</b>	Unlinkability
<b>Accountability</b>	Anonymity, Undetectability, Unobservability
<b>Non-repudiation</b>	Anonymity, Unobservability

In the following Table 5.2, we specify each conflict as identified in the literature from seven studies. For instance, Mouratidis *et al.* (2013) research conflicts surrounding confidentiality vs unlinkability; integrity vs anonymity; integrity vs unlinkability; integrity vs unobservability; availability vs unobservability. The type of notational language used to highlight these conflicts between requirements include the Goal Model and Secure Tropos. Additionally, Ramadan *et al.* (2018), identify conflicts around confidentiality vs unobservability; confidentiality vs undetectability; accountability vs anonymity; non-repudiation vs anonymity; non-repudiation vs unobservability; and BOD vs Unlinkability. The BPMN represents types of languages in which conflict is found for these requirements (see Table 5.2). On the other hand, Diamantopoulou, Vasiliki *et al.* (2017) pinpoint conflicts around confidentiality vs pseudonymity; authentication vs pseudonymity; authentication vs unobservability; and availability vs undetectability. No modelling language was identified (see Table 5.2). These are the initial instances in which we begin to see the Secure Tropos type models from the literature, which will be previewed below.



**Table 5.2 Literature Review – Conflict Requirements**

<b>Paper</b>	<b>Conflicts</b>	<b>Language</b>
<b>Mouratidis <i>et al.</i>, 2013</b>	<ul style="list-style-type: none"> <li>- Confidentiality vs Unlinkability</li> <li>- Integrity vs Anonymity</li> <li>- Integrity vs Unlinkability</li> <li>- Integrity vs Unobservability</li> <li>- Availability vs Unobservability</li> </ul>	<b>Secure Tropos</b>
<b>Kalloniatis <i>et al.</i>, 2013</b>	<ul style="list-style-type: none"> <li>- Integrity vs Unlinkability</li> <li>- Integrity and Anonymity</li> <li>- Availability vs Unlinkability</li> </ul>	<b>Secure Tropos</b>
<b>Mellado <i>et al.</i>, 2014</b>	<ul style="list-style-type: none"> <li>- Availability vs Anonymity</li> </ul>	<b>Secure Tropos</b>
<b>Shei <i>et al.</i>, 2015</b>	<ul style="list-style-type: none"> <li>- Confidentiality vs Anonymity</li> </ul>	<b>Secure Tropos</b>
<b>Ramadan <i>et al.</i>, 2018</b>	<ul style="list-style-type: none"> <li>- Confidentiality vs Unobservability</li> <li>- Confidentiality vs Undetectability</li> <li>- Accountability vs Anonymity</li> <li>- Non-repudiation vs Anonymity</li> <li>- Non-repudiation vs Unobservability</li> <li>- BOD vs Unlinkability</li> </ul>	<b>BPMN</b>
<b>Diamantopoulou, Vasiliki <i>et al.</i>, (2017)</b>	<ul style="list-style-type: none"> <li>- Confidentiality vs Pseudonymity</li> <li>- Authentication vs Pseudonymity</li> <li>- Authentication vs Unobservability</li> <li>- Availability vs Undetectability</li> </ul>	<b>No model</b>
<b>Matyás &amp; Kur, 2013</b>	<ul style="list-style-type: none"> <li>- Authentication vs Anonymity</li> </ul>	<b>No model</b>

Elaborating on Mouratidis *et al.* (2013), the actions taken to identify conflicts include defining the organisational context and in particular organisational goals, relevant actors, their plans, resources and security and privacy goals (see Table 5.3; Figure 5.1). Furthermore, five actors are identified, which are relevant to the case study as shown in Figure 5.1 – EPOS Ltd, Night Club Ltd, EPOS Software, Cashier and Card Payment System. The main organisational goal for EPOS Ltd was to provide clients with EPOS infrastructure, as part of the Night Club Ltd depends on EPOS Ltd to *Manage Tills* and *Receive Licence*. In doing so, EPOS was required to *Provide Sales Management* and *Provide Licencing*, two goals for which EPOS Ltd depends on for the EPOS Software. To achieve these dependency goals, EPOS Software has three main goals, *Manage Sales Transactions*, *Manage Licence* and *Manage Inventory*.

Next, in order to satisfy the achievement of the *Manage Sales Transactions* goal, a number of sub-goals need to be satisfied such as *Log Sale*, *Record Sale Item*, *Manage Payment Type*, *Record Sale Quantity*, *Generate Receipt* and *Calculate Sale Price*. Some of these goals can be further refined to include relevant plans. For example, the *Manage Payment Type* goal is decomposed to three plans: *Cash Payment*, *Voucher Payment* and *Card Payment*. For the first two plans, the EPOS Software actor depends on the *Cashier* to record the cash and voucher transactions, while for the last plan the EPOS Software actor depends on *the Card Payment System* to authorise the card payment. Once all the relevant actors, plans, resources and dependencies have been identified, the relevant security and privacy goals are identified and modelled (see Table 5.3; Figure 5.1).

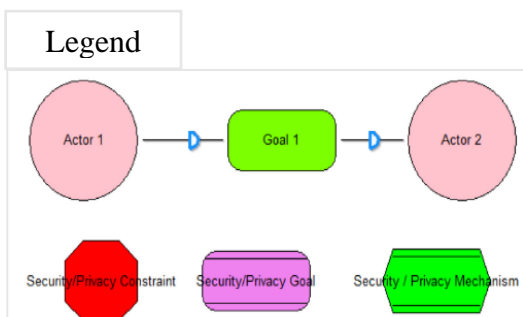
Returning to the EPOS Ltd case study, a number of security goals (i.e. confidentiality, integrity and availability) and privacy goals (i.e. identification, unobservability and unlinkability) have been identified, as shown in Figure 5.1.

Also, the three actors – EPOS Ltd, EPOS Software and the Card Payment System, have their own relevant goals which are required to be fulfilled. Firstly, in order to have a secure card payment, managing sales transaction between EPOS Ltd and the Card Payment System via the EPOS Software, is needed. Additionally, to have a secure verified transaction, integrity is necessary, to guarantee secure transaction as a security goal, and some privacy goals such as unobservability to ensure that the card information of users is not revealed in the EPOS Software. Moreover, unlinkability is likely to arise to ensure no linking of the user’s card information to any another transaction. Furthermore, confidentiality as a security goal and unlinkability as a privacy goal are also necessary, to ensure no linking of the user’s card information to any another transaction (see Table 5.3; Figure 5.1).

Additionally, EPOS Ltd is necessary in order to provide licensing from the EPOS Software. To provide licensing, we need to have this information available (security goal) and in the same manner ensure unobservability as a privacy goal of revealing this information. This can, however, result in conflict between those requirements (see Table 5.3; Figure 5.1).

**Table 5.3 Security and Privacy requirements for EPOS Night Club**

<b>Mouratidis <i>et al.</i>, 2013</b>	<ul style="list-style-type: none"> <li>- <b>Confidentiality vs Unlinkability</b></li> <li>- <b>Integrity vs Unlinkability</b></li> <li>- <b>Integrity vs Unobservability</b></li> <li>- <b>Availability vs Unobservability</b></li> </ul>	<b>Secure Tropos</b>
---------------------------------------	---	----------------------



Mouratidis *et al.*, 2013

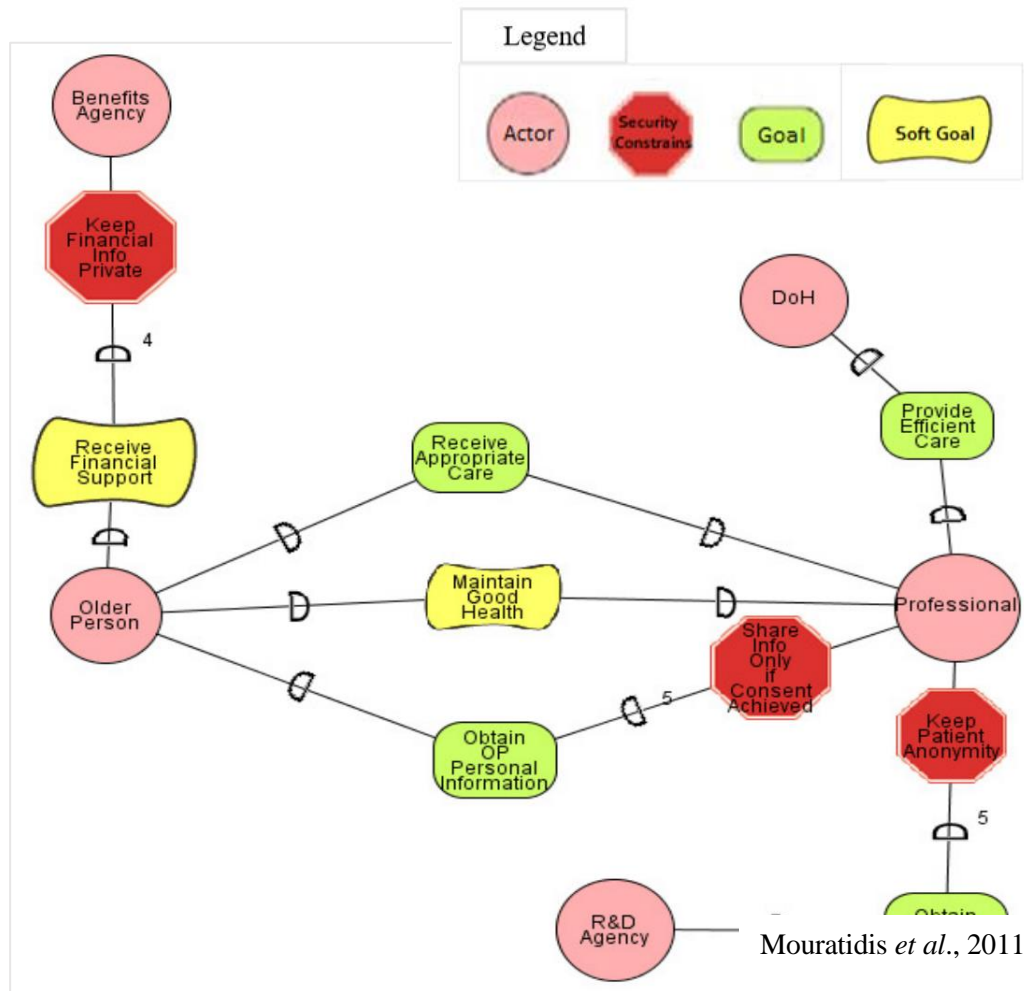
**Figure 5.1 EPOS Night Club**

Furthermore, in the case study presented in Mouratidis *et al.* (2013), we have a health insurance scenario, the actors of the system and dependency relationships between them (see Table 5.4; Figure 5.2). The diagram enables software system developers to understand the security concerns of each actor and model these concerns with appropriate security constraints.

To support older persons (actors) to have appropriate care while obtaining their personal information, requires security constraints in the form of integrity. As we have seen previously, the professional actor is required to fulfil certain constraints – integrity as a security requirement and anonymity as a privacy requirement. This, however, is likely to result in conflict, as to satisfy both requirements, it is necessary to maintain integrity in order to share sensitive information, while anonymity is necessary for patient information. The possibility therefore arises of conflicts between the two requirements, integrity and anonymity (see Table 5.4; Figure 5.2).

**Table 5.4 Security and Privacy requirements for Health Insurance**

Mouratidis et al., 2011	- Integrity vs Anonymity	Secure Tropos
-------------------------	--------------------------	---------------



**Figure 5.2 Health Insurance**

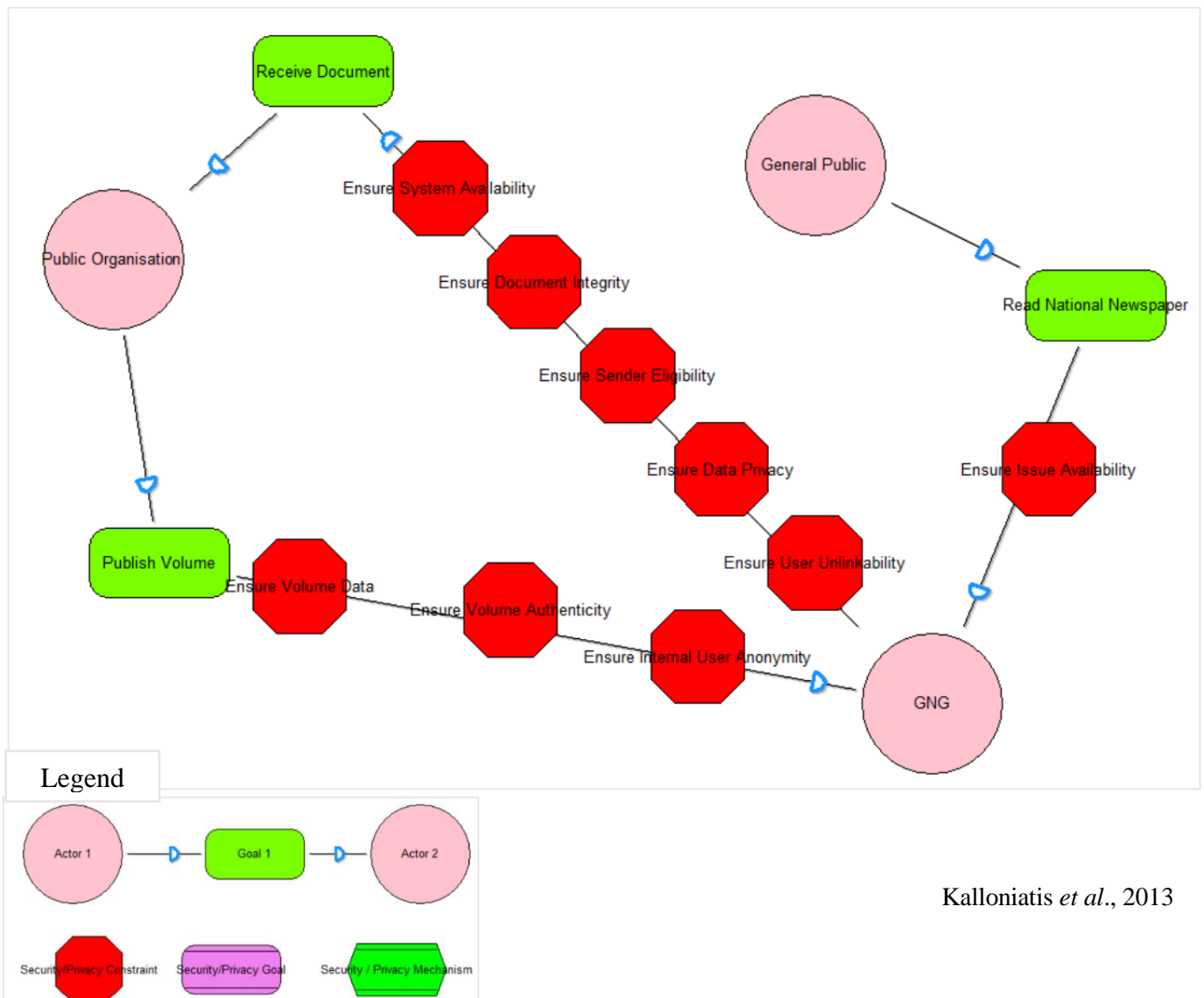
In the scenario presented by Kalloniatis *et al.* (2013), the Greek National Gazette (GNG) decided to provide a service for electronic submission of manuscripts submitted for publication. The process starts when a document is sent by a public/private sector organisation/company to the GNG (see Figure 5.3). The first step of this activity aims to detect the relevant security and privacy goals. Confidentiality, integrity and availability (security goals) and anonymity and unlinkability (privacy goals) were identified. The confidentiality goal is mandatory in order to ensure external users' eligibility. Integrity is of vital importance as well, since it must be ensured that non-authorised alterations of the documents, issues and volumes are not allowed.

Furthermore, availability will ensure that the system provides the proper mechanisms in order to be able to accept documents for publication, as well as provide the published volumes to the Greek citizens. Ensuring anonymity for the GNG’s internal users is also important since the published volumes should not include any identifiable information of the users who worked in the publication process. Furthermore, the volumes are required to only be signed by the General Secretary and the respective politicians regarding the published documents in each volume. Finally, unlinkability between the GNG and external users should be realised when GNG’s authorisation system sends the authentication means to external users in order to gain access to the submission system.

As indicated above, the GNG depends on the Public Organisation Actor to receive the document to be published. On the other hand, the Public Organisation Actor depends on the GNG actor to publish the document. Both these dependencies introduce a number of security and privacy constraints as shown in Table 5.5 and Figure 5.3. For example, the ‘Receive Document’ dependency introduces the following constraints, ‘Ensure System Availability’, ‘Document Integrity’, ‘Sender Eligibility’, ‘Data Privacy’ and ‘User Unlinkability’ when providing authentication means to eligible users (this can result in conflicts between availability and unlinkability; and integrity and unlinkability). On the other hand, the ‘Publish Volume’ dependency introduces the following constraints, ‘Ensure Volume Integrity’, ‘Volume Authenticity’ and ‘Internal User Anonymity’, likely resulting in potential conflict between integrity and anonymity (see Table 5.5).

**Table 5.5 Security and Privacy requirements for Greek National Gazette**

<b>Kalloniatis et al., 2013</b>	- <b>Integrity vs Unlinkability</b>	<b>Secure Tropos</b>
	- <b>Integrity and Anonymity</b>	
	- <b>Availability vs Unlinkability</b>	



Kalloniatis *et al.*, 2013

**Figure 5.3 Greek National Gazette**

Additionally, security and requirements engineering are two of the most important factors of success in the development of a software product line (SPL) (Mellado *et al.*, 2014). Mellado *et al.* (2014) present two applications – eCRM-I and eCRM-II (Electronic Customer Relationship Management). They specify the security requirements of a software product line of a CRM (Customer Relationship Management) system, which may have several different configurations for three different public institutions of the public social security system of Spain. Both eCRM-

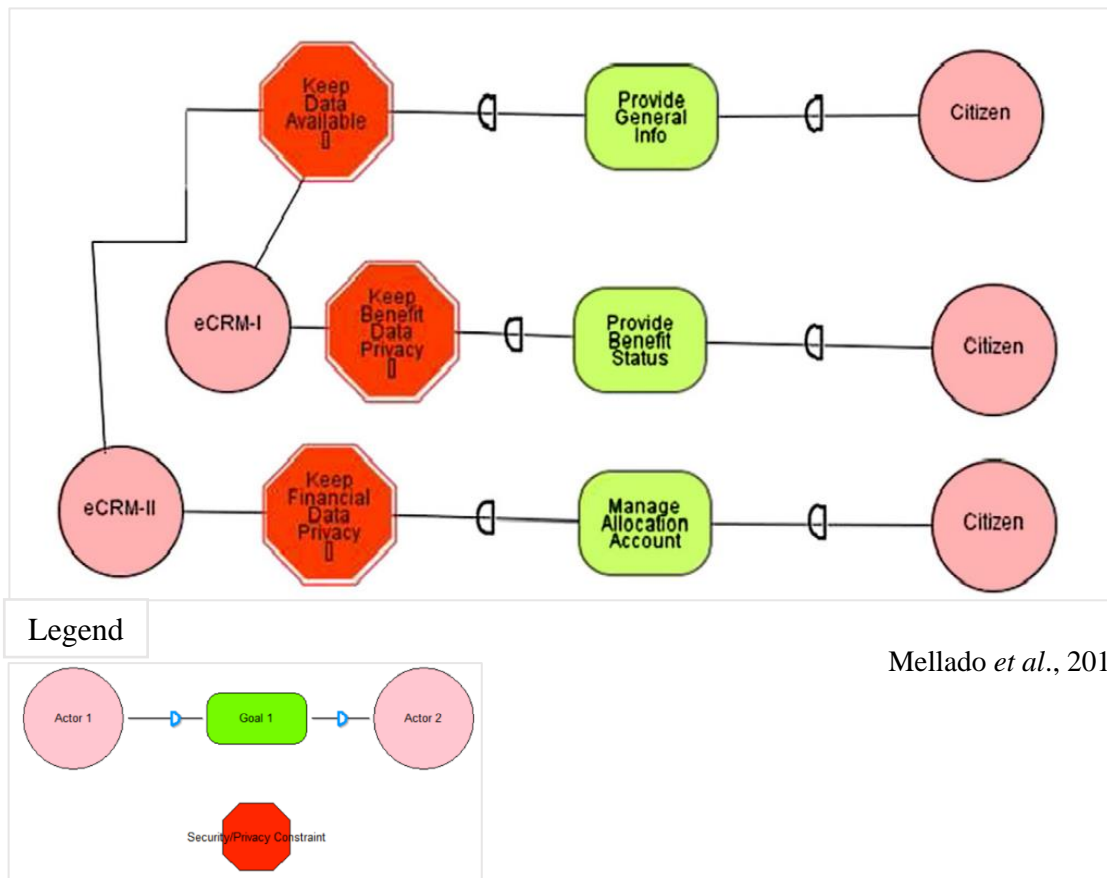


I and eCRM-II inherit common goals, constraints, plans and resources. The actor ‘eCRM (SPL)’ has strategic goals and intentions (see Table 5.6; Figure 5.4).

In this example, the ‘eCRM (SPL)’ has a common service goal to citizens, that is to: ‘Provide general information about social security issues’, and two optional service goals, to: ‘Provide the status of a citizen’s benefit’ and/or ‘Manage the allocation account contribution to the Social Security’. Security constraints, shown in the model as ‘Keep data available’, availability, and privacy constraints, ‘Keep financial data privacy’ and ‘Keep benefit data privacy’, present some issues. They need to keep citizen data available, while at the same time keeping financial data private or anonymous. Therefore, possible conflicts can arise between availability and anonymity, based on the nature of those requirements (see Table 5.6; Figure 5.4).

**Table 5.6 Security and Privacy requirements for Customer Relationship Management**

<b>Mellado <i>et al.</i>, 2014</b>	- <b>Availability vs Anonymity</b>	<b>Secure Tropos</b>
------------------------------------	------------------------------------	----------------------



Mellado *et al.*, 2014

**Figure 5.4 Customer Relationship Management**

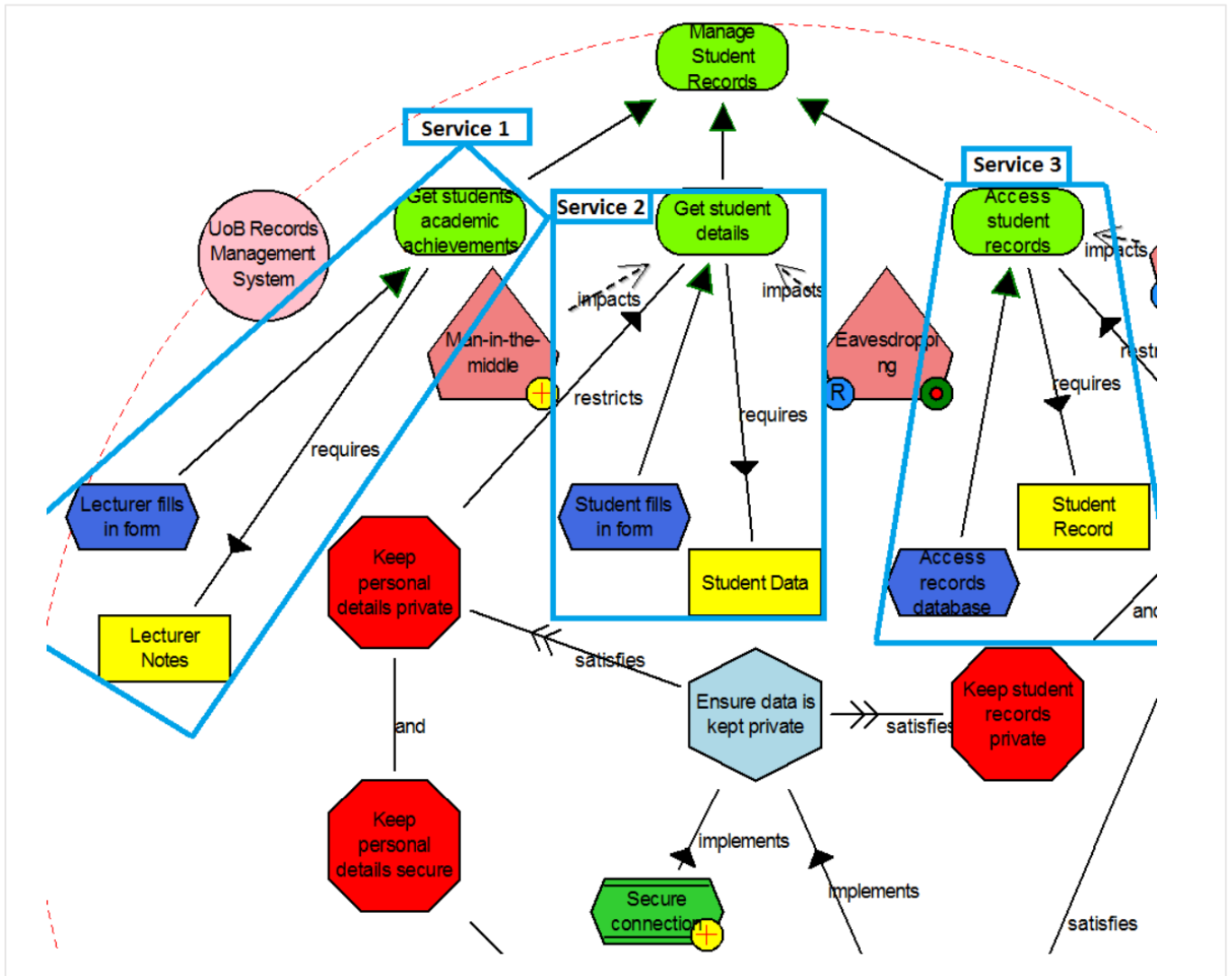
Shei *et al.* (2015) present a simple example denoting the identification of a service based on the goal ‘Get student details’, the plan ‘Student fills in form’ and the resource ‘Student Data’ contributes toward the definition of the service, as shown in Figure 5.5. The functionality of the service is to obtain details from students. The plan indicates that the service will include the capacity to obtain student data from forms which are filled in by the student, possibly through a form defined by the interface. The required input will be student data which the resource describes in full detail, including properties such as the owner of the data, how the data is stored and the specifications of the data.

The primary goal ‘Manage Student Records’ has three sub-goals, in this example the ‘Get Student Details’ sub-goal is examined in more detail. This subgoal requires the resource ‘Student Data’ and the plan ‘Student Fills in Form’ in order to satisfy its requirements. It also has the security constraints of keeping personal details private and secure and is impacted by two threats: ‘Man-in-the-Middle’ and ‘Eavesdropping’ (see Table 5.7; Figure 5.5) (Shei *et al.*, 2015).

Furthermore, the ‘Assessing Student Record’ goal has a privacy constraint, which is to keep the student record private, while the ‘Get Student Details’ goal has two constraints i.e. to keep personal details private as a privacy constraint, and to keep personal details secure, as a security constraint. We must keep student personal data private (anonymous) whilst also keeping those personal data secure (confidential). However, in order to apply the security mechanism (to maintain confidentiality), the data could possibly be revealed by breaching anonymity and confidentiality, through disclosure. This therefore produces a potential conflict between the security and privacy requirements (see Table 5.7; Figure 5.5).

**Table 5.7 Security and Privacy requirements for UoB Records Management system**

<b>Shei <i>et al.</i>, 2015</b>	<b>- Confidentiality vs Anonymity</b>	<b>Secure Tropos</b>
---------------------------------	---------------------------------------	----------------------



Shei *et al.*, 2015

**Figure 5.5 UoB Records Management system**

Additionally, a pictorial representation by Ramadan *et al.* (2018; 2020) of the E-Health’s organisation and security requirements view is shown in Figure 5.6. Here, we have three actors: patient, system portal and tele-medicine. The model explains how a patient makes use of a tele-medicine device to receive a healthcare service remotely. Moreover, a patient can

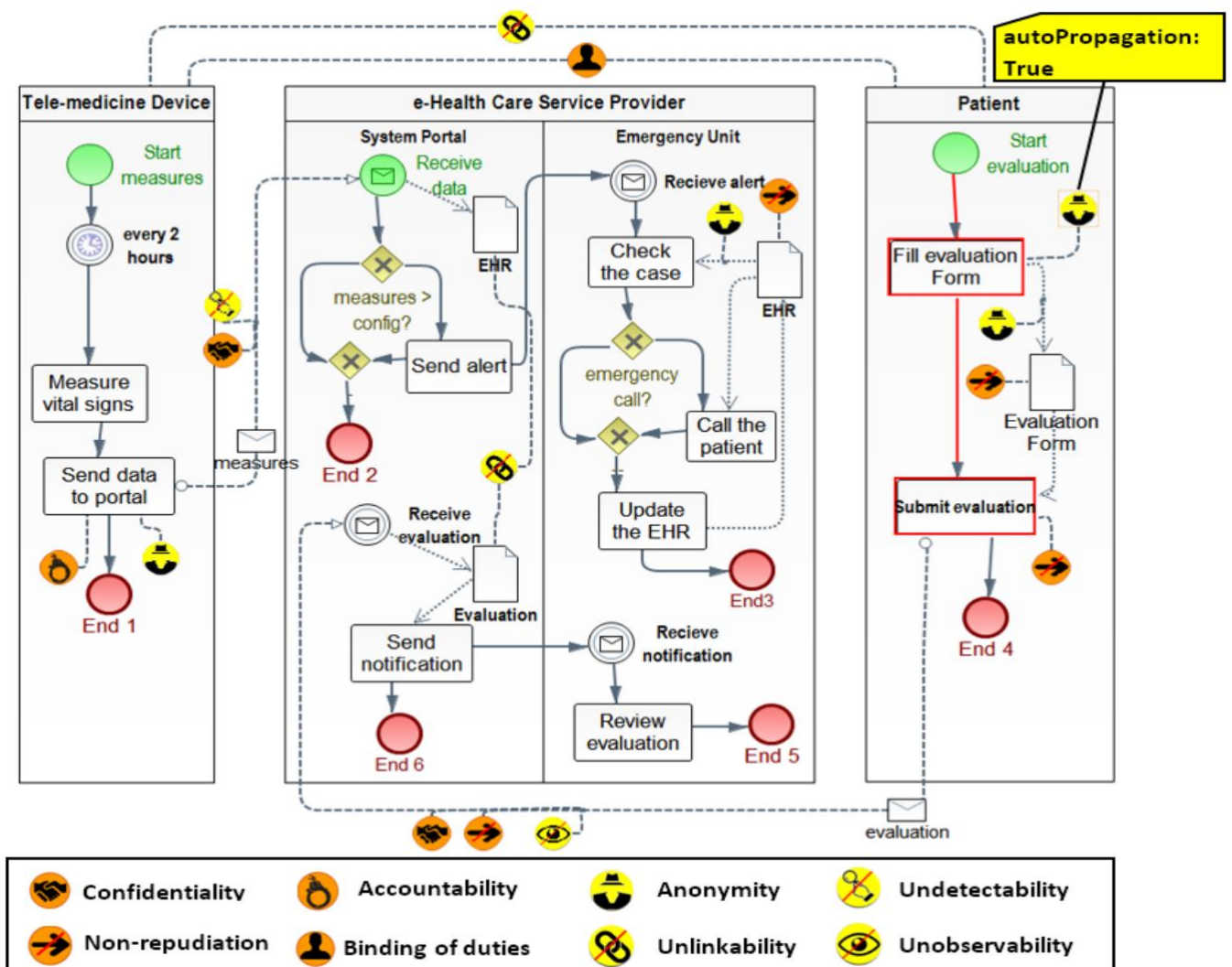
evaluate the service through an online portal. Firstly, requirements are identified, then the next step is to model this example, using SecBPMN2 (see Table 5.8).

The content of such communication is messages. For example, the tele-medicine device sends the patient's biometric information to the system portal. Key pointers of the model show that:

- Atomic activities are then represented with tasks, for example to send an alert.
- Data objects provide information about what activities are required to be performed, and/or what they produce, for example electronic healthcare record (EHR).
- A data association is a directional association used to model how data is written to or read from a data object. For instance, the 'Check the case' task needs the EHR data object to be read.
- Events are represented with circles.
  - Start events and End events mark the initial and terminal points.
  - Catch events represent points in a business process where an event needs to happen, for example at a certain time.
- Confidentiality is associated to message flows, meaning that the content of the message is to be preserved and is not to be accessed by unauthorised users.
- Accountability is associated with submitting an evaluation, meaning that the task's executor must be monitored.
- Our new data minimisation concept, discussed below, are represented with yellow icons.
- To allow users to enrich business process models with data-minimisation requirements, we extended BPMN's artefact class with four concrete data-minimisation concepts, namely: Anonymity, undetectability, unlinkability and unobservability.

Table 5.8 Security and Privacy requirements for Healthcare Management

<p>Ramadan <i>et al.</i>, 2018</p>	<ul style="list-style-type: none"> <li>- Confidentiality vs Unobservability</li> <li>- Confidentiality vs Undetectability</li> <li>- Accountability vs Anonymity</li> <li>- Non-repudiation vs Anonymity</li> <li>- Non-repudiation vs Unobservability</li> <li>- BOD vs Unlinkability</li> </ul>	<p>BPMN</p>
------------------------------------	---	-------------



Ramadan *et al.*, 2018

Figure 5.6 Healthcare Management

Additionally, Diamantopoulou *et al.* (2017) evaluates the effectiveness of a Security and Privacy Requirements Engineering methodology, namely Secure Tropos on nine principles of the Theory of Notation. They identify conflicts likely to arise.

**Table 5.9 Security and Privacy requirements for ‘Supporting the design of privacy-aware**

<b>Diamantopoulou <i>et al.</i>, 2017</b>	<ul style="list-style-type: none"> <li>- <b>Confidentiality vs Pseudonymity</b></li> <li>- <b>Authentication vs Unobservability</b></li> <li>- <b>Authentication vs Pseudonymity</b></li> <li>- <b>Availability vs Undetectability</b></li> </ul>	<b>No model</b>
---	---	-----------------

### **Confidentiality vs Pseudonymity**

While pseudonymity gives users the freedom to work under an alias or aliases, without having to provide personal information sufficient to determine their identity, there still is the possibility of the user’s identity being identified, and thus a breach of confidentiality, for example, ensuring that an entity cannot be linked with a real identity during online interactions. Therefore, it is difficult to satisfy confidentiality as a security requirement with pseudonymity as a privacy requirement. Conflict is likely to arise.

### **Authentication vs Unobservability**

Unobservability ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used. The strength of unobservability depends on the strength of:

- i) The sender/recipient’s anonymity set.
- ii) The sender/recipient’s undetectability set.

Users’ privacy is enforced since they can use a resource or service anonymously and without being detected, in that the state of IOIs should be indistinguishable from any IOI (of the same

type), when the user wants to send messages that are not discernible e.g. random noise. This task can therefore clash with achieving authentication, as this is the process of determining whether an entity is, in fact, what or who he is declared to be.

### **Authentication vs Pseudonymity**

Pseudonymity is the utilisation of an alias instead of personally identifiable information. The issue might therefore arise, where the entity cannot be linked with a real identity during online interactions, and therefore authentication is violated. To ensure that an entity cannot be linked with a real identity during online interactions, authenticated services can be used, without disclosing the identifiable information. The Public-key Authenticated Encryption with Keyword Search (PAEKS), in which the data sender not only encrypts the keyword but also authenticates it, means that the server cannot encrypt a keyword itself. Therefore, cannot launch an attack by guessing the keyword.

### **Availability vs Undetectability**

The strength of undetectability depends on the number of nodes belonging to the undetectability set. Undetectability ensures that an entity cannot identify which user among a user pool is accessing the service. The system must enforce users' privacy by allowing them to use a service without being detected by a malicious third party. On the other hand, the system has to ensure that data is always accessible and can easily be provided to authorised entities which are likely to have been detected some time or another. A conflict is therefore likely to arise here between requirements.

Furthermore, Matyás & Kur (2013) identify conflicts between authentication and anonymity (see Table 5.10). This is likely to arise, as in achieving anonymity, authentication is likely to be



disrupted – determining whether an entity is, in fact, what/who it is declared to be, can be difficult to establish. The intrusion detection system (IDS) monitors other nodes for packet dropping by checking whether an incoming packet is re-sent in a reasonable time frame. A privacy mechanism that uses anonymity mixing will interfere with such detection because a packet might be delayed for some time.

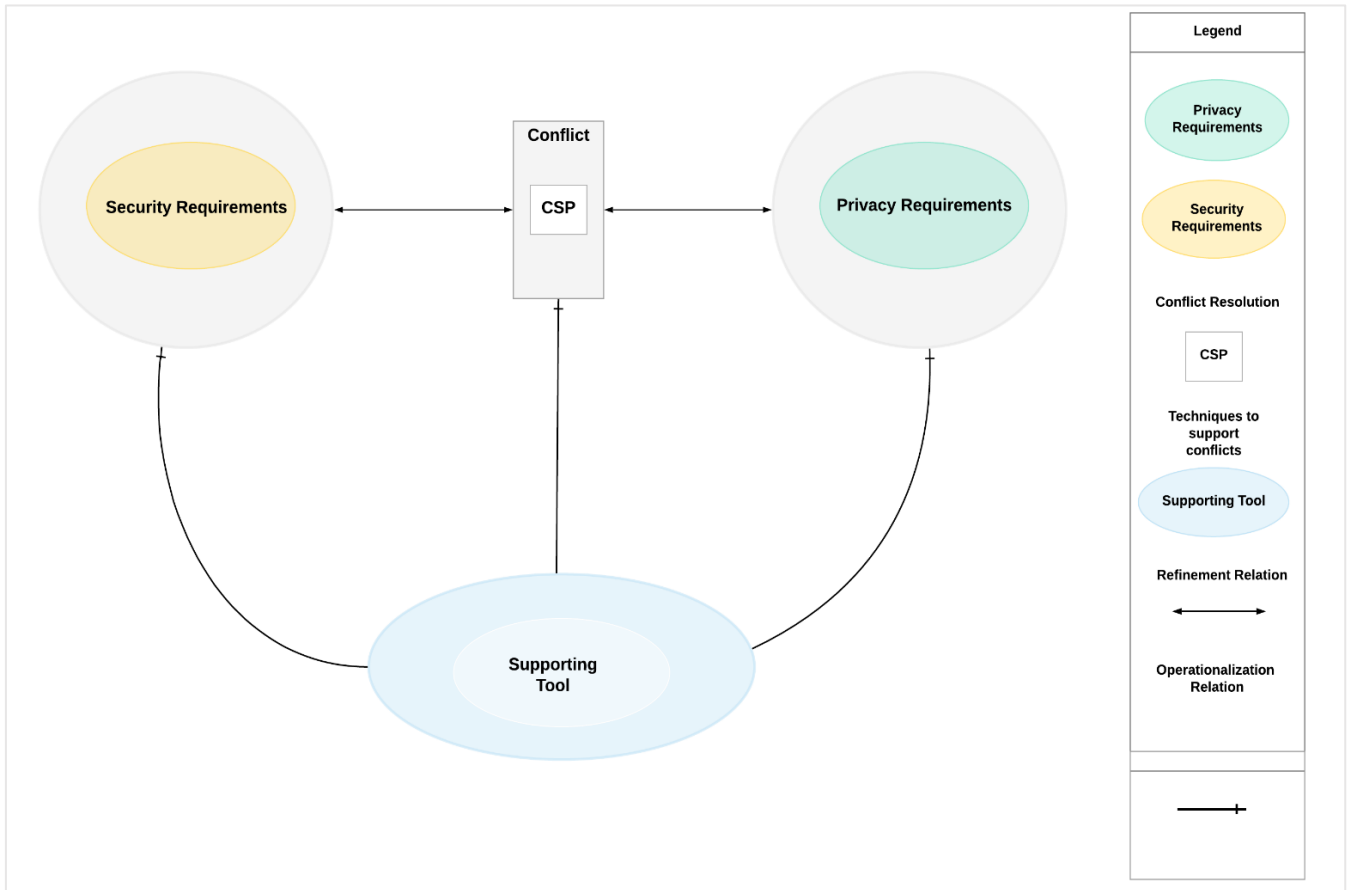
This watch is enabled by the encrypted packet format that contains per-hop changing cryptographic message authentication code verifiable by X or Z. You can design this such that X and Z do not need to share any key. However, this problem evasion will decrease the performance (decrease the security functionality or increase the protocol complexity) of the IDS, the privacy mechanism, or both.

**Table 5.10 Conflicts between intrusion detection and privacy mechanisms**

<b>Matyás &amp; Kur, 2013</b>	- <b>Authentication vs Anonymity</b>	<b>No model</b>
-------------------------------	--------------------------------------	-----------------

### **5.3 Conflict Resolution Model**

Here we identify both types of requirements and link them with the supporting tool to resolve the potential conflict through an automated model using Cyber-Physical Systems (CPS). In Figure 5.7, we demonstrate pictorially a conflict arising between privacy and security requirements, which the model assists us in visualising. After the conflict is identified, relevant supporting tools, as discussed in the previous chapters, are added in the privacy pattern library, which is used to possibly mitigate the conflict and support conflict resolution (see Figure 5.7).



**Figure 5.7 Conflict Resolution Model**

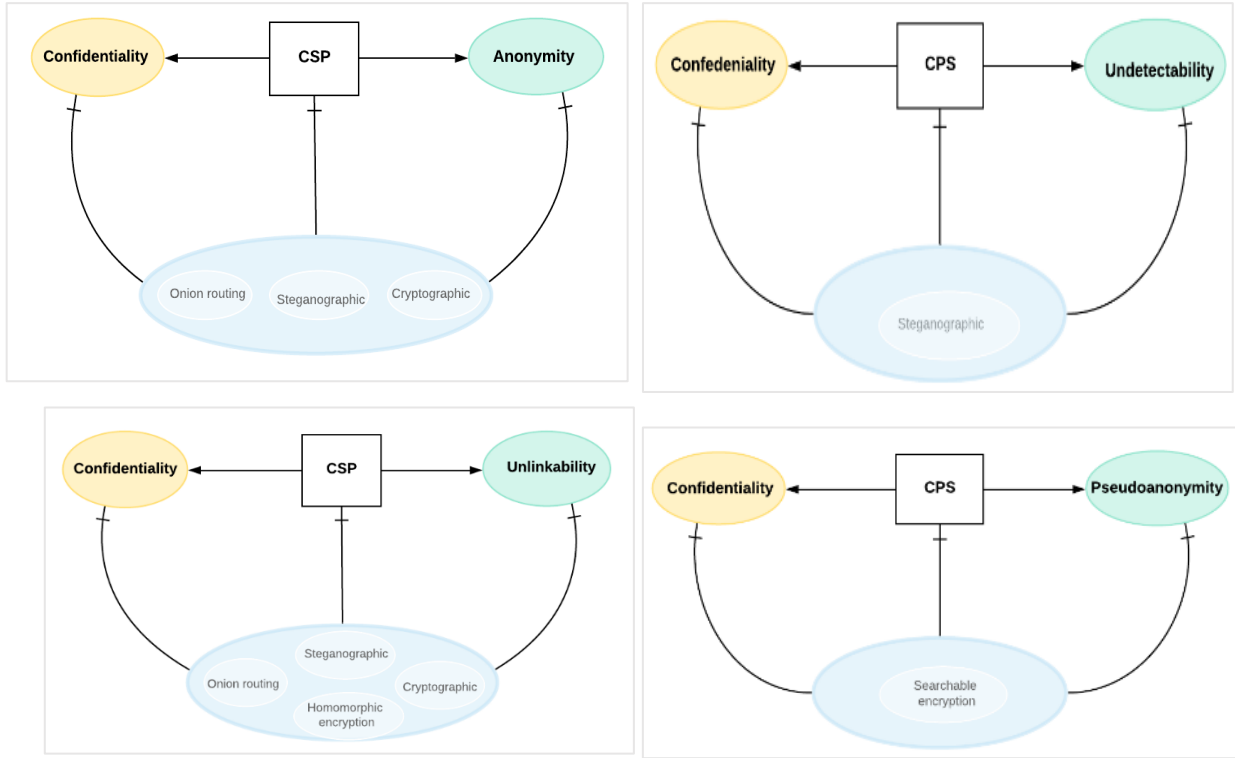
Furthermore, a combined approach depicting the suggested supporting tools for each conflict is shown in Figure 5.8. Conflict arising between confidentiality and anonymity, unlinkability, undetectability, pseudonymity, show the appropriate conflict resolution supporting tools. Among these are Onion Routing, Steganographic technologies, Searchable encryption, Homomorphic encryption, and Cryptography (see Figure 5.8).

Additionally, Figures 5.8 to 5.14 reveal the same process, for conflicts between security and privacy requirements, supporting tools are added in the privacy pattern library to mitigate the conflict and support conflict resolution. For conflicts arising between integrity and anonymity, unlinkability or unobservability, supporting tool Cryptography can be used (see Figure 5.9). On

the other hand, the binding of duties requirement is likely to conflict with unlinkability, therefore data hiding is suggested (see Figure 5.10).

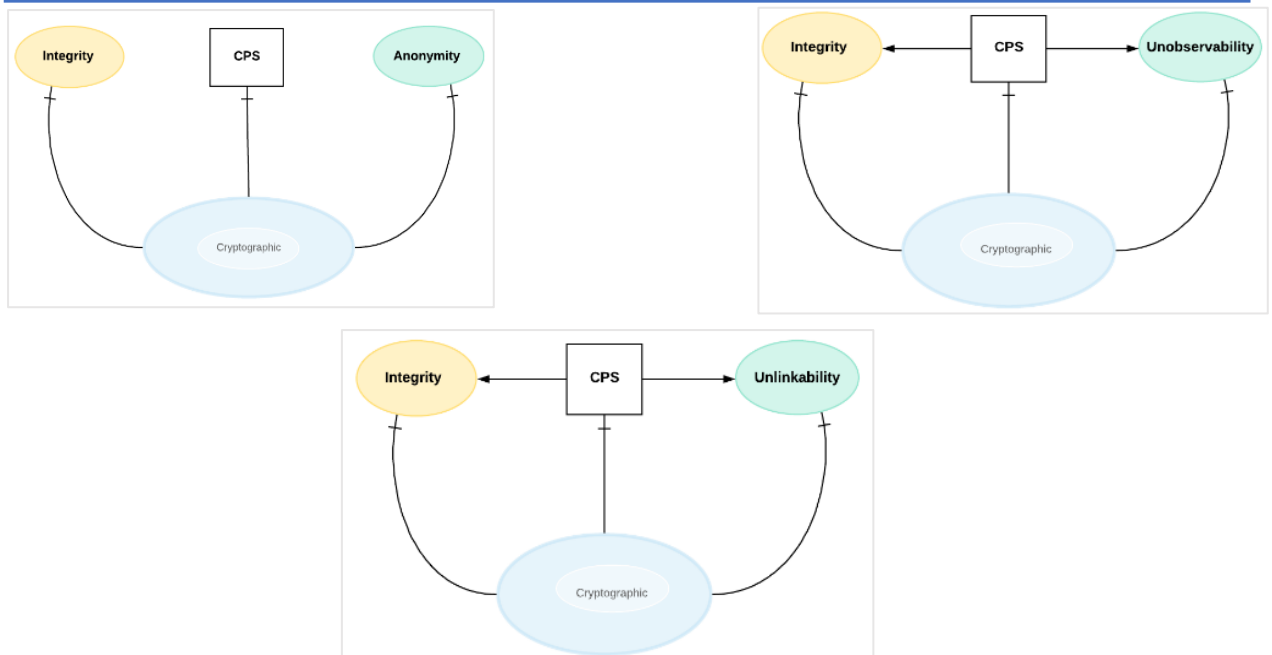
Additionally, conflicts arising between availability and anonymity, unlinkability, undetectability or unobservability will employ supporting tools such as P2P and redundancy (see Figure 5.11). Similarly, in instances of non-repudiation conflicting with anonymity or unobservability, the use of dummy traffic is implemented (see Figure 5.12). The same can be said for conflicts between accountability and anonymity, undetectability or unobservability, in which cases dummy traffic, IDEMIX and data hiding are introduced (see Figure 5.13). Lastly, requirement authentication can might conflict with anonymity, unobservability and pseudonymity, resulting in the use of cryptography, dummy traffic, public key and searchable encryption to resolve conflict (see Figure 5.14). For instance, Figure 5.14 depicts a pattern for authentication and its resolution options when in conflict with anonymity, unobservability and pseudonymity. The first figure, in the top left of Figure 5.14, shows that cryptography can help mitigate and resolve conflicts of authentication and anonymity. For example, if a system demands the user to provide evidence of who he or she is (authentication), to avoid further risk of infringing privacy requirements already agreed around anonymity, the identity/message/transaction of the user can be encrypted.

***CONFIDENTIALITY: Anonymity, Unlinkability, Undetectability, Pseudonymity***



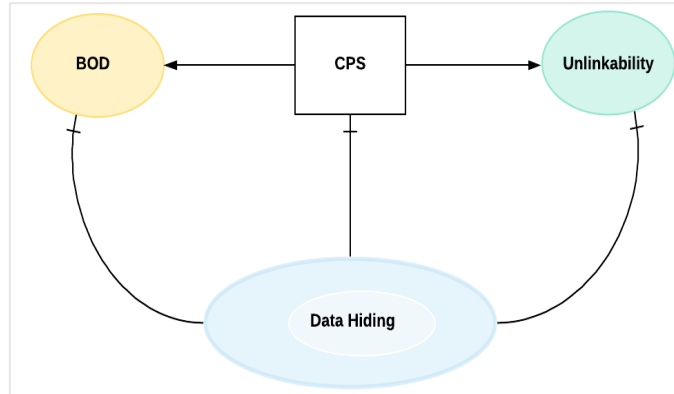
**Figure 5.8 Confidentiality conflicts with privacy requirements**

***INTEGRITY: Anonymity, Unlinkability, Unobservability***



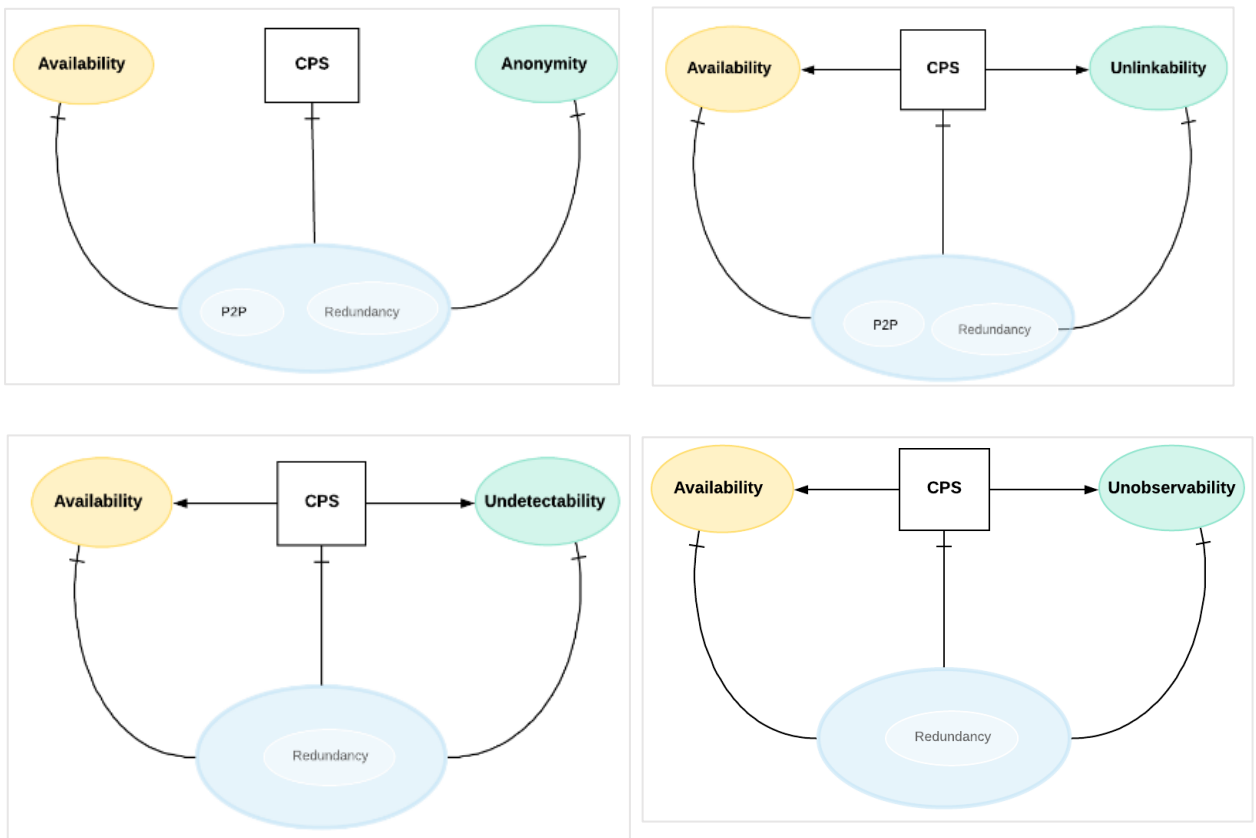
**Figure 5.9 Integrity conflicts with privacy requirements**

*BINDING of DUTIES: Unlinkability*



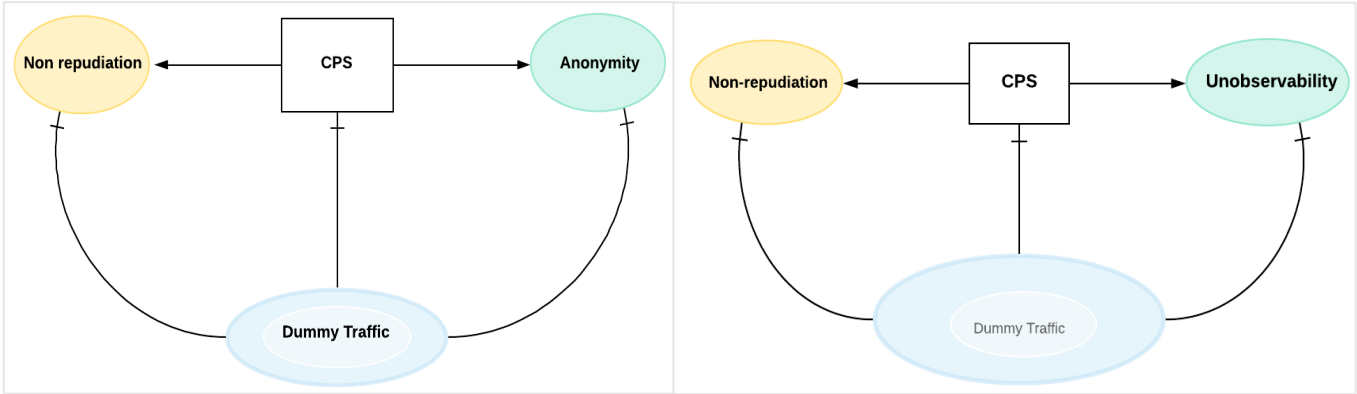
**Figure 5.10 Binding of duties conflicts with privacy requirements**

*Availability: Anonymity, Unlinkability, Undetectability, Unobservability*



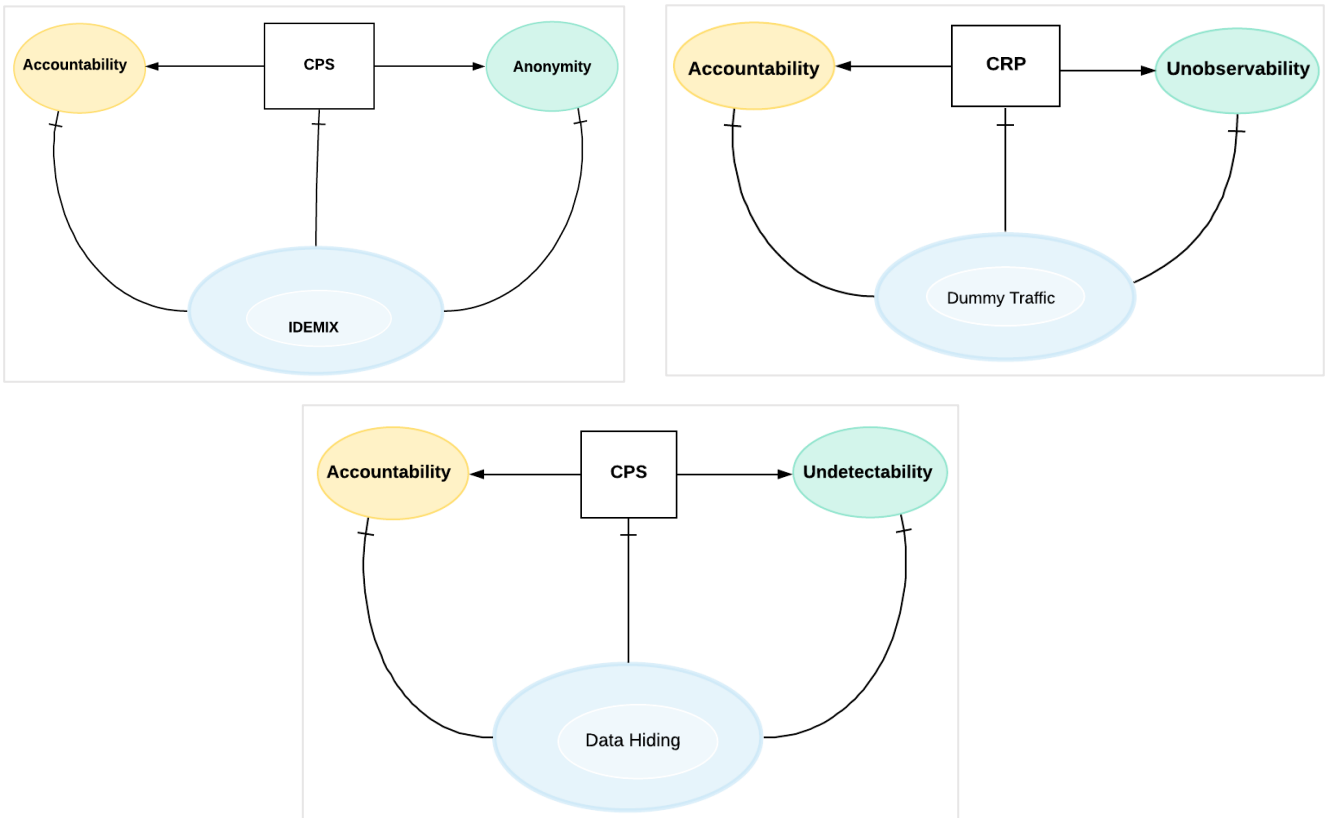
**Figure 5.11 Availability conflicts with privacy requirements**

***NON-REPUDIATION: Anonymity, Unobservability***



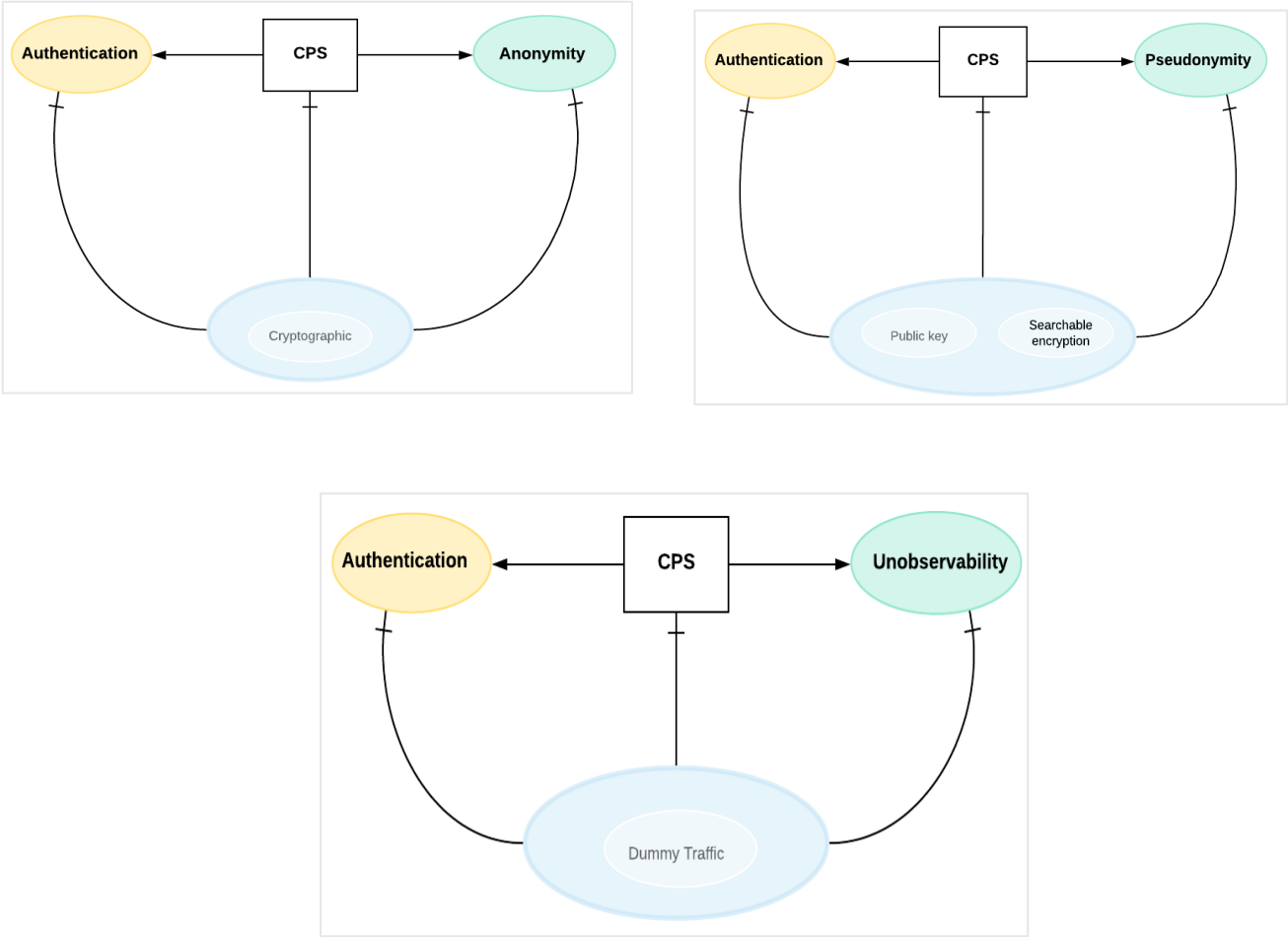
**Figure 5.12 Non-repudiation conflicts with privacy requirements**

***Accountability: Anonymity, Undetectability, Unobservability***



**Figure 5.13 Accountability conflicts with privacy requirements**

***AUTHENTICATION: Anonymity, Unobservability, Pseudonymity***



**Figure 5.14 Authentication conflicts with privacy requirements**

## 5.4 Chapter Summary

This chapter describes the models of conflicts between privacy and security requirements, supported by literature reviews. We present a model framework, explaining how it works in Section 5.2, demonstrating the reciprocity between privacy and security requirements. While this could lead to conflicts based on the nature of the requirements, the model assists in visualising the requirements that could possibly cause conflicts and suggesting supporting tools. We have a wider description about modelling conflicts, which contain both identification of conflicts and solutions. Scenarios from the literature are presented, and conflicts identified to determine their nature. at this point, we take the analysis a little further, by now suggesting supporting tools for conflict resolution in order to mitigate conflicts, as shown in Section 5.3. Modelling the solution using a graphic demonstration is conducted. In the following chapter, the model is then applied to a real live scenario using a pilot study. This will test the model's robustness and capabilities for mitigating conflicts between requirements.



# CHAPTER 6

## PILOT STUDY

### 6.1 Introduction

This pilot study is inspired by the work of Ramadan *et al.* (2018). The main difference of Ramadan's work to this research is that it is conducted using the BPMN-oriented security engineering approach, while here, it is transferred to Secure Tropos. This research uses the E-health scenario of Ramadan *et al.* (2018) and applies the ConfIS framework to identify requirements, and conflict, in order to mitigate conflicts. The ConfIS framework will be further elaborated on in this chapter. This is a framework for Privacy/Security Conflicts Identification and Solution developed within this PhD thesis research and proposed to add further knowledge to the literature in software engineering and security and privacy requirements in particular.

To illustrate conflicts between requirements, we propose Secure Tropos models for specifying privacy and security requirements (Section 6.2). The case study chosen in E-health represents a goal model in the context of healthcare management (Section 6.3), which is inspired by a study by Ramadan *et al.* (2018) who detected conflicts between data-minimisation and security requirements in business process models. Here patients have some doubts about the privacy of their information. Their disquiet about the way in which and for what purpose their health information is being retained and used, could possibly obstruct the organisation's documentation responsibilities to confirm complete accountability. While this research focuses on the early requirements stage, the scenario from Ramadan *et al.* (2018) helps us to better understand conflicts between requirements.

To apply the framework in Section 6.4, firstly we identify requirements (security and privacy) which are likely to be in conflict. We will therefore model the system based on requirements documents as an input. Secure Tropos is chosen because other approaches lack methodologies that consider security issues in a graphical way, especially at the early stage and all through the development process. Secure Tropos is also elaborated upon in Chapter 2, Section 2.4.3.1.

In the next step we will detect conflicts (Section 6.4.2) using the mapping matrix described in depth in Chapter 4 (Section 4.2-1-1). This is used to help identify the potential conflicts that can possibly arise. Moreover, this ConfIS framework resolving method is evaluated through the use of a pilot study in which, in Section 6.4.3, supporting tools are identified for mitigating conflicts.

## **6.2 Secure Tropos Framework**

The ConfIS framework can be applied to any types of modelling language, but in this research, we apply it to Secure Tropos. Very few security engineering methodologies take into account trust aspects. This modelling absence affects decision-making on the security measures imposed on the system. In particular, such measures might be excessive in some cases and inadequate in others (Massacci and Zannone, 2008). For instance, system designers may not introduce security measures since they may implicitly assume trust relationships among users, who are not in the domain. Alternatively, system designers may introduce expensive mechanisms for protecting a trusted system that has not been perceived as a trusted system by the designers themselves. To solve this problem, designers should model organisational settings in terms of social relationships among the actors involved in the system (Massacci and Zannone, 2008).

Although the application of Secure Tropos to different case studies (Massacci, Prest & Zannone, 2005) has revealed its ability to identify conflicts among functional and security requirements at an organisational level, we notice that conflicts might be concealed in requirements specified at different levels (Giorgini, Massacci & Zannone, 2005). Essentially, modelling and analysing only the structure of the organisation could be insufficient for stating that the system is secure. In fact, retrospectively untrusted agents can play trusted roles within the organisation in order to gain personal advantage from their position.

This shows that comparing the structure of the organisation with the concrete circumstances of the organisation (i.e. the agents who play roles in the organisation and relations among them) is needed to bring conflicts to light. This research intends to show that the Secure Tropos concepts and primitives are sufficient to capture high-level functional and security requirements.

Furthermore, Secure Tropos is a security-oriented extension of the requirements engineering methodology Tropos (Diamantopoulou *et al.*, 2018; Mouratidis *et al.*, 2003a Mouratidis & Giorgini, 2007; Mouratidis *et al.*, 2013; Pavlidis & Islam, 2011). Secure Tropos introduces a number of security-related concepts to the Tropos methodology, and is mainly based on four stages which covers all requirement stages:

- Early requirements analysis, aimed at defining and understanding a problem by studying its existing organisational setting;
- Late Requirements analysis, conceived to define the system-to-be in the context of its operational environment;
- Architectural design, which deals with the definition of the system's global architecture in terms of subsystems; and

- Detailed design phase, aimed at specifying each architectural component in further detail in terms of inputs, outputs, control and other relevant information.

The main unique points of this methodology as compared to other security-oriented software engineering approaches are that:

- Social issues of security are analysed during the early requirements stage;
- Security is considered simultaneously with the other requirements of the system-to-be; and
- The methodology supports not only requirements stages but also design stages.

### **6.3 E-health Scenario**

This scenario represents a business process in the context of healthcare management. A patient makes use of a tele-medicine device to receive an over-distance healthcare service. They can also evaluate the service through an online evaluation portal. During the use of the service, certain security and privacy requirements are expected to be fulfilled, keeping in line with GDPR regulations and laws.

A pictorial step-by-step representation by Ramadan *et al.* (2018) of the E-Health's organisation and security requirements view is shown in Figure 5.6 in Chapter 5. Here, we have three actors: Patient, System Portal and Tele-Medicine.

## **6.4 Applying ConfIS framework to E-health scenario**

### **6.4.1 Phase 1: Identify Requirements: (Security and Privacy Requirements)**

The model explains how a patient makes use of a tele-medicine device to receive a healthcare service remotely. The patient can evaluate the service through an online evaluation portal.

The tele-medicine device sends the data to the system portal. This step could have conflicts between requirements, such as accountability of the data and keeping those data anonymous when the data is sent to the system portal (see Figure 6.1). Thereafter, the patient can evaluate the service by accessing the evaluation form from the tele-medicine device. This step requires patient privacy in accordance with privacy laws and regulations to anonymise the data before it is shared, while the information should be non-repudiated at the tele-medicine device (see Figure 6.1 and 6.2).

After completing the form, the patient can submit it to the tele-medicine device. This process requires confidentiality and unobservability, to send the forms to the system's portal at the final step (see Figures 6.1 and 6.2). The final step is to send the evaluation form from the tele-medicine device to the system portal, while confidentiality of information is maintained, and there is undetectability (see Figure 6.1 and 6.2).

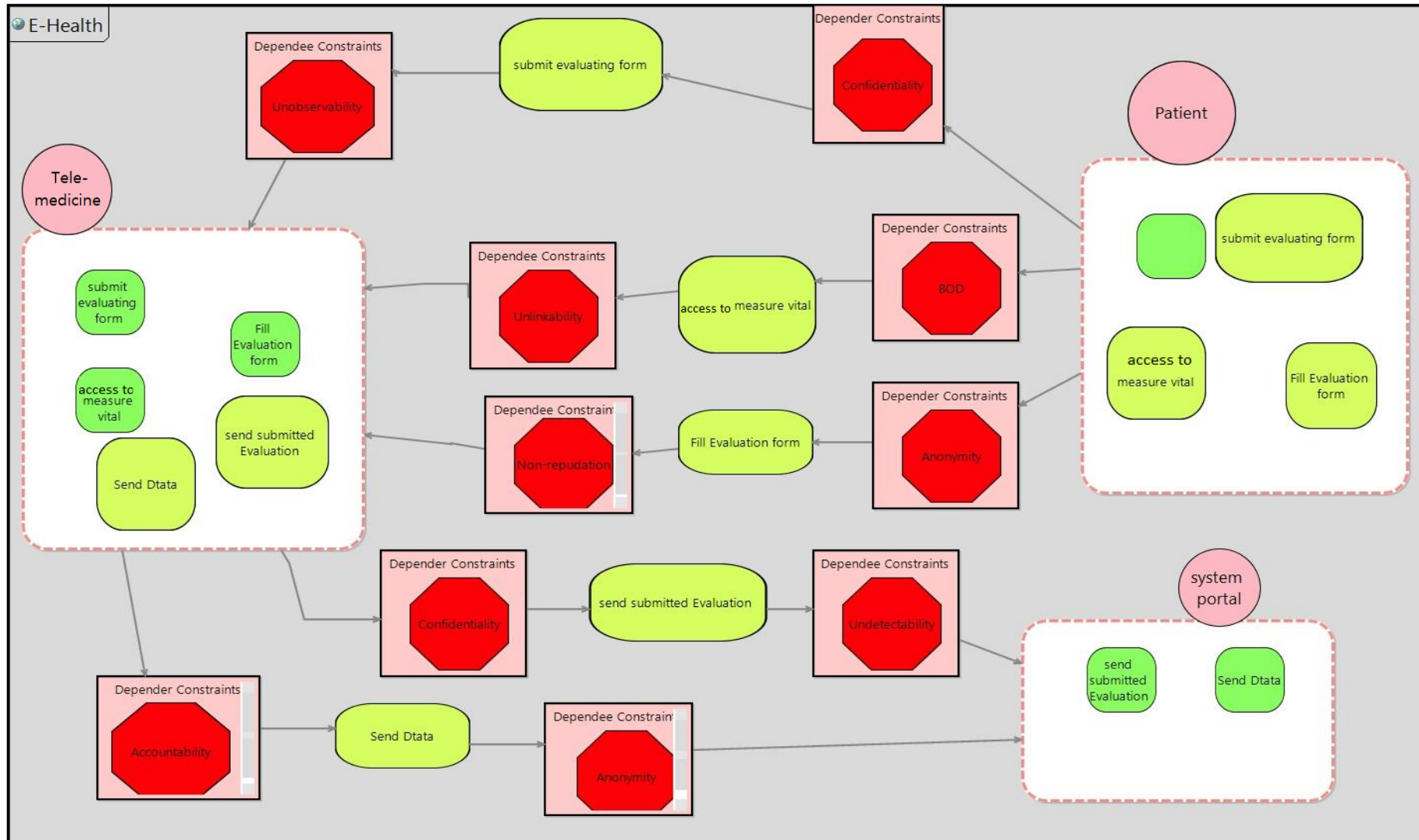
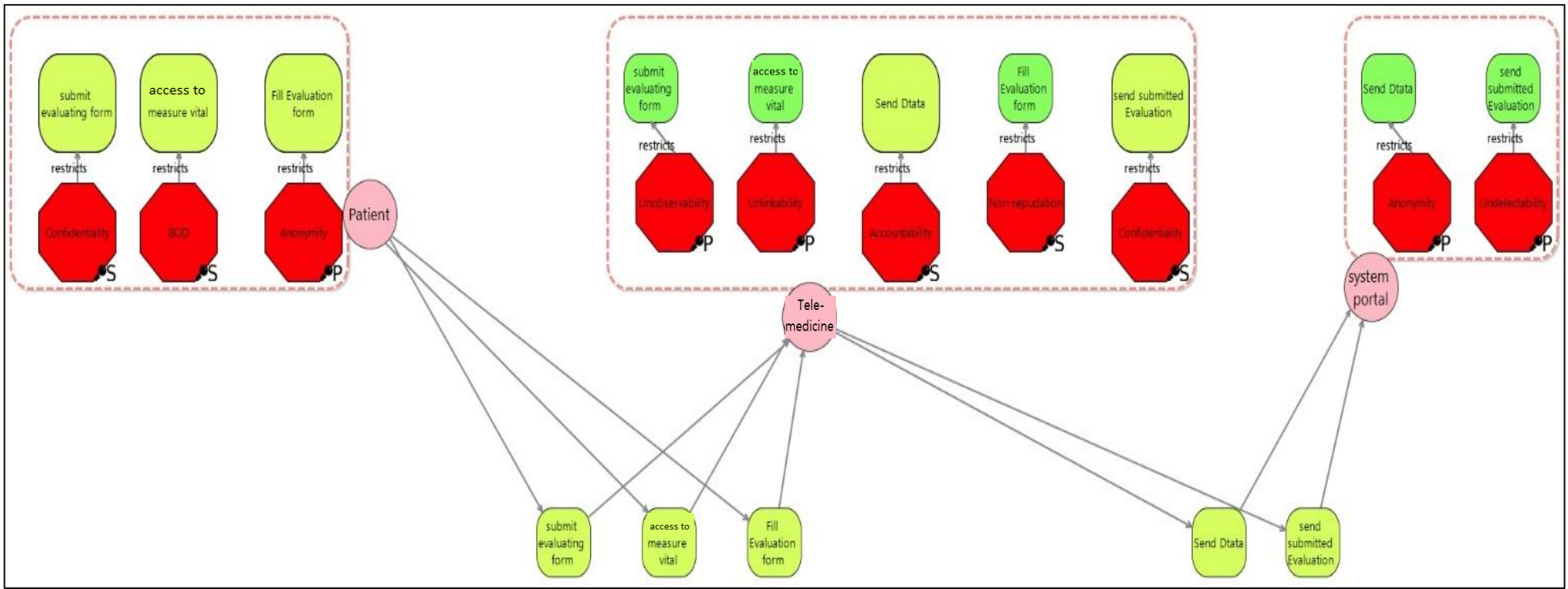


Figure 6.1 Organisational View

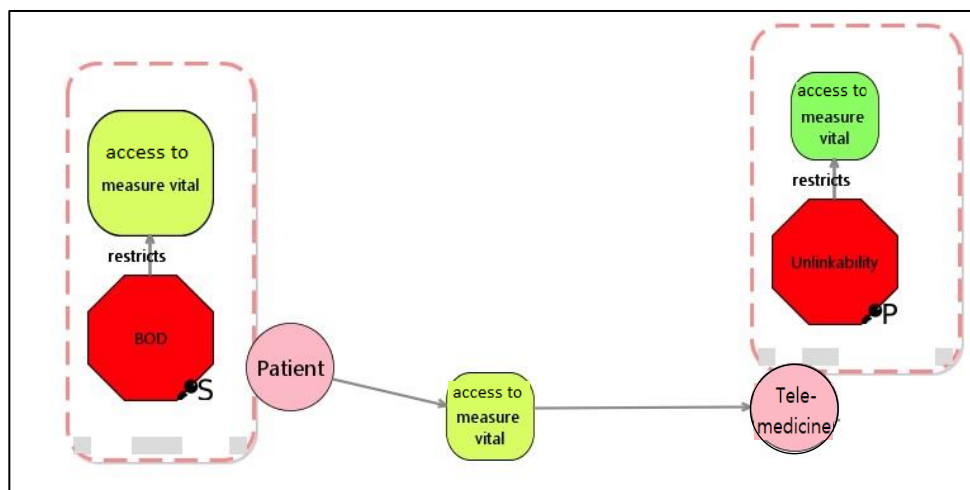


**Figure 6.2 Security Requirements View**

### 6.4.2 Phase 2: Identify Conflicts between Requirements:

Here, we identify requirements that could possibly in conflict Table 4.1 which are also based on the matrix Table 4.2 that we described in Chapter 4, Section 4.2-1-1. Furthermore, we specify requirements in order to resolve conflicts.

Binding and separation of duties can conflict with anonymity if any of the activities to which they are applied are also required to be executed anonymously. For instance, it will be hard, in case of binding of duties, to prove whether or not two fully-anonymously executed activities are executed by the same person. A potential conflict between the binding of duties and unlinkability is also possible: unlinkability is linked to two pools and indicates that the two process executions should not be linked to each other as related. Therefore, it may conflict with binding of duties.

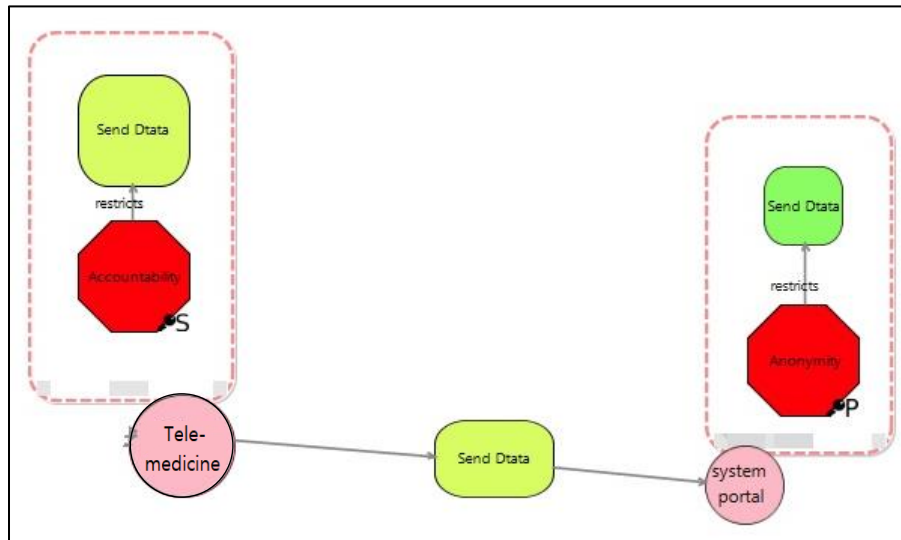


**Figure 6.3 Patient access to measure vital**

Furthermore, conflicts between security and privacy requirements occur in two ways: first, requirements related to the same asset in the system may be conflicting. For example, consider accountability and anonymity linked with the ‘Send data to portal’ task in Figure 6.4. For



accountability, the system needs to track the executor of this task's responsibility, while anonymity specifies that the executor should be fully anonymous against insider adversaries.

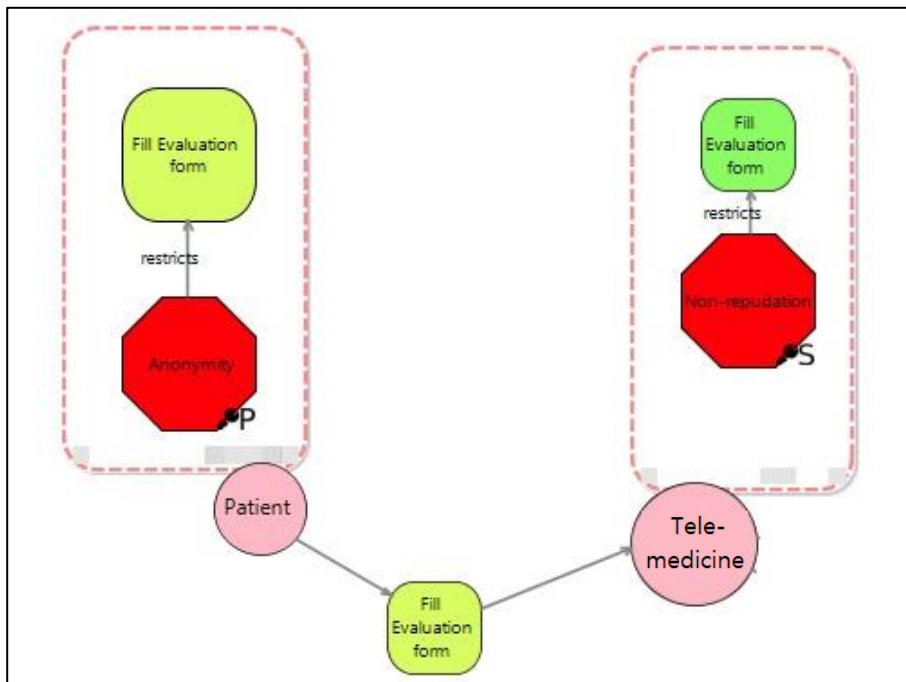


**Figure 6.4 Tele-medicine Sending data to system portal**

Second, requirements related to different, dependent assets may be conflicting. For example, in Figure 6.5 consider anonymity and non-repudiation requirements linked with the 'Fill evaluation form' task and the 'Evaluation form data' object, respectively. The former imposes that an executor to the 'Fill evaluation form' task should be fully anonymous against insider adversaries; the latter indicates that an accessor to the 'Evaluation form data' object should not be able to deny that she accessed the evaluation form. Since the 'Fill evaluation form' task writes data to the evaluation form, a conflict is reported.

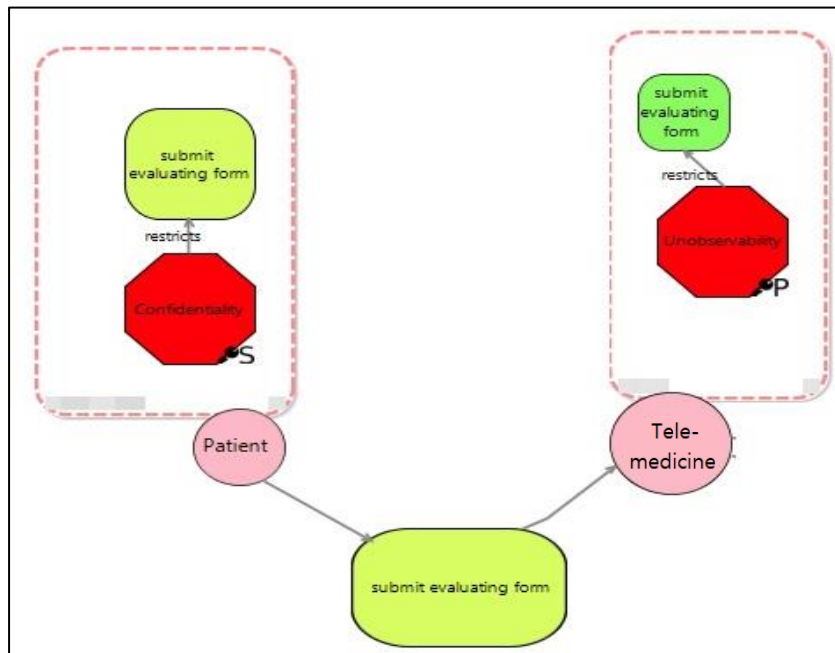
Furthermore, there could possibly be a conflict between the anonymity requirement for the 'Fill evaluation form' task and the non-repudiation requirement for the 'Submit evaluation' task. For instance, imagine a flow between two tasks in which the first task allows a customer to anonymously use a service and the second task allows the service provider to prevent a customer from being able to deny his payment for receiving a service. In this situation, it may

be sufficient for a service provider to prove that a customer performed the payment task without revealing which service the customer is paying for, and as a consequence, preserve the customer anonymity. Such potential conflicts should be reported and discussed early on in order to mitigate them.



**Figure 6.5 Completing the evaluation form**

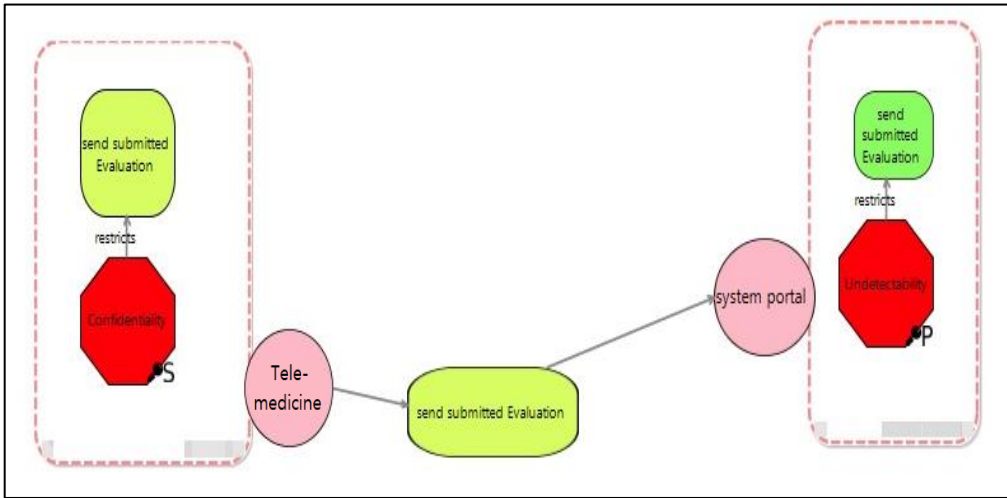
Additionally, the scenario model includes an unobservability requirement linked with the message flow between the 'Submit evaluation' task and 'Receiving the evaluation form' (see Figure 6.6). This requirement specifies that outsider adversaries should not be able to detect the true messages being sent over the message flow from false ones, also ensuring confidentiality, which with associated to the message flows, ensuring that the content of the message is preserved and not accessed by unauthorised users.



**Figure 6.6 Submitting the evaluation form**

Such potential conflicts would need to be reported and discussed early on in order to mitigate them. Also, linked with the confidentiality requirement is undetectability as this links the message flow between the ‘Send data to portal’ task and the ‘Receive data’ start event (see Figure 6.7). This specifies that outsider adversaries must not be able to distinguish true messages sent over the message flow between the ‘Send data to portal’ task and the ‘Receive data’ event from a false one. In other words, at a specific time, an outsider adversary should not be able to detect whether or not the tele-medicine device is sending data, thereby disrupting the confidentiality of the patient.

;



**Figure 6.7 Sending the submitted evaluation form**

We listed security and privacy requirements that are in conflict in Table 6.1.

**Table 6.1 Conflicting Requirements (security/privacy)**

<i>Security Requirements Conflict with:</i>	<i>Privacy Requirements</i>
BOD	Unlinkability
Accountability	Anonymity
Non-repudiation	Anonymity
Confidentiality	Undetectability
	Unobservability

### **6.4.3 Phase 3: Resolve conflicts based on Support Techniques:**

In Table 6.2, the patient requests to misuser by the tele-medicine device, and as a result conflicts between unlinkability and BOD can arise. The CPS reveals the supporting tool – data hiding to be used to mitigate the conflict (see Table 6.2). Next, when the tele-medicine device sends the data to the system portal, conflicts may arise between accountability and anonymity. At this point, we can apply mitigating tools – cryptographic, steganographic technologies, IDMIX and/or Onion Routing (see Table 6.3).

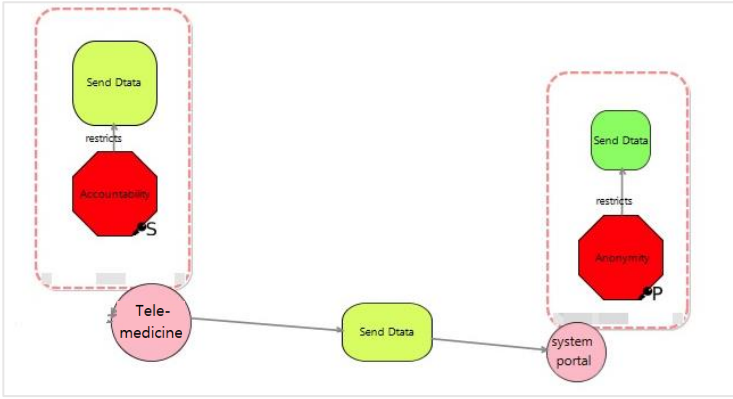
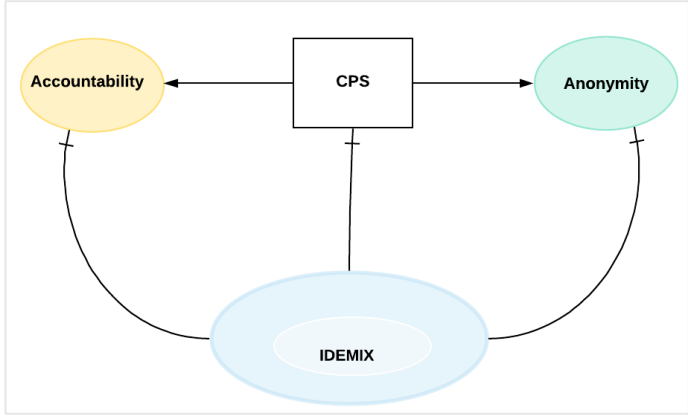
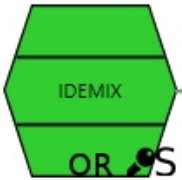
Next, the patient will evaluate this service by accessing the evaluation form from the tele-medicine device. This step requires the evaluator of the patient to be anonymous, while the information should be non-repudiated using the tele-medicine device. Mitigating this conflict calls for the use of Dummy Traffic supporting tools (Table 6.4). When the patient proceeds to now submit the evaluation form to the tele-medicine device, confidentiality of those forms and unobservability at the tele-medicine device is crucial, in sending the forms to the system portal in the final step (Table 6.5).

The recommended supporting tool for that stage is therefore Dummy Traffic. Lastly, in sending the evaluation forms from the tele-medicine to the system portal, the information must still be kept confidential and undetectable. In order to support both requirements, the optimum tool is Steganographic technologies (Table 6.6). Justification for use of these tools is further elaborated in Section 4.2-1-3.

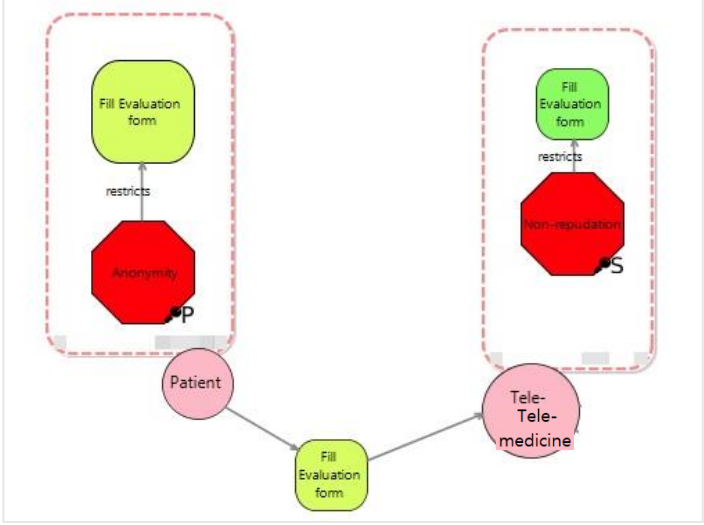
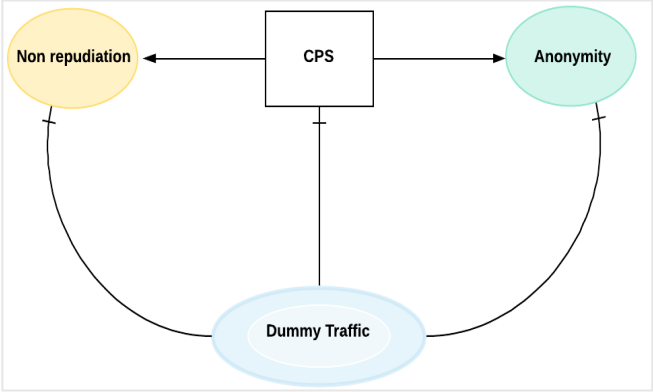

**Table 6.2 Unlinkability vs BOD**

<b>NAME</b>	<b>UNLINKABILITY VS BOD</b>
<b>PROBLEM</b>	Patient requests to misuse by the tele-medicine device
<b>MODEL</b>	
<b>FORCE</b>	This step could result in conflicts between binding of duties as this goal could be achieved by multiple patients at the same time, and unlinkability as this goal might not be linked with other goals
<b>SOLUTION</b>	<p>We apply the Data Hiding supporting tool:</p>
	<p>- Mechanism to be added in SecTro:</p>

**Table 6.3 Anonymity vs Accountability**

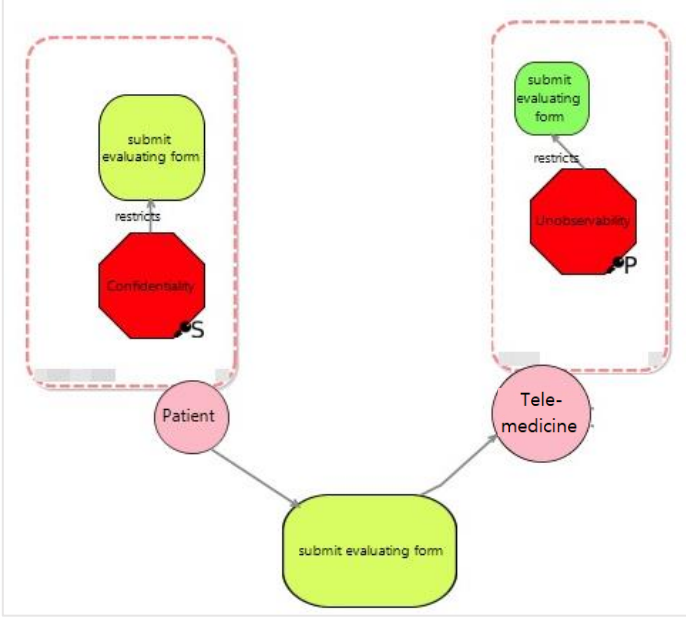
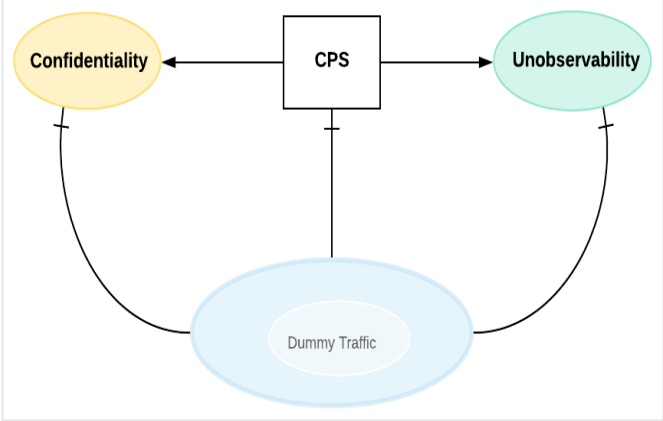

<i>NAME</i>	<b>ANONYMITY VS ACCOUNTABILITY</b>
<i>PROBLEM</i>	The tele-medicine device will send the data to the system portal
<i>MODEL</i>	 <p>The diagram illustrates the data flow from a Tele-medicine device to a system portal. A central 'Send Data' node (green oval) connects the 'Tele-medicine' node (pink circle) to the 'system portal' node (pink circle). On the Tele-medicine side, a 'Send Data' node (green oval) is connected to an 'Accountability' restrictor (red octagon) labeled 'AS'. On the system portal side, a 'Send Data' node (green oval) is connected to an 'Anonymity' restrictor (red octagon) labeled 'AP'. Both restrictors are connected to their respective 'Send Data' nodes with arrows labeled 'restricts'. The entire process is enclosed in a dashed red box.</p>
<i>FORCE</i>	This step could have conflicts between requirements such as accountability and keeping those data anonymous when it is sent to the system portal
<i>SOLUTION</i>	<p>In order to support both requirements accountability and anonymity, we can apply one of the tools: Cryptographic, Steganographic technologies, Onion Routing, IDMIX</p>  <p>The diagram shows a central 'CPS' box connected to two ovals: 'Accountability' (yellow) on the left and 'Anonymity' (teal) on the right. Below these is a large light blue oval labeled 'ID MIX'. Arrows point from 'CPS' to both 'Accountability' and 'Anonymity'. Curved arrows also connect 'ID MIX' to both 'Accountability' and 'Anonymity'.</p>
	<p>- Mechanism to be add in SecTro:</p>  <p>A green hexagon with 'ID MIX' in the top half and 'OR AS' in the bottom half.</p>

**Table 6.4 Anonymity vs Non-repudiation**


<b>NAME</b>	<b>ANONYMITY VS NON-REPUDIATION</b>
<b>PROBLEM</b>	Patient can evaluate this service by accessing the evaluation form from the tele-medicine device
<b>MODEL</b>	
<b>FORCE</b>	This step requires the evaluator of the patient to be anonymous, while the information should be non-repudiated via the tele-medicine device
<b>SOLUTION</b>	<ul style="list-style-type: none"> <li>- In order to support both requirements, the best supporting tool is Dummy Traffic</li> </ul> 
	<ul style="list-style-type: none"> <li>- Mechanism to be added in SecTro: </li> </ul>



**Table 6.5 Confidentiality vs Unobservability**

<i>NAME</i>	<b>CONFIDENTIALITY VS UNOBSERVABILITY</b>	
<i>PROBLEM</i>	Patient can submit evaluation form to the tele-medicine device	
<i>MODEL</i>		
<i>FORCE</i>	This step requires confidentiality of those forms and unobservability at the tele-medicine, in order to send those forms to the system portal in the final step	
<i>SOLUTION</i>	<p>In order to support both requirements, the optimum tool is Dummy Traffic</p>  <p>- Mechanism to be added in SecTro:</p> 	

**Table 6.6 Confidentiality vs Undetectability**

<i>NAME</i>	<b>CONFIDENTIALITY VS UNDETECTABILITY</b>
<i>PROBLEM</i>	Send the evaluation forms from tele-medicine device to system portal
<i>MODEL</i>	
<i>FORCE</i>	The information must be kept confidential from previous step, as well as being undetectable
<i>SOLUTION</i>	<p>In order to support both requirements, the optimum tool is steganographic technologies</p>
<p>- Mechanism to be added in SecTro: </p>	

## 6.5 Chapter Summary

In this chapter, we apply a step-by-step Secure Tropos framework to the E-health scenario. First, we justify the use of Secure Tropos in Section 6.2, proceed to identify the privacy and security requirements, and then the potential conflicts likely to arise in Section 6.4. Organisational pictorial views are shown throughout the E-health scenario stages, to take in the ‘big picture’ quickly, helping to simplify the information and reduce complex data to easily digestible illustrations. Lastly, bearing in mind the limitation (forces) in mitigating conflicts of which the analyst must be aware, appropriate supporting tools are determined in Section 6.4.3. In the next chapter, we will apply the framework proposed here and its applicability to the DEFEND project – its themes and services, showing the importance of the platform which empowers organisations across various sectors, to assess their compliance status with EU-GDPR regulation on data protection and privacy. The strength of the ConfIS framework is that it has provided a systematic method of conflict resolution by firstly identifying security and privacy requirements, before moving on to detect potential conflicts and seek tools to mitigate these. Nothing like this has been performed in the previous literature or research. At the same time, the framework also revealed the possibility of seeking methods or steps to specify how and why which security and privacy requirements are identified.

# CHAPTER 7

## APPLYING THE FRAMEWORK ON A REAL CASE STUDY (DEFEND PROJECT)

### 7.1 Introduction

In this chapter, we apply the ConfIS framework to the DEFEND<sup>3</sup> project. Firstly, the DEFEND project themes and services are elaborated upon in Sections 7.2 to 7.4, showing the importance of the platform which empowers organisations across various sectors, to assess their compliance status with GDPR regulations in terms of the European Union (EU) law on data protection and privacy. Secondly, in Section 7.5, we apply the ConfIS framework to an E-Health scenario, modelling the example using a SecTro tool which is also available to the public. Thereafter, we apply the framework to some examples from the E-health scenario in Section 7.6. The examples identify the requirements, the potential conflicts arising, organisational and privacy by design views of the analysis, all embedded within the three-part phase of the ConfIS framework. Lastly, conflict resolutions are mapped out in Phase 3, and discussions presented on all examples, presenting a step-by-step comprehensive explanation of the framework. Next, Section 7.7 presents the benefits of using the proposed ConfIS framework within the boundaries of DEFEND and finally, we conclude in Section 7.8.

---

<sup>3</sup> <https://www.defendproject.eu/>

## 7.2 Data: Privacy and Security within DEFEND

The CSIUS group (Centre for Secure, Intelligent and Usable Systems) at the University of Brighton have a mission to integrate tools and development in support of General Data Protection Regulation (GDPR), which is applied on the DEFEND project<sup>4</sup> (Piras *et al.*, 2019).

DEFEND is a European partnership that will provide a platform to empower organisations in different regions to consider and comply with the European Union's General Data Protection Regulation (GDPR). We will describe more about participation in this project later in this chapter.

## 7.3 DEFEND Project

In order to successfully plan the achievement of GDPR compliance and raise recognition of its diverse features, DEFEND will deliver a platform that empowers organisations across various sectors, to assess compliance status. The DEFEND project was chosen as a case study, as it is a platform which is designed to empower organisations in different sectors to assess and comply to the European Union's General Data Protection Regulation (GDPR). It is therefore very relevant and appropriate to test the ConfIS framework. Furthermore, the University of Brighton has contributed a significant role in contributing to the DEFEND project, and as such, this research seeks to improve the present software tools and frameworks, thereby seeking to develop and improve data privacy governance. In particular, the project's technical focus is on delivering the novel Data Privacy Governance for Supporting GDPR (DEFEND) platform, which supports organizational-focused privacy governance and addresses challenges faced by organisations when complying with GDPR.

---

<sup>4</sup> <https://www.defendproject.eu/university-of-brighton/>

DEFEND permits the design and analysis of models following a Privacy-by-Design approach. This is applied to Planning and Operational Levels, via three management areas – Data Scope, Data Process and Data Breach. The platform will influence existing software, tools and methodologies regarding the implementation of the platform software components. The DEFEND platform is already being tested in pilot studies and focus groups, collaborating with partners from four EU countries in four different areas, namely healthcare, banks, energy and local public administration. Afterwards, it will be tested in an effective environment, under three scenarios, via two different viewpoints – the first focusing on the GDPR compliance process for end-users, and secondly, GDPR associations for external stakeholders.

In terms of the new EU GDPR regulation law on data protection and privacy, it is intended to give more control to individuals over their personal data. The university's Centre for Secure, Intelligent and Usable Systems (CSIUS) is heading the EU-funded Data Governance for Supporting GDPR project, DEFEND. This will provide an innovative data privacy governance platform, to empower organisations across different sectors to assess and comply with the law, increasing their operational capacity.

The Defend project seeks to deliver a platform that empowers organisations across various sectors, to assess compliance status. Developing the ConfIS framework supports this initiative, by determining if there are security and privacy requirements involved, and any pending conflicts that can be detected and mitigated early on. Through each segment of the case scenario, the security and privacy requirements are identified which has not been undertaken previously in the DEFEND project. For purposes of this thesis, the ConfIS framework is applied to the DeFend project, with emphasis on Data Scope Management (DSM) at the operational

level namely Data Protection Impact Assessment (DPIA), Security and Privacy Threats, and Privacy by Design.

To apply and evaluate the framework to a relevant, real case, this research had the possibility to do that by collaborating with DEFEND EU project researchers. Specifically, a scenario was chosen with a representative and a relevant number of requirements from real settings, and DEFEND gave the possibility to consider different potential ones, coming from important sectors such as the ones of healthcare, banking, public administration and smart energy (according to the related DEFEND pilots).

Specifically, this research selected the healthcare scenario of DEFEND, because it dealt with critical user information and the personal data of patients. It furthermore found potentiality for identifying and solving different privacy/security requirements and conflicts. Accordingly, this thesis has different parts that introduce DEFEND, its context, and in particular the healthcare scenario to which the framework was applied for evaluating it, by involving privacy/security experts.

Therefore, the input taken from DEFEND, for the ConfIS framework, has been mainly the healthcare scenario. Based on that, it has been possible to apply the framework and its phases. It has also been possible to discover and categorise privacy and security requirements, identify potential conflicts and find solutions for them.

Furthermore, experts are used in the evaluation and a real scenario to discuss the framework and its aspects with them. Therefore, the framework has been used in parallel with the DEFEND platform due to the technical aspects and needs of the project. It has shown potentiality for

enhancing the platform, as future potential work, concerning privacy/security requirements classification and conflict resolution.

### **7.3.1 Themes and services:**

The goal of this deliverable is to identify the privacy and security requirements for the DEFEND platform in general, and where appropriate, for specific areas. The requirements identified by Tsohou (2020), on the basis of GDPR, describe for each of the 12 themes identified in the DEFEND platform, which requirements need to be met in order for it to meet its objectives. The security requirements identified by the DEFEND consortium describe which measures need to be taken and implemented for the platform to be protected against possible security risks and threats (Piras *et al.*, 2019).

On the basis of the predefined platform and its 12 key themes, Tsohou (2020) has identified the relevant corresponding GDPR obligations. Subsequently, she has transposed these obligations into legal requirements. In this sense, 'transposing', on the basis of the general and abstract GDPR obligations, taking into account the specific DEFEND platform components, identifying concrete, practical legal requirements for the platform to meet, in order for it to support organisations in complying with the GDPR (see Table 7.1).



**Table 7.1 GDPR 12 Theme Obligations**

<b>Theme ID</b>	<b>Theme Description</b>
<i>Theme 1</i>	Developing a GDPR Privacy Plan
<i>Theme 2</i>	Creating a Third-Party Management Programme
<i>Theme 3</i>	Managing Privacy Complaints and Individual Rights
<i>Theme 4</i>	Managing Privacy Incidents and Breach Notification
<i>Theme 5</i>	Implementing PbD/Privacy Engineering
<i>Theme 6</i>	Data De-Identification/Anonymisation
<i>Theme 7</i>	Meeting Regulatory Reporting Requirements
<i>Theme 8</i>	Addressing International Data Transfers
<i>Theme 9</i>	Creating Data Inventory and Maps
<i>Theme 10</i>	Conducting Privacy Risk Assessments (PIAs/DPIAs)
<i>Theme 11</i>	Obtaining and Managing User Consent
<i>Theme 12</i>	Selection of Appropriate Security Technical and Organisational Measures
<i>General Platform</i>	General DEFEND Platform Requirements

(Tsohou, 2020)

After presenting themes of the DEFEND platforms, those themes are divided into a related service, shown in Table 7.2.

**Table 7.2 GDPR Service Description and Related Themes**

<b>Service Description / ID</b>	<b>Related Themes</b>
Data Scope Management ( <i>DSM</i> )	Theme 2, Theme 5, Theme 9, Theme 10, Theme 12
Data Process Management ( <i>DPM</i> )	Theme 2, Theme 3, Theme 6, Theme8, Theme9, Theme 11, Theme 12
Data Breach Management ( <i>DBM</i> )	Theme 4, Theme 7
GDPR Reporting	Theme 1, Theme 2, Theme 3, Theme 4, Theme 7, Theme 9, Theme 10
GDPR Planning	Theme 1, Theme 2

(Tsohou, 2020)

## 7.4 Data Scope Management

For the purposes of this research, the DEFEND Data Scope Management service (DSM) aspect of the DEFEND project will be elaborated. The aim of DSM is to support organisations at an operational level to ensure continuous GDPR compliance through Model-Based Privacy by design analysis. Here, we present important Privacy by Design activities and strategies, then describe DSM, its design, flow and a preliminary case study and evaluation performed with pilots from the healthcare scenario.

### 7.4-1 Activities and Strategies (AS) for PbD

The GDPR Action Plan of Activities and Strategies (AS), which are important for PbD and relevant for this research, identifies AS4: GDPR Data Syntheses, Graphical Representations and Model-Based, Visual Support.

It is beneficial to provide further support and guidance with graphical representations and synthesis of GDPR information analysed and collected. These should be provided to business analysts, privacy/security experts and other end-users involved, based on the completion of the GDPR Self-Assessment, and at support to other activities (e.g., Data Protection Impact Assessment, data minimization analysis, creation of GDPR action plans). While, privacy/security analysis threat analysis, continuous risk assessment configurations, and other critical activities and analyses, could be performed supported by visual model-based techniques enhanced and adapted for GDPR purposes (Piras *et al.*, 2020, p. 186-201).

AS5: Data Protection Impact Assessment (DPIA), Preventive/Reacting Analyses and GDPR Action Plan.

It is important to analyse, in a preliminary way, GDPR lacks, vulnerabilities and assets that can be affected by data issues/breaches, and which preliminary mitigation mechanisms to adopt, and if preventive/reactive actions are in place (e.g., data breach plans). These analyses should be performed for producing a DPIA and a GDPR Action Plan, for identifying current gaps of compliance of an organization, on which to perform further PbD analysis (Piras *et al.*, 2020, p. 186-201).

AS6: Privacy/Security Model-Based, and Pattern-Based, Analysis.

The GDPR Action Plan of AS5 identifies the gaps, but it is at high-level, thus, needs to be enacted by further critical analysis, performed by privacy/security analysts, supported by visual model-based techniques enhanced and adapted for GDPR (AS4). This concerns analysis of the organization context, data/assets/accountability mapping with also analysis of risks, threats and measures in place, privacy/security requirements constraints and conflict resolution, supported via libraries of patterns and modelling techniques specifically designed for GDPR (Piras *et al.*, 2020, p. 186-201).

Based on the above set of Activities and Strategies (AS), the Data Scope Management service (DSM) for the DEFEND platform to support PbD shows that it enables organisations to execute DPIA (AS5) and elaborates other information collected for supporting the organisations with data synthesis and graphical representations (AS4) through a set of DSM tools. Moreover, DSM helps organisations to perform threats analysis (AS4, AS6), data minimisation analysis (AS4), privacy/security analysis and design with tool-supported modelling techniques (AS4, AS6) as well as continuous risk assessment (AS4, AS6).

Furthermore, AS5 identifies the gaps, but it is at a high level, and thus needs to be enacted by further critical analysis, performed by privacy/security analysts, supported by visual model-based techniques enhanced and adapted for GDPR (AS4). This concerns analysis of the organisation context, data/assets/accountability, also mapping with analysis of risks, threats and measures in place, privacy/security requirements constraints and conflict resolution, supported via libraries of patterns and modelling techniques specifically designed for GDPR.

AS6 in detail, in DSM, is performed via Organisational Structure Analysis, Data Mapping and Risk Models Analysis, Privacy/Security Requirements Analysis, Requirements Conflicts Analysis and Resolution based on Patterns, Threat Analysis, Attacks Analysis and Security Measures Identification based on Patterns. This research focuses more on AS6, building from AS4 and AS5 security and privacy requirements, which is needed to comply with GDPR and

PbD regulation. Conflicts between privacy and security requirements are very common, and they exist in almost every sector – banking, education and health care. These sectors are required to ensure users’ privacy whilst also maintaining system security and invulnerability. Failure in handling requirement conflicts is one of the main reasons for failure in software projects which is caused by cost and lack of time. It is essential to detect and resolve conflicts in early phases in order to prevent re-iterations of all phases.

#### **7.4-2 Model DSM Themes with Related Requirements**

We apply the framework to the DSM service. In order to gain an overview of the related service and themes, we have a list of approximately 300 requirements in Microsoft Excel format for the whole project (Piras *et al.*, 2020; Tsohou, 2020). Referring to the twelve themes of the DSM platform as shown in Table 7.2, each theme has its own requirements, while some requirements can also be related to different themes. In terms of the latter, we can possibly link this to related requirement in the particular theme, which makes this an action of linking between the theme’s requirements and related requirements. This helps us to detect/identify any type of conflicts that could arise.

#### **7.5 Data Scope Management (DSM) storyline**

This section outlines a motivation scenario that is used in our case study to apply the ConfIS framework, aiming to achieve conflict resolution. This is presented to participants of the pilot study, showing the DEFEND platform in order to show them how to receive a tool to support conflict problems (Piras *et al.*, 2020).

One of the most critical aspects is managing patient medical records. There should be verification from the supervisor of any changes to the records, and to establish a retention

period for this data. Furthermore, the data must not be stolen or compromised – for instance, in relation to potential threats and data breaches – therefore, a hospital needs to put in place monitoring systems for those potential problems.

Third parties are also involved in the organisational processes (e.g. external laboratories for medical examinations). Therefore, these must also be considered in achieving GDPR compliance. Based on the answers to these issues, the platform generates Data Protection Impact Assessments (DPIA) and a risk assessment, by highlighting the importance of achieving confidentiality and integrity in patient medical records. This is followed by validation processes and a proposed GDPR plan.

In this scenario, the hospital analyst improves their graphical representation by modelling how a medical doctor can change the patients' medical record (for instance by adding examination results received by third parties) and obtain validation from a supervisor. The system helps a hospital security analyst in modelling potential threats that could affect the confidentiality and integrity of this important data, along with privacy and security measures that could mitigate those potential problems.

For instance, a hospital can achieve GDPR compliance by using the DEFEND platform (Piras *et al.*, 2019). When a hospital starts using the platform, it provides the system with relevant organisational information by compiling questionnaires. Based on the information collected and elaborated upon in the assessment, the platform displays graphically (models) the organisational structure of the hospital, with the main actors and interactions. The system also helps in modelling the data retention periods for any kind of data managed in this process. Based on the self-assessment, Data Protection Impact Assessments (DPIA), risk assessment,

processes modelled for changing and validating data, and related modelling of potential threats, the system generates a configuration for monitoring those threats.

A hospital security analyst reads these assessments and optionally improves them by adding further specific information. After all these complex analyses, the system can monitor threats (by using other services and components of the DEFEND platform). This example will be used in the case study section, in which we demonstrate an intrinsic conflict discovered and resolved by this process.

### 7.5-1 Linking Scenarios with Associated Requirements

In this section, we will list each scenario with its requirements. In Table 7.3 the medical scenario is presented. Through each segment A to M, the security and privacy requirements are identified in this thesis, which has not been undertaken previously in the DEFEND project. This helps the analyst to allocate each scenario to the related requirements. By applying this newly introduced step, this thesis helps to identify potential conflicts which can be more easily identified thereafter (see Table 7.3).

**Table 7.3 Identifying Privacy/Security Requirements in Medical Scenario**

SEGMENT	SCENARIO	REQ.ID	SECURITY/PRIVACY REQ.
A.	ONE OF THE MOST CRITICAL ASPECTS IS TO MANAGE THE PATIENT MEDICAL RECORD AND TO HAVE VERIFICATION, FROM A SUPERVISOR, FOR ANY CHANGES HAPPENING TO IT (FOR INSTANCE ADDING A NEW EXAMINATION RESULT, ETC.), AND TO ESTABLISH A	REQ09.06 REQ09.24 REQ09.25 REQ05.08	ACCOUNTABILITY AUDIBILITY ANONYMITY

	RETENTION PERIOD FOR THIS DATA		
<b>B.</b>	FURTHERMORE, THIS DATA HAS NOT TO BE STOLEN OR TO BE COMPROMISED; FOR INSTANCE, IN RELATION TO POTENTIAL THREATS AND DATA BREACHES; THEREFORE, THE HOSPITAL NEEDS TO PUT IN PLACE MONITORING OF THOSE POTENTIAL PROBLEMS; IN THE ORGANISATIONAL PROCESSES, THIRD PARTIES ARE ALSO INVOLVED (EXTERNAL LABORATORIES FOR MEDICAL EXAMINATIONS), THEREFORE IT IS NECESSARY TO ALSO CONSIDER THIS FOR ACHIEVING GDPR COMPLIANCE	REQ12.01 REQ12.02 REQ02.12 REQ02.03	ACCOUNTABILITY AUDIBILITY ANONYMITY NON-REPUDIATION
<b>C.</b>	ON THE BASIS OF THE ANSWERS TO THE PREVIOUS SEGMENTS, THE PLATFORM GENERATES DPIA, RISK ASSESSMENT, BY HIGHLIGHTING THE IMPORTANCE OF FULFILLING THE CONFIDENTIALITY AND INTEGRITY OF THE PATIENT MEDICAL RECORD, THROUGH VALIDATION PROCESSES, AND PROPOSES A GDPR PLAN1 FOR THIS	REQ05.07 REQ10.01 REQ10.11	ACCOUNTABILITY INTEGRITY CONFIDENTIALITY

<b>D.</b>	THE HOSPITAL ANALYST IMPROVES THE GRAPHICAL REPRESENTATION BY MODELLING HOW A MEDICAL DOCTOR CAN CHANGE THE PATIENT MEDICAL RECORD (FOR INSTANCE BY ADDING EXAMINATION RESULTS RECEIVED BY THIRD PARTIES AS EXTERNAL LABS) AND OBTAINING VALIDATION FROM A SUPERVISOR	REQ09.15 REQ02.03 REQ02.04 REQ02.06 REQ09.04 REQ02.11	AUTHORISATION- ACCOUNTABILITY ANONYMITY
<b>E.</b>	FINALLY, THE SYSTEM HELPS A HOSPITAL SECURITY ANALYST TO MODEL POTENTIAL THREATS THAT COULD AFFECT CONFIDENTIALITY AND INTEGRITY OF THIS IMPORTANT KIND OF DATA, AND PRIVACY AND SECURITY MEASURES THAT COULD MITIGATE/SOLVE THOSE POTENTIAL PROBLEMS	REQ05.07 REQ12.01 REQ12.02	CONFIDENTIALITY INTEGRITY AUDIBILITY NON-REPUDIATION
<b>F.</b>	OUTPUT: HIGH-LEVEL ANALYSIS/DESIGN RESULTS AND HIGH-LEVEL CONFIGURATIONS (E.G. FOR MONITORING POTENTIAL THREATS)	REQ12.01 REQ12.02 REQ12.06	AUDIBILITY NON-REPUDIATION
<b>G.</b>	A HOSPITAL WANTS TO ACHIEVE GDPR COMPLIANCE BY USING THE DEFEND PLATFORM	REQ10.05 REQ05.08 REQ05.09	ACCOUNTABILITY INTEGRITY ANONYMITY
<b>H.</b>	THE HOSPITAL STARTS USING THE DEFEND PLATFORM AND PROVIDES THE SYSTEM WITH RELEVANT ORGANISATIONAL INFORMATION, BY COMPILING QUESTIONNAIRES	REQ09.02	ACCOUNTABILITY AVAILABILITY



<b>I.</b>	THE PLATFORM, ON THE BASIS OF THE INFORMATION COLLECTED WITHIN THE ASSESSMENT AND THE ELABORATED PLAN, SHOWS GRAPHICALLY (MODELS) THE ORGANISATIONAL STRUCTURE OF THE HOSPITAL, WITH THE MAIN ACTORS AND INTERACTIONS	REQ09.01 REQ09.03 REQ09.04	ACCOUNTABILITY CONFIDENTIALITY ANONYMITY
<b>J.</b>	THE SYSTEM ALSO HELPS TO MODEL THE DATA RETENTION PERIODS FOR ANY KIND OF DATA MANAGED IN THIS PROCESS	REQ09.06 REQ09.01 REQ12.06 REQ02.02	ACCOUNTABILITY AUDIBILITY CONFIDENTIALITY ANONYMITY
<b>K.</b>	THE SYSTEM, ON THE BASIS OF THE SELF ASSESSMENT, DPIA, RISK ASSESSMENT, PROCESSES MODELLED FOR CHANGING DATA AND VALIDATING THEM, AND RELATED MODELLED POTENTIAL THREATS, GENERATES A CONFIGURATION FOR PERFORMING MONITORING OF THOSE THREATS	REQ10.11 REQ10.06 REQ12.06	AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY
<b>L.</b>	A HOSPITAL SECURITY ANALYST WILL READ THESE ASSESSMENTS AND OPTIONALLY IMPROVE THEM BY ADDING FURTHER SPECIFIC INFORMATION	REQ10.02	AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY
<b>M.</b>	AFTER ALL THESE COMPLEX ANALYSES, THE SYSTEM IS ABLE TO PERFORM MONITORING OF THREATS (BY USING OTHER SERVICES AND COMPONENTS OF THE DEFEND PLATFORM)	REQ10.06 REQ10.11	AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY ANONIMITY

## 7.5-2 Assigning Appropriate Tools to Privacy and Security Requirements in the Medical Scenario

In this section, we sort each segment of the scenario, connecting them with the most appropriate tools to fulfil the type of requirement demanded. For a more understandable approach and clearer analysis, segments are grouped per security and privacy requirement. In addition, we link each group (1-12) with the requirements ID for the medical scenario under study (see Table 7.4; Appendix A). For instance, in Table 7.4, security requirements and the appropriate tools to mitigate each conflict are identified for groups 1-8. For instance, confidentiality is identified for segments C, E, F, I, J, K, L, M (group 1):

*C- the platform generates DPIA, risk assessment, by highlighting the importance of fulfilling the confidentiality and integrity of the patient's medical record;*

*E- the system helps the hospital security analyst in modelling potential threats that could affect confidentiality and integrity of this important kind of data;*

*F- OUTPUT: high-level analysis/design results and high-level configurations (e.g. for monitoring potential threats);*

*I- the platform, on the basis of the information collected, the assessment and the plan elaborated, shows graphically (models) the organisational structure of the hospital, with the main actors and interactions included;*

*J- the system also helps to model the data retention periods for any kind of data managed in this process;*

*K- the system, on the basis of the self-assessment, DPIA, risk assessment, processes modelled for changing data and validating them, and related modelled potential threats, generates a configuration for performing monitoring of those threats;*

*L- the hospital security analyst reads this configuration and optionally improves it by adding further specific information;*

*M- finally, the system is able to perform monitoring of threats (by using other services and components of the DEFEND platform).*

Here, confidentiality in each segment is necessary to ensure GDPR compliance and no breach of confidentiality of patient data. To ensure this, the appropriate tools to mitigate conflict (as shown in Chapter 4) which may arise include cryptographic, accesses control enforcement, symmetric key and public key encryption, steganographic technologies, homomorphic encryption, Onion Routing and/or searchable encryption.

Additionally, groups 9-12 map out the relevant segments of the scenario, identifying privacy requirements and relevant tools. For instance, anonymity is identified for segments A, B, C, D, E, F, G, I, J and M in Group 9, with their relevant IDs specifying the level of priority for each (Appendix A):

*A- one of the most critical aspects is to manage the patient's medical record and to receive verification, from a supervisor, for any changes happening to it (for instance adding a new examination result, etc.) and to establish retention period for this data;*

*B- furthermore, this data has not to be stolen or to be compromised; for instance, in relation to potential threats and data breaches. Therefore, the hospital must put in place monitoring of those potential problems; organisational processes are involved as well as third parties (external laboratories for medical examinations), therefore it is necessary to consider this for achieving GDPR compliance;*

*C- on the basis of the answers to the above, the platform generates DPIA, risk assessment, by highlighting the importance of fulfilling the confidentiality and integrity of the patient's medical record, also through validation processes, and proposes a GDPR plan;*

- D- the hospital analyst improves the graphical representation by modelling how a medical doctor can change the patient's medical record (for instance by adding examination results received by third party external labs) and obtaining a validation for these changes from a supervisor;*
- E- finally, the system helps a hospital security analyst in modelling potential threats that could affect confidentiality and integrity of this important kind of data, and privacy and security measures that could mitigate/solve those potential problems;*
- F- OUTPUT: high-level analysis/design results and high-level configurations (e.g. for monitoring potential threats).*
- G- A hospital wants to achieve GDPR compliance by using the DEFEND platform;*
- I- the platform, on the basis of the information collected, the assessment and the plan elaborated, shows graphically (models) the organisational structure of the hospital, with the main actors and interactions;*
- J- the system helps to model the data retention periods for any kind of data managed in this process;*
- M- after these complex analyses, the system is able to perform monitoring of threats (by using other services and components of the DEFEND platform).*

Here, the privacy requirement of anonymity in each segment is necessary as the system will need to allow the user access for providing sensitive information without unveiling their identity. To ensure this, without any conflicts arising, the appropriate tools necessary include cryptographic technology, Steganographic Technologies, Onion Routing, Trusted Third

Parties, Dummy Traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), and K-anonymity (Chapter 4.2-1-3).

With regards to the requirements ID, we identify security/privacy requirements for each ID, illustrated in Appendix A. The list has been prioritised using the MoScow Technique, which helps the analyst to identify the level of priority for each requirement. For instance, the priority level for ID REQ09.24, ‘*The DEFEND Platform shall support the creation of the record of processing activities when the organization acts as data controller*’, is mandatory and is identified as *Must*. On the other hand, for ID REQ09.04, ‘*The DEFEND platform shall allow for graphical representation of specific relationships between third parties (e.g., joint controller) and the organization*’, the level of priority is *Should* as it would be quite helpful to the analyst but not obligatory (Appendix A; Table 7.4).

**Table 7.4 Identify Requirements and Tools to Mitigate Conflict**

GROUP	SEGMENT	REQUIRED ID	SECURITY REQ.	TOOLS
1	C, E, F, I, J, K, L, M	REQ05.07, REQ10.01, REQ10.11, REQ12.01, REQ12.02, REQ12.06	<b>Confidentiality</b>	Cryptographic, accesses control enforcement, Symmetric key, public key encryption, Steganographic technologies, Homomorphic encryption, Onion Routing, Searchable encryption
2	A, C, E, F, G	REQ05.07, REQ10.01, REQ10.11, REQ12.01, REQ12.02, REQ12.06, REQ09.06, REQ09.24, REQ09.25, REQ05.08	<b>Integrity</b>	Cryptographic, Accesses Control Enforcement, Message Authentication Codes (MAC) Redundancy and Comparison
3	A, B, C, D, E, G, H, I, J, K, L, M	REQ09.06, REQ09.24, REQ09.25, REQ05.08, REQ12.01, REQ12.02, REQ02.12, REQ02.03, REQ05.07, REQ10.01, REQ10.11, REQ09.15	<b>Accountability</b>	ADOPT

		REQ02.03, REQ02.04, REQ02.06, REQ09.04, REQ02.11		
4	A, B, E, J	REQ09.06, REQ09.24, REQ09.25 REQ05.08, REQ12.01, REQ12.02 REQ02.12, REQ02.03, REQ05.07	<b>Audibility</b>	Cryptographic, Steganographic Technologies, Onion Routing
5	B, E	REQ12.01, REQ12.02, REQ02.12 REQ02.03, REQ05.07	<b>Non-repudiation</b>	Onion Routing, Dummy traffic
6	A, B, C, D, F, K, L, M	REQ12.01, REQ12.02, REQ12.06 REQ09.06, REQ09.24, REQ09.25 REQ05.08, REQ02.12, REQ02.03 REQ10.01	<b>Authorisation</b>	Accesses Control Enforcement
7	B, D, F	REQ12.01, REQ12.02, REQ02.12, REQ02.03, REQ09.15, REQ02.03 REQ02.04, REQ02.06, REQ09.04 REQ02.11, REQ12.06	<b>Authentication</b>	Trusted third parties, Message Authentication Codes (MAC)
8	H	REQ09.02	<b>Availability</b>	Redundancy
<b>GROUP</b>	<b>SEGMENT</b>	<b>REQ ID</b>	<b>PRIVACY REQ.</b>	<b>TOOLS</b>
9	A, B, C, D, E, F, G, I, J, M	REQ09.06, REQ09.24, REQ09.25 REQ05.08, REQ02.12, REQ02.03 REQ09.15, REQ02.03, REQ02.04 REQ02.06, REQ09.04, REQ02.11 REQ05.07, REQ12.01, REQ12.02 REQ12.06, REQ10.01, REQ10.11	<b>Anonymity</b>	Cryptographic, Steganographic Technologies, Onion routing, Trusted Third Parties Dummy traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), K-anonymity.

10	A, D, E, F	REQ09.06, REQ09.24, REQ09.25 REQ05.08, REQ09.15, REQ02.03, REQ02.04, REQ02.06, REQ09.04 REQ02.11, REQ05.07, REQ12.06 REQ12.01, REQ12.02	<b>Unlinkability</b>	Cryptographic, Steganographic Technologies, Homomorphic encryption, Onion Routing, Data Hiding, K-anonymity, Trusted Third Parties, Dummy Traffic.
11	A, C, D, F	REQ12.01, REQ12.02, REQ12.06 REQ09.15, REQ02.03, REQ02.04 REQ02.06, REQ09.04, REQ02.11 REQ05.07, REQ10.01, REQ10.11 REQ09.06, REQ09.24, REQ09.25 REQ05.08	<b>Unobservability</b>	Dummy Traffic
12	D, F	REQ12.01, REQ12.02, REQ12.06 REQ09.15, REQ02.03, REQ02.04 REQ02.06, REQ09.04, REQ02.11	<b>Undetectability</b>	Dummy Traffic, Steganographic Technologies

### 7.5-3 Identifying Conflicts between Requirements

In Table 7.5, we analyse segments of the medical case scenario, but now incorporating the possibility of both security and privacy requirements, and conflicts that may arise, finalising by recommending supporting tools. For instance, Segment A shows that the medical case firstly requires verification from a supervisor with the relevant medical information upon adding new examination results to the database. Due to this, conflict is likely to arise between authorisation (where the identified entity is provided with permission to access data or functional resources) and unobservability (denied from knowing for certain that a user is accessing a service, and the inability to track a user’s actions while using the service). Furthermore, conflicts arise between accountability (holding entities responsible for their actions or lack thereof) and anonymity (allowing entities to use the service without having to reveal their identity). Lastly audibility (ensuring that a trace can be done on the entity’s activities within the system) and anonymity

are just some ways in which conflicts can occur. Due to these conflicts, ADOPT and dummy traffic can be introduced as supporting tools to mitigate these risks as it is expected to remain within the confinements of GDPR stipulations.

Furthermore, in Segment B, the system must manage the data so it cannot be stolen/compromised and ensure this also with third party interventions, undertaking a certain amount of monitoring of these potential problems and dealing with them. Conflict is likely to arise between authentication (determining whether the user is in fact who he is declared to be) and anonymity (allowing the user to use the service without having to reveal his identity). The optimal supporting tool to deal with this issue is Onion Routing, as we discussed in Chapter 4. This research does not cover a set criterion for choosing any specific supporting tool, but rather Chapter 4 introduces each tool in depth and presents its use for mitigating conflicts between requirements.

Next, based on the answers to these issues, the platform generates data protection impact assessments (DPIA) and a risk assessment, by highlighting the importance of achieving key requirements – integrity, anonymity, confidentiality, undetectability and unobservability. This can be achieved by applying cryptography, onion Routing, steganographic technologies and likely dummy traffic.

In Segment D, the hospital analyst then improves the graphical representation by modelling how a medical doctor can change the patient medical record (for instance by adding examination results received by third parties) and obtain validation from a supervisor. Hypothetically, the system can help a hospital security analyst in modelling potential threats that could affect the confidentiality and integrity of this important kind of data, along with



privacy and security measures ADOPT and dummy traffic, which could mitigate those potential problems.

Furthermore, the hospital can achieve GDPR compliance by using the DEFEND platform. When the hospital starts using the platform it provides the system with relevant organisational information by compiling questionnaires. Based on the information collected and elaborated on in the assessment, the platform displays graphically (or models) the organisational structure of the hospital, with the main actors and interactions.

A hospital security analyst reads these graphs and optionally improves them by adding further specific information. After all these complex analyses, the system can monitor threats, by using other services and components of the DEFEND platform and incorporating the previously mentioned tools (see Table 7.5).

**Table 7.5 Privacy/Security Requirements and Supporting Tools to Mitigate Conflict**

SEGMENT	REQ.ID	SCENARIO	SECURITY REQ.	PRIVACY REQ.	REQUIREMENTS IN CONFLICT	SUPPORTING TOOL
<b>A</b>	REQ09.06, REQ09.24, REQ09.25, REQ05.08		ACCOUNTABILITY AUDIBILITY AUTHORISATION INTEGRITY	ANONYMITY UNLINKABILITY UNOBSERVABILITY		
<b>A.1</b>		verification from a supervisor			AUTHORISATION VS UNOBSERVABILITY	
<b>A.2</b>		adding a new examination result			ACCOUNTABILITY VS ANONYMITY  AUDIBILITY VS ANONYMITY	ADOPT, DUMMY TRAFFIC
<b>B</b>	REQ12.01, REQ12.02, REQ02.12, REQ02.03		ACCOUNTABILITY AUDIBILITY NON-REPUDIATION AUTHORISATION AUTHENTICATION	ANONYMITY		IDEMIX, ONION ROUTING, CRYPTOGRAPHIC, ACCESSES CONTROL ENFORCEMENT
<b>B.1</b>		data must not be stolen or compromised				
<b>B.2</b>		need to monitor those potential problems				
<b>B.3</b>		organisational processes are involved; also third parties			AUTHENTICATION VS ANONYMITY	CRYPTOGRAPHIC

C.	REQ05.07, REQ10.01, REQ10.11		ACCOUNTABILITY INTEGRITY CONFIDENTIALITY AUTHORISATION	ANONYMITY UNOBSERVABILITY		
C.1		DPIA, risk assessment			INTEGRITY VS ANONYMITY	CRYPTOGRAPHIC
C.2		fulfilling confidentiality & integrity of patient medical record			CONFIDENTIALITY VS ANONYMITY	CRYPTOGRAPHIC, ONION ROUTING, STEGANOGRAPHIC TECHNOLOGIES
					INTEGRITY VS ANONYMITY	CRYPTOGRAPHIC, STEGANOGRAPHIC TECHNOLOGIES
					CONFIDENTIALITY VS UNDETECTABILITY	ONION ROUTING, DUMMY TRAFFIC
D.	REQ09.15, REQ02.03, REQ02.04, REQ02.06, REQ09.04, REQ02.11	graphical representation by modelling how a medical doctor can change the patient medical record	AUTHORISATION ACCOUNTABILITY AUTHENTICATION	ANONYMITY UNLINKABILITY UNOBSERVABILITY UNDETECTABILITY	ACCOUNTABILITY VS ANONYMITY	ADOPT, DUMMY TRAFFIC
					AUTHENTICATION VS ANONYMITY	
E.	REQ05.07, REQ12.01, REQ12.02	security analyst to model potential threats	CONFIDENTIALITY INTEGRITY ACCOUNTABILITY AUDIBILITY NON-REPUDIATION	ANONYMITY UNLINKABILITY	CONFIDENTIALITY VS ANONYMITY	CRYPTOGRAPHIC CRYPTOGRAPHIC ADOPT, DUMMY TRAFFIC DUMMY TRAFFIC CRYPTOGRAPHIC
					INTEGRITY VS ANONYMITY	
					ACCOUNTABILITY VS ANONYMITY	
					NON-REPUDIATION VS ANONYMITY	
					INTEGRITY VS UNLINKABILITY	

<b>F.</b>	REQ12.01, REQ12.02, REQ12.06	high-level analysis/design results and high-level configurations	CONFIDENTIALITY INTEGRITY AUTHORISATION AUTHENTICATION	ANONYMITY UNLINKABILITY UNOBSERVABILITY UNDETECTABILITY	CONFIDENTIALITY VS ANONYMITY INTEGRITY VS ANONYMITY  INTEGRITY VS UNLINKABILITY	CRYPTOGRAPHIC, ONION ROUTING, STEGANOGRAPHIC TECHNOLOGIES CRYPTOGRAPHIC
<b>G.</b>	REQ10.05, REQ05.08, REQ05.09	achieve GDPR compliance by using the DEFEND platform	ACCOUNTABILITY INTEGRITY	ANONYMITY	ACCOUNTABILITY VS ANONYMITY  INTEGRITY VS ANONYMITY	ADOPT  CRYPTOGRAPHIC
<b>H.</b>	REQ09.02		ACCOUNTABILITY AVAILABILITY			N/A
<b>I.</b>	REQ09.01, REQ09.03, REQ09.04		ACCOUNTABILITY CONFIDENTIALITY	ANONYMITY		ADOPT, CRYPTOGRAPHIC
<b>J.</b>	REQ09.06, REQ09.01, REQ12.06, REQ02.02		ACCOUNTABILITY AUDIBILITY CONFIDENTIALITY	ANONYMITY		ONION ROUTING, CRYPTOGRAPHIC
<b>K.</b>	REQ10.11, REQ10.06, REQ12.06		AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY			N/A
<b>L.</b>	REQ10.02		AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY			N/A

M.	REQ10.06, REQ10.11	the system is able to perform monitoring of threats by using other services and components of the DEFEND platform	AUTHORISATION CONFIDENTIALITY ACCOUNTABILITY	ANONIMITY	AUTHORISATION VS ANONIMITY  CONFIDENTIALITY VS ANONIMITY  ACCOUNTABILITY VS ANONIMITY	ACCESSES CONTROL ENFORCEMENT  CRYPTOGRAPHIC, ONION ROUTING  IDEMIX
----	-----------------------	---	--	-----------	---	--

## 7.6 Applying the ConfIS framework to case study example

In this section, we present as an example part of the scenario, showing an organisational view of managing patients' records. Furthermore, Phases 1, 2 and 3 of the ConfIS framework are introduced, in which the scenario's security and privacy requirements are identified, followed by an understanding of their associated conflicts and lastly the development of a conflict resolution pattern.

**Table 7.6 Example- Phase 1: Mapping Security and Privacy Requirements**

Scenario	Security Req.	Privacy Req.
One of the most critical aspects is to manage the patient's medical record and to receive verification from a supervisor for any changes happening to it (for instance adding a new examination result) and to establish a retention period for the data	Accountability Audibility	Anonymity

Based on the storyline described in the example, we find that there are some security and privacy requirements involved. Therefore, to determine which requirements are in conflict, we model each scenario in a bubble. For instance, we give scenario A in example, we break into each task to assign a related requirement for it, as shown above in Table 7.6 and 7.7.

There is a need to fulfil the anonymity requirement for the 'Update Patient Medical Record' process. This process must ensure that nobody knows which medical doctor made the change to the records. In addition, the accountability constraint is related to the validate aspect of the 'Validate Medical Exam' process, i.e., a supervisor needs to validate the change. However, for this, the supervisor needs to know which medical doctor made the change; thus, there is a

conflict between accountability and anonymity, because the supervisor cannot know, due to the anonymity requirement, who the medical doctor is, so accountability cannot be fulfilled.

Next, the employee is expected to fulfil confidentiality and integrity while sending medical results; we fulfil this with the cryptographic mechanism. This is related to fulfilling GDPR principles, and an example is accountability, where it is necessary to record which medical doctor made the change.

Each scenario has security and privacy requirements, and subsequent conflicting requirements. For instance, anonymity as a privacy requirement conflicts with accountability as a security requirement. In Figure 7.1, we model the motivation example in SecTro to pinpoint the case study. Here, the employee at the external medical laboratory performs a medical examination on a pregnant patient. He obtains the results and sends them to the medical doctor of the maternity ward. The medical doctor then manages the patient’s medical record by obtaining the new medical result and updating the patient’s medical records. There is a security policy put in place by the hospital to ensure that the updating and supervision of patients’ medical records comply with security policies. The medical doctor will then send these new medical results to the supervisor for validation (see Figure 7.1)

**Table 7.7 Identifying Requirements for each Scenario**

<b>Scenario</b>	<b>Potential Requirement Conflict</b>
Update Patient Medical Record	Anonymity
Validate Add Medical Exam	Accountability
Sending Medical Result	Confidentiality and integrity
Update Patient Medical Record	Accountability

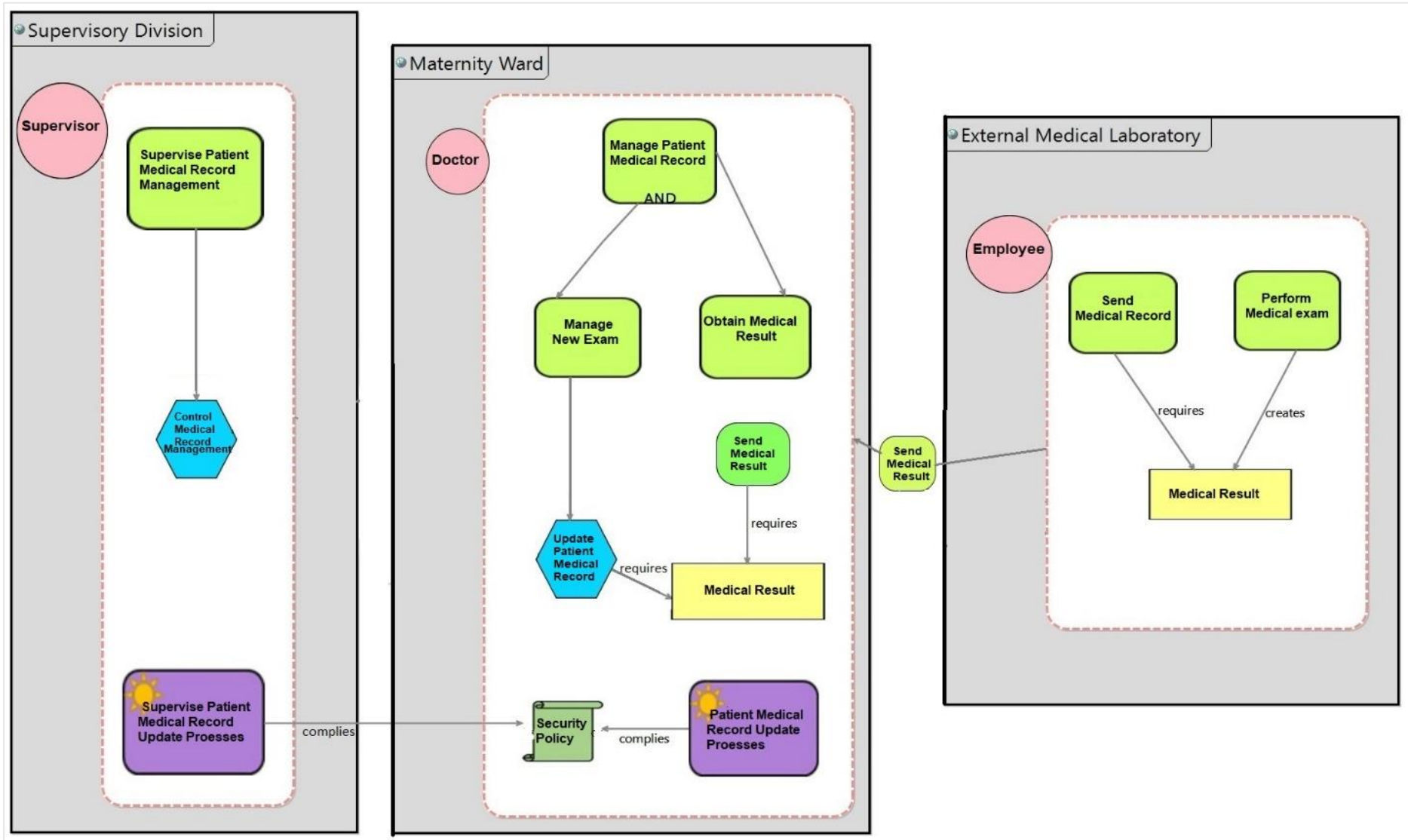


Figure 7.1 Organisational View of Managing Patient Records



### **7.6-1 Phase 2: Identify Conflicts between Requirements and Conflict Decisions**

To identify conflicts, we divide each scenario to illustrate possible conflicts. Therefore, for each case, we assign the requirements that are involved, as shown in Tables 7.4 and 7.5. Based on the ‘Managing Patient Records’ scenario mentioned above, we illustrate security and privacy requirements for each activity. For instance, the lab must perform a medical examination before sending the results to the medical doctor (security requirements: confidentiality and integrity). In addition, the medical results will be sent to the medical doctor to update the patient’s medical record; this action must be compatible with the GDPR accountability principle.

While the medical doctor is updating the patient’s medical record, this action should be anonymous. This, however, could lead to conflicts between accountability and anonymity. In order to process the updated results, they should be verified by the supervisor, therefore, this requirement involves accountability as a security requirement. Updating the patient medical record involves anonymity, to keep the patient record private, according to Privacy-by-Design principles. On the other hand, this update must be accountable to the supervisor to keep the system secure and accurate; the supervisor must be aware of the last update being made and by whom.

At this point, conflicts could occur between anonymity as a privacy requirement and accountability as a security requirement. That a task can require more than one requirement will lead to a potential conflict between requirements, especially based on privacy and security requirements. It can be difficult to fulfil both requirements simultaneously. For instance, accountability is the requirement that holds entities responsible for their actions

while anonymity allows entities to use resources or services without having to reveal their identity.

In Figure 7.2, we provide an overview of the Privacy-by-Design view of ‘Managing Patient Records’. In this view, we allocate security and privacy requirements for each goal. As discussed above, we have already identified a conflict between accountability related to the supervisor and anonymity related to the medical doctor. In this phase, we only highlight the conflict issue.

As seen in Figure 7.2, the employee performs the medical examination and obtains the patient’s results. Here confidentiality and integrity are necessary. The results are prepared by the employee and sent to the medical doctor, who then manages the patient’s medical record by obtaining new medical results and updating them. Here anonymity might be hindered, as in updating records, the medical doctor’s identity might have to be revealed. Furthermore, it is important to maintain accountability, meeting GDPR regulations, as this will hold the user of the system (the medical doctor) responsible for their actions – or lack thereof – in making changes to the patient’s records. The medical doctor will send these new medical results to the supervisor for validation. As the supervisor must validate any new medical examinations, accountability i.e. holding entities responsible for their actions or lack thereof, is necessary. This, however, will create a conflict, if anonymity is to be achieved.

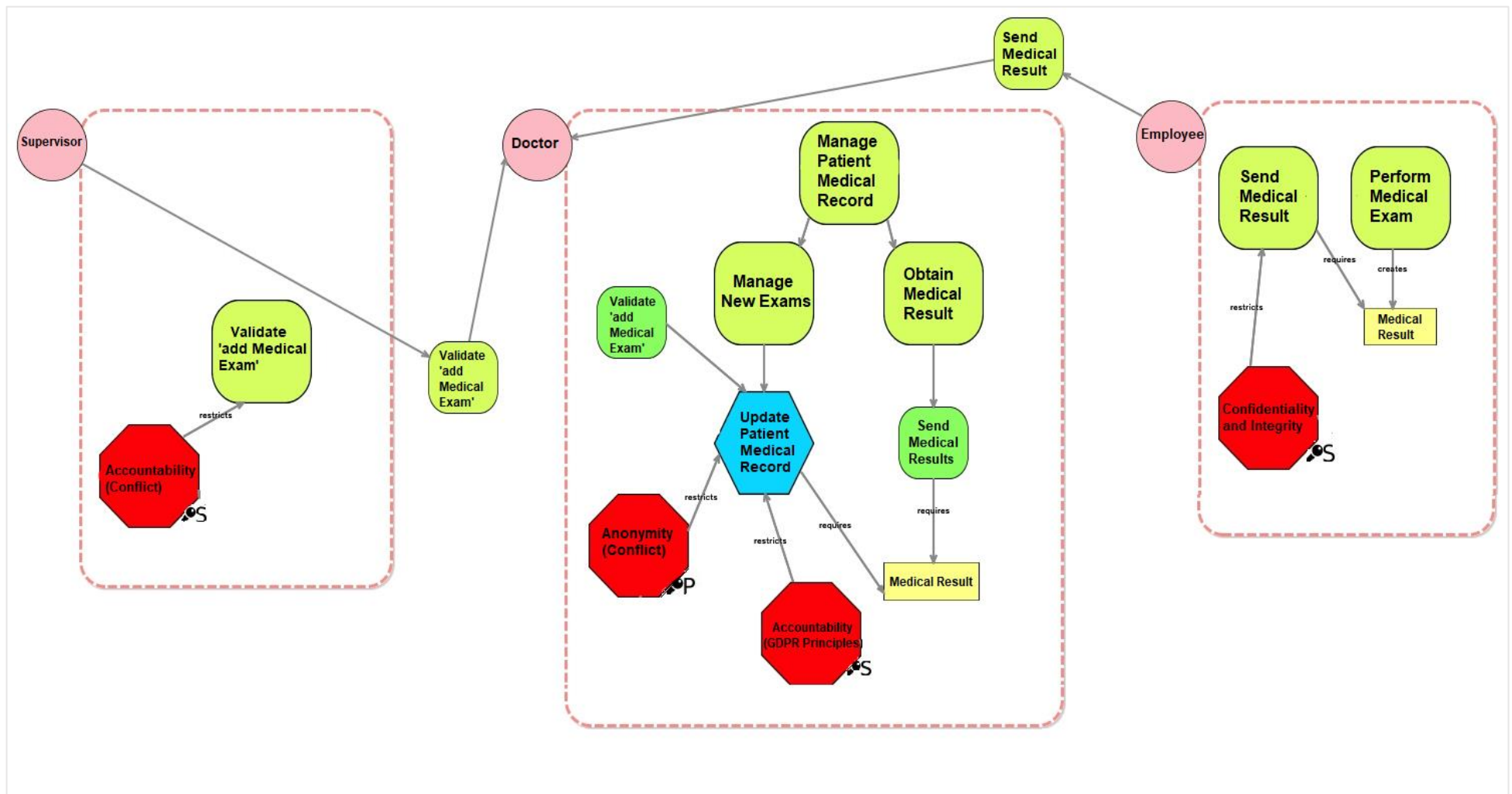
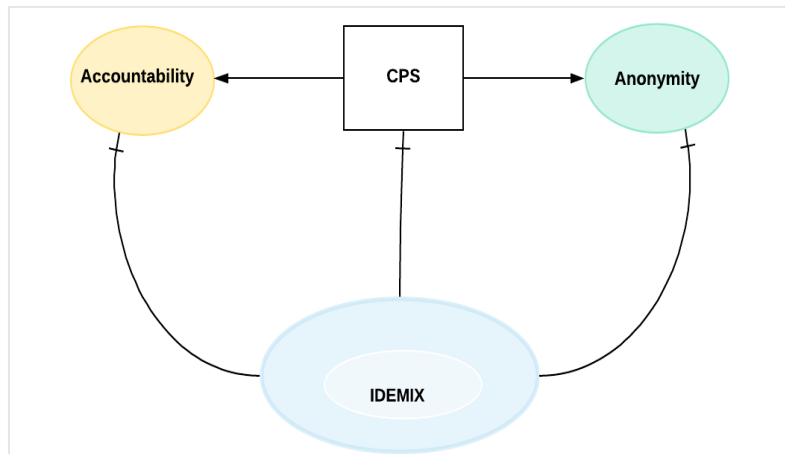


Figure 7.2 Privacy by Design View of Managing Patient Record

### 7.6-2 Phase 3: Conflict Resolution

In this phase, we present each conflict case in order to define the problem and identify the restrictions we need to follow, in order to mitigate the conflict. In the solution, we connect each case of conflict between requirements with a suitable supporting tool (see Figure 7.3).



**Figure 7.3 Accountability conflicts anonymity**

In order to resolve the conflict via supported tools, a relevant tool that could satisfy both requirements is recommended. By applying this scenario in SecTro, we add the tool to the Privacy Pattern library.

Firstly, we identify security and privacy measures. This step aims to identify security and privacy measures that support the satisfaction of relevant security and privacy constraints. Measures are identified with the support of security and privacy experts as well as cloud experts. Moreover, plans are identified for each actor to support the operationalisation of the identified measures. Next, we identify security and privacy mechanisms; this step aims to identify security and privacy mechanisms that support the implementation of the relevant plans. Similarly, measures and mechanisms can be identified with the support of security and privacy

experts as well as cloud experts and usage can be made of the security and privacy catalogue, if such a catalogue exists (Mouratidis *et al.*, 2013).

The Design Pattern Library (DPL) is an add-on for ‘SecTro2’, introduced in version 2.0 of the tool. The DPL allows the capture of modelling structures on the model and their saving for latter reuse. Such a mechanism enables various experts to capture their knowledge and transfer it to the developer of the system.

The main features of DPL are:

- the design patterns are modelled in ‘SecTro2’ using the same concepts available to the developer so no extra tools are required;
- saved design patterns are associated with a number of attributes which describe each individual pattern or group of patterns;
- saved design patterns can be selectively exported as a ‘well-formed’ XML (Extensible Markup Language) file which can be imported into DPL by other team members and developers. This XML file also can be used in other tools supporting such functionality; and
- the database, which holds saved design patterns, is a single file database, located in the user’s Documents folder. This file can be easily saved elsewhere for backup purposes and then restored back to the user’s Documents folder. The DPL functionality can be accessed from the main menu when the modelling component is active.

Depending on the type of saved design patterns, some will be greyed out as shown in Figure 7.4. During the creation of each of the design patterns, a view to which they belong to is assigned automatically.

Therefore, the design pattern selection window will recognise the currently active view and allow inserting design patterns which only belongs to this view. Furthermore, the:

- ‘Import’ button allows the user to import design patterns from XML file;
- ‘Export’ button allows the user to export design patterns to XML file for easy sharing;
- ‘Show pattern hierarchy’ checkbox enables/disables hierarchical design pattern view.

The hierarchy is extracted from the ‘Related patterns’ attribute.

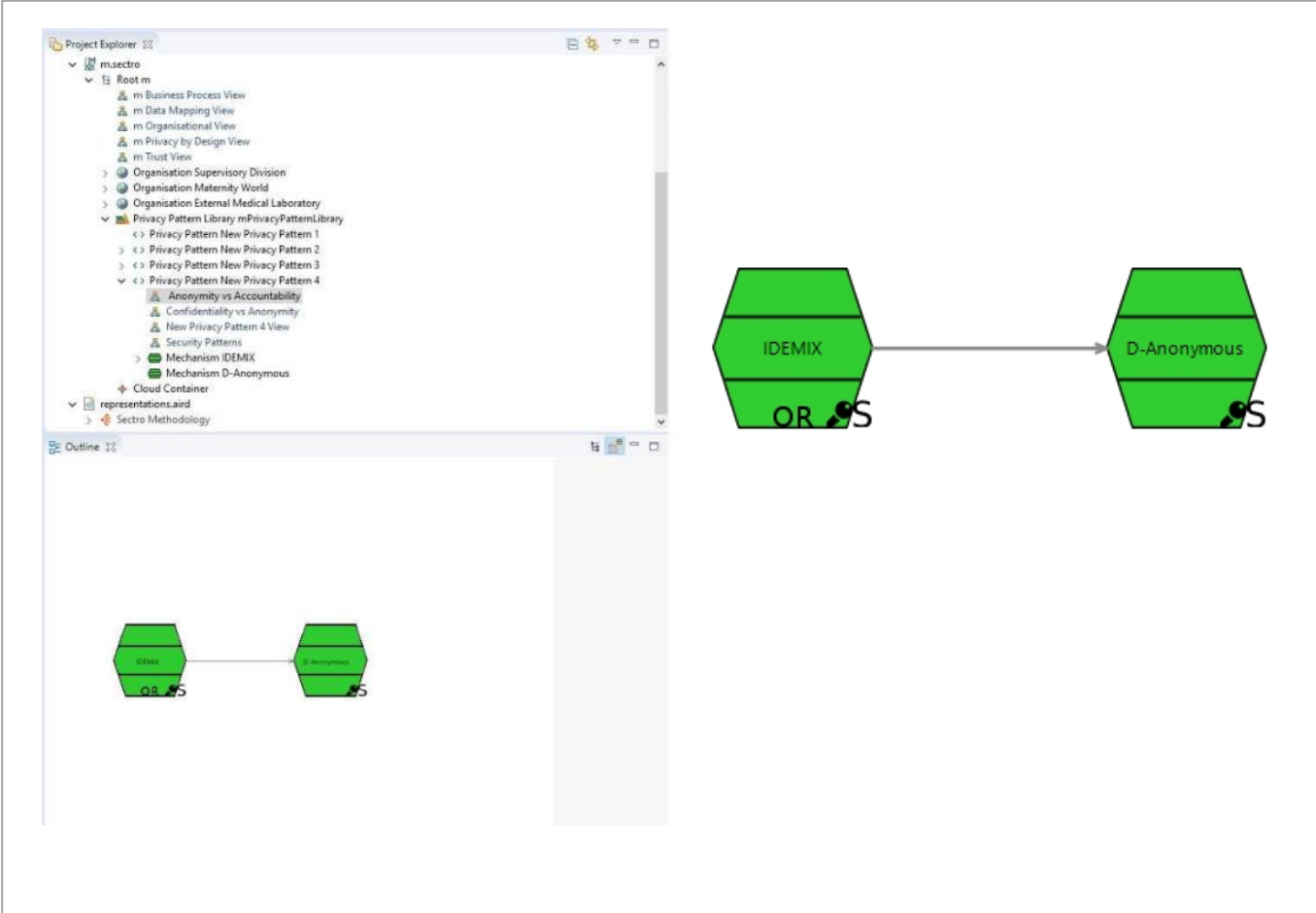
In this case, we identify two supporting tools, but determine that IDEMIX – which is a solution for minimising the release of personal information and can be based on one of many proposed techniques for anonymising the transport medium used between users and service providers – is the most appropriate (Camenisch and Van Herreweghen, 2002). Figure 7.4 shows how we add the supporting tool, and consequently Figure 7.5 shows the Privacy-by-Design view, after adding the new concepts to identify conflict between requirements and importing a suitable mechanism to satisfy those requirements.

In Figure 7.5, the employee performs the medical examination and obtains the results of the patient. Here confidentiality and integrity are necessary, for which the cryptography supporting tool is introduced. The results are prepared and sent to the medical doctor. The medical doctor manages the patient’s medical record by obtaining new medical results and updating patient medical records. Here anonymity can be hindered, as in updating records, the medical doctor’s identity might have to be revealed as accountability is also required. Furthermore, it is important to maintain accountability through meeting security policies, as this would hold the users of the system (medical doctor and supervisor) responsible for their actions taken in making changes to and validating patients’ records. Supporting tools IDEMIX and the Data Record Action mechanisms are introduced to mitigate these conflicts.

### **7.6-3 Discussion**

With regard to Figure 7.5 previously mentioned, the need to fulfil the anonymity requirement for the ‘Update Patient Medical Record’ process is fulfilled via the IDEMIX solution mechanism. In addition, the accountability constraint is related to the validate aspect of the ‘Add Medical Exam’ process, i.e. a supervisor needs to validate the change. However, for this, the supervisor needs to know which medical doctor made the change; thus, there is a conflict between accountability and anonymity, because the supervisor cannot know, due to the anonymity requirement, who the medical doctor is, so accountability cannot be fulfilled. We solve this by introducing the IDEMIX mechanism, which will be used by the supervisor, so that accountability can be fulfilled. IDEMIX is a solution for minimising the release of personal information and can be based on one of many proposed techniques for anonymising the transport medium used between users and service providers. IDEMIX is an optimising cryptographic compiler that achieves an unprecedented level of assurance, without sacrificing practicality for a comprehensive class of cryptographic protocols. This protocol satisfies the conditions for anonymous authenticated and accountable transactions between users and the service providers.

The employee is expected to fulfil confidentiality and integrity while sending medical results; we fulfil this with the cryptographic mechanism. This is related to fulfilling GDPR principles, and an example is accountability, where it is necessary to record which medical doctor made the change, and we fulfil this via the Record Data Action mechanism.



**Figure 7.4 Adding the Supporting Tool in Privacy Pattern Library**



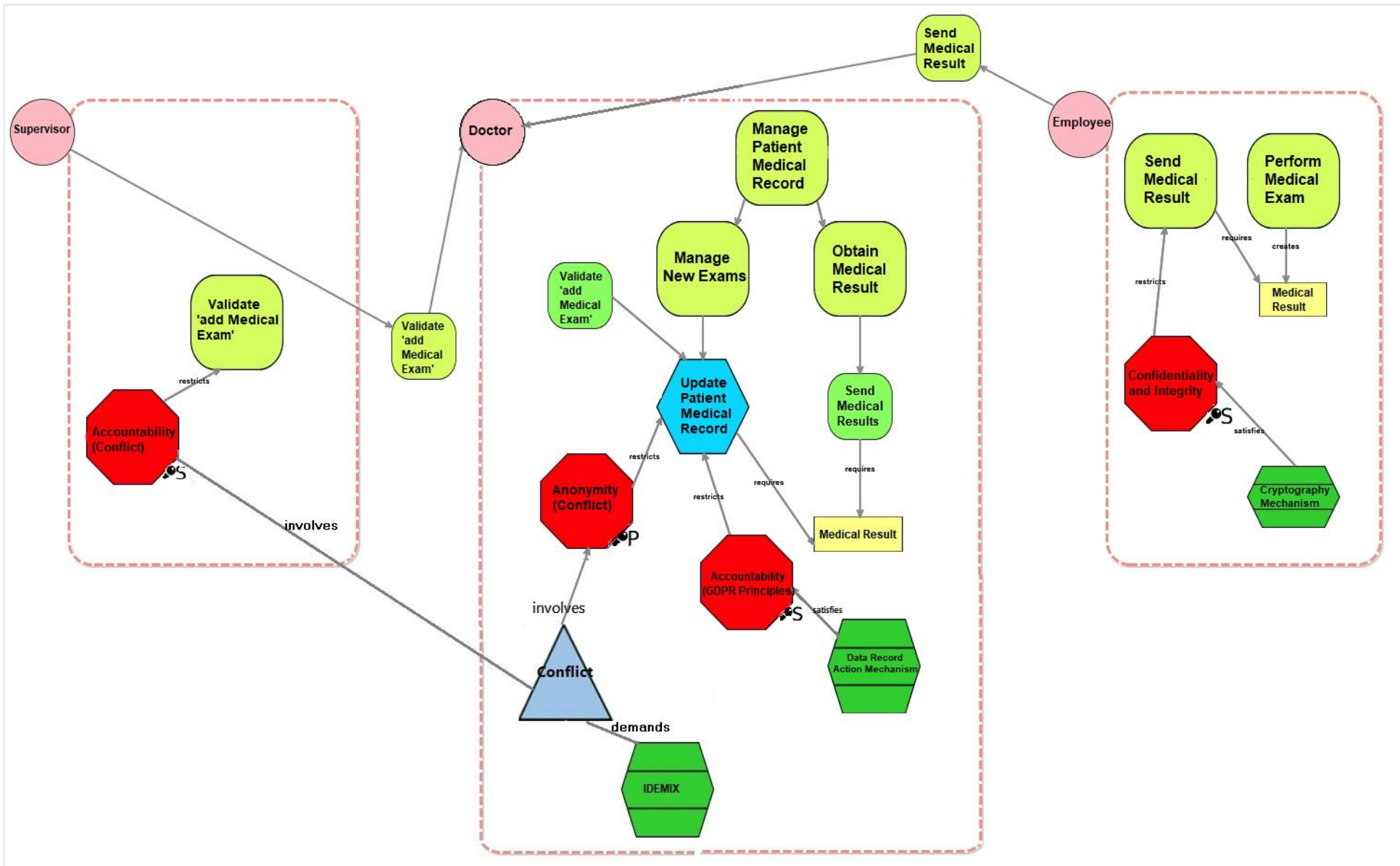


Figure 7.5 Integrating Conflict Resolution in Privacy-by-Design view

## **7.7 Benefits of applying the ConfIS framework within DEFEND boundaries**

The framework has helped to support DEFEND by identifying and resolving conflicts between security and privacy requirements. It develops a 3-Phase framework to identify, analyse and resolve conflicts between security and privacy requirements using supporting tools. The benefits of using ConfIS include a framework that defines and separates security and privacy. This will enable software engineers to analyse each one of these dimensions in further detail and understand the relationship between them. Furthermore, the framework enables software engineers to understand how security requirements and privacy requirements can co-exist within a system design. Therefore, any issues that need addressing (in terms of potential conflicts) can be identified at an early stage of the development process. Additionally, the framework allows an automated detection of security and privacy requirements conflicts assisting the analyst in decision-making. The framework identifies, characterises and defines similar resolution strategies that consider security and privacy requirements within one approach. This is important to overcome the limitations and issues discussed above and to provide novel methods to resolve conflicts between security and privacy requirements. This enables software engineers to resolve such conflicts and therefore the possibly of reducing potential development costs.

Furthermore, the ConfIS framework was tested on the DEFEND platform, as the platform is developed and endorsed by the EU to enable organizations to assess and comply to the EUs GDPR. The framework provides a structure to organisations, where they can determine security and privacy requirements of the case scenarios, and if there are any pending conflicts, so they can be detected and mitigated early on.

## 7.8 Chapter Summary

In this chapter we apply the ConfIS framework proposed in the previous chapter. Firstly, the DEFEND project (Piras *et al.*, 2019), with regards to privacy and security, its themes and services are elaborated upon in Sections 7.2 to 7.4, showing the importance of the platform which empowers organisations across various sectors, to assess their compliance status with GDPR regulation at the European Union (EU) law on data protection and privacy. Next, in Section 7.5, we apply the ConfIS framework to an E-Health scenario, modelling the example using a SecTro tool.

We link the scenarios with their associated requirements, identifying privacy and security requirements and the relevant supporting tools to mitigate conflicts within individual requirements, and between privacy and security requirements. Thereafter, we applied the framework to some examples from the E-health scenario, in Section 7.6. The examples identify the requirements, the potential conflicts arising, organisational and privacy by design views of the analysis, all embedded within the three-part phase of the proposed ConfIS framework. Lastly, conflict resolution patterns are mapped out in Phase 3, and discussions presented on all examples, presenting a step-by-step comprehensive explanation of the framework. Lastly, Section 7.7 presents the benefits of using the proposed ConfIS framework within the boundaries of DEFEND.

In the next chapter, the framework is applied to a real case study, having collaborated with the DEFEND project to integrate supporting tools that align with GDPR. The chapter will describe it in further depth, observing the outcomes of applying the framework and how it helps in reducing conflicts.

# CHAPTER 8

## EVALUATION & ANALYSIS

### 8.1 Introduction

Following Miles and Huberman's (1994) recommendation regarding analysis and the drawing of conclusions, this chapter firstly discusses the ethics involved in thematic analysis, review and data management of participants' information, and documenting consent (Section 8.2). Next, preliminary evaluation of the framework and integrating resolving conflicts is investigated in Section 8.3. Three participants who are software engineering experts are brought in at this stage to view the evaluation process. We then update the evaluation by taking into account the pros and cons of each scenario. Next, Section 8.4 covers the actual evaluation with the participants by using the focus group method and discusses how to apply the framework phases. By the end of the session, they fill out an evaluation questionnaire which covers all phases of the framework. We follow up through interviews, with participants receiving an in-depth evaluation feedback. To undertake the evaluation, we prepare content material as a toolkit to support the participants in understanding the framework. Its strategy and results are presented in this section. Next, in Section 8.5, we highlight other analysis methods, and their shortcomings, which thematic analysis (TA) seeks to fulfil hence the chosen evaluation method. Section 8.6 presents an in-depth analysis of TA, and the required steps. Thematic analysis and its application are used in answering the research questions. Thereafter, applying thematic analysis to evaluate the participants' responses in the case study using ConfIS framework, are examined in Section 8.7. As a result of this, the evaluation chapter focuses primarily on the ConfIS framework, as it is already confirmed, that it is built within the premise of the DEFEND project, as highlighted in Chapter 7. Lastly, a summary of the chapter is presented in Section 8.8.

## 8.2 Ethics in Evaluation

In focus groups, the collaboration and interaction of participants may result in the production of data that no other method can produce (Frantzana, 2019).

Following the same pattern with the interview questions, as a basis of the focus group discussion topics, we used hypotheses based on our previous results and results from other similar studies.

Any empirical research activity involving human subjects must take ethical aspects into consideration. Some aspects are regulated by national laws, while others are not regulated at all. Andrews and Pradhan (2001) identified ethical issues in software engineering and found existing policies to be insufficient. Hall and Flynn (2001) surveyed ethical practice and awareness in the UK, and found alarming unawareness, and nothing indicates this country as an exception. Vinson and Singer (2008) initiated a discussion on ethical issues, continued to discuss cases of ethical issues, and provided practical guidelines for the conduct of empirical studies. They identified four key principles:

- Subjects must give informed consent to their participation, implying that they should have access to all relevant information about a study before making a decision about whether to participate. Their decision must be explicit and free, also with respect to implicit dependencies on managers, professors etc.
- The study should have scientific value in order to motivate subjects to expose themselves to the risks of the empirical study, even if these are minimal.
- Researchers must take all possible measures to maintain confidentiality of data and sensitive information, even when this is in conflict with the interests of the publication.

- Weighing risks, harms and benefits, the beneficence must outweigh, not only for the individual subjects, but also for groups of subjects and organisations.

These principles are turned into more practical guidelines below, related to planning, conduct and reporting of an experimental study. We also refer to Sieber (1993) for a checklist of risks for subjects to be addressed in experimentation.

### **8.2.1 Ethics Review and Data Management Plan**

According to the policy for ethical research of the University of Brighton, parts of the research methods and data of a research study are subject to ethical review because of the involvement of human participants. The research involves accessing participants via an online environment or internet setting – experts in software engineering; some from the University of Brighton and collaborative researchers outside the university. This involves a focus group setting by using Microsoft Teams to explain and introduce the framework to them. By the end of the session, the researcher will ask the participants to fill out a questionnaire, followed by individual recorded interviews also via Microsoft Teams. This will be scripted. The participants will be informed at the beginning of the focus group, questionnaire and interviews that all of the data related to their participation is confidential. Only the researcher will be allowed to use or read the answers that are presented by the participant.

Ethical review self-assessment forms and a data management plan were submitted to the Ethics and Integrity Officer of the University. The ethical review forms included details of the project and self-assessment questions. The data management plan includes information about the purpose of the project, the type of data, data storage and preservation, confidentiality and data sharing. All the documents are reviewed and approved by the Research Ethics Board of the University (Frantzana, 2019).

### **8.2.2 Consent Documentation**

Following the regulations for ethical research, consent documentation is produced and provided to the participants of the interviews and of the focus group discussions conducted for this study. This documentation included an information sheet to fully inform the participants of the terms and conditions of their participation and a consent form, which was signed by the participants prior to data collection. The information sheet covered the purpose and details of the research, the procedure of participation, benefits and risks, terms for withdrawal, strategies for ethical use of the data and contact details for participants wishing to raise concerns and questions.

The consent documentation was read before the beginning of each interview, it was verbally agreed by the participants who were interviewed either in-person or through a video call, and it was audio recorded together with the interview of each participant. The participants who chose to be interviewed via email received the consent in written form, which they had to sign and return before they received the interview questions. The participants of the focus group discussions were provided with consent documentation in written form, which was read and signed by them before the beginning of the discussion. All versions of consent documentation by the University of Brighton Ethical office are available in Appendix B.

### **8.3 Preliminary Evaluation**

We ran a pilot evaluation with three participants, by preparing a presentation to describe the framework phases followed with a scenario from the DEFEND project to show the participant how to apply the framework in a real case study. All participants receive a toolkit that includes framework phases – mapping between requirements, models of conflicts and supporting tools

to resolve conflicts, – and describe the inputs and outputs of each phase. In addition, DEFEND scenario and screenshots of integrating conflict concept and supporting tool in SecTro are also utilised.

By using the focus group method, the moderator emphasises the motivation of this research, followed with research questions, and explains the framework, phase by phase. Thereafter, we apply the framework to one scenario from the DEFEND project. The moderator then gives the participant a chance to discuss the phases and how it is applied to a scenario, therefore the participant applies the framework phases to a second and third example; this is to ensure that they have a full understanding of the framework phases. The final stage is to complete a questionnaire that has been included in the toolkit, which covers an evaluation of all framework phases.

I received feedback from the participants. On the positive side, the length of the presentation was compatible with expectations, while simplifying the framework phases and instructions on how to apply it to a scenario were well received. Furthermore, the analysis of each scenario to identify conflicts was made clear and was therefore well understood. Briefly describing the DEFEND project helped the participants to better understand the project and grasp the research. Moreover, the participants reflected that the researcher's presentation skills were excellent, she had a clear voice, willingness to explain each slide and attention to the concern/questions of the participants, ensuring that they understood each part of the research. Additionally, the participants felt that the results were clearly presented, and their combination with the handout was particularly useful. Overall, the participants agreed that the research field is quite interesting.



## **Participants' Recommendations for Improvement**

Participants' suggestions for improving the framework application include:

- that the motivation of the research was not hugely convincing;
- in the toolkit we describe inputs and outputs for each phase, but participants suggested also adding it to the presentation;
- we listed specific types of requirements on which this research focused – participants recommended justifying this in the presentation so that the rest of the participants had a better understanding of the research;
- in the presentation, the resolution of some pictures was unclear, and in the presentation, acronym of concepts was used, but this needed to be made clear and understandable.

All this feedback was taken into consideration and led to revisions of the pilot evaluation before the actual evaluation begins.

### **8.4 Actual Evaluation**

In this section, we describe the preliminary evaluation we carried out. Here we report the evaluation strategy and results. The framework supports the investigation of this kind of analysis based on the importance of usable systems and promotes the process of human centred design as a way to achieve them. The 'Human Oriented' method is useful to design evaluation in a human centred way, to obtain feedback from experts of security and privacy engineering. We sampled fifteen participants, who are researchers of privacy and security engineering. They work within different universities from various countries including the United Kingdom, Italy, Greece, Germany, Saudi Arabia and China; this gives scope for a variety of perspectives (heterogenous). Each are presented with an evaluation form for relevant feedback, as depicted in Appendix C.

### 8.4.1 Evaluation Strategy

To achieve a comprehensive evaluation, we use qualitative and quantitative analyses. For the qualitative aspect, we designed a focus group session, with participants who are experts and researchers. Before we undertook the evaluation, we constructed a pilot focus group evaluation with three participant groups – PhD student, PhD doctor and Research Fellow. This revealed to us the possibilities of improving the focus group evaluation according to the participants' feedback. Moving forward, we could perform the full-scale focus group evaluation.

The rationale of the problem is to allow the participants to interact with a task in order to find out how the researcher can identify conflicts between requirements. We describe the ConfIS framework with an example provided (as discussed in this paper) and provide the participants with a handout containing a description of the focus group sessions, as well as the input and outputs for each phase of the framework. To be more specific, Phase 1 contains the list of security and privacy requirements, as well as Mapping Security and Privacy Requirements. Phase 2 provides a supporting tool, represented by tables and patterns. Finally, Phase 3 offers a conflict resolution table and screenshot of how we add the conflict concept and detect conflicts with supporting tools to solve this conflict. After the participants have grasped the full idea and learned how to use the framework, we asked them to apply ConfIS to the same task with which we started the presentation. This method gives us a comparison between using the framework or without using ConfIS. By the end of the session, participants are required to complete a survey, evaluating the framework phase by phase. By completing the survey, we will have an answer to RQ2: *How to design a framework that can support the analyst to resolve conflicts?*

This evaluation strategy will cover the qualitative evaluation. By holding the focus group and during discussions in this session, we can observe how the participants understand the framework. However, answering the survey employs a quantitative evaluation.

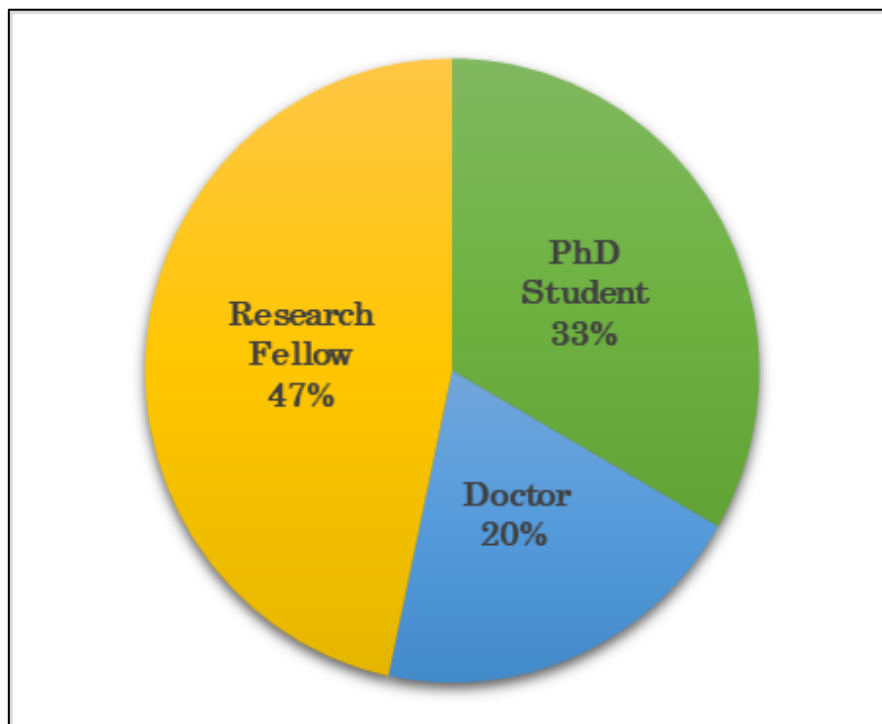
#### **8.4.2 Evaluation Results**

The survey consists of fifteen participants, of which 100% are respondents. Encouraging responses of the design include showing huge effort, with a well and confident presentation, remarkably interesting field and helpful work, and utilising real cases within EU projects. Its clarity in understanding the research objective was deemed a supportive method which could be used in an iterative way, and for each phase there is good support for the analyst. Additionally, it brings about a revelation of many more alternatives that can arise for the designer. The tables are a valuable form of presentation, but models could be a better way to visualise potential analysis of elements and solutions, speeding up the process. The evaluation was in general a positive experience, and the evaluator clearly presented the framework and its main objectives.

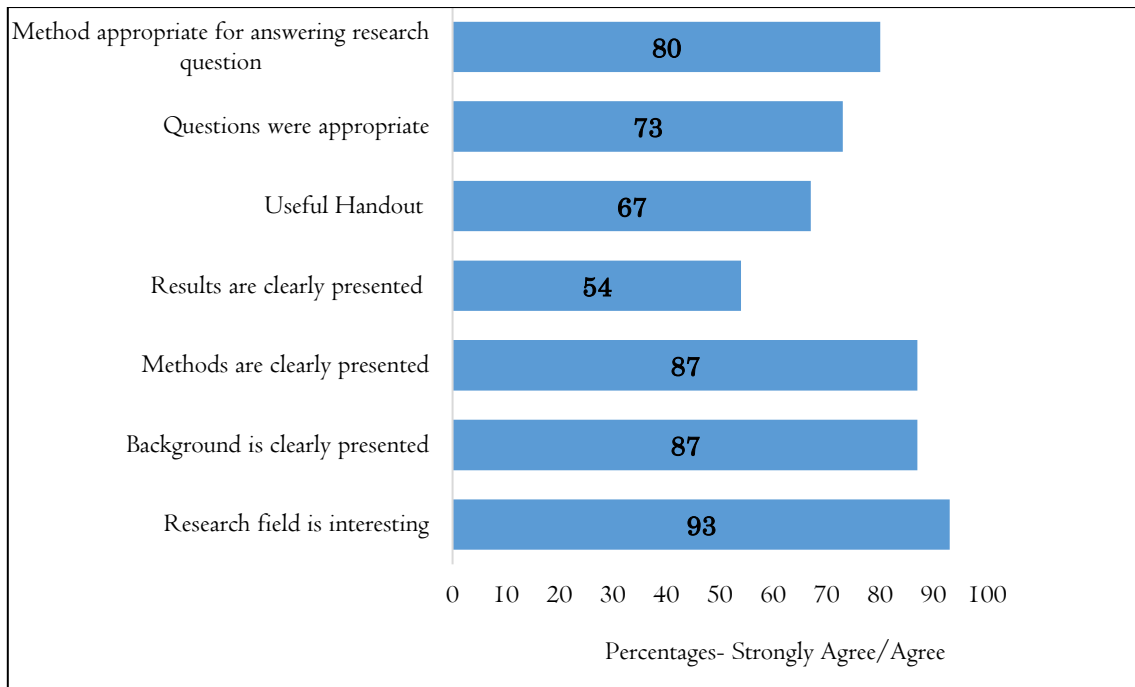
Furthermore, suggested areas for improvement include considering additional features/phases such as prioritisation and conflicts. The material and tools used to resolve conflicts can be more informative especially for those without more knowledge of the field, for which the use of more examples would be useful. Another improvement suggested was to specify the basis of any choice of solution; when the participant identifies conflicts and then chooses a possible solution, specifying how to choose one if there is more than one option. Moreover, creating a more structured evaluation that guides the subjects in their evaluation should be noted. Participants were a little unsure of the utility (or the ordering) of the conflict identification phase. The identification of the enforcement technologies that ‘resolve’ the identified conflicts

eliminates the conflict and therefore some participants did not see the reason for identifying them, if there were no more conflicts to search.

A summary analysis of the evaluation survey reveals that the majority of respondents were research fellows (47%), followed by PhD students (33%) and doctor (20%) (see Figure 8.1). All participants found the research design questions were appropriate, useful, well presented (87%) and the research field quite interesting (93%) in gaining their feedback. On the other hand, just 54% agreed that the results were clearly presented; this leaves room for improvement (see Figure 8.2).



**Figure 8.1 Survey Respondents**

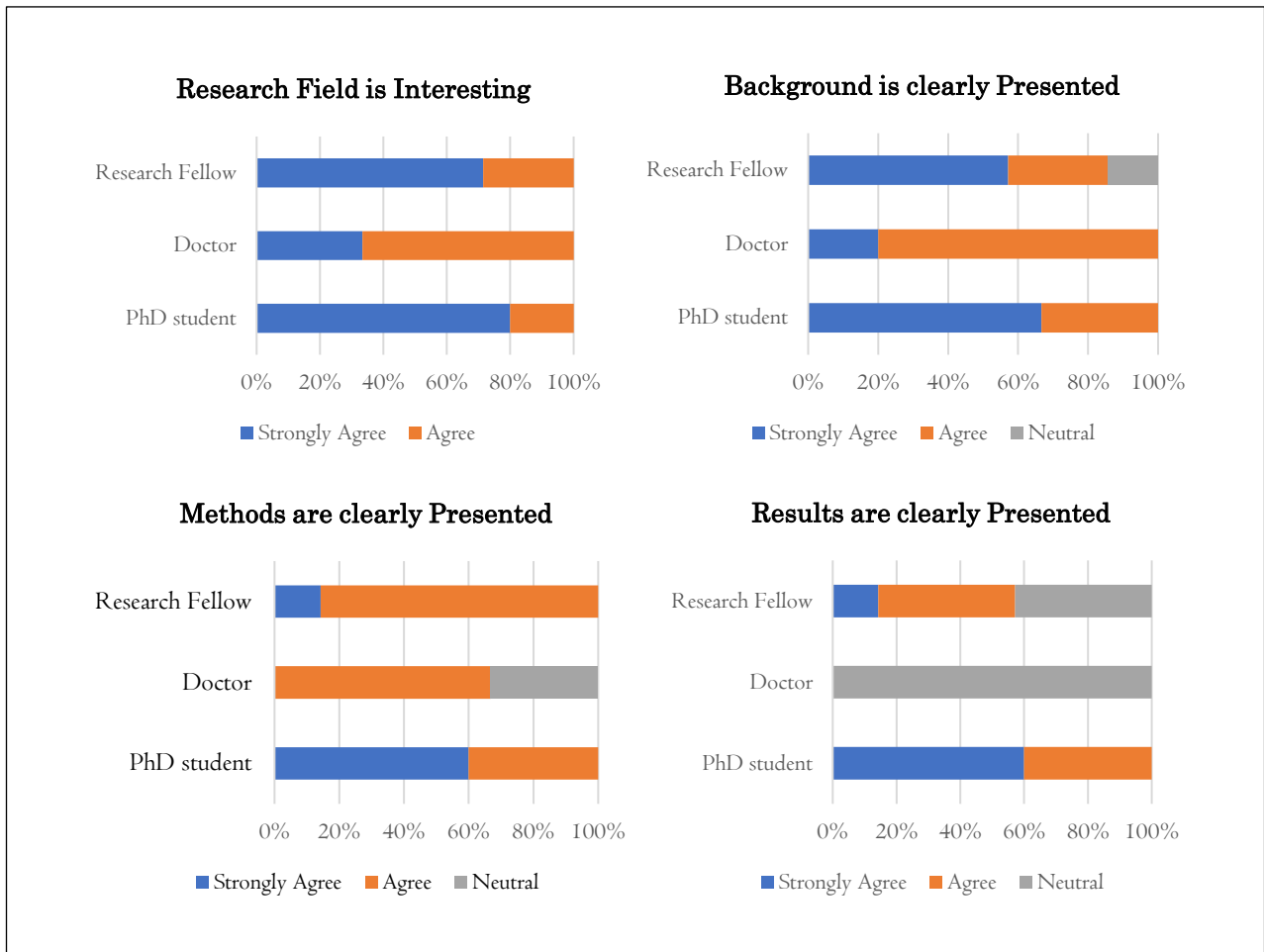


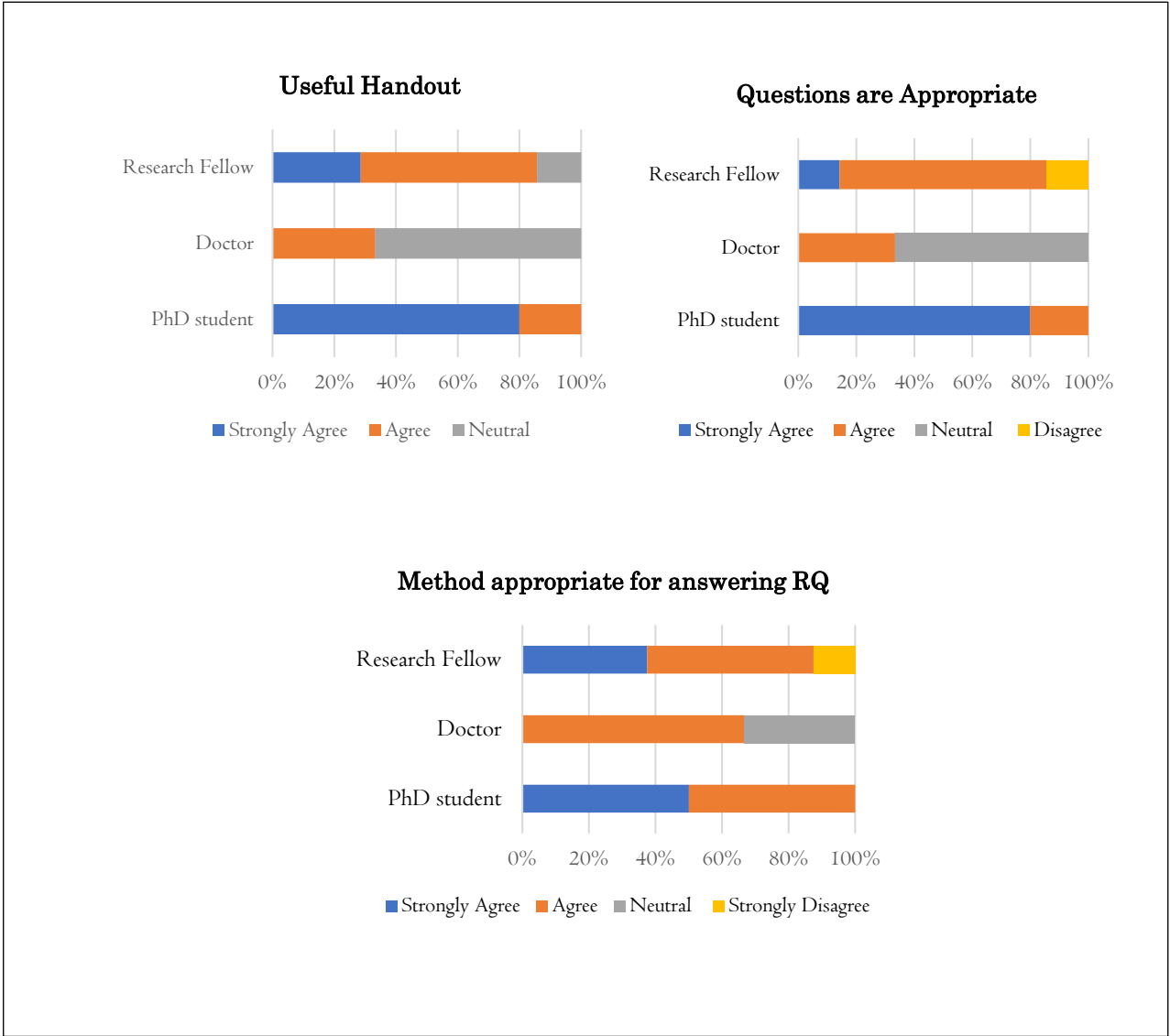
**Figure 8.2 Research Design Questions**

Further analysis into the survey reveals responses per profession. In this research, the student is currently pursuing a PhD degree, PhD doctor has already accomplished his degree, while the position of a research fellow normally requires possession of a doctoral degree and is in an academic research position at a university. It is therefore safe to say that the responses of each group are highly valuable, with that of the research fellow being more significant than that of a PhD doctor, with PhD student last due to the amount of expertise attained and level of qualification (see Figure 8.3).

More than 80% of the research fellows who participated highly agreed with the research design saying that the research field is interesting, background and methods are clearly presented and appropriate for answering the research questions, the handout is useful and questions are appropriate. Furthermore, 100% of the PhD doctors who participated highly agreed that the research field is interesting, and that the background is clearly presented. Moreover, over 60% (the majority) did agree to the method being clearly presented and appropriate for answering

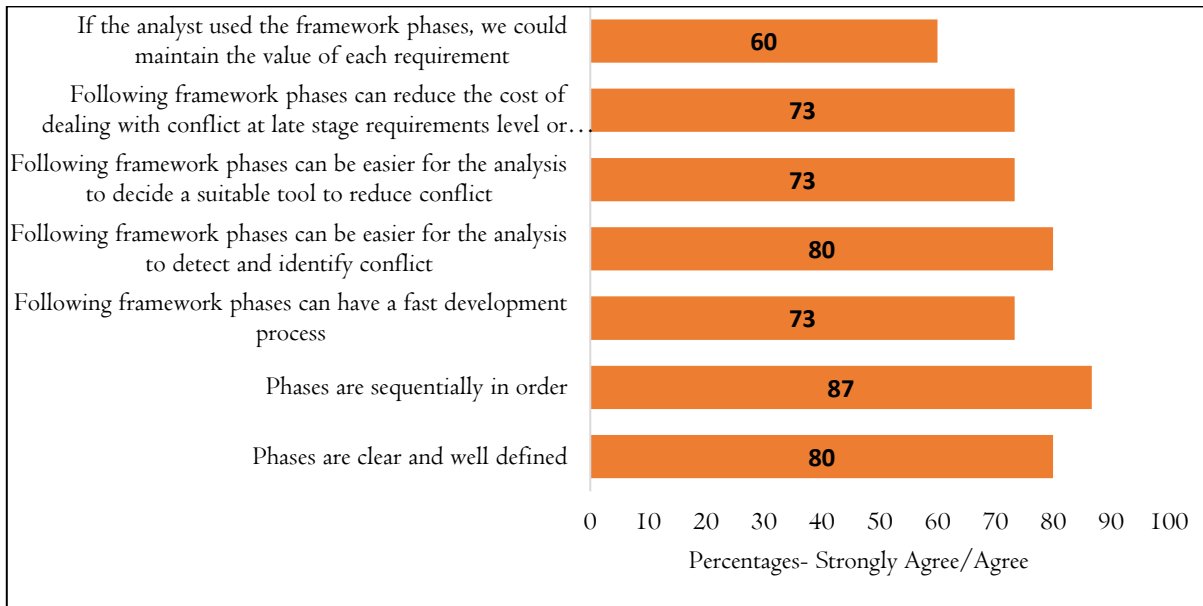
the research questions. A neutral response was provided, however, to whether the results were clearly presented, the usefulness of the handout and appropriateness of questions. Additionally, most PhD students, over 60%, agreed with the research design (see Figure 8.3). In instances of participants disagreeing with it to some degree, these results are specified in the graphs below.





**Figure 8.3 Research Design Per Respondent Group**

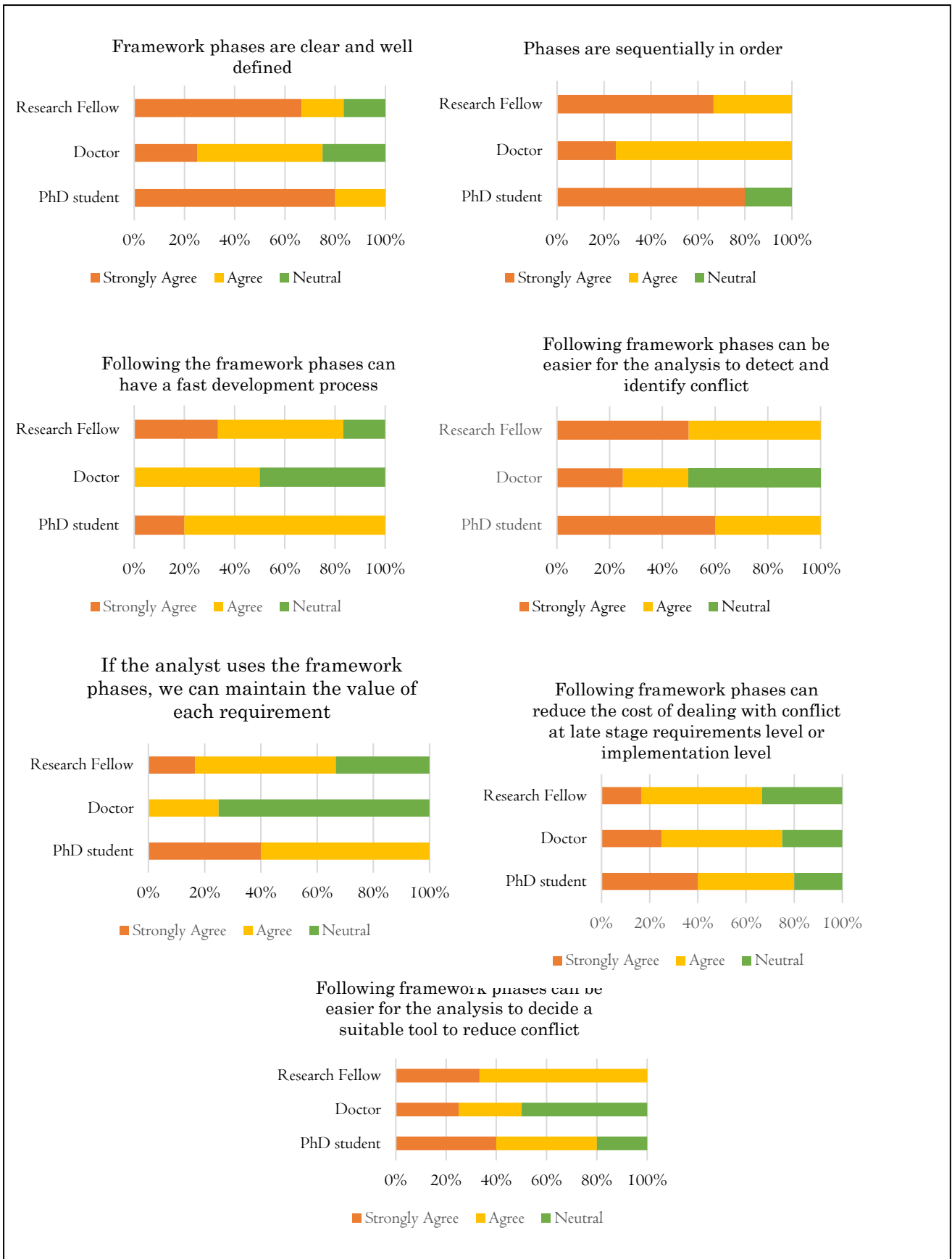
Additionally, the general framework was well received by the majority, proving to be sequentially in order (87%), clear and well defined (80%), easy to analyse (80%) and for making feasible decisions such as reducing cost, conflict and faster development processing (73%) (see Figure 8.4).



**Figure 8.4 General Framework**

The majority share, well over 70% of research fellows, agreed with the general framework. They approve of the statements that the relevant phases are clear, well defined, sequentially in order, can have a fast development process, are easy for identifying conflict, reducing it and its relevant costs, and maintaining the value of each requirement. The same can be said for PhD doctors, with the exception of 50% indicating a neutral response to the statement that framework phases have a fast development process, are easy for detecting/identifying and reducing conflict, and for maintaining the value of each requirement. Additionally, more than 80% of PhD students agreed with the design of the general framework and its phases (see Figure 8.5).



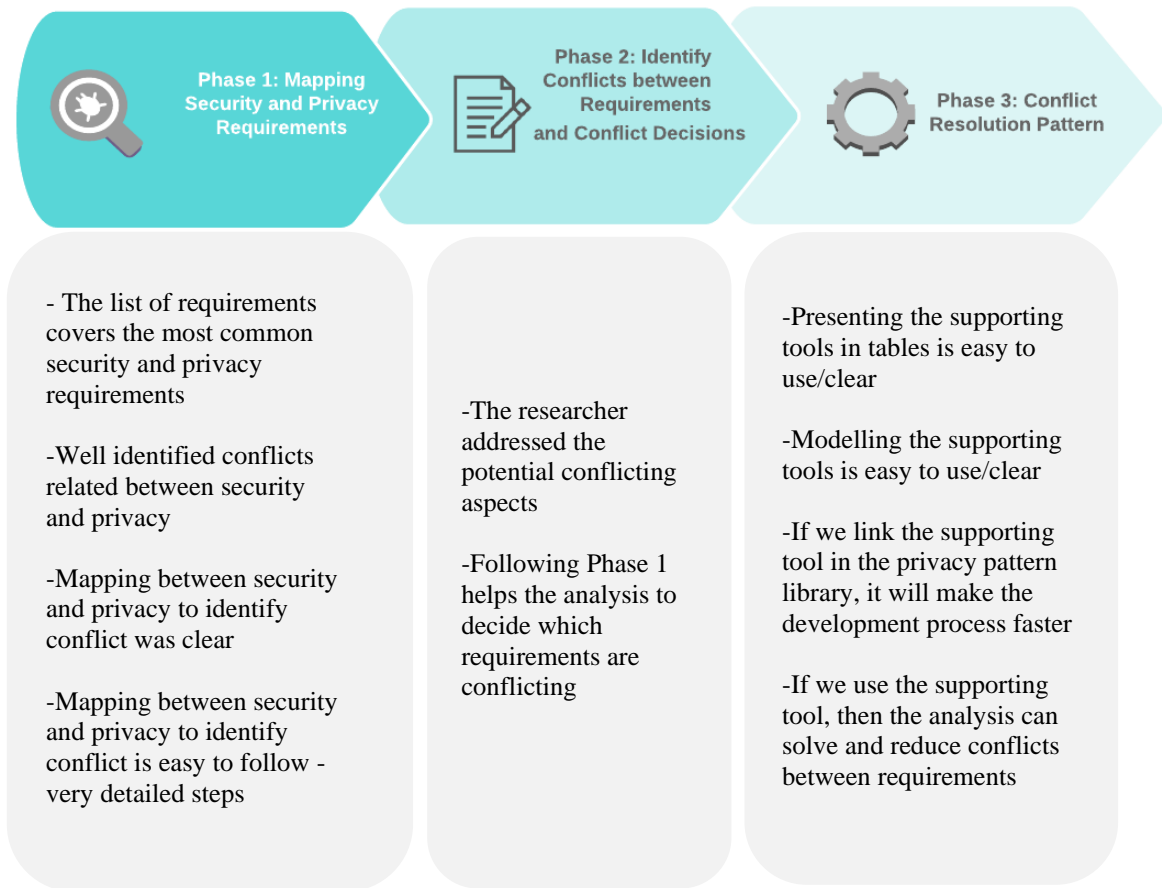


**Figure 8.5 General Framework Per Respondent Group**

Moreover, Figure 8.6 shows the breakdown of Phases 1, 2, and 3, with the relevant survey questions for each phase.

Some of the steps are semi-automated, while others are manual steps, based on the analyst's point of view. First, the conflicts between requirements are identified, based on a matrix presented by a previous study (Alkubaisy, Cox & Mouratidis, 2019). Hence, we sort the requirements that could lead to a potential conflict. After identifying the requirements which are in conflict, the analyst must decide whether this kind of conflict would affect the system, based on the presented scenarios. Therefore, the first phase of the framework is performed manually by the software requirements analyst. Phase 2 identifies the potential conflicts between requirements that were detected in the previous phase. The final phase proposes conflict resolution patterns by matching the problem to a resolution pattern for each conflict that the analyst might face. These patterns act as a reference for the analyst to resolve conflicts between requirements. The final phase of our framework is automated by using SecTro tool (by importing a privacy pattern library) (see Figure 7.4) from Chapter 7, Section 7.6-3.

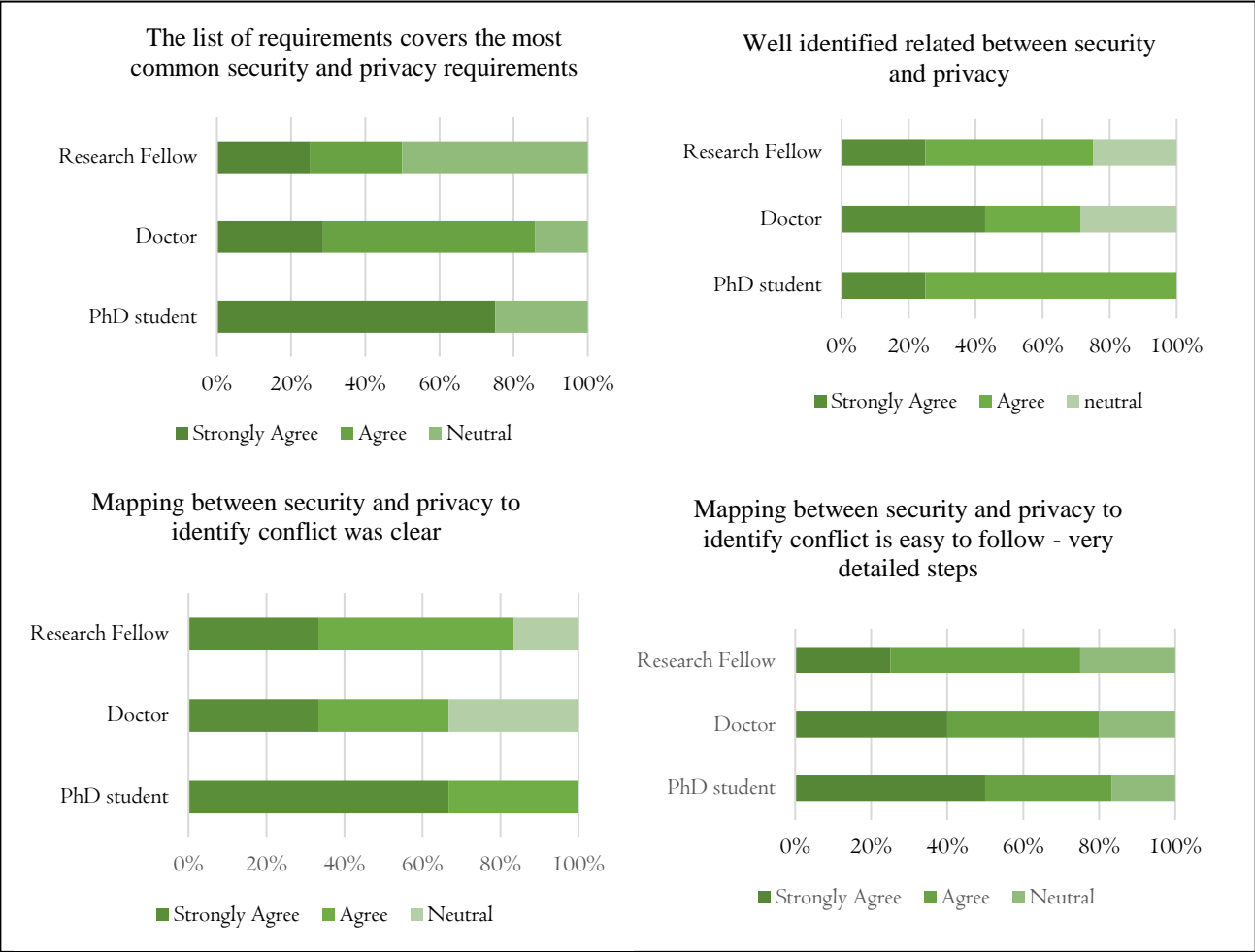
Phase 1, mapping security and privacy requirements, showed 70-87% of participants agreeing to the presentation of Phase 1 (see Table 8.1; Figure 8.7). Phase 2 was well received with the majority (80-86%) agreeing that the researcher adequately addressed conflicts between requirements and decisions. Additionally, feedback on Phase 3 showed varying responses (67-87%), yet the participants still agreed that there was an ease to understanding conflict resolutions patterns and its supporting tools (see Table 8.1).



**Figure 8.6 ConfIS Framework Phases and Survey Questions**

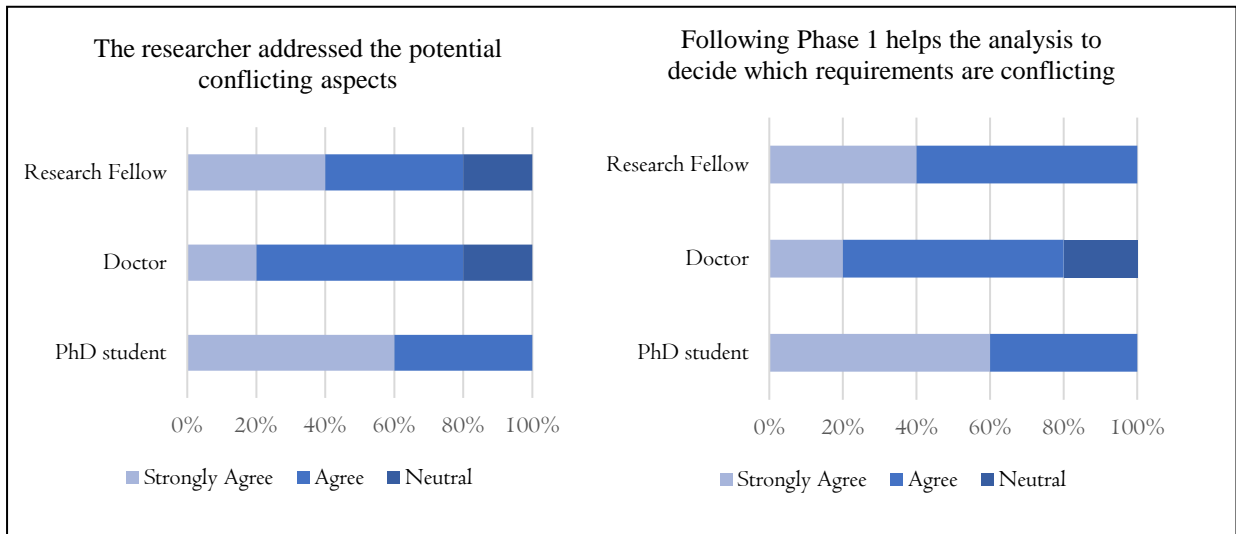
**Table 8.1 ConfIS Framework Phases and Survey Responses**

<i>Phase 1: Mapping Security and Privacy Requirements</i>	70-87% (strongly/agree)
<i>Phase 2: Identify Conflicts between Requirements and Conflict Decisions</i>	80-86% (strongly/agree)
<i>Phase 3: Conflict Resolution Patterns</i>	67-87% (strongly/agree)

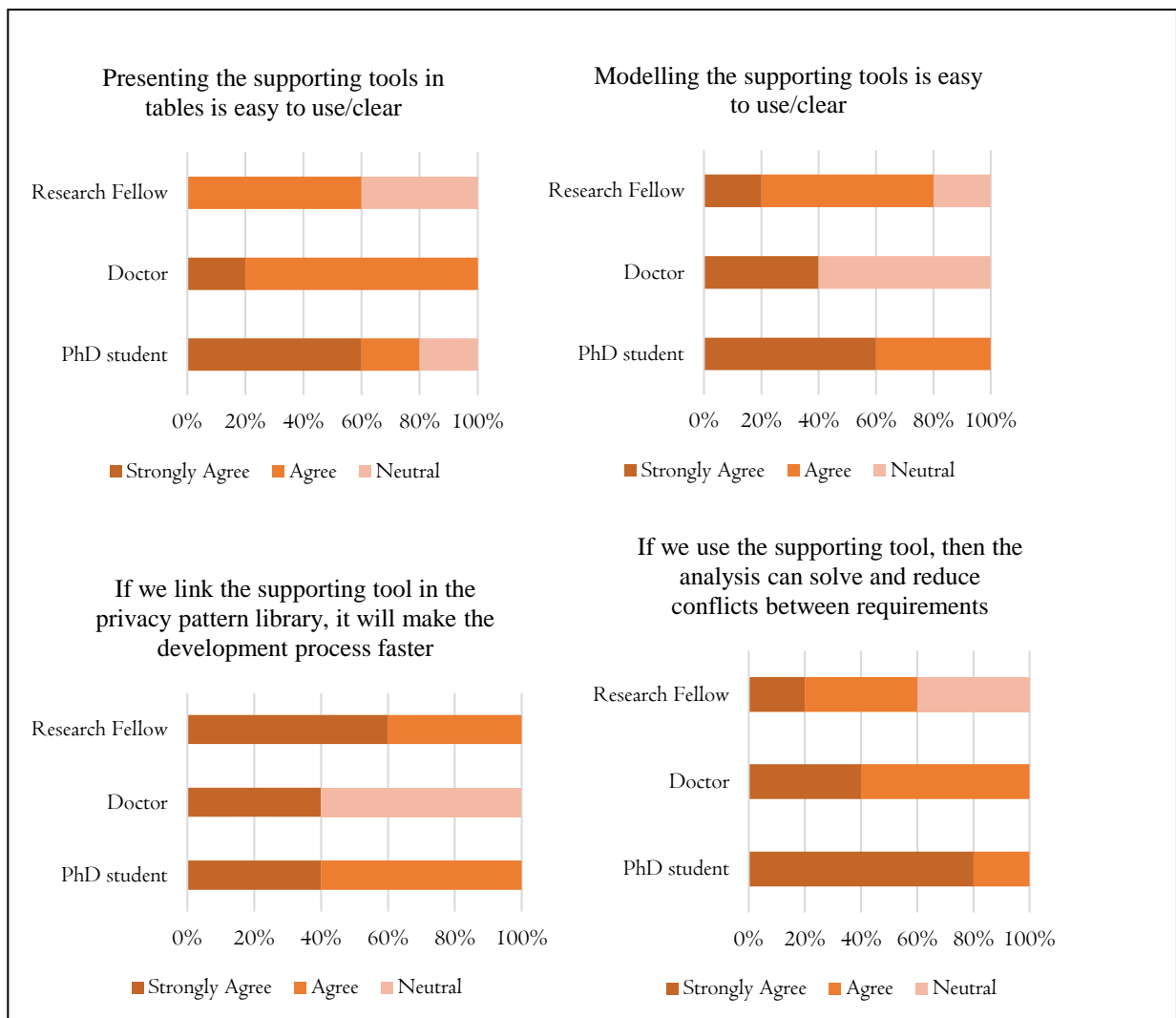


**Figure 8.7 ConfIS Framework Phase 1 Per Respondent Group**

Phase 2 further gained the interest of the participants, with over 80% of research fellows, academic doctors and PhD students widely agreeing that the researcher has addressed potential conflicting aspects, and that Phase 1 does help in the analysis with decision-making (see Figure 8.8). Furthermore, the majority of PhD students (80%) and research fellows (60%) did agree that the supporting tools were well presented, modelled, easy to use/clear, could speed up the development process and reduce conflicts. Doctors, however, were predominantly neutral (60%) in their decisions regarding its ease of use, and the speediness of the development process (see Figure 8.9).



**Figure 8.8 ConfIS Framework Phase 2 Per Respondent Group**



**Figure 8.9 ConfIS Framework Phase 3 Per Respondent Group**

## **Analysis of ConfIS Framework Phases' Focus Group Results and Survey Responses**

Ven & Delbecq (1972) found that a two-stage combination of focus group and the nominal group technique (NGT), coined as 'nominal focus group', was particularly effective as an evaluation method.

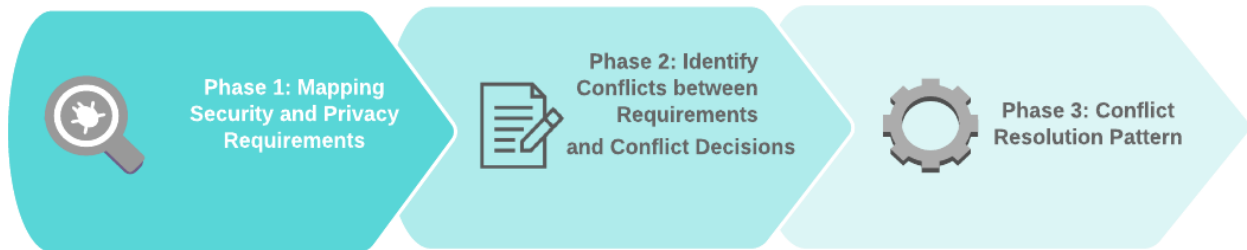
The nominal group process is a structured meeting which seeks to provide an orderly procedure for obtaining qualitative information from target groups who are most closely associated with a particular issue. It allows the meetings' participants to determine which issues require further, more in-depth inquiry and to draw attention to issues that may have been previously unidentified.

In the basic method, the numbers each solution receives are totalled, and the solution with the highest (i.e., most favoured) total ranking is selected as the final decision. There are variations on how this technique is used. For example, it can identify strengths versus areas in need of development, rather than being used as a decision-making voting alternative.

This evaluation method is used in this research to rank in order of importance the participants' responses to Phases 1 and 2. In order of importance for Phase 1, the top three security requirements are seen to be integrity, confidentiality and accountability, while anonymity, unobservability and pseudonymity are ranked top highest in privacy requirements. Participants' responses to identifying possible conflicts between requirements as depicted in Phase 2, show accountability and anonymity mostly chosen, followed by auditability and anonymity and accountability and undetectability. Anonymity accounts for a large portion of Phase 2 (see Figure 8.10).

Participants' responses to applying the proposed SecTro framework, and its supporting tools in Phase 3, show that the framework supports the identification of conflicts among requirements and suggested tools. Participants praise work well done and its contribution to

the academic community. The research opens possibilities for future work; more requirements could be added and a comparison between tools can be conducted, to determine the best tool. Furthermore, for larger models, it can be difficult to use to identify conflicts, but having the process automated would render it easier and quicker.



Using Nominal Ranking Evaluation Method

<p><b>Participants identify the relevant security and privacy requirements:</b></p> <p><u>Security</u>          Integrity          Confidentiality          Accountability          Availability          Authentication          Authorisation          Non-repudiation</p> <p><u>Privacy</u>          Anonymity          Unobservability          Pseudonymity          Unlinkability</p>	<p><b>Participant Responses:</b></p> <p>Accountability and Anonymity</p> <p>Auditability and Anonymity</p> <p>Accountability and Undetectability</p> <p>Anonymity and Confidentiality</p> <p>Anonymity and Integrity</p>	<p><b>Participant Recommendations on applying Proposed Framework SecTro (supporting tools);</b></p> <p>Framework supports in easily identifying conflicts among requirements</p> <p>Larger models: difficult to be used to find conflicts. Having it automated would be easier to resolve conflict</p> <p>Well done, work is very important, really impressed of progress. Has a lot of contribution to academic community          Issue could be resolving these conflicts - could be a lot of work</p> <p>What the results could show with the tools to use - interesting work</p> <p>Start with the requirements we currently have, and maybe add more later on - future research work - post doc stage</p> <p>Comparison between tools can be done later on. To determine the best tool.</p>
---	--	---

**Figure 8.10 ConfIS Framework and Focus Group Response using Ranking Evaluation Method**

## 8.5 Evaluation Methods

There are three key methods which help in evaluating design ideas: thematic analysis (TA) for the focus group evaluation, and evaluation to analyse the questionnaire. These methods can be implemented individually or in a sequence-based number of steps on the number of creative ideas and the type of the evaluation required. For the purposes of this research, thematic analysis will be used which will be thoroughly elaborated upon in Section 8.6. There are however other forms of qualitative focus group research analysis methods, namely content analysis, narrative analysis, and discourse analysis, which cannot be ignored in choosing a method.

Content analysis recognizes patterns in participant transcripts, by creating a set of guidelines for coding the text. This method is reasonably cheap, can be reliable as it follows a systematic process, and can be used widely in media research. The initial coding of texts is however key, and appropriate rationale for choice of codes must be justified, as it can impact the findings of the research. TA provides a broader and more in depth understanding of the research than content analysis, as the latter investigates more so on the frequency of occurrence of different categories. TA recognizes themes in the data and builds up the analysis from this.

Narrative analysis engages storytelling, where the social interactions of participants and interviewer are studied, rather than solely looking at the information collected reflective of answering the research questions. The story is the investigative spotlight, which is not what this research seeks to accomplish. Additionally, discourse analysis seeks to understand the participants more so than what they would have to say about the research at hand.

TA therefore fills the shortcomings of each method and justifies its choice of use for this research. In particular, the other methods are not constructed to create themes as a way of capturing what has been learnt from the data. This is explicitly what TA does.



## 8.6 Thematic Analysis

Thematic analysis is a widely used method in qualitative research. First named as an approach in the 1970s (Merton, 1975), it is used as a method for identifying, analysing and reporting patterns (themes) within qualitative data (Braun & Clarke, 2006). It is used commonly because of the wide variety of research questions and topics that can be addressed, and through this flexibility, it allows for rich, detailed and complex description of the data.

The method also often goes further than this, however, and interprets various aspects of the research topic (Boyatzis, 1998). Qualitative approaches are incredibly diverse, complex and nuanced (Holloway & Todres, 2003), while Braun, Clarke & Terry (2014) argue that thematic analysis should be seen as a foundational method for qualitative analysis.

Indeed, Holloway and Todres (2003, p. 347) identify “thematizing meanings” as one of a few shared generic skills across qualitative analysis. For this reason, Boyatzis (1998) characterises thematic analysis not as a specific method but as a tool to use across different methods. Similarly, Ryan and Bernard (2000) locate thematic coding as a process performed *within* ‘major’ analytic traditions (such as grounded theory), rather than a specific approach in its own right. Braun, Clarke and Terry (2014) argue that thematic analysis should be considered a method in its own right.

In applying thematic analysis to a research study, some of its many benefits are listed in Table 8.2 below:

**Table 8.2 Advantages of Thematic Analysis**

---

Flexibility.
Relatively easy and quick method to learn and do.
Accessible to researchers with little or no experience of qualitative research.
Results are generally accessible to educated general public.
Useful method for working within participatory research paradigm, with participants as collaborators.
Can usefully summarise key features of a large body of data, and/or offer a ‘thick description’ of the data set.
Can highlight similarities <i>and</i> differences across the data set.
Can generate unanticipated insights.
Allows for social as well as psychological interpretations of data.
Can be useful for producing qualitative analyses suited to informing policy development.

---

Braun, Clarke & Terry (2014)

While this is so, the method does not come without some disadvantages, which must be noted. Many of the disadvantages depend more on poorly conducted analyses or inappropriate research questions, than on the method itself. Furthermore, the flexibility of the method – which allows for a wide range of analytic options – means that the potential range of things that can be said about the data is broad. While this is an advantage, it can also be a disadvantage in that it makes developing specific guidelines for higher-phase analysis difficult and can be potentially paralysing to the researcher trying to decide which aspects of the data to focus on. Another issue to consider is that a thematic analysis has limited interpretative power beyond mere description if it is not used within an existing theoretical framework that anchors the analytic claims that are made (Braun, Clarke & Terry, 2014).

Thematic analysis has more or less six clearly defined steps required to ensure clarity and rigour in the process (Braun & Clarke, 2006; Nowell *et al.*, 2017). To apply this to the current research study, we need to follow these steps in order, therefore we can discuss our findings briefly (Braun & Clarke, 2006):

1. Familiarisation
2. Coding
3. Generating and reviewing themes
4. Synopsis

Firstly, when conducting data analysis, the researcher must make informed decisions regarding the coding, theming, decontextualising and recontextualising of the data (Sarks & Trinidad, 2007). The researcher must familiarise themselves with the data, and then generate initial codes. Coding can be done manually or with a software program such as NVivo.

The activity of coding involves identifying interesting features of the data systematically across the data set. Initially, codes are attached to units of data that could vary in size (i.e. phrase, sentence, paragraph) but usually codes encompass a complete thought. The code serves as a tag used to retrieve and categorise similar data so that the researcher can pull out and examine all of the data across the dataset associated with that code (Castleberry & Nolen, 2018). The action of coding requires the researcher to ask specific questions of the data such as what is happening in the text, who are the actors and what are their roles, when is it happening (preceding event, during event, reaction to event, etc.), where is it happening, what are the explicit and implicit reasons why it is happening, and how is it happening (process or strategy)

to name but a few (Bernard, Wutich & Ryan, 2017). A coding strategy can be established before coding begins (a priori) based on a careful review of previous research or theory.

After codes are generated, searching for broader level themes involves sorting the different codes into potential themes. To help with this, visuals such as mind maps can be used. Next, reviewing themes is important, as it involves refining. Here, themes can generate other themes, whereas existing themes may need to be broken down into smaller components. Revisions that are made at this stage are produced on a thematic map. After this, the researcher defines and names the relevant themes; this step captures the essence of what each theme is about and what aspect of the data each theme captures. Lastly, producing the report involves final analysis and write-up.

A theme may be initially generated inductively from the raw data or generated deductively from theory and prior research (Boyatzis, 1998). With an inductive approach, the themes identified are strongly linked to the data themselves and may bear little relation to the specific questions that were asked of the participants. Inductive analysis is a process of coding the data without trying to fit it into a pre-existing coding frame or the researcher's analytic preconceptions. In this sense, this form of thematic analysis is data-driven (Braun & Clarke, 2006). In contrast, deductive analysis is driven by the researchers' theoretical or analytic interest and may provide a more detailed analysis of some aspect of the data but tends to produce a less rich description of the overall data (Braun & Clarke, 2006). Researchers must distinguish whether they are conducting an inductive or deductive thematic analysis as this will inform how themes are theorised (Braun & Clarke, 2006).

In order to produce a richer description of the overall data, and for a more data driven analysis, the inductive approach is used in this research. Here, themes identified are strongly linked to

the data themselves, and data is coded without any pre-existing coding frame and/or analytic preconceptions.

Another justification for using thematic analysis is embedded in its benefits. It brings a highly flexible approach that can be modified for the needs of many studies, providing a rich and detailed, yet complex, account of data. As thematic analysis does not require the detailed theoretical and technological knowledge of other qualitative approaches, it offers a more accessible form of analysis, particularly for those early in their research career. Also, those who might be unfamiliar with qualitative methods may find that thematic analysis is easily grasped and can be relatively quick to learn, as there are few procedures (Braun & Clarke, 2006; King, 2004). Furthermore, it is a useful method for examining the perspectives of different research participants, highlighting similarities and differences, and generating unanticipated insights. Thematic analysis is also useful for summarising key features of a large data set, as it forces the researcher to take a well-structured approach to handling data (King, 2004).

Although there are several advantages to using this form of qualitative research method, we must not ignore its drawbacks. A simple thematic analysis is disadvantaged when compared to other methods, as it does not allow the researcher to make claims about language use (Braun & Clarke, 2006). Furthermore, while thematic analysis is flexible, this flexibility can lead to inconsistency and a lack of coherence when developing themes derived from the research data (Holloway & Todres, 2003).

## **8.6.1 Thematic Analysis Steps**

### **8.6.1-1 Familiarisation**

Firstly, the research problem is introduced to the focus group, so that they gain a general understanding of the project. Here fifteen participants are interviewed, and their responses are presented in Appendix B. Their response pertains to a task given to solve the pre-proposed framework. Next, the same task is reintroduced, where the proposed research framework is now applied, to ascertain the difference pre- and post-framework applications. Participants' comments are made regarding stakeholder expectations, security and privacy requirements, and conflicts identified, the design of the ConfIS framework discussed, and further suggestions and comments for improving the framework are gathered.

Furthermore, time is spent getting to know the data and a thorough overview of the participants, before analysing begins. This involves transcribing audio, reading through the text and taking initial notes, and generally looking through the data to get familiar with it.

### **8.6.1-2 Coding**

The next step is to code the data. Coding means highlighting sections of the text – usually phrases or sentences – and developing shorthand labels or ‘codes’ to describe their content. A coding strategy can be established before coding begins (a priori) based on a careful review of previous research or theory. Here, coding qualitative data is firstly done by separating responses into pre- and post-framework sections for easier analysis. The various colours represent various phrases in different colours corresponding to different codes. Each code describes the idea or feeling expressed in that part of the section.

At this stage, we want to be thorough: we go through the transcript of every interview and highlight everything that jumps out as relevant or potentially interesting. As well as

highlighting all the phrases and sentences that match these codes, we will keep adding new codes as we go through the text.

After we have been through the text, we collate together all the data into groups identified by a code pre- and post-framework. These codes allow us to gain a condensed overview of the main points and common meanings that recur throughout the data. See Tables 8.3 (as it presents the codes introduced pre/post framework as per TA steps) and 8.4 for a breakdown of the relevant codes chosen as a result of participant responses to evaluating the ConfIS framework.

**Table 8.3 Coding Participant Responses – Evaluating ConfIS Framework**

<i>Expectations</i>		<i>Expectations</i>	
<i>Pre-Framework</i>	GDPR	<i>Post-Framework</i>	Contribution
	Security Requirements		Design
	Privacy Requirements		Improvements
	Associated Conflicts		Short Term
	Standards		Longer Term
	Compromising		Task Scenario
	Progress		Focus Group
			Analysis
	Matrix		

**Table 8.4 Interview Extracts – Evaluating *ConfIS* Framework**

Pre-framework					During/ Post-framework		
	Early comments on framework	Applying security requirements	Applying privacy requirements	Conflict between security and privacy	Mapping – Matrix	Recommendations	Comments
1	Stakeholders expect GDPR compliant better quality for requirements	Confidentiality Integrity	Anonymity	Auditability conflict anonymity data disclosure – limitations, pattern user Auditability req. by authorities Oblige to standards	Framework supports in easy identifying conflicts among requirements  Three potential conflicts identified starting the phase – situation and framework, the mapping, etc. then examples, as it will be clear what you are doing  Designing and delivery platform for organisation for achieving GDPR compliance, being very complex, privacy/security requirements that need to be considered So even more important to have your framework introduced.	Support provided by the framework can be included Privacy/security requirements related to the task integrity confidentiality of the respondents anonymity availability (replies to question have to be available when needed and accessed) pseudonymity	Introduction, tasks with framework was good
2	Stakeholder involvement Regulatory GDPR security objectives	Monitorability Transparency Accountability	Anonymity	Link between requirements achieve security some privacy could compromised.		Not relevant (all others): auditability and accountability (as anonymity, non- repudiation, authentication and authorisation	



	and goals privacy goals expertise opinion Knowledge gap between regulations			Accountability with anonymity Accountability with undetectability Not adequate accountability		unlinkability (date object and subject) not linking it in terms of disclosure to anyone
3	Most requirements are related and should be considered	Authentication	Anonymity Pseudonymity	Anonymity conflict with confidentiality/ integrity / accountability Based on tools used etc. Anonymity and Accountability Pseudonymise is not in conflict with any form of security	Making changes without knowing what the document includes can be an issue All changes documented should be logged/recorded somewhere – accountability applying framework between anonymity and confidentiality then check whether there is a conflict  once detected: tool will look for solutions and solve by finding patterns that work for both requirement and updating the model. Larger models: difficult to be used to find conflicts Situation is a conflict or not, having it automated, will be easier, list of requirements that might be in conflict to save time etc.	(Term patterns more so than model) (Also put arrows in the diagrams) Methodological level is great Technical side - post doctorate to continue future research Aware that list of tools might be outdated.  RQ3,4 – more for evaluation Separate RQ for thesis from RQ from evaluation  Evaluating the tool, you can use RQ3,4  Mitigate or resolve? Just use one.  RQ2 separate into 2 questions- RQ1how to

					Automated tasks should be done to resolve conflict	identify conflict between sec. and priv. requirements. RQ2How to resolve conflict bet sec. and privacy requirements	
4		Confidentiality Accountability Integrity	Anonymity		Questions about framework: Accountability required? Yes  Supervisor just controlling approval of records  Applying framework between anonymity and confidentiality	Ensure to show which phase of software engineering will do this?	
5	Compliant with GDPR	Confidentiality Integrity Availability authorisation	Ask about privacy requirement - . who has access authorisation and for what reason. How long the different platforms keep the data, remaining	Security of information of medical records more important than usability		Focus and highlight the matrix, connection between Requirement and technical solutions, mitigating conflicts – most important in presentation and research.	Well done, work is very important, really impressed with progress. Has a lot of contribution to academic community

			GDPR compliant				
6	Compliant with GDPR analyse conflict-research	Confidentiality Integrity Availability Authorisation	Access to privacy requirements Authorisation How long different platforms keep the data sharing with sub-contracting keeping GDPR compliant	Security of information of medical records more important than usability			Good project and way in approaching it
7		Accountability, Confidentiality, Authorisation, Authentication, Non-repudiation, integrity, Availability	Anonymity Unobservability Unlinkability			Start with the requirement we currently have and maybe add more later on. Issue could be that resolving these conflicts could be a lot of work What results could transpire with the tools to use - interesting work  Finally, one tool is chosen which is the best one.	Well done, clear and lot of work

8		<p>Accountability, Confidentiality, Authorisation, Authentication, Non- repudiation, integrity, Availability</p>	<p>Anonymity Unobservability Unlinkability</p>			<p>Comparison between tools can be done later. To determine the best tool. But show that aware of this and can be done for future work.</p>	<p>Design by using Sectro tool? Identify differences between conflict? If you design for each e.g. of conflict, if you did design different diagrams with Sectro?</p> <p>While designing if you find differences between the way in which each pair of requirements can be similar or not?</p> <p>Selection of this collection of conflict. How did you decide choosing them? Randomly or not. - Based on research Requirements used for DEFEND project and whether we will add more requirements? - Author response – the list is not exhaustive - Prof.’s method – could add more privacy requirements such as data protection etc. Security requirements ok, but add some more</p>
---	--	--	--	--	--	---	---

							security requirements in order to have more identification of conflicts
9-13	<p>Which is most important to us and what is the balance between these requirements? Depends on the stakeholder objective. Meaning of SOB and BOD (B-Mapping table)</p> <p>Security requirements in conflict with each other? Like</p>	<p>Accountability Integrity Authentication Integrity Availability Withdrawability (the right to withdraw) Non-repudiation</p>	<p>Anonymity Unlinkability Unobservability Pseudonymity</p>	<p>Solution for when you have these conflicts is not on the organisation of the diagram, but on the selection of the right security mechanism that allows both conflicting requirements? - Yes. (accountability vs anonymity - concept, problem, force, solution slide)</p> <p>Conflicts no longer exist as now there are security</p>	<p>Suggestions: Matrix - privacy patents/context concern Contacts should specify 100% possibility to apply the patent (or pattern?) Privacy is far more context-dependent than security. Solution provided is a way to achieve the goal in a way that doesn't influence/harm the other goal. If losing pattern it's ok as almost already to the end. Integrity with anonymity - Don't see any conflicts really, the same with Unlinkability and observability. Supporting tool for Anonymity - choose cryptographic tool</p>	<p>Focus group - could be more guided, more interaction, so easier to follow etc. Analysis could be more fully automated Make each step clear, five steps - fully automated, manual and rest semi-automated Analysis - matrix - provide scenarios to show how the conflicts happen with each other. Future work: usually not all the requirements are relevant; so not all conflicts will arise. Where is the benefit to the user of using the framework?</p>	

separation of SOB and BOD? (B-Mapping table slide)- Yes

Matrix triangular? A conflicts with B, B conflicts with A? (B-Mapping table slide)- Yes

Depends on system? As anonymity may/not be a privacy requirement as you are gathering consent then don't need anonymity requirement so there might be anonymity requirement

mechanisms that can enforce both of these types of conflicting requirements? (supporting document to presentation on word document) -Yes but can't guarantee 100%. The framework presents a warning for the analyst that there is potential conflict, so to maintain/manage conflicts.

Basically, dealing with conflicts with technical solution? Yes  
Is this mechanism used right now? - plenty of research on this but didn't take into account the conflict between those requirements.

a participant notes satisfying security requirement more so than privacy requirement, since the privacy requirement is quite difficult to be tackled by just technical solutions.

How selected type of privacy requirement - state of the art? - recent work, but others could be added later on as the research is not limited.

Possibility of doing focus group with non-experts also.

	<p>but in this case something went wrong in the elicitation of the requirement? Is this what you want to show in your framework? (Phase 1 example)- Author suggests next slide for clarifying the answer to this question.- (Model example SecTro)</p>					
14	<p>Can the analyst edit the data? Need to have zudibility</p>	<p>Usability Confidentiality Access Authentication Authorisation Integrity Audibility</p>	<p>Anonymity Pseudonymity</p>	<p>Confidentiality Conflict anonymity Integrity conflicts Anonymity Confidentiality Conflict pseudonymity</p>	<p>There is a contribution in the framework, and it does help the analysis to identify and solve conflicts using models and tools. Phase 2 is critical because there is no specific structure to follow, it</p>	<p>Extend the list of requirements (to cover more about security and privacy)</p>

						would be better if this phase is automated	
15	How to store, share, and save data? Who can access those data?	Usability Confidentiality Access Authentication Authorisation Integrity	Anonymity Pseudonymity	Confidentiality Conflict anonymity Integrity conflicts Anonymity Confidentiality Conflict Pseudonymity		The privacy and security requirements of a large system may form a complex network of dependencies and contradictions. It makes sense to model these requirements and provide tooling support that helps to identify conflicting requirements to make suggestions on how to resolve them.	Prioritise the tools or have a method of choosing the most suitable solution. What is the most applicable tool to solve this conflict?



### 8.6.1-3 Generating and reviewing themes

Next, given the codes we have created, we can identify patterns and start developing themes. These are generally broader than codes, and usually combine several codes into a single theme. In Table 8.5, a summary of the themes is generated before and after the implementation of the framework. This comprises the stakeholder, data protection, software requirements, conflicts (pre- framework); and the *ConfIS* framework, future contributions and mentions of the research methodology (post framework). Table 8.6 gives further in-depth extracts of the participant responses per code and theme (see Tables 8.5 and 8.6).

**Table 8.5 Turning Codes into Themes**

	<b>Theme</b>	<b>Code</b>	<b>ID</b>
<b>Pre-Framework</b>	Stakeholder	Expectations	1,2,9-13
	Data Protection	GDPR	1,2,5,6,9-13,14,15
	Software Requirements	Security requirements	1,2,3,4,5,6,7,8,9-13,14,15
		Privacy requirements	1,2,3,4,5,6,7,8,9-13,14,15
	Software Requirement Conflicts	Associated Conflicts	1, 2, 3, 5, 6,9-13,14,15
		Standards	
Compromising Progress		1, 2, 3, 5, 6,9-13,14,15	
<b>Post-Framework</b>	<i>ConfIS</i> Framework	Contribution	1,2,3,4,5,6,7,8,9-13,14,15
		Design	
		Improvements	
	Future Contribution	Short Term	3, 4, 7,8,9-13,14,15
		Longer Term	
	Research Methodology	Task Scenario	3, 9-13
Focus Group			
Analysis Matrix			

**Table 8.6 Interview Extracts: Turning codes into Themes**

	<b>Code</b>	<b>ID participants</b>	<b>Participants' responses</b>	<b>Theme</b>
<b>Pre-Framework</b>	<b>Expectations</b>	1,2,9-13	Stakeholders expect GDPR compliance. Which is most important and what is the balance between these requirements? This depends on the stakeholder's objective.	<b>Stakeholders</b>
	<b>GDPR</b>	1,2,5,6,9-13,14,15	Stakeholders expect GDPR compliance as it creates a better quality of requirements; designing and delivering a platform for the organisation for achieving GDPR compliance; in achieving regulatory GDPR and security/privacy objectives and goals, expertise is needed to fill the knowledge gap in these regulations; the expectation of being compliant with GDPR regulations is important; how long do the different platforms keep data and how is the data stored while being GDPR compliant; standards for data sharing with sub-contracting work while remaining GDPR compliant. Suggestions: Matrix-privacy patients/context concern. Contacts should specify 100% possibility to apply the patent (or pattern?). Can the analyst edit the data; how to store, share and save data? Who can access those data?	<b>Data Protection</b>
	<b>Security Requirements</b>	1,2,3,4,5,6,7,8,9-13,14,15	Applying Nominal Ranking Evaluation Method: <u>Security</u> Integrity Confidentiality Accountability Availability Authentication Authorisation Non-repudiation	<b>Software Requirements</b>
	<b>Privacy Requirements</b>	1,2,3,4,5,6,7,8,9-13,14,15	<u>Privacy</u> Anonymity Unobservability Pseudonymity Unlinkability	

	<b>Associated Conflicts</b>	1, 2, 3, 5, 6,9-13,14,15	Applying Nominal Ranking Evaluation Method: accountability and anonymity auditability and anonymity accountability and undetectability anonymity and confidentiality anonymity and integrity	<b>Software Requirement Conflicts</b>
	<b>Conflict additional:</b>  <b>Standards/ Compromising/ Progress</b>	1, 2, 3, 5, 6,9-13,14,15	Obligation to maintain <b>standards</b> ; In achieving security some privacy could <b>compromised</b> ; Solution for when you have these conflicts is not on the organisation of the diagram, but on the selection of the right security mechanism that allows to both conflicting requirements. Conflicts no longer exist as <b>now</b> there are security mechanisms that can enforce both of these types of conflicting requirements (supporting document to presentation on Word document) but cannot guarantee 100%. The framework presents a warning for the analyst that there is potential conflict, so to maintain/manage conflicts. Basically, dealing with conflicts with technical solution? Yes <b>Plenty of research done</b> on this but doesn't quite take into account the conflict between requirements	
<b>Post Framework</b>	<b>Contribution/ Design/ Improvements of Framework</b>	1,2,3,4,5,6,7,8,9-13,14,15	<b>Framework Contribution:</b> Framework supports in easily identifying conflicts among requirements; so even more important to have your framework introduced; designing and delivering a platform for organisations for achieving GDPR compliance, but also its being very complex, privacy and security requirements that need to be considered; introduction, tasks with framework was good; once detected tool will look for solutions and solve by finding patterns that work for both requirements and updating the model; methodological level is great; mitigating conflicts – most important in presentation and research; well done, work is very important, really impressed with progress; has a lot of contribution to academic community; analyse conflict-research;	<b>ConfIS Framework</b>

good project and way to approach it; well done, clear and lot of work;

**Framework Design:**

Identify differences between conflict?  
 If you design for each e.g. of conflict if you did design different diagrams with SecTro?

While designing if you find differences between the way in which each pair of requirements can be similar or not?  
 Selection of this collection of conflict.

How did you decide between them?  
 Randomly or not? - Based on research Requirements used for DEFEND project, and will more be added? Duaa response – the list is not exhaustive  
 Professor’s method – could add more privacy requirements like data protection etc.

Security requirements in conflict with each other? Like separation of SOB and BOD?  
 (B- Mapping table slide) - Yes;  
 Matrix triangular? A conflict with B, B conflicts with A? (B - Mapping table slide) –Yes;

Framework provided is a way to achieve the goal in a way that does not influence/harm the other goal.

**Improvements for Framework:**

Support provided by the framework can be included;  
 be aware of the list of tools that might be outdated;  
 losing pattern is ok as almost already to the end;  
 there is a contribution in the framework and it does help the analyst to identify and solve conflicts using models and tools;  
 prioritise the tools or have a method to choose the most suitable solution;  
 what is the most applicable tool to solve this conflict;  
 security requirements ok, but should add some more.

<b>Short/ Longer term Contributions</b>	3, 4, 7,8,9- 13,14,15	<b>Short Term:</b> RQ3,4 – more for evaluation Separate RQ for thesis from RQ from evaluation	<b>Future Contribution</b>
---	--------------------------	---	--------------------------------

Evaluating the tool, you can use RQ3,4  
Mitigate or resolve? Just use one.  
RQ2 separate into 2 questions – RQ1 how  
to identify conflict between security and  
privacy requirements.  
RQ2 How to resolve conflict between  
security and privacy requirements; ensure  
you show which phase of software  
engineering you will use to do this?; start  
with the requirement you currently have  
and maybe add more later on.  
Issue could be that resolving these  
conflicts could be a lot of work  
What the results could have with the tools  
to use – interesting work  
Finally, one tool is chosen which is the  
best one; But comparison between tools  
can be done later to determine the best  
tool.  
Show that we are aware of this and can be  
done for future work; future work: usually  
not all the requirements are relevant; so  
there may not be conflicts arising.  
Where is the benefit to the user of using  
the framework?  
Satisfying security requirements more so  
than privacy requirements since the  
privacy requirement is quite difficult to  
tackle by just technical solutions.

**Longer Term:**

Larger models: difficult to use in finding  
conflicts.  
Situation is a conflict or not, having it  
automated, will be easier to save time etc.  
Automated should be done say to resolve  
conflict.  
How the certain privacy requirements were  
selected – state of the art? – recent work,  
but others could be added later as the  
research is not limited.  
Possibility of doing focus group with non-  
experts also;  
Phase 2 is critical because there is no  
specific structure to follow, it would be  
better if this phase is automated; extend  
the list of requirements (to cover more  
about security and privacy);  
the privacy and security requirements of a  
large system may form a complex network

		of dependencies and contradictions, it makes sense to model these requirements and provide tooling support that helps in identifying conflicting requirements to make suggestions on how to resolve them; Technical side – post doctorate to continue future research.	
<b>Task Scenario/ Focus Group/ Analysis/ Matrix</b>	3, 9-13	For the <b>task scenario</b> – Making changes without knowing what the medical doctor has really included first can be an issue; all changes documented should be logged/recorded somewhere; <b>focus group</b> – could be more guided, more interaction, so easier to follow; <b>analysis</b> could be more fully automated; make each step clear, five steps – fully automated, manual and rest can be semi-automated; Analysis – <b>matrix</b> – provide scenarios to show how the conflicts happen with each other; (term patterns more so than model) (also put arrows in the diagrams).	<b>Research Methodology</b>

#### 8.6.1-4 Synopsis

Thematic analysis – a widely used method in qualitative research – is applied. This method is usually applied to a set of texts, such as interview transcripts, which in this instance, is retrieved from the focus group. Fifteen participants consented to the study, comprising various academic backgrounds – PhD students, PhD doctors and research fellows were interviewed using recorded video conferencing. These participants comprise an international audience, which made video conferencing the ideal method of data collection, especially during the Covid-19 lockdown period.

Participants work within different universities from various countries including the United Kingdom, Italy, Greece, Germany, Saudi Arabia and China; which gives scope for a variety of perspectives (heterogenous).

The TA analysis was utilized pre and post framework to simply differentiate the effects of use of the ConfIS framework compared to its non-use. Prior to introducing the framework, from the scenario presented to the participants, they utilized their judgement to identify privacy and security requirements, and the associated conflicts likely to arise. Mitigating tools however were not suggested by participants. When the framework was introduced, this better supported participants in identifying requirements in the scenario that might have been missed and seeing the conflicts likely to arise. Furthermore, the framework supported in suggesting the utilisation of several conflict mitigation tools.

In analysing participant responses, we collate all the data into groups identified by codes pre- and post-framework. Thereafter, the data is closely examined, to code the data. Here, sections of the text are highlighted and colour-coded which helps to describe the relevant or potentially interesting data (see Table 8.7). The codes derived pre-framework are expectations, GDPR, security requirements, privacy requirements, associated conflicts, standards, compromising and progress. The associated post-framework codes are contribution, design, improvements, short term, longer term, task scenario, focus group, analysis and matrix (see Table 8.7). These codes allow us to gain a condensed overview of the main points and common meanings that recur throughout the data.

Next, we identify common themes – topics, ideas and patterns of meaning that occur repeatedly. These themes help us to understand the data and answer the research questions. In Table 8.7, a summary of the themes is generated pre- and post-framework implementation. For some themes, several codes are summed into single themes. Pre-framework, the relevant themes are stakeholder, data protection, software requirements and software requirement conflicts; whilst post-framework themes are *ConfIS* framework, future contribution and

research methodology. Tables 8.4 and 8.5 give further in-depth extracts of the participant responses per theme and associated code.

Prior to introducing the *ConfIS* framework, participants gave their views on expectations, pertaining mainly to the relevant stakeholders. Similarly, GDPR, which held much significant importance, was embedded within the theme of data protection. Security and privacy requirements codes were captured within the software requirements theme; and all associated conflicts, standards to maintain, the possibility of compromising and progress, were captured within software requirement conflicts. Post-framework, contribution, design and improvements codes were identified within the *ConfIS* framework theme, whilst future contributions had short (immediate) and longer term (possibly for post doctorate research) contribution codes. Lastly, comments on the present research methodology theme were given which identified task scenario, focus group, analysis and matrix as relevant codes.

Addressing each theme in turn was of utmost importance. We describe how often the themes occur and what they mean, including examples from the data as evidence pre- and post-framework (see Tables 8.7 and 8.8). Firstly pre-framework, the nominal ranking evaluation method is used to rank in order of importance the participants' responses to software requirements as required by the task. The top three security requirements are seen to be integrity, confidentiality, and accountability, while anonymity, unobservability and pseudonymity are ranked the highest in privacy requirements (see Table 8.9).

Next, participants' responses to identifying possible conflicts between requirements, show that accountability and anonymity were the most highly voted for, followed by auditability and anonymity and accountability and undetectability. Anonymity accounts for a large portion of Phase 2 especially given data protection, GDPR standards, which organisations are required to



uphold by law. Furthermore, participants were concerned that while achieving security, privacy requirements could be compromised or vice versa. The proposed framework deals with this by guiding the selection of the right mechanisms for dealing with conflicts between requirements. While this is so, no framework can guarantee 100% reliability that the conflicts will be resolved, as the framework presents a warning for the analyst that there is a potential conflict, so that conflicts can be detected and managed. There exists research on this subject, but without taking into account conflicts between requirements and mitigating tools (Maxwell, Antón & Swire, 2011; Schon, Thomaschewski & Escalona, 2017).

Data protection within GDPR plays a key role in software requirements, as it guides the stakeholders' expectations (decisions and objectives). In achieving regulatory GDPR and security/privacy objectives/goals, expert opinion is needed to fill the knowledge gap. The expectation of being compliant with GDPR regulations is important. Designing and delivering a framework for organisations for achieving GDPR compliance is important but so also is identifying and managing conflicts, as this makes for a better quality of software requirements.

**Table 8.7 Ranking Themes: Pre-Framework**

Rank	Theme	Code
1.	Pre-Framework	Software Requirements
		Security Requirements
2.	Pre-Framework	Software Requirement
		Conflicts
		Standards
		Compromising
3.	Pre-Framework	Progress
		Data Protection
4.	Pre-Framework	GDPR
		Stakeholder
		Expectations

Post-framework, the ConfIS framework has obviously been the most talked about. Participants add that using the framework for the task was positive, as its methodology provides solutions by finding patterns that work for both requirements. It aids the analyst to easily identify and mitigate conflicts among requirements. Furthermore, the framework provides a way to achieve the requirement aim in a way that does not negatively influence/harm other requirement goals, helping the analyst to identify and solve conflicts using models and tools.

Participants did have some suggestions regarding the framework, such as short to longer term contributions, including additional requirements that can be added to the framework, being aware of tools that might be outdated, and if the framework accommodates this. Additionally, introducing a method that guides the analyst in prioritising the most suitable tools to mitigate conflicts, and interviewing non-experts would be quite interesting, as different groups can give a variety of views.

Moreover, further justification for the proposed framework is given by the likelihood of having the framework automated, which would save users time in solving conflicts. The privacy and security requirements of a large system may form a complex network of dependencies and contradictions, so it makes sense to model these requirements, and provide tools as support to help identify conflicting requirements and make suggestions on their resolution.

**Table 8.8 Ranking Themes: Post-Framework**

Rank	Theme	Code
1.	Post-Framework	Contribution
		Design
		Improvements
2.	Post-Framework	Future Contribution
		Short Term Longer Term
3.	Post-Framework	Task Scenario
		Research Methodology
		Focus Group
		Analysis Matrix

### 8.7 Application to ConfIS Framework

The four research questions put forward by this PhD research were as follows:

RQ 1- How would you classify security and privacy requirements that are in conflict?

RQ 2 - How to design a framework that can support the analyst to (identify/resolve) conflicts?

RQ 3 - Is tool support useful for the requirements analyst in identifying and solving conflicts between security and privacy?

RQ 4 - Does the proposed solution mitigate conflicts?

We explain the main solutions and show how the analysis has aided in answering the research questions. This is further elaborated in Chapter 9 – Key Findings. Participants’ responses are given in Appendix C. The supporting document and framework (see Appendix D) present support in identifying and resolving conflicts and providing solutions to mitigate conflicts. In pursuit of answering *RQ 1- How would you classify security and privacy requirements that are in conflict?* **Phase 1: Mapping Security and Privacy Requirements** provides a table of the **List of Security and Privacy Requirements** supported by literature reviews and further research. A generic list is stipulated in Table 8.9.

**Table 8.9 Generic List of Common Security and Privacy Requirements**

<b>Security Requirements</b>	<b>Privacy Requirements</b>
Availability	Anonymity
Non-repudiation	Unlinkability
Confidentiality	Pseudonymity
Integrity	Unobservability
Authentication	Undetectability
Authorisation	
Accountability	
Auditability	

On the other hand, *RQ 2 - How to design a framework that can support the analyst to (identify/resolve) conflicts?* is supported by a mapping matrix as stipulated in **Phase 1: Mapping Conflicts between security and privacy requirements** (see Appendix C). Additionally, *RQ 3 - Is tool support useful for the requirements analyst in identifying and solving conflicts between security and privacy?* **Phase 2: Identify Conflicts between Requirements and Conflict Decisions: 2.1 Supporting Tools, and 2.3 Privacy Pattern Library** (see Appendix D) provide the relevant tools required by the analyst.

Lastly, *RQ 4 - Does the proposed solution mitigate conflicts?* the framework and its supporting document as specified in Appendix C, does seek to mitigate conflicts, given that prior steps Phase 1 and 2 are fulfilled, **Phase 3: Conflict Resolution Patterns** (its table and design view) contribute to mitigating conflicts. All research questions are further elaborated on in the conclusion Chapter 9.

## **8.8 Summary**

This chapter has firstly addressed the ethical perspective of data collection and management in the experimentation, review and data management of participants' information, and documenting consent (section 8.2). Next, preliminary evaluation of the framework and integrating resolving conflicts has been investigated in section 8.3. The evaluation is updated by incorporating the recommendations from the preliminary findings. Thereafter, section 8.4 covered actual evaluation with participants using the focus group method and discussed how to apply the framework phases. At the end of the session, participants completed an evaluation questionnaire which covered all phases of the framework. We followed up by interviews, with participants giving in-depth evaluation feedback.

To complete this evaluation, we prepared content material as a toolkit to support the participants to understand the framework. Its strategy and results are presented. Next, in section 8.6, thematic analysis and the steps are used as an evaluation tool to answer the research questions. Thereafter, applying thematic analysis to evaluate the participants' responses in the case study using ConfIS framework, is examined in section 8.7.

# CHAPTER 9

## CONCLUSION AND FURTHER WORK

### 9.1 Introduction

This thesis reports the development and implementation of an innovative approach to conflict resolution in requirements engineering. The ConfIS framework can be used to identify and mitigate conflicts between security and privacy requirements in the field of software engineering. This chapter reviews the research aims, discusses issues arising from the research and threats to its validity, sets out the novel contributions of the research to knowledge and proposes a plan for future research activities.

### 9.2 Research aims revisited

Requirements engineers live in a world in which inconsistencies are the rule, not the exception. There are many kinds of inconsistencies, many of which originate from the elicitation of goals and requirements from multiple stakeholders.

Conflicts between privacy and security requirements are very common, and they no doubt exist in every sector including banking, education and health care. These sectors are particularly under pressure to ensure user privacy whilst maintaining system security and invulnerability.

Several studies show that the time and cost involved in handling requirements conflicts is one of the main reasons for the failure of software projects (Butt *et al.*, 2011). It is essential, therefore, to detect and resolve conflicts in early phases of software development in order to prevent re-iterations of conflict and redevelopment at all phases (Heisel and Souquires, 2001). (Boehm and Papaccio, 1988) has highlighted the potential escalation of costs that can occur should the coding process have to be repeated during development phases, and thus the earlier

conflict can be resolved, the lower the cost in terms of time and money for the developer and end user.

This thesis addresses the problem of identifying and resolving conflicts between security and privacy requirements. It developed a three-phase framework to identify, analyse and resolve conflicts between security and privacy requirements. The proposed framework was implemented using SecTro, a CASE Tool for Modelling Security in Requirements Engineering using Secure Tropos. Secure Tropos is a software system which ensures that software is developed according to the user's needs in conjunction with security and privacy. The tool supporting the methodology described in this thesis has been implemented as an extension of the Secure Tropos tool and methodology and evaluated within the DEFEND European Union project.

### **9.3 Issues arising from conducting the research**

- At the start of this research, we were searching for risks relating to security and privacy, however, this is a huge area and too unwieldy for a single piece of research, so it was decided to narrow the research by investigating conflicts between security and privacy requirements. This topic was honed during the course of investigation and research (Chapter 2).
  
- In Chapter 2, we discussed the prioritisation of requirements – a crucial step in order to sort the privacy and security requirements into their relative order of importance. By applying the framework to DEFEND, however, this step was not ultimately needed because the requirements are already prioritised within DEFEND. The discussion

remains within the thesis as it represents an important element of work, but it is worth noting that the step will not always be required.

- Due to the Covid-19 pandemic and national lockdowns, the primary research data collection method was undertaken remotely instead of a live session with students, academics and research fellows. Whilst engagement with academics and researchers alone is not ideal, the national lockdown rendered it impossible to meet face-to-face with professionals to discuss the framework. Business support professionals represent the link between academia and practice; not only are they familiar with ‘high level’ concepts but also with practical application issues – most support professionals have an academic background and can critically evaluate concepts and their meaning in practice. Five online evaluation sessions took place, each of which was broken into smaller groups. In order to gain maximum benefits from the participants’ experiences and enable the participants enough time to understand, apply and test the framework, they were divided into groups of three for each session. Live sessions would have had a different set of benefits, as in presenting the framework participants could all gather in one session, allowing them the chance to discuss the work together. On the other hand, running multiple sessions improved my presentation skills and gave me the opportunity to refine the framework and research as it developed. In addition, the use of online sessions benefitted the framework by allowing a more diverse pool of participants, because in this case the framework would be verified by expertise from different countries that they might have different perspective or point of view. Rather than it be evaluated in our university community. In addition, thus ensuring greater applicability and generalisability. This worldwide evaluation accessed individuals in



the field of requirement engineering from the UK, China, Italy, Germany, Greece and Saudi Arabia.

- Additionally, in the evaluation (Chapter 8) two examples could be introduced with the online session, where the first example is introduced to the participants, and they discuss with everyone and myself. They can predict requirements which are in conflict and so on, then I explain it using the ConfIS framework to resolve conflicts. Then for the second scenario, the participants discuss in their groups, having the toolkit to assist them throughout the ConfIS phases. After that they discuss their outcomes with me. The final step was to fill up questioner about the framework and each phase.

## **9.4 Limitations**

### **9.4.1 List of requirements not exhaustive**

In Phase 1 of the ConfIS framework, a list of security and privacy non-functional requirements are defined. These are chosen, guided by the literature review, because they are the most widely used and discussed. This list, however, should not be considered to be exhaustive, but merely to be illustrative of the value of the framework.

Furthermore, the research focused on conflicts surrounding non-functional requirements (NFR), describing how a system should work, and its properties or characteristics. Conflicts regarding functional requirements (FR), however, were not explored. NFRs and FRs are fully reliant on each other, the two being necessary complements to each other, but the study aimed to use NFRs in this instance to determine trends and updates in ever-changing policies (such as GDPR), increasing the need for interoperability with other software or hardware systems,

and external factors such as safety and privacy regulations. Conflicts surrounding functional requirements and the effect of their interrelation may be explored in future work.

#### **9.4.2 Side- effects of deploying framework and resolution strategies**

Additionally, there can potentially be project risks involved. This is because in improving one aspect of security could result in causing a privacy leakage elsewhere because of the proposed solution. There could also be risk in deploying a different conflict resolution tool (as shown in Phase 3 of the framework), where the outcome might be unknown. Therefore, promoting higher security could reduce privacy to beyond an acceptable level. Perhaps extra staff would also be needed as contractors who have expertise in managing the new technologies as they are applied to address these security-privacy conflicts. This could result in higher project costs and even delays to the schedule.

#### **9.5 Contribution**

This research has introduced a novel way to model and analyse conflicting requirements, taking the existing work in the literatures further, and advancing the current state of art.

Firstly, engaging a critical literature review analysis allowed us to list the relevant security and privacy frameworks. This gave insights into the gaps which this research attempted to fill, giving an overall idea about the frameworks that care about security and privacy. This revealed the need to build a three-part ConfIS framework and mapping matrix for identifying conflicts between privacy and security requirements in non-functional requirements. This had not been done previously. In addition, we listed modelling language to understand and have a background about modelling, in order to decide which language, we will apply to the ConfIS framework.

Secondly, the design of ConfIS was applied in Secure Tropos, but it is not specific for this language only. The framework presented in this work contributes towards a multitude of different areas of interest, including security requirements, and the attempt to import of privacy patterns giving rise to Privacy Enhancing Technologies connected with privacy requirements that they satisfy.

Furthermore, we narrow security and privacy requirements to fifteen requirements based on the most frequented use of those requirements in the literatures. In addition, we do a mapping matrix to allocate the requirements which are likely to be in conflict: This revealed the most likely requirements to be in conflict which can act as a proactive precautionary measure to minimize conflict, analysts can take even before conflict does arise in a real case scenario. For instance, the mapping matrix revealed security requirement *availability* is likely to conflict with several privacy requirements- *anonymity*, *unlinkability*, *pseudonymity*, *unobservability* and *undetectability*.

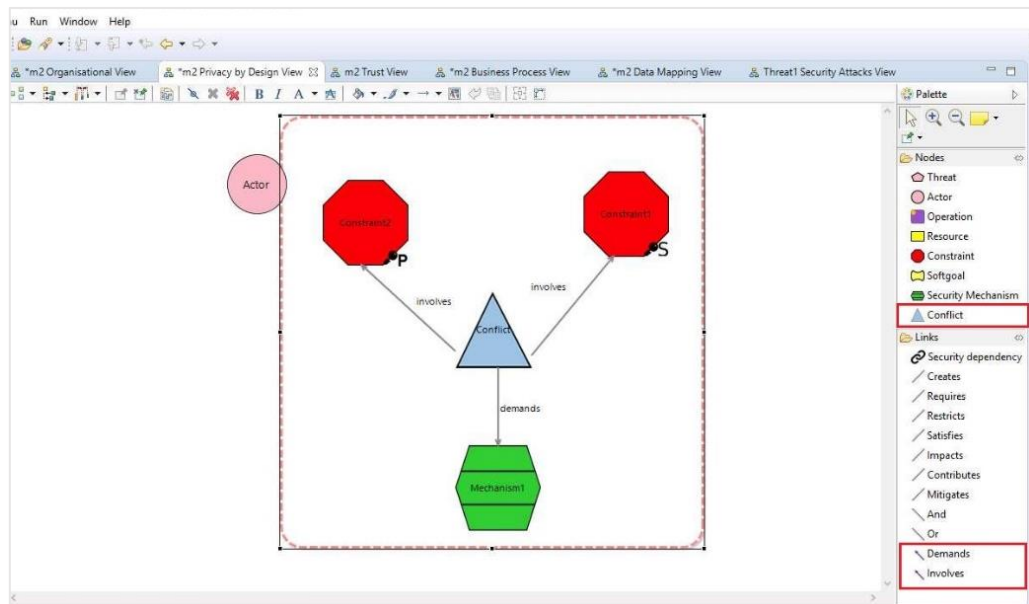
Additionally, in the conflict resolution phase, we incorporate supporting tools to understand how these tools can be helpful to resolve the conflicting issue between privacy and security requirements. Though the supporting tools list is not exhaustive, we have listed essential ones that would be relevant and useful to the fifteen privacy and security requirements conflict resolution. The pictorial models of resolving conflict are also helpful to an analyst to resolve conflicts. The models are clear, easy to be understand and applicable. This is supported by our focus group study, where we apply the ConfIS framework in two case studies, one in the pilot chapter, and another in a real case study to verify and validate the effectiveness of the

framework. Specifically, this framework is evaluated and tested by experts in requirements engineering.

This framework minimises the time for analysts to identify and resolve conflicts between security and privacy requirements. The manual step introduces the use of the mapping matrix to identify conflicts faster, and then applies the supporting tool to resolve conflicts in the Privacy Pattern library utilizing computer- aided software engineering (CASE) tools. This further would benefit the analyst in terms of their having a mechanism which provides flexibility by allowing the analyst to select the most suitable tool to resolve conflicts.

ConfIS framework contributes to Secure Tropos as follows:

- The extension of the already established Secure Tropos modelling language. To identify conflicts between requirements, we added the conflict notation in privacy by design view in Secure Tropos (see Figure 9.1)



**Figure 9.1 Conflict notation in Privacy by Design View**

- The framework also covers conflict concerns via the introduction of new concepts (e.g. conflict). In addition, new relationship types have been introduced (involve and demand) to existing concepts to allow for a more accurate and quantifiable.
  
- A semi-automated step by adding the supporting tool to resolve conflicts in the Privacy Pattern library. This would benefit the analyst in terms of their having a mechanism which also provides flexibility by allowing the analyst to select the most suitable tool to resolve conflicts.

The research, furthermore, presents a framework for security and privacy requirements identification and analysis for supporting GDPR compliance. The framework makes use of a utility tool for the resolution of conflict between security and privacy requirements – a common occurrence within software development. The framework consists of three phases: the first phase identifies security and privacy requirements separately; the second phase ascertains the conflicting scenarios; and in the third phase, we model a pattern to link and resolve two conflicting requirements, and a suitable supporting tool to aid this process.

The ConfIS framework furthermore was built with GDPR standards in mind, hence conforming to GDPR compliance. The framework is therefore not purely theoretical but is also embedded in a much larger practical European Union - EU DEFEND project, which can support EU regulators in nationwide decision making, with regards to data and compliance.

## **9.6 Future Work**

Suggestions for future research fall into five categories, which include:

➤ **Building upon findings of the research.**

The list of software requirements selected for the study was not exhaustive and thus future research could test the framework using other requirements such as resilience, liability, trust (Jaiswal, & Gupta, 2017), fairness (Ramadan, 2020) and privacy preserving (Yahuza *et al.*, 2020). Viability and robustness could be tested in a wider variety of business contexts, while the expansion of the framework could identify and show the relationships between more than two requirements.

➤ **Generalisation**

Furthermore, while the ConfIS framework offers an original contribution to knowledge by mapping privacy and security NFRs, the inclusion of FRs and a broader exploration of its validation would be a useful development.

➤ **Addressing limitations of the research.**

The inclusion of non-academic users from the professional or technical environment would be a useful addition for future work. As discussed above, while the input of academic users was valuable, providing a rich source of feedback on the ConfIS framework, further diversity would provide holistic feedback and better represent the views of the end user to complement the more theoretical, academic opinion. Additionally, more in-depth understanding of the framework and its application in practice could have been gained through follow-up interviews with selected participants. Focus groups can have their limitations and, even with an experienced moderator, one or two people may dominate the group and sway the opinions of

others, so follow-up interviews would contribute more specialised in-depth knowledge about the framework which may have been missed in the focus group meeting.

➤ **Constructing the same research in a new context, location and/or culture**

This research was designed to complement the DEFEND project; part of an EU project to support organisations by defining essential tools and methods that enable organisations and authorities to monitor their actions for GDPR-compliance. While organisations in EU countries are likely to be affected by GDPR, it would be interesting to apply the ConfIS research framework to a non-EU region or country, such as the Middle East, in order to assess its applicability within a different cultural and policy setting.

➤ **Re-assessing and expanding the framework**

Expanding the ConfIS framework would include the consideration of additional features and phases. This includes the prioritisation of conflicts where conflicts are ranked in a certain order based on various criteria, with the aim of improving conflict resolution. The current research has shown that there is no single supporting tool which is suitable for resolving all types of conflict, however, attempts could be made to build a framework which incorporates all security and privacy requirements alongside their supporting tools, as well as further automation of the identification process.

Moreover, the ConfIS framework presents the tools to mitigate the associated conflict between privacy and security requirements. The suggested tools present a baseline for the analyst to work with, in choosing from this list the best tool choice over another with several options. His optimum solution (supporting tool) can be the most relevant tool that suits most requirements being in conflict, easy of installing and using the tool, or one which ensures best results by

minimizing conflict between requirements as much as possible, or one which ensures a minimum cost of using the tool. The analyst needs first to set up a criterion to choose the best optimum solution, which will differ from analyst to analyst.



## REFERENCES

- Abadi, M. & Glew, N. (2002) 'Certified email with a light on-line trusted third party: Design and implementation', *Proceedings of the 11th international conference on World Wide Web*. Hawaii, May 2002, Association for Computing Machinery, pp. 387–395. Available at: <https://doi.org/10.1145/511446.511497> (Accessed:10 March 2019 ).
- Alberts, C., Woody, C. & Dorofee, A. (2014) *Introduction to the Security Engineering Risk Analysis (SERA) Framework*. Pittsburgh, PA: Carnegie Mellon University.
- Albrecht, J. P. (2016) 'How the GDPR will change the world', *European Data Protection Law Review*, 2(3), pp. 287–289.
- Aldekhail, M., Azzedine, C. & Djamal, Z. (2016) 'Software Requirements Conflict Identification: Review and Recommendations,' *International Journal of Advanced Computer Science & Applications*, 7(10), pp. 326–335.
- Aldekhail, M. and Ziani, D., 2017. Intelligent method for software requirement conflicts identification and removal: proposed framework and analysis. *International Journal of Computer Science and Network Security*, 17(12), pp.91-95.
- Ali, R., Dalpiaz, F. and Giorgini, P., 2010. A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 15(4), pp.439-458.
- Alkubaisy, D. (2017) 'A framework managing conflicts between security and privacy requirements', *2017 11<sup>th</sup> international conference on Research Challenges in Information Science (RCIS)*, Institute of Electrical and Electronics Engineers, pp. 427–432. doi: [10.1109/RCIS.2017.7956571](https://doi.org/10.1109/RCIS.2017.7956571).
- Alkubaisy, D., Cox, K. & Mouratidis, H. (2019) 'Towards detecting and mitigating conflicts for privacy and security requirements,' in Kolp, M. *et al.* (eds.) *Proceedings: RCIS 2019 - IEEE 13th international conference on Research Challenges in Information Science: Towards a design science for information systems*. Brussels, 29–31 May 2019, Belgium: Institute of Electrical and Electronics Engineers Computer Society. Available at: <https://doi.org/10.1109/RCIS.2019.8876999> (Accessed 05 Dec 2020).
- Alqassem, I. and Svetinovic, D., 2014, December. A taxonomy of security and privacy requirements for the Internet of Things (IoT). In *2014 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1244-1248). IEEE.
- Andrews, A. A. & Pradhan, A. S. (2001) 'Ethical issues in empirical software engineering: The limits of policy', *Empirical Software Engineering*, 6, pp. 105–110. Available at: <https://doi.org/10.1023/A:1011442319273> (Accessed 20 October 2016).
- Anon. (2019) *Official Legal Text* [online]available from <<https://gdpr-info.eu/>> [20 June 2020]
- Antón, A. I. (1996) *Goal-based requirements analysis*. Colorado Springs, CO: IEEE. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=491438> (Accessed: 5 April 2016).

Antón, A. I. & Earp, J. B. (2000) ‘Strategies for developing policies and requirements for secure electronic Commerce Systems’, Submitted to: *1<sup>st</sup> workshop on security and privacy in e-commerce at CCS2000*. New York, NY: Springer. Available at: <https://www.cc.gatech.edu/~aianon/assets/strategiesfordevelopingpolicies.pdf> <http://www4.ncsu.edu/~aianon/pubs/ccs2000.pdf> (Accessed: 20 January 2016).

Aos, A. Z. *et al.* (2009) ‘Approved undetectable-antivirus steganography for multimedia information in PE-file’, *2009 International association of computer science and information technology – Spring Conference*. Singapore, 2009, pp. 437–441. doi: 10.1109/IACSIT-SC.2009.103.

Argyropoulos, N., Shei, S., Kalloniatis, C., Mouratidis, H., Delaney, A., Fish, A. and Gritzalis, S., 2017, January. A semi-automatic approach for eliciting cloud security and privacy requirements. In *Proceedings of the 50th hawaii international conference on system sciences*.

Atzori, L., Iera, A. and Morabito, G., 2010. The internet of things: A survey. *Computer networks*, 54(15), pp.2787-2805.

Ayala-Rivera, V. and Pasquale, L., 2018, August. The grace period has ended: An approach to operationalize GDPR requirements. In *2018 IEEE 26th International Requirements Engineering Conference (RE)* (pp. 136-146). IEEE.

Baldoni, M. *et al.* (2018) ‘Computational accountability in MAS organizations with ADOPT’, *Applied Sciences*, 8(4), p. 489.

Basin, D., Doser, J. & Lodderstedt, T. (2006) ‘Model driven security: From UML models to access control infrastructures,’ *ACM Transactions on Software Engineering and Methodology*, 15(1), pp. 39–91. Available at: <http://portal.acm.org/citation.cfm?doid=1125808.1125810> (Accessed: 10 March 2016).

BBC News (2016) *TalkTalk fined £400,000 for theft of customer details*. 5 October 2016. Available at: <https://www.bbc.co.uk/news/business-37565367> (Accessed June 2019).

Bellotti, V. & Sellen, A. (1993) ‘Design for privacy in ubiquitous computing environments’, *ECSCW’93 Proceedings of the third conference on European conference on computer-supported cooperative work*. Milan, 13–17 September 1993. Norwell, MA: Kluwer Academic Publishers, pp. 77–92.

Bernard, H., Wutich, A. & Ryan, G. (2017) *Analyzing qualitative data: Systematic approaches*. 2nd edn. Thousand Oaks, CA: Sage Publications.

Berthold, O. & Langos, H. (2002) ‘Dummy traffic against long term intersection attacks’, in Dingledine, R. & Syverson, P. (eds.) *Proceedings of privacy enhancing technologies workshop (PET 2002)*. Lecture Notes in Computer Science 248. Berlin, Germany: Springer-Verlag, pp. 110–128.

Bhagwan, R., Savage, S. & Voelker, G. M. (2003) ‘Understanding availability’, in Kaashoek, M. F. & Stoica, I. (eds.) *Peer-to-Peer Systems II. IPTPS 2003. Lecture Notes in*

*Computer Science*, vol 2735. Heidelberg, Germany: Springer. doi.org/10.1007/978-3-540-45172-3\_24.

Bhavsar, R. *et al.* (2019) 'Resolving conflicts in requirement engineering through agile software development: A comparative case study', in Bhattacharyya, S. *et al.* (eds.) *International Conference on Innovative Computing and Communications*. Singapore: Springer, pp. 349–357.

Birrel, N. D. & Ould, M. A. (1986) *A practical handbook for software development*. Massachusetts, MA: Cambridge University Press.

Biswas, C., Gupta, U. D. & Haque, M. (2019) 'An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography', *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. Cox's Bazar, Bangladesh: Institute of Electrical and Electronics Engineers, 2019, pp. 1–5. doi: 10.1109/ECACE.2019.8679136.

Boehm, B.W. and Papaccio, P.N., 1988. Understanding and controlling software costs. *IEEE transactions on software engineering*, 14(10), pp.1462-1477.

Boyatzis, R. E. (1998) *Transforming qualitative information: Thematic analysis and code development*. Thousand Oaks, CA: Sage Publications.

Breaux, T.D., Antón, A.I. and Spafford, E.H., 2009. A distributed requirements management framework for legal compliance and accountability. *computers & security*, 28(1-2), pp.8-17.

Braun, V. & Clarke, V. (2006) 'Using thematic analysis in psychology', *Qualitative Research in Psychology*, 3(2), pp. 77–101.

Braun, V., Clarke, V. & Terry, G. (2014) 'Thematic analysis', in Poul Rohleder, P. & Lyons, A. (eds.) *Qualitative research in clinical and health psychology*, London: Palgrave Macmillan, pp. 95–113. doi:10.1007/978-1-137-29105-9\_7.

Bresciani, P. *et al.* (2004) 'Tropos: An agent-oriented software development methodology', *Journal of Autonomous Agents and Multi-Agents Systems*, 8(3), pp. 203–236.

Brooks, S., Brooks, S., Garcia, M., Lefkovitz, N., Lightman, S. and Nadeau, E., 2017. *An introduction to privacy engineering and risk management in federal systems* (pp. 1-49). US Department of Commerce, National Institute of Standards and Technology.

Bryl, V. *et al.* (2006) 'Designing Security Requirements Models Through Planning', in Dubois, E. & Pohl, K. (eds.) 18th International Conference, CAiSE 2006. Heidelberg, Germany: Springer, pp. 33–47. Available at: <http://disi.unitn.it/~bryl/bryl-mass-mylozann-06-CAiSE.pdf> (Accessed: 28 December 2016).

Buede, D. M., "The Engineering Design of Systems: Models and Methods", 2nd ed., John Wiley & Sons, Inc., New Jersey, 2009.

Butler, S. A. (2002) 'Security attribute evaluation method: a cost-benefit approach', *24th International Conference on Software Engineering (ICSE '02)*. 2002, New York, NY: Association for Computing Machinery, pp. 232–249.

Butt, W.H., Amjad, S. and Azam, F., 2011, June. Requirement conflicts resolution: Using requirement filtering and analysis. In *International Conference on Computational Science and Its Applications* (pp. 383-397). Springer, Berlin, Heidelberg.

Camenisch, J. & van Herreweghen, E. (2002) 'Design and implementation of the idemix anonymous credential system', *Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY: Association for Computing Machinery, pp. 21–30. doi:<https://doi.org/10.1145/586110.586114>.

Castleberry A. & Nolen A. (2018) 'Thematic analysis of qualitative research data: Is it as easy as it sounds?', *Currents in Pharmacy Teaching and Learning*, 10(6), pp. 807–815.

Castro, J., Kolp, M. & Mylopoulos, J. (2002) 'Towards requirements-driven information systems engineering', *Information Systems*, 27, pp. 365–389.

Carson, R.S., 2015, October. Implementing structured requirements to improve requirements quality. In *INCOSE International Symposium* (Vol. 25, No. 1, pp. 54-67).

Cavoukian, A., 2012. Privacy by design: origins, meaning, and prospects for assuring privacy and trust in the information era. In *Privacy protection measures and technologies in business organizations: aspects and standards* (pp. 170-208). IGI Global.

Chen, L., Babar, M.A. and Nuseibeh, B., 2012. Characterizing architecturally significant requirements. *IEEE software*, 30(2), pp.38-45.

Choy, D. M. (2000) *Integrated method and system for controlling information access and distribution*, U.S. Patent No. 6,141,754.

Chung, L. (1993) 'Dealing with security requirements during the development of information systems', *5th International Conference on Advanced Information Systems Engineering, CAiSE 1993*. Paris, France, June 8–11. London: Springer-Verlag, pp. 234–251. doi.org/10.1007/3-540-56777-1\_13

Chung, L. & Supakkul, Sam. (2004). Representing NFRs and FRs: A Goal-Oriented and Use Case Driven Approach. 29-41. 10.1007/11668855\_3.

Clandinin, D. J. & Connelly, F. M. (2000) *Narrative inquiry: Experience and story in qualitative research*. San Francisco, CA: John Wiley & Sons.

Colesky, M., Hoepman, J.H. and Hillen, C., 2016, May. A critical analysis of privacy design strategies. In *2016 IEEE Security and Privacy Workshops (SPW)* (pp. 33-40). IEEE.

Creswell, J. W. & Creswell, D. J. (2017) *Research design: Qualitative, quantitative, and mixed methods approaches*. Los Angeles, CA: Sage Publications (Accessed: 13 May 2019).

Creswell, J., Research design: Qualitative, quantitative, and mixed methods approaches. *Management Review*, 14, pp.490-495.

Crotty, M. (1998) *The foundations of social research: Meaning and perspective in the research process*. London: Sage Publications.

Dalpia, F., Franch, X. and Horkoff, J., 2016. istar 2.0 language guide. *arXiv preprint arXiv:1605.07767*.

Deng, L. & Kuzmanovic, A. (2011) *Pseudonymous public keys based authentication*. Justia Patents. U.S. Patent Application No. 13/154,125, 2011. Available at: <https://patents.justia.com/patent/20110302412> (Accessed:14 January 2019).

Dey, A., Abowd, G. & Salber, D. (2001) 'A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications', *Human-Computer Interaction*, 16(2), pp. 97–166. Available at: [http://www.informaworld.com/openurl?genre=article&doi=10.1207/S15327051HCI16234\\_02&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3](http://www.informaworld.com/openurl?genre=article&doi=10.1207/S15327051HCI16234_02&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3) (Accessed: 10 December 2015).

Diamantopoulou, V. *et al.* (2017) 'Supporting the design of privacy-aware business processes via privacy process patterns,' *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, Brighton, 10–12 May. Brighton: IEEE Press, pp. 187–198, doi: 10.1109/RCIS.2017.7956536.

Diamantopoulou, V. & Mouratidis, H. (2018) 'Applying the physics of notation to the evaluation of a security and privacy requirements engineering methodology', *Information & Computer Security*, 26(4), pp. 382–400.

Diaz, C. & Preneel, B. (2004) 'Taxonomy of mixes and dummy traffic', in Deswarte Y. *et al.* (eds.) *Information Security Management, Education and Privacy*. IFIP International Federation for Information Processing, vol 148. Boston, MA: Springer, pp. 217–232.

Dittel, A., 2016. Data security requirements under the General Data Protection Regulation. *Journal of Data Protection & Privacy*, 1(1), pp.41-45.

Doan, T. *et al.* (2004) *UML design with security integration as first class citizen*. Storrs, CT: University of Connecticut. Available at: [https://www.academia.edu/16795103/UML\\_Design\\_with\\_Security\\_Integration\\_as\\_First\\_Class\\_Citizen](https://www.academia.edu/16795103/UML_Design_with_Security_Integration_as_First_Class_Citizen) (Accessed: 20 September 2016).

Domec, J.C., Lachenbruch, B., Meinzer, F.C., Woodruff, D.R., Warren, J.M. and McCulloh, K.A., 2008. Maximum height in a conifer is associated with conflicting requirements for xylem design. *Proceedings of the National Academy of Sciences*, 105(33), pp.12069-12074.

Drijvers, M. (2014) *Efficient delegation of Idemix credentials*. Master's Thesis. Radboud University, Nijmegen. Available at: <https://docplayer.net/89600460-Efficient-delegation-of-idemix-credentials.html> (Accessed: 2 February 2019)

- Dubrovsky, V.J., Kiesler, S. and Sethna, B.N., 1991. The equalization phenomenon: Status effects in computer-mediated and face-to-face decision-making groups. *Human-computer interaction*, 6(2), pp.119-146.
- Duncan, G. T. *et al.* (2001) Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies. Chapter 2.3. Disclosure Limitation Methods and Information Loss for Tabular Data.
- Dybå, T. *et al.* (2011) ‘Qualitative research in software engineering’, *Empirical Software Engineering*, 16(4), pp. 425-429.
- Edmunds, H., *The Focus Group Research Handbook*, NTC Business Books, Lincolnwood, IL, 1991.
- Egyed, A. & Boehm, B. (1998) ‘A comparison study in software requirements negotiation’, *Proceedings of the 8<sup>th</sup> annual international symposium on systems engineering*, INCOSE’98.
- Egyed, A. & Grünbacher, P. (2004) ‘Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help’, *Institute of Electrical and Electronics Engineers Software*, 21, pp. 50–58.
- Elahi, G. & Yu, E. (2007) ‘A goal oriented approach for modeling and analyzing security trade-offs’, in Parent, C. *et al.* (eds.) *Conceptual Modeling - ER 2007 Lecture Notes in Computer Science*, volume 4801 Heidelberg, Germany: Springer, pp. 375–390. Available at: [http://link.springer.com/10.1007/978-3-540-75563-0\\_26](http://link.springer.com/10.1007/978-3-540-75563-0_26) (Accessed: 12 September 2016).
- Eisenring, M., Thiele, L. and Zitzler, E., 2000. Conflicting criteria in embedded system design. *IEEE Design & Test of Computers*, 17(2), pp.51-59
- Espina, D. (2016) ‘How do you manage conflicting stakeholder demands?’, *Scope* <http://pm.stackexchange.com/questions/1399/how-do-youmanage-conflicting-stakeholder-demands> (Accessed: 22 June 2017 )
- Fabian, B. *et al.* (2010) ‘A comparison of security requirements engineering methods’, *Requirements Engineering*, 15(1), pp. 7–40.
- Faily, S. *et al.* (2015) ‘Usability and security by design: a case study in research and development.’ (2015).
- Federal Trade Commission (2000) *Fair information practice principles*.
- Firesmith, D. (2004) ‘Prioritizing requirements’, *Journal of Object Technology*, 3(8), pp. 35–48.
- France, R. B. *et al.* (2004) ‘A UML-based pattern specification technique’, *IEEE Transactions on Software Engineering*, 30(3), pp. 193–206.
- Frantzana, A. (2019) *Women’s representation and experiences in the high-performance computing community*. Doctoral Thesis. University of Edinburgh. Available at:

[https://www.researchgate.net/publication/335756173 Women's representation and experiences in the high performance computing community](https://www.researchgate.net/publication/335756173_Women's_representation_and_experiences_in_the_high_performance_computing_community) (Accessed:16 January 2020).

Freitas, H., Oliveira, M., Jenkins, M. and Popjoy, O., 1998. The Focus Group, a qualitative research method. *Journal of Education*, 1(1), pp.1-22.

Fridrich, J. (2013) *Steganography in digital media: Principles, algorithms, and applications*. Cambridge: Cambridge University Press.

Ganji, D., Mouratidis, H., Gheytaasi, S.M. and Petridis, M., 2015, September. Conflicts between security and privacy measures in software requirements engineering. In *International Conference on Global Security, Safety, and Sustainability* (pp. 323-334). Springer, Cham.

Ganji, D., 2019. *A model-driven framework to support analysis and implementation of information security management systems* (Doctoral dissertation, University of Brighton).

Gentry, C. (2010) 'Computing arbitrary functions of encrypted data', *Communications of the ACM*, 53(3), pp. 97–105.

Gharib, M., Giorgini, P. and Mylopoulos, J., 2017, November. Towards an ontology for privacy requirements via a systematic literature review. In *International conference on conceptual modeling* (pp. 193-208). Springer, Cham.

G. Hayes and K. El-Khatib, "Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol," 2013 Third International Conference on Communications and Information Technology (ICCIT), Beirut, Lebanon, 2013, pp. 179-184, doi: 10.1109/ICCITechnology.2013.6579545.

Giorgini, P., Massacci, F. & Zannone, N. (2005) 'Security and trust requirements engineering', in Aldini A., Gorrieri R. & Martinelli F. (eds.) *Foundations of security analysis and design III*, Heidelberg, Germany: Springer, pp. 237–272.

Gillin, D., 2000. The federal trade commission and Internet privacy. *Marketing Research*, 12(3), p.39

Gjermundrød, H., Dionysiou, I. and Costa, K., 2016, June. privacyTracker: a privacy-by-design GDPR-compliant framework with verifiable data traceability controls. In *International Conference on Web Engineering* (pp. 3-15). Springer, Cham.

Greer, D. & Bustard, D. W. (1997) 'SERUM-Software engineering risk: Understanding and management', *The International Journal of Project & Business Risk*, 1(4), pp. 373–388.

Haley, C. B. *et al.* (2008) 'Security requirements engineering: A framework for representation and analysis', *IEEE Transactions on Software Engineering*, 34 (1), pp. 133–153. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4359475> (Accessed: 12 March 2016).

Hall, T. & Flynn, V. (2001) 'Ethical issues in software engineering research: A survey of current practice', *Empirical Software Engineering*, 6, pp. 305–317. doi: <https://doi.org/10.1023/A:1011922615502>.

Hayes, G. & El-Khatib, K. (2013) 'Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol.' *2013 Third International Conference on Communications and Information Technology (ICCIT)*. Beirut: IEEE, pp. 179–184.

He, Q. & Antón, I. (2003) 'A framework for modelling privacy requirements in role engineering', *International Workshop on Requirements Engineering for Software Quality (REFSQ)*. 16–17 June 2003, Klagenfurt/Velden, Austria, pp. 115–124.

He, Q. & Antón, A. I. (2007) A Framework for Modeling Privacy Requirements in Role Engineering. [Online]. Available from: A Framework for Modeling Privacy Requirements in Role Engineering.

Heisel, M. and Souquieres, J., 2001. A heuristic algorithm to detect feature interactions in requirements. In *Language Constructs for Describing Features* (pp. 143-162). Springer, London.

Holloway, I. & Todres, L. (2003) 'The status of method: Flexibility, consistency and coherence', *Qualitative Research*, 3, pp. 345–357. doi:10.1177/1468794103033004.

Hong, J. I. & Landay, J. A. (2004) 'An architecture for privacy-sensitive ubiquitous computing', *Proceedings of the 2nd international conference on mobile systems, applications, and services – MobiSYS '04*. New York, NY, USA: Association for Computing Machinery Press, pp. 177–189. Available at: <http://portal.acm.org/citation.cfm?doid=990064.990087> (Accessed: 4 April 2016).

Hood, C., Wiedemann, S., Fichtinger, S. and Pautz, U., 2007. *Requirements management: The interface between requirements development and all other systems engineering processes*. Springer Science & Business Media.

Horkoff, J. *et al.* (2019) 'Goal-oriented requirements engineering: an extended systematic mapping study,' *Requirements Engineering*, 24(2), pp. 133–160.

IBM Security, (2019) Cost of a data breach report. Study conducted by the Ponemon Institute. 2019.

INCOSE, I., 2012. Guide for the application of Systems Engineering in Large Infrastructure Projects. *main contributors*.

Islam, S. *et al.* (2012) 'Model based process to support security and privacy requirements engineering', *International Journal of Secure Software Engineering*, 3(3), pp. 1–22.

Jaiswal, S. & Gupta, D. (2017) 'Security requirements for internet of things (IoT)', in Modi, N., Verma, P. & Trivedi, B. (eds.) *Proceedings of international conference on communication and networks*. Singapore: Springer, pp. 419–427.



Jannat, U.K., 2019. Identifying the Conflicts in the Software Requirement Engineering: A Literature Review.

Jensen, C. *et al.* (2005) *STRAP: A structured analysis framework for privacy*. Available at: <https://smartech.gatech.edu/bitstream/handle/1853/4450/05-02.pdf> (Accessed: 18 August 2016).

Jensen, C. & Potts, C. (2007) 'Experimental evaluation of a lightweight method for augmenting requirements analysis', in *Proceedings of the 1st ACM international workshop on empirical assessment of software engineering languages and technologies*. New York, NY: Association for Computing Machinery Press, pp. 49–54.

Jiang, X., Hong, J. I. & Landay, J. A. (2002) 'Approximate information flows: Socially-based modeling of privacy in ubiquitous computing', in Borriello G. & Holmquist L. E. (eds) *Proceedings of Ubicomp 2002*. Berlin: Springer, pp. 176–193.

Johanson, B., Fox, A. & Winograd, T. (2002) 'The interactive workspaces project: Experiences with ubiquitous computing rooms', *IEEE Pervasive Computing*, 1(2), pp. 67–74. Available at: <http://dl.acm.org/citation.cfm?id=612846> (Accessed: 7 March 2016).

Jurjens, J. (2004) *Secure systems development with UML*. Berlin, Germany: Springer Science & Business Media.

Kalloniatis, C. *et al.* (2004) 'Security requirements engineering for e-government applications: Analysis of current frameworks', in Traunmüller R. (ed.) *Electronic government*. 3183(1), pp. 66–71. Available at: [http://link.springer.com/10.1007/978-3-540-30078-6\\_11](http://link.springer.com/10.1007/978-3-540-30078-6_11) (Accessed: 14 December 2015).

Kalloniatis, C., Kavakli, E. & Gritzalis, S. (2008) 'Addressing privacy requirements in system design: the PriS method', *Requirements Engineering*, 13(3), pp. 241–255. Available at: <http://link.springer.com/10.1007/s00766-008-0067-3> (Accessed: 20 December 2015).

Kalloniatis, C., Kavakli, E. & Gritzalis, S. (2011) 'Designing privacy aware information systems', in Mouratidis, H. (ed.) *Software Engineering for Secure Systems: Industrial and Research Perspectives*. Pennsylvania, PA: IGI Global Publishing Company, pp. 212–231. Available at: <http://www.irma-international.org/viewtitle/48411/> (Accessed: 10 January 2016).

Kalloniatis, C., Kavakli, E. & Kontelis, E. (2010) 'Pris Tool: A case tool for privacy-oriented requirements engineering', *Journal of Information System Security*, 6(1), pp. 4–19.

Kalloniatis, C., Kavakli, E. and Gritzalis, S., 2009, September. Methods for designing privacy aware information systems: a review. In *2009 13th Panhellenic Conference on Informatics* (pp. 185-194). IEEE.

Kamberelis, G. and Dimitriadis, G., 2005. Collectively Remembering Tupac: The Narrative Mediation of Current Events, Cultural Histories, and Social Identities. *Afterlife as Afterimage: Understanding Posthumous Fame*, edited by Steve Jones and Joli Jensen, pp.143-70.

- Kalloniatis, C., Mouratidis, H. and Islam, S., 2013. Evaluating cloud deployment scenarios based on security and privacy requirements. *Requirements Engineering*, 18(4), pp.299-319.
- Kar, P. and Bailey, M., 1996, July. Requirements management working group: characteristics of good requirements. In *INCOSE International Symposium* (Vol. 6, No. 1, pp. 1225-1233).
- Katasonov, A. and Sakkinen, M., 2006. Requirements quality control: a unifying framework. *Requirements Engineering*, 11(1), pp.42-57
- Katz, N. & McNulty, K. (1994) *Conflict resolution*. Available at: <https://www.maxwell.syr.edu/uploadedFiles/parcc/cmc/Conflict%20Resolution%20NK.pdf>.80 (Accessed: 15 December 2018)
- Kaur, H. & Sharma, A. (2016) ‘A measure for modelling non-functional requirements using 155 extended use case’, *2016 3rd international conference on computing for sustainable global 156 development* (INDIACom), pp. 1101–1105.
- Kavakli, E. *et al.* (2006) ‘Incorporating privacy requirements into the system design process’, in Gritzalis, S. (ed.) *Internet Research*, 16(2), pp. 140–158. Available at: <http://www.emeraldinsight.com/doi/abs/10.1108/10662240610656483> (Accessed: 20 January 2018)
- Kim, M. *et al.* (2007) ‘Managing requirements conflicts in software product lines: A goal and scenario based approach’, *Data & Knowledge Engineering*, 61(3), pp. 417–432.
- King, N. (2004) ‘Using templates in the thematic analysis of text’, in Cassell, C. & Symon, G. (eds.) *Essential guide to qualitative methods in organizational research*. London, UK: Sage, pp. 257–270.
- Koorn, R. *et al.* (2004) *Privacy-enhancing technologies: White paper for decision-makers*, Technical report. Netherlands: Ministry of the Interior and Kingdom Relations. Available at: [https://is.muni.cz/el/1433/podzim2005/PV080/um/PrivacyEnhancingTechnologies\\_KPMGstudy.pdf](https://is.muni.cz/el/1433/podzim2005/PV080/um/PrivacyEnhancingTechnologies_KPMGstudy.pdf) (Accessed: 18 February 2016).
- Kontio, K., ABB, J.R., Koskinen, T. and HUT, M.N., 2003. Lightweight Contextual Design—a Case Study in Process Control Environment. *Human-Computer Interaction: Theory and Practice*, 1, p.138
- Koops, B.J. and Leenes, R., 2014. Privacy regulation cannot be hardcoded. A critical comment on the ‘privacy by design’ provision in data-protection law. *International Review of Law, Computers & Technology*, 28(2), pp.159-171.
- Kothari, C. R. (2004) *Research methodology: Methods and techniques*. New Delhi: New Age International Publishers.

Kousalya, P. *et al.* (2012) ‘Analytical hierarchy Process approach – An application of engineering education’, *Mathematica Aeterna*, 10, pp. 861–878.

Krueger, R.A., Casey, M.A., *Focus Groups: A Practical Guide for Applied Research*, Sage Publications, Thousand Oaks, CA, 2000.

Lam, T. C. *et al.* (2007) ‘Timed zero-knowledge proof (TZKP) protocol’, submitted to IEEE Real-Time and Embedded Technology and Application Symposium.

Langford J. & McDonough D, 2003, *Focus groups: Supporting effective product development*, Taylor and Francis, New York.

Leydesdorff, L. (2010) ‘Redundancy in systems which entertain a model of themselves: Interaction information and the self-organization of anticipation’, *Entropy*, 12(1), pp. 63–79.

Liu, L., Yu, E. & Mylopoulos, J. (2002) ‘Analyzing security requirements as relationships among strategic actors’, *2nd Symposium on Requirements Engineering for Information Security (SREIS’02)*. Raleigh, North Carolina: Institute of Electrical and Electronics Engineers, pp. 1–14. Available at: <http://www.cs.toronto.edu/pub/eric/SREIS02-Sec.pdf> (Accessed: 11 March 2016).

Liu, L., Yu, E. & Mylopoulos, J. (2003) ‘Security and privacy requirements analysis within a social setting’, *Proceedings 11th IEEE International Requirements Engineering Conference 2003*. Washington, DC: Institute of Electrical and Electronics Engineers, pp. 151–161. Available at: <http://www.cs.toronto.edu/pub/eric/RE03.pdf> (Accessed: 11 March 2016).

Liu, X.F. and Yen, J., 1996, March. An analytic framework for specifying and analyzing imprecise requirements. In *Proceedings of IEEE 18th International Conference on Software Engineering* (pp. 60-69). IEEE.

Lopez, J., Oppliger, R. & Pernul, G. (2004) ‘Authentication and authorization infrastructures (AAIs): A comparative survey’. *Computers & Security*, 23(7), pp. 578–590.

Mairiza, D. & Zowghi, D. (2010) ‘An ontological framework to manage the relative conflicts between security and usability requirements’, *2010 Third international workshop on managing requirements knowledge (MARK)*. New York, NY: Institute of Electrical and Electronics Engineers.

Mairiza, D. & Zowghi, D. (2010) ‘Constructing a catalogue of conflicts among non-functional requirements’, *International conference on evaluation of novel approaches to software engineering*. Berlin, Germany: Springer.

Mairiza, D., Zowghi, D. & Gervasi, V. (2013) ‘Conflict characterization and analysis of non functional requirements: An experimental approach’, *IEEE 12th International conference on intelligent software methodologies, tools and techniques (SoMeT)* Budapest: Institute of Electrical and Electronics Engineers, pp. 83–91.

- Mairiza, D., Zowghi, D. & Gervasi, V. (2014) 'Utilizing TOPSIS: A multi criteria decision analysis technique for non-functional requirements conflicts', *Communications in Computer and Information Science*, 432, pp. 31–44.
- Mairiza, D., Zowghi, D. & Nurmuliani, N. (2009) 'Managing conflicts among non-functional requirements', *Australian Workshop on Requirements Engineering*. Sydney, Australia: University of Technology.
- Massacci, A. *et al.* (2008) 'Response of the photosynthetic apparatus of cotton (*Gossypium hirsutum*) to the onset of drought stress under field conditions studied by gas-exchange analysis and chlorophyll fluorescence imaging'. *Plant Physiology and Biochemistry*, 46(2), pp. 189–195. doi: 10.1016/j.plaphy.2007.10.006.
- Massacci, F., Prest, M. & Zannone, N. (2005) 'Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation'. *Computer Standards & Interfaces*, 27(5), pp. 445–455.
- Massacci, F. and Zannone, N., 2008. Detecting Conflicts between Functional and Security Requirements with Secure Tropos: John Rusnak and the Allied Irish Bank. *Social modeling for requirements engineering*. MIT Press, Cambridge.
- Matsumoto, Y., Shirai, S. & Ohnishi, A. (2017) 'A method for verifying non-functional requirements'. *Procedia Computer Science*, 112, pp. 157–166.
- Matyas, V. and Kur, J., 2013. Conflicts between intrusion detection and privacy mechanisms for wireless sensor networks. *IEEE Security & Privacy*, 11(5), pp.73-76.
- Maxwell J. C., Antón, A. I. & Swire, P. (2011) 'A legal cross-references taxonomy for identifying conflicting 160 software requirements', *2011 IEEE 19th international requirements engineering conference*, 161, pp. 197–206.
- Mayer, N. *et al.* (2008) 'Adapting secure Tropos for security risk management in the early phases of information systems development, in Bellasèhne, Z. & Léonard, M. (eds.) *Advanced Information Systems Engineering*. Berlin, Germany: Springer, pp. 541–555.
- Mead, N. R. & Stehney, T. (2005) 'Security quality requirements engineering (SQUARE) methodology'. *ACM SIGSOFT Software Engineering Notes*, 30(4), pp. 1–7.
- Mellado, D., Mouratidis, H. and Fernández-Medina, E., 2014. Secure Tropos framework for software product lines requirements engineering. *Computer Standards & Interfaces*, 36(4), pp.711-722.
- Menezes, A. J., van Oorschot, P. C. & Vanstone, S. A. (2018) *Handbook of applied cryptography*. Boca Ratan, FL: CRC Press.
- Merton, R. K. (1975) 'Thematic analysis in science: Notes on Holton's concept'. *Science*, 188(4186), pp. 335–338.
- Miles, M. B. & Huberman, A. M. (1994) *Qualitative data analysis. An expanded sourcebook*. London: Sage Publications.

Moffett, J. D. & Nuseibeh, B. A. (2003) *A framework for security requirements engineering*. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.607> (Accessed: 7 March 2016).

Mouratidis, H., Argyropoulos, N. and Shei, S., 2016. Security requirements engineering for cloud computing: The secure tropos approach. In *Domain-specific conceptual modeling* (pp. 357-380). Springer, Cham.

Mouraditis, H. (2004) *A security-oriented approach in the development of multi-agent systems: Applies to the management of the health and social care needs of older people in England*. PhD thesis. University of Sheffield.

Mouratidis, H. *et al.* (2013) 'A framework to support selection of cloud providers based on security and privacy requirements'. *Journal of Systems and Software*, 86(9), pp. 2276–2293.

Mouratidis, H. & Giorgini, P. (2007) 'Secure Tropos: A security-oriented extension of the Tropos methodology'. *International Journal of Software Engineering and Knowledge Engineering*, 17(02), pp. 285–309. Available at: <http://www.worldscientific.com/doi/abs/10.1142/S0218194007003240> (Accessed: 10 February 2016).

Mouratidis, H., Giorgini, P. & Manson, G. (2003a) 'An ontology for modelling security: the Tropos project, in Palade, V., Howlett, R. & Jain, L. (eds.) *Proceedings of the KES 2003 Invited Session Ontology and Multi-Agent Systems Design (OMASD'03)*. University of Oxford, UK: Springer, pp. 1387–1394.

Mouratidis, H., Giorgini, P. & Manson, G. (2003b) 'Integrating security and systems engineering: Towards the modelling of secure information systems', *15<sup>th</sup> international conference CAiSE '03, LNCS 2681*. Klagenfurt, Austria: Springer, pp. 63–78.

Mylopoulos, J., Chung, L. & Nixon, B. (1992) 'Representing and using nonfunctional requirements: A process-oriented approach'. *Institute of Electrical and Electronics Engineers Transactions on Software Engineering*, 18(6), pp. 483–497.

Myers, M., *Qualitative Research in Information Systems*, <http://www.qual.auckland.ac.nz/>, 2004. Nambisan, S., *Information Systems as a Reference Discipline for New Product Development*, *MIS Quarterly*, 27(1):1–18, 2003.

Nissenbaum, H. (2004) 'Privacy as contextual integrity'. *Washington Law Review*, 79(1), pp. 119–157.

Nithya, V. & Subha, R. (2013) 'Privacy requirement engineering based on modified evidence combination approach'. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(2), pp. 1–6.

Nowell, L. S. *et al.* (2017) ‘Thematic analysis: Striving to meet the trustworthiness criteria’. *International Journal of Qualitative Methods*, 16, pp. 1–13. doi:10.1177/1609406917733847.

Nuseibeh, B. & Easterbrook, S. (2000) ‘Requirements engineering: A roadmap’, *Proceedings of the conference on the future of Software engineering - ICSE '00*. New York, NY: Association for Computing Machinery Press, pp. 35–46. Available at: <http://portal.acm.org/citation.cfm?doid=336512.336523> (Accessed: 15 April 2016).

OASIS (2005) *Security and privacy considerations for the OASIS security assertion markup language (SAML) V2.0*. Available at: <https://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>. (Accessed: 3 December 2015).

Olbrechts, A. (2019) *Guidelines 4/2019 on Article 25 Data Protection by Design and by Default* [online] available from <[https://edpb.europa.eu/our-work-tools/public-consultations-art-704/2019/guidelines-42019-article-25-data-protection-design\\_en](https://edpb.europa.eu/our-work-tools/public-consultations-art-704/2019/guidelines-42019-article-25-data-protection-design_en)> [11 January 2020]

O. Nyumba, T., Wilson, K., Derrick, C.J. and Mukherjee, N., 2018. The use of focus group discussion methodology: Insights from two decades of application in conservation. *Methods in Ecology and evolution*, 9(1), pp.20-32.

Paja, E., Dalpiaz, F. & Giorgini, P. (2013) ‘Managing security requirements conflicts in socio-technical systems’, in Ng W., Storey V. C. & Trujillo J. C. (eds.) *Conceptual Modeling*, Lecture Notes in Computer Science, vol 8217. Heidelberg, Germany: Springer, pp. 270–283.

Pan, Y. (2012) *Comparison of i\*-based and use case-based security modelling initiatives for software requirements engineering: An empirical comparison of secure Tropos and misuse cases*. Master’s Thesis. Norwegian University of Science and Technology.

Pan, J.Z., Staab, S., Aßmann, U., Ebert, J. and Zhao, Y. eds., 2012. *Ontology-driven software development*. Springer Science & Business Media.

Pandey, D., Suman, U. & Ramani, A. K. (2011) ‘Security requirement engineering issues in risk management’. *International Journal of Computer Applications*, 17(5), pp. 11–14.

Panusuwan, V., Batlagundu, P. & Mead, N. (2009) *Privacy risk assessment case studies in support of SQUARE*. Available at: <http://www.sei.cmu.edu/reports/09sr017.pdf> (Accessed: 12 April 2016).

Partnership Against Cancer (2014) CPAC Privacy and Security Framework. Available at: <http://www.partnershipagainstcancer.ca/wp-content/uploads/sites/5/2015/03/Privacy-and-Security-Framework-Overview.pdf> (Accessed: 7 March 2016).

Pasquale, L. *et al.* (2016) ‘Automating trade-off analysis of security requirements’. *Requirements Engineering*, 21(4), pp. 481–504.

Pavlidis, M. & Islam, S. (2011) 'SecTro: A CASE tool for modelling security in requirements engineering using secure Tropos'. *CEUR Workshop Proceedings*, 734, pp. 89–96.

Phillips, D. C. & Burbules, N. C. (2000) *Postpositivism and educational research*. Lanham, MD: Rowman & Littlefield.

Piras, L. *et al.* (2019) 'DEFEND architecture: A privacy by design platform for GDPR compliance', in Gritzalis S. *et al.* (eds.) *Trust, privacy and security in digital business*. Proceedings of 16th international conference, TrustBus 2019. Cham, Switzerland: Springer, pp. 78–93.

Piras, L. *et al.* (2020) 'DEFEND DSM: A data scope management service for model-based privacy by design GDPR compliance', in Gritzalis S. *et al.* (eds.) *Trust, privacy and security in digital business*. Lecture Notes in Computer Science, vol 11711. Cham, Switzerland: Springer, pp. 186–201.

Poort, E.R. and de With, P.H.N., 2004, June. Resolving requirement conflicts through non-functional decomposition. In *Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004)* (pp. 145-154). IEEE.

Ramadan, Q. (2020) *Data protection assurance by design: Support for conflict detection, requirements traceability and fairness analysis*. PhD Thesis. University of Koblenz and Landau.

Ramadan, Q. *et al.* (2018) 'Detecting conflicts between data-minimization and security requirements in business process models, in Pierantonio, A. & Trujillo, S. (eds.) *European Conference on Modelling Foundations and Applications*. Lecture Notes in Computer Science, vol 10890. Cham, Switzerland: Springer, pp. 179–198.

Ramadan, Q. *et al.* (2020) 'A semi-automated BPMN-based framework for detecting conflicts between security, data-minimization and fairness requirements'. *Software and Systems Modeling*, 19, pp.1191–1227. doi:10.1007/s10270-020-00781-x.

Ray, I. *et al.* (2004) 'Using UML to visualize role-based access control constraints', *Proceedings of the ninth ACM symposium on access control models and technologies*. New York, NY: Association for Computing Machinery Press, pp. 115–124.

Regnell, B., Berntsson Svensson, R. & Olsson, T. (2008) 'Supporting roadmapping of quality requirements'. *Institute of Electrical and Electronics Engineers Software*, 25(2), pp. 42–47.

Robinson, W. N. (2004) 'Surfacing requirements interactions', in do Prado Leite, J. C. S. & Doorn, J. H. (eds.) *Perspectives on software requirements*. Boston, MA: Springer, pp. 69–90.

Robertson, S. and Robertson, J., 2012. *Mastering the requirements process: Getting requirements right*. Addison-wesley.

- Rubinstein, S. M. *et al.* (2012) ‘Spinal manipulative therapy for acute low-back pain’. *The Cochrane Database of Systematic Reviews*, 2012(9), CD008880. Available at: <https://doi.org/10.1002/14651858.CD008880.pub2> (Accessed:23 February 2019).
- Runeson, P. & Höst, M. (2009) ‘Guidelines for conducting and reporting case study research in software engineering’. *Empirical Software Engineering*, 14(2), p.131.
- Roman, R., Zhou, J. and Lopez, J., 2013. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10), pp.2266-2279.
- Ryan, G. W. & Bernard, H. R. (2000) ‘Data management and analysis methods’, in Denzin, N. K. & Lincoln, Y. S. (eds.) *Handbook of qualitative research*. 2nd edn. Thousand Oaks, CA: Sage, pp. 769–802.
- Sadana V. & Liu X. F. (2007) ‘Analysis of conflict among non-functional requirements using integrated analysis of functional and non-functional requirements’, *Proceedings of the 31st annual international computer software and applications conference (COMPSAC 2007)*. Institute of Electrical and Electronics Engineers.
- Salado, A. & Nilchiani, R. (2014) ‘The concept of order of conflict in requirements engineering’. *Institute of Electrical and Electronics Engineers Systems Journal*, 10(1), pp. 25–35.
- Salnitri, M. *et al.* (2020) ‘Modelling the interplay of security, privacy and trust in sociotechnical systems: A computer-aided design approach’. *Software and Systems Modeling*, 19(2), pp. 467–491.
- Sarks, H. & Trinidad, S. B. (2007) ‘Choose your method: A comparison of phenomenology, discourse analysis, and grounded theory’. *Qualitative Health Research*, 17, pp. 1372–1380. doi:10.1177/1049732307307031.
- Schär, B. (2015) *Requirements engineering process: HERMES 5 and SCRUM*. Master’s Thesis. University of Applied Sciences and Arts.
- Schon, E. -M., Thomaschewski, J. & Escalona, M. J. (2017) ‘Agile requirements engineering: A systematic literature review’. *Computer Standards and Interfaces*, 49, pp. 79–91.
- Sepúlveda, M. A. *et al.* (2014) ‘Domestic dogs in rural communities around protected areas: Conservation problem or conflict solution?’. *PLOS ONE*, 9(1), e86152, Available at: <http://dx.doi.org/10.1371/journal.pone.0086152> (Accessed:08 April 2017).
- Shapiro, S. S. (2012) ‘Situating anonymization within a privacy risk model’, *2012 IEEE International Systems Conference*. Vancouver, BC: Institute of Electrical and Electronics Engineers, pp. 1–6.
- Shei, S., Delaney, A.J., Kapetanakis, S. and Mouratidis, H., 2015. Visually Mapping Requirements Models to Cloud Services. In *DMS* (pp. 108-114).



Sieber J. E. (1993) 'Ethical considerations in planning and conducting research on human subjects', *Academic Medicine*, 68(9), pp. S9–13. [Ethical\\_considerations\\_in\\_planning\\_and\\_conducting.27.pdf](#) (Accessed: 20 November 2018)

Sindre, G. & Opdahl, A. L. (2005) 'Eliciting security requirements with misuse cases'. *Requirements Engineering*, 10(1), pp. 34–44.

Skou, N., 2003, October. Microwave remote sensing: Needs and requirements concerning technology. In *33rd European Microwave Conference Proceedings (IEEE Cat. No. 03EX723C)* (Vol. 2, pp. 863-866). IEEE.

Smith, S., Beaulieu, A. & Phillips, W. G. (2011) 'Modeling and verifying security protocols using UML 2', *2011 IEEE International Systems Conference*. Montreal, QC: Institute of Electrical and Electronics Engineers, pp. 72–79.

Sommerville, I., 2011. Software engineering 9th Edition. *ISBN-10, 137035152*, p.18.

Stewart, D.W., Shamdasani, P.N. and Rook, D.W. 2007. The Focus Group Moderator. In: *Focus Groups, Applied Social Research Methods*. 2nd ed. Thousand Oaks, CA: SAGE Publications, Ltd. pp. 69-87. Available at: <http://www.doi.org/10.4135/9781412991841> [Accessed 12 April 2020]

Syverson, P. *et al.* (2001) 'Towards an analysis of onion routing security', in Federrath, H. (ed.) *Designing privacy enhancing technologies*. Berlin, Germany: Springer, pp. 96–114.

Tange, K. P. *et al.* (2020) 'A systematic survey of industrial internet of things security: Requirements and fog computing opportunities'. *Institute of Electrical and Electronics Engineers Communications Surveys & Tutorials*, 22(4), pp. 2489–2520.

Tellis, W. (1997) 'Introduction to case study', *The Qualitative Report*, 3(2), pp. 1–14. Available at: <http://www.nova.edu/ssss/QR/QR3-2/tellis1.html> (Accessed: 12 March 2018).

Tellis, W., 1997. Application of a case study methodology. *The qualitative report*, 3(3), pp.1-19.

Tremblay, M.C., Hevner, A.R. and Berndt, D.J., 2010. Focus groups for artifact refinement and evaluation in design research. *Communications of the association for information systems*, 26(1), p.27.

Tsohou, A. *et al.* (2020) 'Privacy, security, legal and technology acceptance elicited and consolidated requirements for a GDPR compliance platform', *Information & Computer Security*, 28(4), pp. 531–553. doi:10.1108/ICS-01-2020-0002.

University of Brighton (2019) *New software to boost data protection and privacy*. Available at: <https://www.brighton.ac.uk/about-us/news-and-events/news/2019/01-29-new-software-to-boost-data-protection-and-privacy.aspx> (Accessed 18 April 2019).

van Lamsweerde, A. (2004) 'Elaborating security requirements by construction of intentional anti-models', *Proceedings of the 26th international conference on software*

*engineering (ICSE'04)*. Washington, DC: Institution of Electrical and Electronics Engineers Computer Society, pp. 148–157.

van Lamsweerde, A., Darimont, R. & Letier, E. (1998) 'Managing conflicts in goal-driven requirements engineering', *IEEE Transactions on Software Engineering*, (24)11, pp. 908–926.

van Lamsweerde, A. & Letier, E. (2000) 'Handling obstacles in goal-oriented requirements engineering', *IEEE Transactions on Software Engineering*, 26(10), pp. 978–1005. doi:10.1109/32.879820.

Van Lamsweerde, A. and Letier, E., 2000. Handling obstacles in goal-oriented requirements engineering. *IEEE Transactions on software engineering*, 26(10), pp.978-1005.

Vartiainen, T., 2008, January. Student life in computing: A variety of conflicting moral requirements. In *Proceedings of the tenth conference on Australasian computing education-Volume 78* (pp. 163-169).

Ven, A. & Delbecq, A. (1972) 'The nominal group as a research instrument for exploratory health studies', *American Journal of Public Health*, 62, pp. 337–342.

Verizon (2020) *2020 Data Breach Investigations Report*. Available at: <https://enterprise.verizon.com/en-gb/resources/reports/dbir/> (Accessed: 18 December 2020).

Vestola, M. (2010) *A comparison of nine basic techniques for requirements prioritization*. Espoo, Finland: Helsinki University of Technology.

Vinson, N. G. & Singer, J. (2008) 'A practical guide to ethical research involving humans', in Shull, F., Singer, J. & Sjøberg, D. I. K. (eds.) *Guide to advanced empirical software engineering*. London: Springer, pp. 229–256. [https://doi.org/10.1007/978-1-84800-044-5\\_9](https://doi.org/10.1007/978-1-84800-044-5_9).

Voigt, P. & Von dem Bussche, A. (2017) *The EU general data protection regulation (GDPR). A practical guide*. Cham, Switzerland: Springer International Publishing.

von Rosing, M., White, S., Cummins, F. and de Man, H., 2015. Business Process Model and Notation-BPMN.

Walden, D.D., Roedler, G.J. and Forsberg, K., 2015, October. INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners. In *INCOSE International Symposium* (Vol. 25, No. 1, pp. 678-686).

Yahuza, M. *et al.* (2020) 'Systematic review on security and privacy requirements in edge computing: State of the art and future research opportunities', *Institute of Electrical and Electronics Engineers Access*, 8, pp. 76541–76567.

Yan, K. *et al.* (2018) 'Steganography security: Principle and practice', *Institute of Electrical and Electronics Engineers Access*, 6, pp. 73009–73022.

Yin, R. K. (1981) 'Life stories of innovations: How new practices become routinized', *Public Administration Review*, (41)1, pp. 21–28.

Yin, R.K., 1981. The case study as a serious research strategy. *Knowledge*, 3(1), pp.97-114.

Yu, E. & Cysneiros, L. M. (2002) 'Designing for privacy and other competing requirements', *Proceedings of the 3<sup>rd</sup> symposium on requirements engineering for information security*. West Lafayette, IN: Purdue University, Article 5, pp. 1–15. Available at: <http://ftp.cs.toronto.edu/pub/eric/SREIS02-Priv.pdf> (Accessed: 15 October 2016).

Yu, E. & Liu, L. (2001) 'Modelling trust for system design using the i\* strategic actors framework', in Falcone, R. Singh, M. & Tan, Y. H. (eds.) *Trust in cyber-societies – Integrating the human and artificial perspectives*. Berlin, Germany: Springer, pp. 175–194. Available at: [http://link.springer.com/10.1007/3-540-45547-7\\_11](http://link.springer.com/10.1007/3-540-45547-7_11) (Accessed: 15 November 2016).

Yu, Eric SK. "Towards modelling and reasoning support for early-phase requirements engineering." *Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering*. IEEE, 1997.

Yu, E., 2011. Modeling Strategic Relationships for Process Reengineering. *Social Modeling for Requirements Engineering*, 11(2011), pp.66-87.

Yu, E. and Mylopoulos, J., 1998, June. Why goal-oriented requirements engineering. In *Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality* (Vol. 15, pp. 15-22).

Zainal, Z. (2007) 'Case study as a research method', *Jurnal Kemanusiaan*, 5(1).

Zave, P. (1997) 'Classification of research efforts in requirements engineering', *Association for Computing Machinery Computing Surveys*, 29(4), pp. 315–321.

Zhou, J., Cao, Z., Dong, X. and Vasilakos, A.V., 2017. Security and privacy for cloud-based IoT: Challenges. *IEEE Communications Magazine*, 55(1), pp.26-33.

# APPENDICES

## Appendix A: List of requirements for E-Health scenario

	REQ ID	DEFEND requirements	PRIORITY
A	REQ09.24	The DEFEND Platform shall support the creation of the record of processing activities when the organisation acts as data controller	Must
	REQ09.25	The DEFEND Platform shall support the creation of the record of processing activities when the organisation acts as data processor	Must
	REQ09.06	The DEFEND Platform shall provide a mechanism that supports the data mapping to the corresponding assets and services of the organisation i.e. the DEFEND platform shall map the processing activities, as well as in which database of the organisation personal data are stored, and the kind of personal data, and the data flows	Must
	REQ05.08	The DEFEND Platform shall allow an organisation to ensure that the organisation implements, per processing activity, appropriate technical and organisational measures which ensure that, by default, only personal data which are necessary for each specific purpose of processing are processed	Must
B	REQ12.01	The DEFEND Platform shall allow and support the organisation in the identification of (technical and organisational) security measures needed for the protection of personal data	Must
	REQ12.02	The DEFEND Platform shall enforce multiple checkpoints to validate protection of personal data against relevant threats	Must
	REQ02.12	The DEFEND Platform shall provide a mechanism to support the identification and recording of third parties appointed in relation to the monitoring/auditing of risks	Should
	REQ02.03	The DEFEND Platform shall provide, in one centralised part of the system, the management of relations with third parties, such as joint-controller, controller-in-common, controller-processor, processor-controller relations	Must
C	REQ05.07	The DEFEND Platform shall allow an organisation to ensure that the organisation implements, per processing activity, appropriate technical and organisational measures which are designed to implement the principle of 'integrity and confidentiality'	Must
	REQ10.01	The DEFEND Platform shall provide to the authorised user roles and upon request the following information: - Data controller identification, - Degree of completion of individual Data Controller DPIA (Compliance Assessment Data Protection)	Must
	REQ10.11	The DEFEND Platform shall provide mechanisms to monitor the progress of individual DPIAs	Should
D	REQ09.04	The DEFEND Platform shall allow for graphical representation of specific relationships between third-parties (e.g. joint controller) and the organisation	Should

	<b>REQ09.15</b>	The DEFeND Platform shall provide a mechanism to create and monitor a process defining who within the organisation is authorised to make changes to personal data and who is responsible for validation of such changes	Must
	<b>REQ02.03</b>	The DEFeND Platform shall provide, in one centralised part of the system, the management of relations with third parties, such as joint-controller, controller-in-common, controller-processor, processor-controller relations	Must
	<b>REQ02.04</b>	The DEFeND Platform shall provide a mechanism to create, store, update and access the identification and relationship type of the third parties with which the organisation has a relationship (such as joint-controller, controller-in-common, controller-processor, processor-controller relations)	Should
	<b>REQ02.06</b>	The DEFeND Platform shall provide a mechanism to graphically visualise any relationships the organisation has with third parties, such as data processors, joint controllers or other data controllers	Should
	<b>REQ02.11</b>	The DEFeND Platform shall provide a mechanism to support the identification and recording of third parties appointed in relation to the implementation of security measures	Should
<b>E</b>	<b>REQ05.07</b>	The DEFeND Platform shall allow an organisation to ensure that the organisation implements, per processing activity, appropriate technical and organisational measures which are designed to implement the principle of 'integrity and confidentiality'	Must
	<b>REQ12.01</b>	The DEFeND Platform shall allow and support the organisation in the identification of (technical and organisational) security measures needed for the protection of personal data	Must
	<b>REQ12.02</b>	The DEFeND Platform shall enforce multiple checkpoints to validate protection of personal data against relevant threats	Must
<b>F</b>	<b>REQ12.01</b>	The DEFeND Platform shall allow and support the organisation in the identification of (technical and organisational) security measures needed for the protection of personal data	Must
	<b>REQ12.02</b>	The DEFeND Platform shall enforce multiple checkpoints to validate protection of personal data against relevant threats	Must
	<b>REQ12.06</b>	The DEFeND Platform shall allow an organisation to map the technical and organisational measures implemented to establish immediately whether a personal data breach has taken place and generate reports documenting decisions, and related tool-supported evaluation/verification	Must
<b>G</b>	<b>REQ10.05</b>	The DEFeND Platform shall provide tools to enable an organisation to carry out a data protection impact assessment in the cases described in Leg.REQ07.01. The Data Protection Impact Assessment shall contain at least the information listed in Article 35(7) of the GDPR	Must
	<b>REQ05.08</b>	The DEFeND Platform shall allow an organisation to ensure that the organisation implements, per processing activity, appropriate technical and organisational measures which ensure that, by default, only personal data which are necessary for each specific purpose of the processing are processed	Must
	<b>REQ05.09</b>	The DEFeND Platform shall allow an organisation to ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons	Must

<b>H</b>	<b>REQ09.02</b>	The DEFeND Platform shall gather input and information about the organisation based on a questionnaire	Must
<b>I</b>	<b>REQ09.01</b>	The DEFeND Platform shall implement a mechanism to discover / create, store, update, access and graphically present, a model of the organisation's structure	Must
	<b>REQ09.03</b>	The DEFeND Platform shall allow for graphical representation of multiple departmental areas of the organisation (e.g. divisions, departments) and the relationships between entities in the organisation	Should
	<b>REQ09.04</b>	The DEFeND Platform shall allow for graphical representation of specific relationships between third-parties (e.g. joint controller) and the organisation	Should
<b>J</b>	<b>REQ09.01</b>	The DEFeND Platform shall implement a mechanism to discover / create, store, update, access and graphically present, a model of the organisation's structure	Must
	<b>REQ09.06</b>	The DEFeND Platform shall provide a mechanism that supports data mapping to the corresponding assets and services of the organisation i.e. the DEFeND Platform shall map the processing activities, as well as in which database of the organisation personal data are stored, and the kind of personal data, and the data flows	Must
	<b>REQ12.06</b>	The DEFeND Platform shall allow an organisation to map the technical and organisational measures implemented to establish immediately whether a personal data breach has taken place and generate reports documenting decisions, and related tool-supported evaluation/verification	Must
	<b>REQ02.02</b>	The DEFeND Platform shall provide to joint controllers information regarding privacy issues in relation to the data subjects for which the company using the DEFeND platform and another company are jointly determining the means and purposes of data processing	Must
<b>K</b>	<b>REQ12.06</b>	The DEFeND Platform shall allow an organisation to map the technical and organisational measures implemented to establish immediately whether a personal data breach has taken place and generate reports documenting decisions, and related tool-supported evaluation/verification	Must
	<b>REQ10.11</b>	The DEFeND Platform shall provide mechanisms to monitor the progress of individual DPIAs	Should
	<b>REQ10.06</b>	The DEFeND Platform shall provide tools to an organisation to identify those types of processing which are likely to result in a high risk to the rights and freedoms of natural persons, taking into account the nature, scope, context and purposes of the processing and public lists of the supervisory authority competent for that organisation	Must
<b>L</b>	<b>REQ10.02</b>	The DEFeND Platform shall allow to authorised users roles and upon request to assign and revoke DPIA to the responsible person within the organisation	Must
<b>M</b>	<b>REQ10.06</b>	The DEFeND Platform shall provide tools to an organisation to identify those types of processing which are likely to result in a high risk to the rights and freedoms of natural persons, taking into account the nature, scope, context and purposes of the processing and public lists of the supervisory authority competent for that organisation	Must
	<b>REQ10.11</b>	The DEFeND Platform shall provide mechanisms to monitor the progress of individual DPIAs	Should

## Appendix B: Ethical Approval



**University of Brighton**

Life, Health and Physical Sciences CREC

424 Watts Building  
Lewes Road  
Brighton  
BN2 4GJ

23/06/2020

**Ref:** 2020-5857-Aikubaisy A Framework Managing Conflicts between Security and Privacy Requirements- Evaluating phase

Dear Duaa

Thank you for your resubmission to the Life, Health and Physical Sciences CREC at the University of Brighton.

The committee are happy to offer a favourable ethical opinion for this study.

Favourable ethical opinion is given on the basis of a project end date of 30/09/2020. If you need to request an extension, please complete a change request form. Please note that the decisions of the committee are made on the basis of the information provided in your application. The CREC must be informed of any changes to the research process after a favourable ethical opinion has been given. Research that is conducted without having been reviewed by the committee is not covered by the University research insurance cover. If you need to make changes to your proposal please complete and submit a change request form in order that the CREC can determine whether the changes will necessitate any further ethical review.

Once your research has been completed, please could you fill in a brief end of project report form. Finally please could I ask that you flag up any unexpected ethical issues, and report immediately any serious adverse events that arise during the conduct of this study.

We wish you all the best with your research and hope that your research study is successful. If the CREC can be of further assistance with your study please contact us again.

Best wishes

Lucy Redhead

Chair, Life, Health and Physical Sciences CREC

**Appendix C: Evaluation form**

**EVALUATION FORM**

**Date:** \_\_\_\_\_

**Please Select Your Status:** PhD Student  PhD doctor  Research fellow

To what extent do you agree or disagree with the following?

*Presentation Content:*

CRITERIA	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE
Research field is interesting					
Background is clearly presented					
Methods is clearly presented					
Results is clearly presented					
Useful handout					
Questions were appropriate					
Method appropriate for answering research question					

*Phase 1: Mapping Security and Privacy Requirements:*

CRITERIA	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE
The list of requirements covers the most common security and privacy requirements					
well identified related between security and privacy					
Mapping between security and privacy to identify conflict was clear					
Mapping between security and privacy to identify conflict is easy to follow very detailed steps					



*Phase 2: Identify Conflicts between Requirements and Conflict Decisions:*

CRITERIA	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE
The researcher addressed the potential conflicting aspects following phase 1 helps the analysis to decide which requirements are conflicting					

*Phase 3: Conflict Resolution Patterns:*

CRITERIA	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE
Presenting the supporting tools in tables is easy to use/clear					
Modelling the supporting tools is easy to use/clear					
If we link the supporting tool in the privacy pattern library, it will make development process faster					
If we use the supporting tool, then the analysis can solve, reduce conflicts between requirements					

*Framework in general*

CRITERIA	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE
The framework phases are clear and well defined					
The framework phases are sequentially in order					
If we follow the framework phases, then we can have a fast development process					
If we follow the framework phases, it can be easier to the					

analysis to detect and identify conflict	
If we follow the framework phases, it can be easier to the analysis to decide a suitable tool to reduce conflict	
If we follow the framework phases, we can reduce cost of dealing with conflict at late stage requirements level or implementation level	
If the analyst used the framework phases, we could maintain the value of each requirement	

*Please enter any comments or suggestions you have in the text box below.*

*Thank you very much for participating with us.*

*Duaa Alkubaisy*

## **Appendix D: Toolkit for the Focus Group session**

### **Supporting Document**

This document is a supporting sheet for presentation of the evaluation, while the researcher present the framework and apply it to scenario, participants can follow up with this supporting documents that contain material of requirements and screenshots of applying the tool in SecTro. The session will contain discussion and sharing ideas about the framework. By the end of this document, participant must fill up the evaluation form and send it back to the researcher to have a complete evaluation session.

#### **This document contains:**

- Inputs and outputs for each phase of the framework
- Motivation scenario to apply the framework phases

#### Phase 1: Mapping Security and Privacy Requirements:

- A- List of Security and Privacy Requirements:
- B- Mapping conflicts between security and privacy requirements
- Organizational view (SecTro)

#### Phase 2: Identify Conflicts between Requirements and Conflict Decisions:

- 2.1 Supporting tools:
- 2.1.1 Tables:
- 2.1.2 Models
- 2.2 Privacy by Design View (SecTro-no conflict concepts)
- 2.3 Privacy pattern library

#### 3. Phase 3: Conflict Resolution Patterns

- 3.1 Conflict resolution table
- 3.2 Privacy by Design View after adding conflicts concepts (SecTro)

**Toolkit for the Focus Group session:**

<b>Phases</b>	<b>Inputs</b>	<b>Outputs</b>
<i>Phase 1: Mapping Security and Privacy Requirements</i>	<ul style="list-style-type: none"> <li>- Mapping matrix</li> <li>- Scenario</li> <li>- Organizational view (SecTro)</li> </ul>	<ul style="list-style-type: none"> <li>- Identify requirements for each scenario.</li> </ul>
<i>Phase 2: Identify Conflicts between Requirements and Conflict Decisions</i>	<ul style="list-style-type: none"> <li>- Supporting tools (tables+ models)</li> <li>- Privacy by Design View</li> </ul>	<ul style="list-style-type: none"> <li>- Adding the Supporting Tool in Privacy Pattern Library</li> </ul>
<i>Phase 3: Conflict Resolution Patterns</i>	<ul style="list-style-type: none"> <li>- Adding the Supporting Tool in Privacy Pattern Library</li> </ul>	<ul style="list-style-type: none"> <li>- Conflict resolution table</li> <li>- Privacy by Design View after adding conflicts concepts</li> </ul>

- **Motivation scenario**



## Phase 1: Mapping Security and Privacy Requirements

### A- List of Security and Privacy Requirements:

Security Requirements	Privacy Requirements
Availability	Anonymity
Nonrepudiation	Unlinkability
Confidentiality	Pseudonymity
Integrity	Unobservability
Authentication	Undetectability
Authorization	
Accountability	
Auditability	

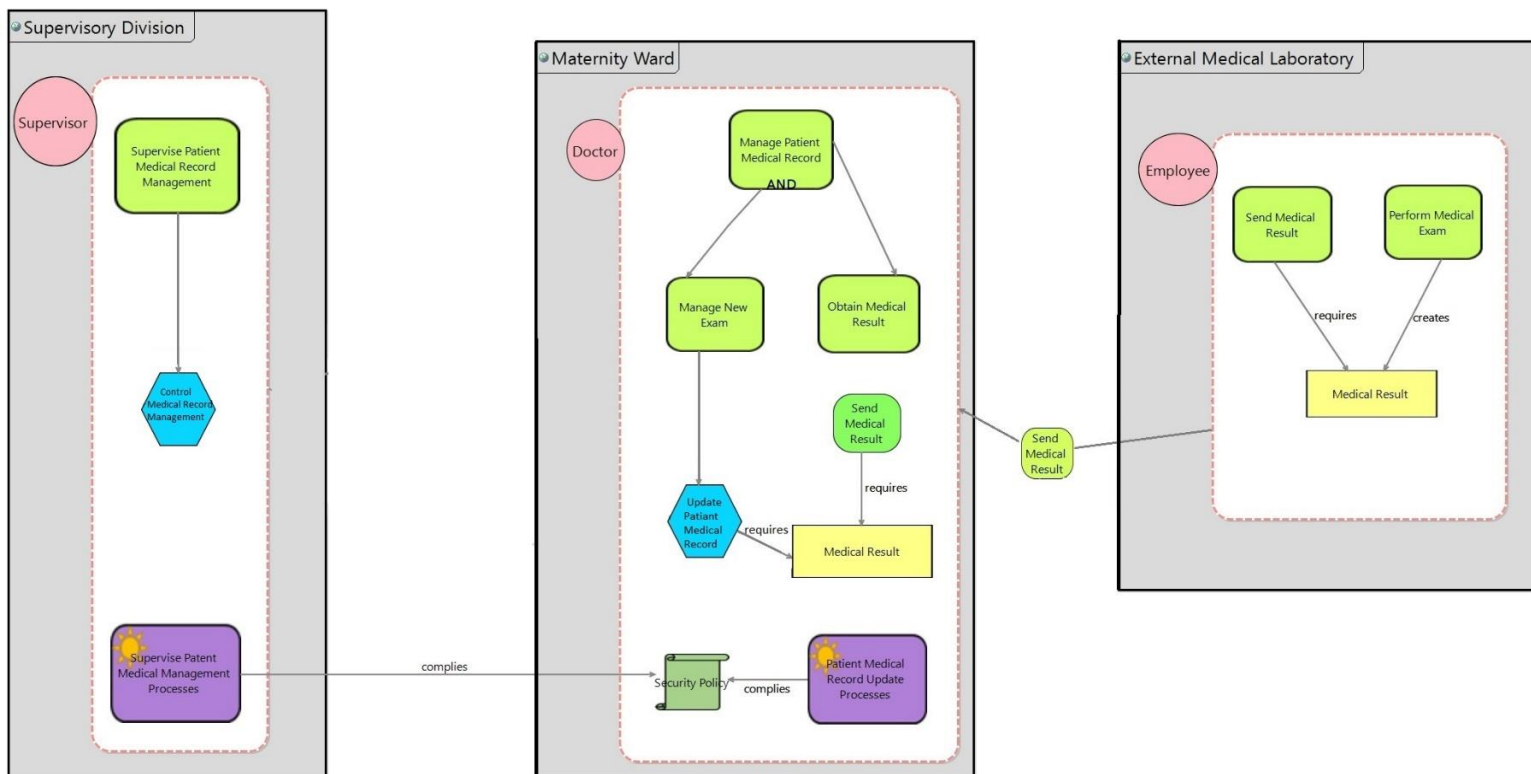
### B- Mapping conflicts between security and privacy requirements

		Security Requirements:								Privacy Requirements:							
		Availability	Non repudiation	Confidentiality	Integrity	Authentication	Authorization	SOD	BOD	Accountability	Auditability	Anonymity	Unlinkability	Pseudonymity	Unobservability	Undetectability	
<b>Security Requirements:</b>	Availability							○				○	○		○	○	
	Non repudiation											○			○		
	Confidentiality											○			○		
	Integrity											○	○		○		
	Authentication						○					○		○	○		
	Authorization							○							○		
	SOD		○					○									
	BOD													○	○		
	Accountability												○			○	○
	Auditability												○			○	○
<b>Privacy Requirements:</b>	Anonymity	○	○	○	○	○				○	○	○	○	○	○		
	Unlinkability	○			○				○			○		○		○	
	Pseudonymity					○			○			○	○				
	Unobservability	○	○	○	○	○	○			○	○	○				○	
	Undetectability	○								○	○		○		○		

*Security requirements conflicts with some privacy requirements:*

Security requirement	Privacy requirements
<b>Confidentiality</b>	Anonymity, Unlikability, Undetectability, Pseudonymise
<b>Integrity</b>	Anonymity, Unlikability, Unobservability
<b>Availability</b>	Anonymity, Unlikability, Undetectability, Unobservability
<b>BOD</b>	Unlikability
<b>Accountability</b>	Anonymity, Undetectability, Unobservability
<b>Non-repudiation</b>	Anonymity, Unobservability

- **Organizational view (SecTro)**



## Phase 2: Identify Conflicts between Requirements and Conflict Decisions:

### 2.1 Supporting tools:

#### 2.1.1 Tables:

*A: tool works with both requirements*

<i>Security requirements</i>	<i>Privacy requirements</i>
<b>Confidentiality</b>	Anonymity, Unlikability, Undetectability, Pseudonymise
<b>Integrity</b>	Anonymity, Unlikability, Unobservability
<b>Availability</b>	Anonymity, Unlikability, Undetectability, Unobservability
<b>BOD</b>	Unlikability
<b>Accountability</b>	Anonymity
<b>Non-repudiation</b>	Anonymity

<i>SECURITY AND PRIVACY REQUIREMENTS</i>	<i>TOOL TO SUPPORT REQUIREMENTS</i>
<b>ANONYMITY VS CONFIDENTIALITY</b>	Cryptographic, Steganographic technologies, Onion routing
<b>UNLINKABILITY VS CONFIDENTIALITY:</b>	Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing
<b>UNLINKABILITY VS INTEGRITY</b>	Cryptographic
<b>PSEUDONYMITY VS CONFIDENTIALITY</b>	Searchable encryption
<b>UNDETECTABILITY VS CONFIDENTIALITY</b>	Steganographic technologies

*B: Tools are suitable for Privacy Requirements*

<i>PRIVACY REQUIREMENTS</i>	<i>TOOL TO SUPPORT REQUIRMENT</i>
<b>ANONYMITY</b>	Cryptographic, Steganographic technologies, Onion routing, trusted third parties Dummy traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), K-anonymity.
<b>UNLINKABILITY</b>	Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing, K-anonymity, data hiding, trusted third parties, dummy traffic.
<b>PSEUDONYMITY</b>	Searchable encryption, Public key
<b>UNOBSERVABILITY</b>	Dummy traffic
<b>UNDETECTABILITY</b>	Dummy traffic, Steganographic technologies

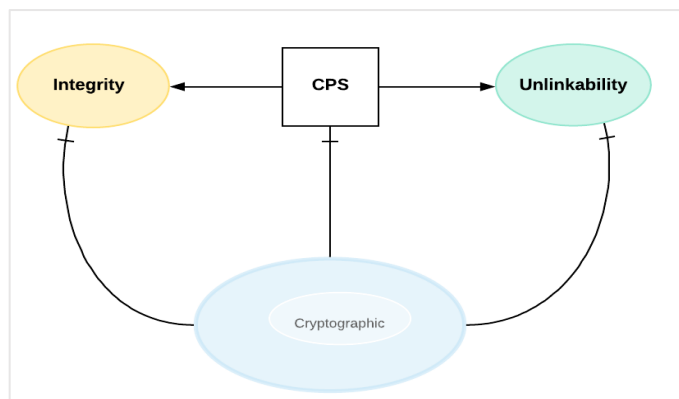
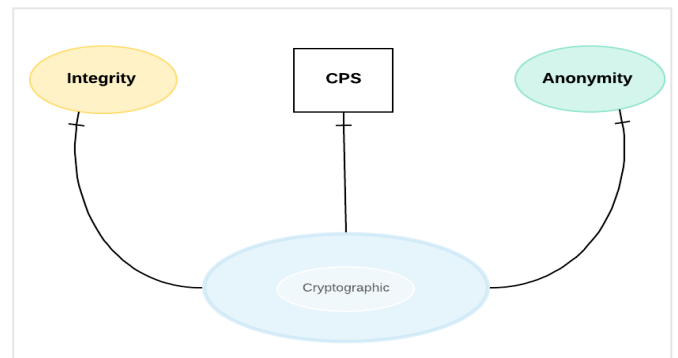
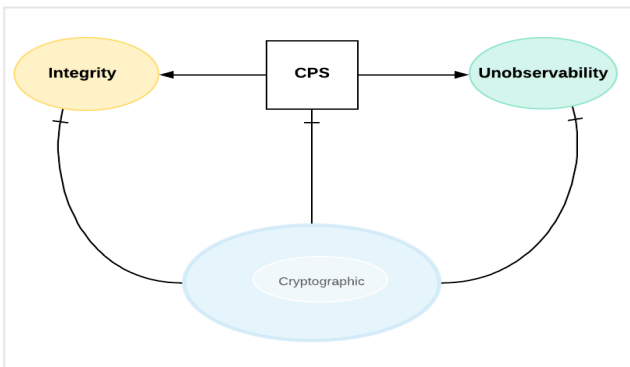


*C: Tools are suitable for Security Requirements*

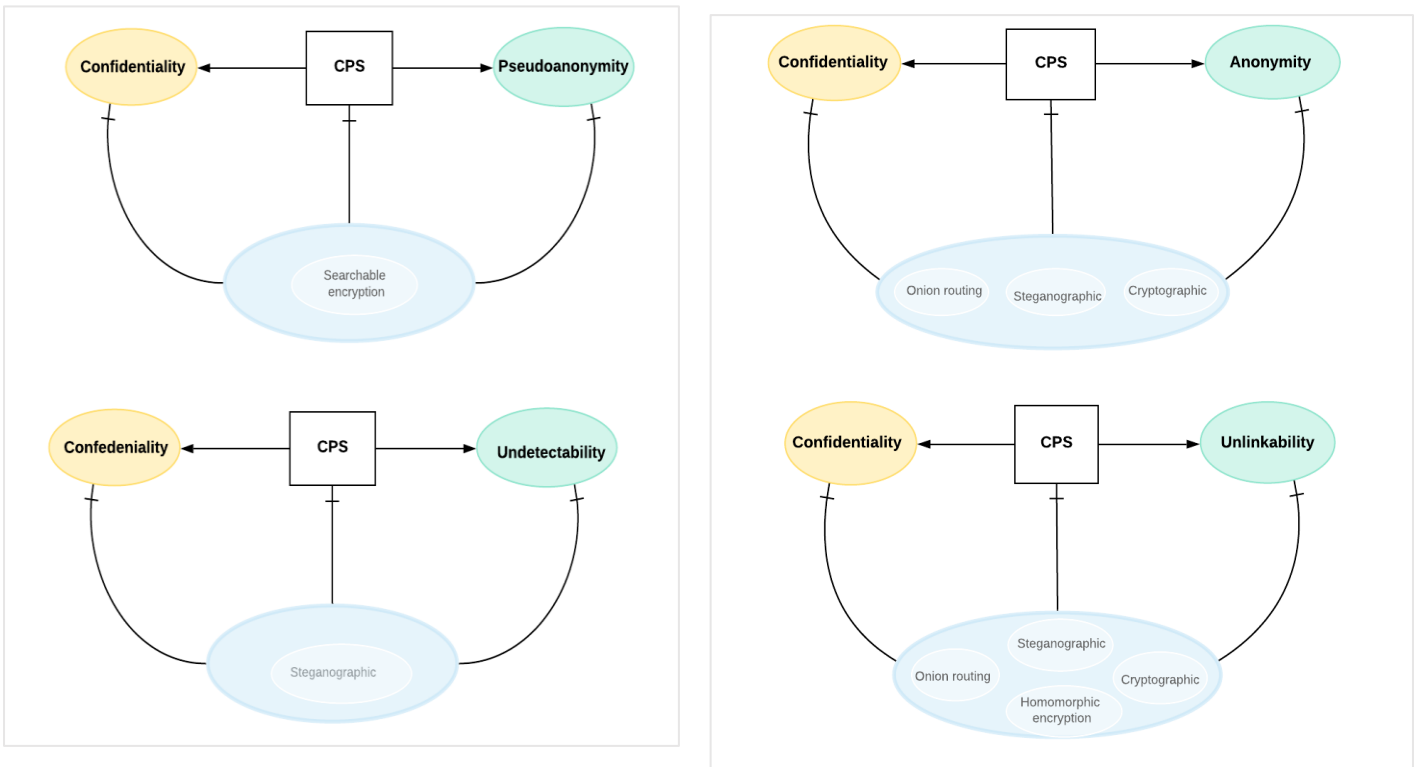
SECURITY REQUIREMENTS	TOOL TO SUPPORT REQUIREMENT
<b>CONFIDENTIALITY</b>	Cryptographic, access control enforcement, Symmetric key and public key encryption, Steganographic technologies, Homomorphic encryption, Onion routing Searchable encryption
<b>INTEGRITY</b>	Cryptographic, access control enforcement (ACE), message authentication codes (MAC) redundancy and comparison
<b>AVAILABILITY</b>	Redundancy to the system
<b>ACCOUNTABILITY</b>	ADOPT, D anonymous, IDEMIX

**2.1.2 Models:**

**A: Integrity** vs: (Anonymity Unlinkability and Unobservability)



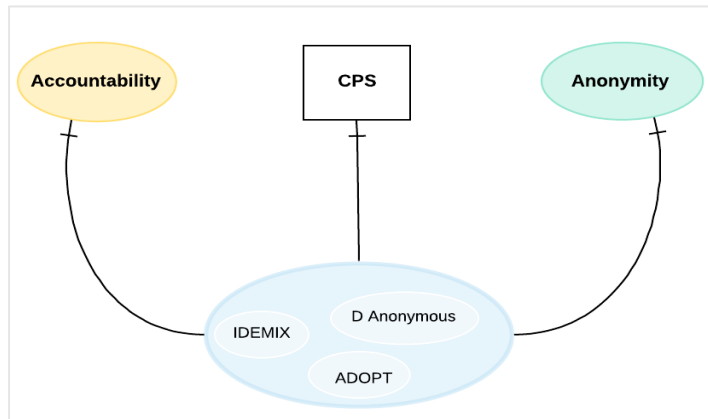
**B: Confidentiality** vs: (Anonymity, Unlinkability, Pseudonymity and Undetectability)



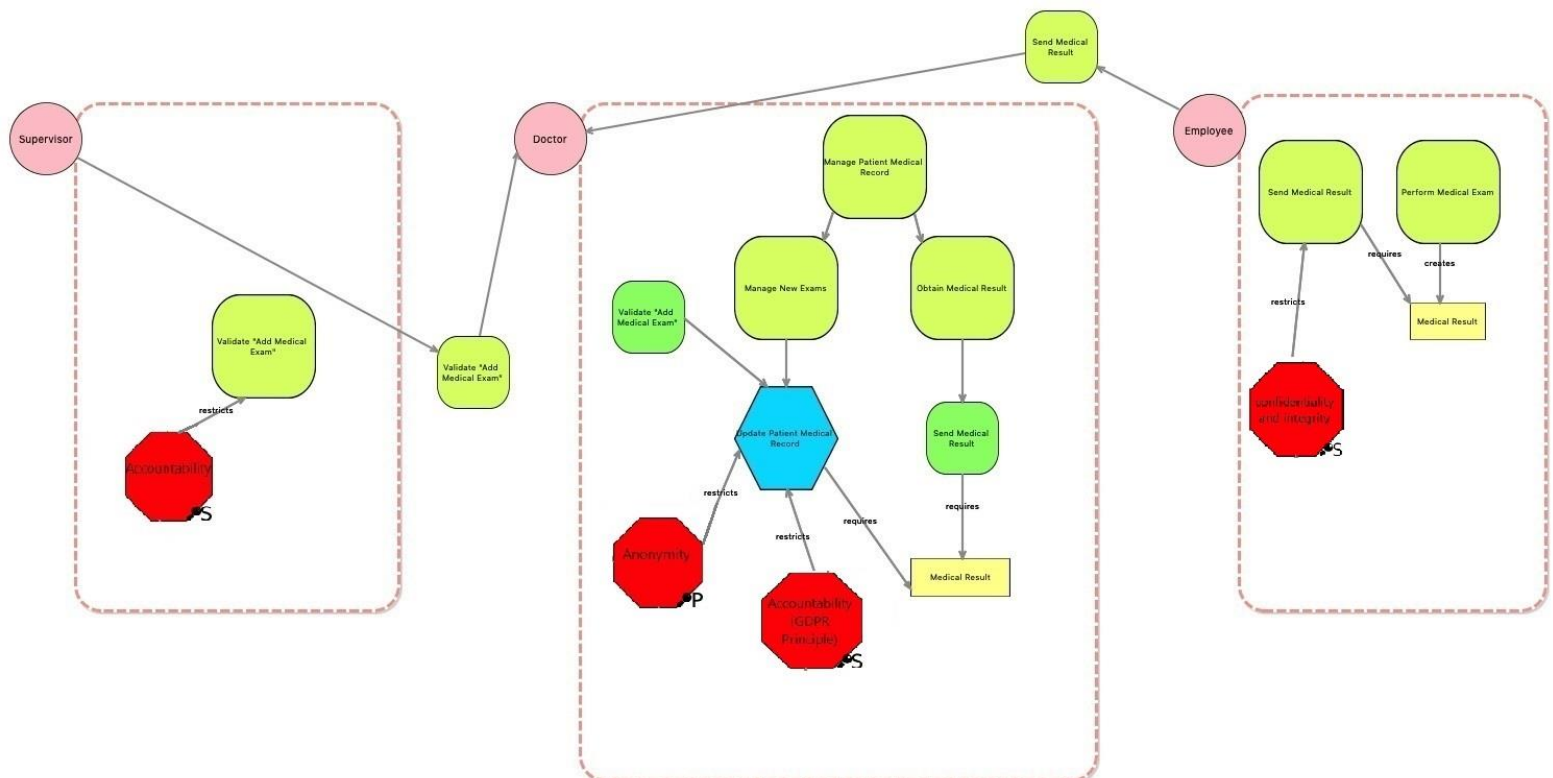
**C: Availability** vs: (Anonymity, Unlinkability, Unobservability and Undetectability)



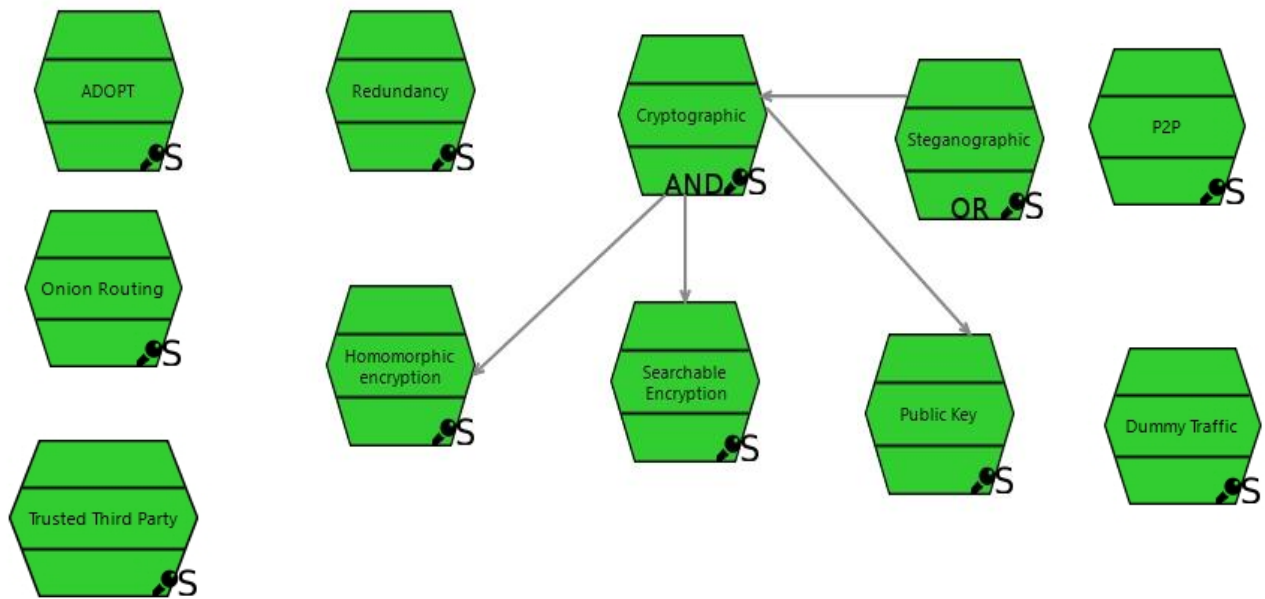
## D: Accountability vs: Anonymity

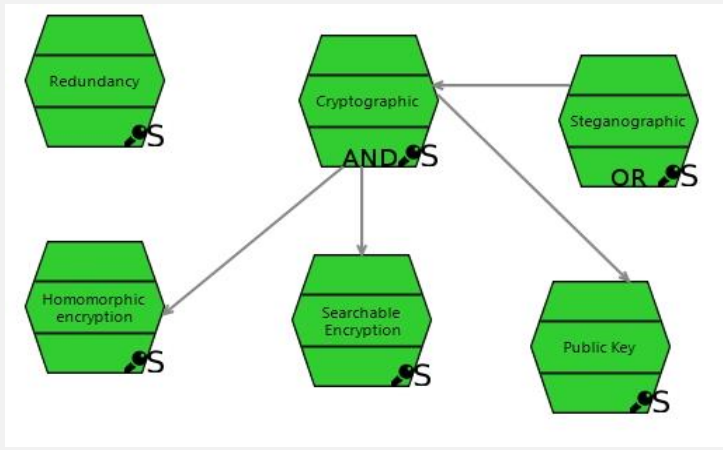



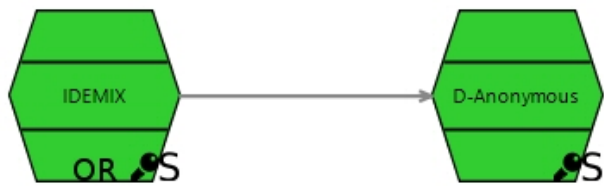
## 2.2 Privacy by Design View (SecTro-no conflict concepts)



## 2.3 Privacy pattern library


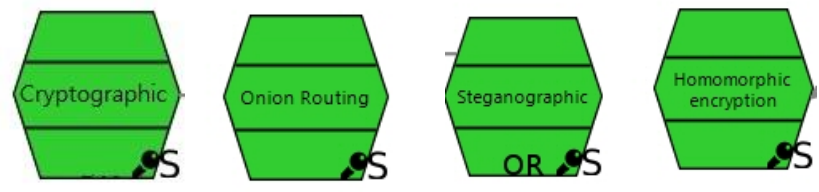
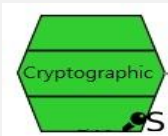

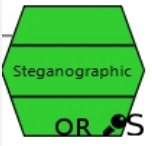


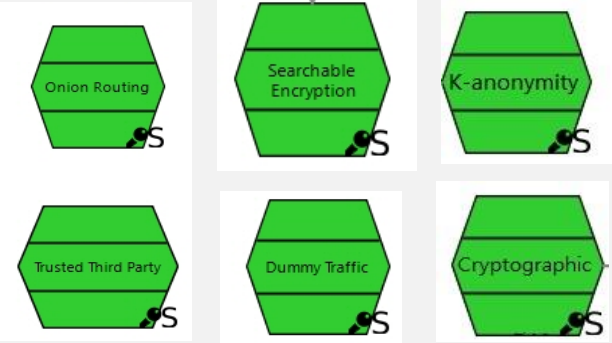
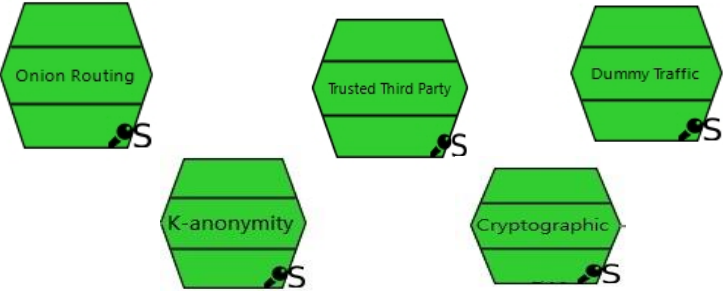



SECURITY REQUIREMENTS	TOOL TO SUPPORT REQUIREMENT
<b>CONFIDENTIALITY</b>	<p>Cryptographic, access control enforcement, Symmetric key and public key encryption, Steganographic technologies, Homomorphic encryption, Onion routing Searchable encryption</p> 
<b>INTEGRITY</b>	<p>Cryptographic, access control enforcement (ACE), message authentication codes (MAC)</p> 

<b>AVAILABILITY</b> <b>ACCOUNTABILITY</b>	Redundancy to the system ADOPT, D anonymous, IDEMIX 
--	--

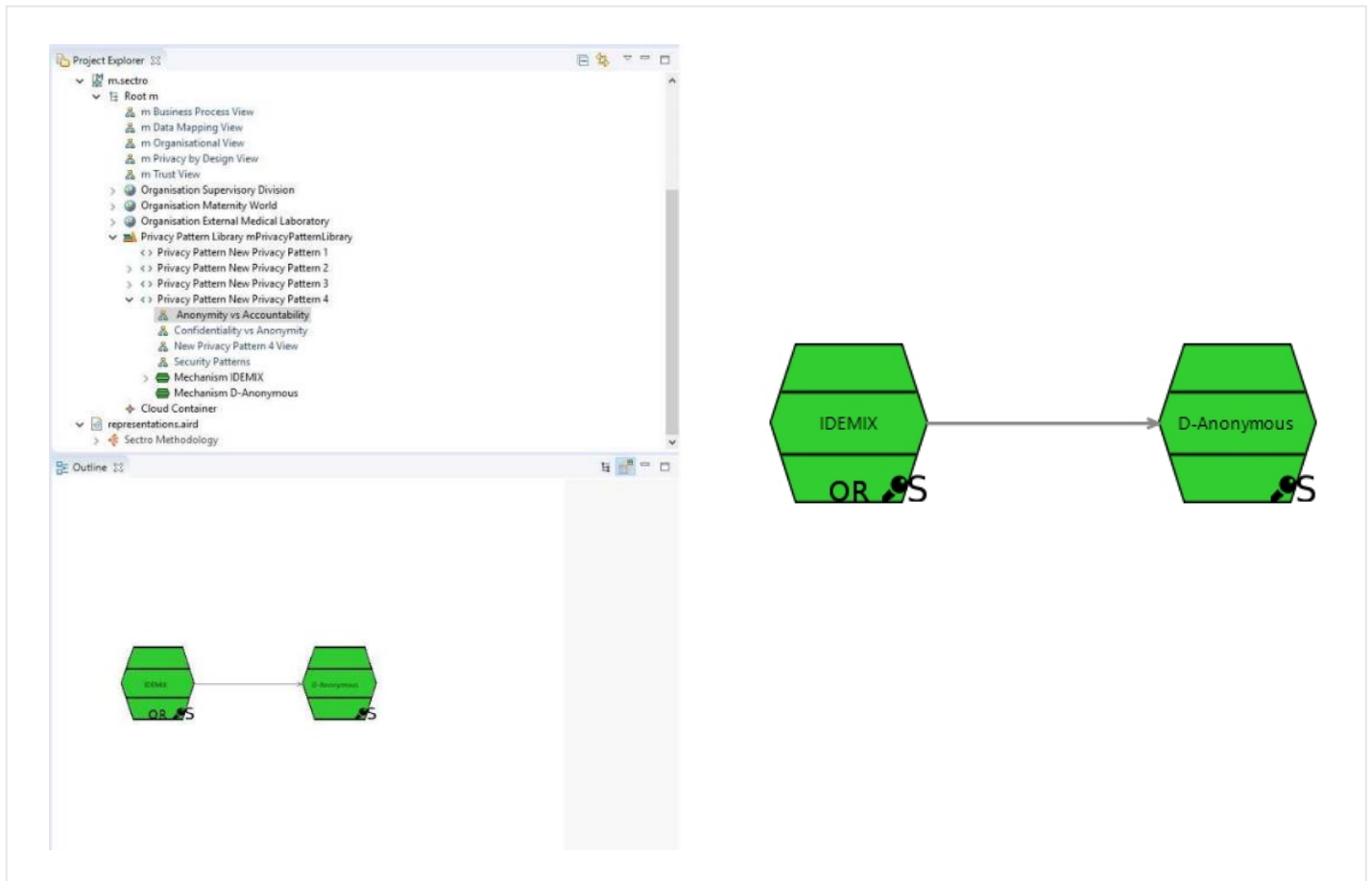
**SECURITY AND  
PRIVACY  
REQUIREMENTS**

**TOOL TO SUPPORT REQUIREMENTS**

<b>ANONYMITY VS CONFIDENTIALITY</b>	Cryptographic, Steganographic technologies, Onion routing 
<b>UNLINKABILITY VS CONFIDENTIALITY:</b>	Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing 
<b>UNLINKABILITY VS INTEGRITY</b>	Cryptographic 
<b>PSEUDONYMITY VS CONFIDENTIALITY</b>	Searchable encryption 
<b>UNDETECTABILITY VS CONFIDENTIALITY</b>	Steganographic technologies 

PRIVACY REQUIREMENTS	TOOL TO SUPPORT REQUIRMENT
<b>ANONYMITY</b>	<p>Cryptographic, Steganographic technologies, Onion routing, trusted third parties, Dummy traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), K-anonymity.</p> 
<b>UNLINKABILITY</b>	<p>Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing, K-anonymity, data hiding, trusted third parties, dummy traffic.</p> 
<b>PSEUDONYMITY</b>	<p>Searchable encryption, Public key</p> 
<b>UNOBSERVABILITY</b>	<p>Dummy traffic</p> 
<b>UNDETECTABILITY</b>	<p>Dummy traffic, Steganographic technologies</p> 

## Adding the Supporting Tool in Privacy Pattern Library



### Phase 3: Conflict Resolution Patterns

#### 3.1 Conflict resolution table

Concept	ACCOUNTABILITY VS ANONYMITY
<b>Problem</b>	Studies show there is a possibility of having conflicts between accountability as a security requirement and anonymity as a privacy requirement, based on the nature of those requirements.
<b>Force</b>	How to satisfy both requirements and resolve the conflict between those requirements.
<b>Solution</b>	<p>The diagram illustrates the relationship between three concepts: Accountability (represented by a yellow oval), CPS (represented by a white rectangle), and Anonymity (represented by a green oval). Arrows point from each of these three concepts down to a central light blue oval. This central oval contains three smaller ovals: IDEMIX, D Anonymous, and ADOPT. This indicates that these three mechanisms serve as solutions to the conflict between Accountability and Anonymity, and are associated with the CPS concept.</p>

### 3.2 Privacy by Design View after adding conflicts concepts (SecTro)

