**University of Southampton**

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

# Designing a Better Internet of Things, Privacy and Compliance by Design as SysML Domain Extension for Consumer Smart Electronics

*by*

**Robert Henry Thorburn**

MA, MBA, MSc

ORCiD: /0000-0001-5888-7036

*A thesis for the degree of*
*Doctor of Philosophy*

November 2022

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

**Designing a Better Internet of Things, Privacy and Compliance by Design as SysML Domain Extension for Consumer Smart Electronics**

by Robert Henry Thorburn

The Internet of Things (IoT), especially for consumer applications, is often described in terms of either its great promise for new and improved services, or its wholesale invasion of user privacy. Investigating this dichotomy, describing the nature there of, and proposing a remedy, jointly constitute the core of the project and contribution presented herein. The IoT is characterised by relentless miniaturisation, cost reduction, and the continued inclusion of new market segments, all in aid of delivering on the promise of truly ubiquitous computing. As one of the most prominent areas for IoT implementation, networked consumer electronics shows a rapid pace of adoption, recasts legacy devices as connected ”smart” devices, and presents an extensive list of privacy and security failures. Making use of connected devices at the edge, consumer IoT implementations supply data to more capable off site systems for analysis and value extraction. This supplies the service provider with valuable data but also affords the customer new services and device functionality. Unfortunately, such devices and systems are all too often rolled out with little to no regard for privacy or regulatory compliance. We contend that the best option for addressing these issues is a new "by design" approach which is based on an investigation of current practice and theory and framed within modern industry best practice. We act on this contention by considering a wide range of related contemporary research and legislation, conducting testbed based research and finally, deriving a new domain extension for the Systems Modelling Language (SysML) connecting formerly discrete privacy and compliance focused elements. Consequently, this domain extension is called DISCREET: **D**oma**I**n exten**S**ion for **C**ompliance and p**R**ivacy by d**E**sign in consum**E**r io**T**.

# Contents

# List of Figures

# List of Tables

# Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as: Robert Thorburn, Andrea Margheri, and Federica Paci. Towards an Integrated Privacy Protection Framework for IoT: Contextualising Regulatory Requirements with Industry Best Practices. In *IET Conference Proceedings*, pages 1–6. The Institution of Engineering & Technology, 2019
Robert Thorburn, Federica Paci, Vladimiro Sassone, and Sophie Stalla-Bourdillon. Connecting Regulatory Requirements to Audit Outcomes: A Model-driven Approach to Auditable Compliance. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 641–642. IEEE, 2021

Signed: Date: 2 November 2022

## Acknowledgements

The road less travelled is without a doubt the best description for my path to and through this PhD. Between people leaving, the turmoil of the pandemic, and the natural progression of any PhD, it has been an eventful few years. Although many people have played some role in getting me here I would like to say a special thankyou to the following seven.

Professor Vladimiro Sassone. Vladi, I will always be grateful to you for taking over as my lead supervisor. After my second lead supervisor left the university I was definitely in need of stability and somebody who could cut through the noise. You not only did both but gave me a significant boost in confidence, much more so than you probably know. Sincerely, thank you.

Professor Sophie Stalla-Bourdillon. Sophie, ever since my Masters degree you have played the role of constant guide to my academic endeavours. Thank you for that guidance, for agreeing to be my supervisor, and for steering me to cyber security. I would not have gotten here without you.

Associate Professor Federica Paci. Federica, although you were only my lead supervisor for nine months your impact on my work has been significant and lasting. This isn't just the case for my PhD but also for the work I will be pursuing next. Thank you.

Dr Cat Morgan. The voice of reason, my unwavering supporter, and above all else, a true friend. We have come a long ways since our Masters year and through it all I have counted on you and leaned on you more than anybody else. Thank you so much for being there for me.

Chris Maidens and Dr Stephen Hart. The coffee break brain trust. Thank you both for the countless hours of talking things through, figuring things out, and general comradery. Doing that over teams hasn't quite been the same, but it's been priceless still.

Dr Ashton Kingdon. Ashton, thank you so much not just for the proofreading and suggestions, but especially for the lockdown teams chats.

*To Susie Thorburn, Carin Thorburn, and Gerhard Thorburn. The ones who are always in my heart.*

# Chapter 1

# Introduction

## 1.1 Motivation

In this work we will show that there is a significant, and worsening, failure on the part of system and service providers to deliver consumer grade Internet of Things (IoT) devices that comply with GDPR requirements and adhere to the principle of Privacy by Design (PbD). We further identify this failure as one of design, that is to say, there is no insurmountable material or technological reason for these failures as they are the direct results of design choices. Furthermore, although others have spotted these same issues they either do not directly attend to the design-based nature thereof, or propose bespoke and contextually limited remedies.

To address this state of affairs a design-based remedy is proposed, taking the form of a domain specific extension to the Systems Modelling Language (SysML). This extension is formulated as a model library, as opposed to a change to the language itself, and as such retains language compliance. As a result, the new domain extension can be used in any language compliant tool, by any system modeller familiar with standard SysML, thereby making a direct and substantive contribution to the advancement of compliant and private systems design within consumer IoT.

To develop these points and deliver on this promised remedy the problem space must be investigated, other proposed solutions understood and best practice incorporated as appropriate. For this, a strong foundation is needed which we establish in this chapter by first presenting our motivation for and then the structure of, this work. To begin though, the nature of the IoT itself must be looked at.

The IoT is often characterised in reference to its broad scope, pervasiveness, and sociotechnical nature, all of which is captured in the following IEEE definition [79]:

> *IoT refers to any system of interconnected people, physical objects, and IT platforms, as well as any technology to better build, operate, and manage the physical*

*world via pervasive data collection, smart networking, predictive analytics, and deep optimization.*

With sensors and actuators being ever cheaper and more capable, the drive to embed these devices into nearly any and all objects within the human sphere [6], is not only gaining pace but has the potential to span all of human endeavour eventually. Although the IEEE definition accurately captures the scope of the IoT, it also sheds light on the difficulty faced by any researcher who wishes to pin the concept down. The IoT has become so broad a concept, has infiltrated so many other fields, and is constituted of so many disparate technologies, that it runs the risk of ceasing to be a functional and distinguishable field. To counter this, some more conceptual work is needed, first to focus on the functioning of the IoT and second, to contextualise the work presented herein within a chosen subset of the IoT. To build out the IEEE definition then, we present a more function-based approach as provided by Al-Fuqaha et al., who identifies six elements within IoT ecosystems [2]. These are:

1. *Identification:* The use of unique device and service identifiers, to ensure component access when needed.
2. *Sensing:* IoT devices gather data for either on-site or off-site data-stores [1].
3. *Communication:* Communications technologies used to connect nodes in an IoT system. Short-range and low powered wireless technologies are typically used for sensors and actuators, while industrial and other on-site devices might use Wi-Fi, GSM[2] or fixed connections.
4. *Computation:* Processing units used by IoT devices in a network, as well as the software applications running on those.
5. *Services:* The functionality or value of an IoT implementation, such as exercise metrics gained from a fitness tracker.
6. *Semantics:* The use of computing power, often off-site, to extract knowledge from the data harvested by IoT devices.

A further challenge for defining the IoT succinctly, is that although the term was first coined in the 1990s [10], the IoT is still actively changing and developing while key aspects such as ubiquitous computing, substantially predates the IoT. Consequently, issues of standardisation in the IoT remain unresolved with knock-on effects into privacy and security [114]. A significant part of this problem is the manner in which the IoT, especially in the home, extracts data for analyses by way of ubiquity and interconnection between devices often not perceived by users

[1] A point of note is that Al-Fuqaha et al. includes both sensors and actuators under the “sensing” heading.

[2] The Global System for Mobile communications, or GSM, is a prominent mobile phone networking technology. In this context though, the reference to GSM should be taken as meaning GSM and its competitors alike.

as “smart[3] devices”. Of course these weaknesses and challenges are often framed by device manufacturers as enhanced features, with the discussion focused on the benefits of new offerings such as recommender services, with no mention of the related private data being extracted and analysed [97]. These challenges are further compounded by the IoT not only including a loosely defined and changing set of technologies, but also by significant differences between device manufacturers. The fluctuating nature of the IoT is clearly on display here as two device manufacturers can present offerings to the market that superficially appear to be similar but which in fact use different and even incompatible protocols, standards, software, data schemas and so forth [47]. Whether dealing with security threats, privacy breaches, or simple interoperability, this entrenched heterogeneity bedevils any notion of single mitigation “silver bullets” [122].

These concerns do, however, not appear to impact on the sales performance of IoT devices in general[4] with the number of Internet-connected objects surpassing the total human population in 2010 [2][5]. It is also expected that these devices will shortly total more than 20 billion [29], and provide ubiquitous computing, data collection and value added services into all aspects of human society [157]. These projections are also supported by field studies, such as that carried out by Al-Fuqaha et al. which found a significant rise in Machine to Machine (M2M)[6] communication identified as relating to IoT devices [2]. In discussing the continuing growth of the IoT, Miettinen et al. rightfully referred to it as “*an ongoing mega trend in computing*” [104].

As the IoT extends its reach in general, it inevitably also finds numerous applications in the home, not only with household appliances but even in children’s toys. Given the manner in which the IoT leverages user data to both provide enhanced services to users and greater value to service providers, applications aimed at children should necessarily raise concern. This is also born out in a large number of documented failures, both by way of negligence and by substandard design. Two of the best known and worst offenders, are “My Friend Cayla” and “Cloud Pets” [73]. “Cloud Pets” are stuffed toy animals capable of connecting to the Internet and accessing cloud services via a mobile phone and associated app. Both the devices and the app were used for data collection, with the app recording the names and images of children as well as the email addresses of their parents or guardians. Users of the app could also record messages to be sent to the toy, for playback via a built in speaker. In 2017 an exposed database belonging to the “Cloud Pets” manufacturer was discovered, containing over 2 million user records of which 40% were complete.

[3]Although used in the IEEE definition, the term “smart” remains some what generic. Herein we take it to mean devices, sensors, or actuators which are capable of at least some data handling and possessing of some form of networking.

[4]The following discussion will introduce individual devices which have fallen foul of both public opinion and the law, though the general trend remains one of increased IoT adoption.

[5]In keeping with the IEEE definition, we include devices connected to the Internet via a gateway as part of the IoT, implying that the true number of IoT devices is actually far higher than the number sighted here.

[6]In general, we can distinguish three types of such connections, namely person to person (P2P), person to machine (P2M) and machine to machine (M2M).

The “My Friend Cayla” doll was similarly able to connect to off-site services, though it also included a microphone. This enabled one of the toys main selling points, which was smart assistant like features for children, including the answering of questions. The device did allow for the Internet connection to be severed, but the microphone could not be disabled. This was particularly problematic since the toy’s Bluetooth connection, which was intended for connecting to a smart phone hosting a related control app, did in fact not enforce any credential checking. As a result, any other smart phone in range, even without the control app, could connect to the toy. These failings, but specifically the always on microphone, eventually led to Germany branding the toy as an illegal surveillance device and banning it.

These two toys and their associated failings demonstrate just some of the ways in which insecure and badly designed data processing can result in negative outcomes for all concerned. It is no wonder then that the US National Institute of Standards and Technology (NIST) views privacy risks for IoT devices as explicitly pertaining to data actions. This includes both accessing and processing data in any way that could adversely affect the data subject [21]. Data actions are however, only part of the problem where consumer electronics are concerned. Since these devices have the appearance of legacy devices and also retain legacy functionality, users do not generally conceive of these devices as anything more than improved TVs, doorbells, or toys. Accordingly, it is unlikely that users view themselves as data subjects in the presence of these devices, a challenge which is compounded by device manufacturers doing little to nothing to address these concerns [103].

Keeping with toys as examples, the above point is demonstrated by the “Hello Barbie” doll, which encourages children to verbally share their thoughts with it. These are then recorded for storage and processing by a third-party data processor[7]. The adults who set up the related device account, and anybody they share their credentials with, can then access these recordings online and even post them to social media[144]. Of course these concerns are amplified when malicious actors are concerned since devices in the proximity of children, such as toys or baby monitors, not only expose personal data but could also allow for attackers to directly interact with children through their speakers [4].

This litany of failures is also not limited to consumer IoT devices, instead the use of toys as examples serves to underscore just how egregious these failings can in fact be. It is also notable that all of the failings named above can be exploited by individuals with little technical proficiency[8] while none of these failings are due to technological shortcomings. In short then, the primary culprit is bad design and implementation. As will be shown in the following discussion there is an acute need for a formalised approach to this problem. The case for this assertion will be further strengthened below, though the anecdotal evidence presented thus far already illustrates the problem. Yet, a substantial number of device manufacturers and vendors appear wholly focused

[7] Using GDPR definitions as introduced in chapter 3; data controllers are the parties in control of data collection and processing, data processors are the parties conducting processing under the direction of controllers (though they can be one and the same), data subjects are the natural persons the data relate to.

[8] There are services such as `shodan.com` which index exposed Internet connected devices, along with a detailed reference of the weaknesses concerned.

on chasing market share even at the cost of PbD and compliance [58]. A worrying further trend, as found by Paul et al., is that this neglect of compliance and PbD is also prevalent in the terms and conditions device vendors present to end users [115]. A finding which is confirmed by the research presented herein and introduced in Chapter 4.

A UK-based example of what can go wrong when PbD is not pursued in a formalised manner, can be found in the government's roll out of smart energy meters. A project which was beset with multiple preventable failures. These included lacking or vague criteria for the implementation and assessment of PbD, suppliers choosing to follow their own requirements over those set by government, and service providers who viewed PbD as a hindrance to their data collection goals [26]. Given the current regulatory landscape, issues of privacy have at a certain level become linked to regulatory compliance, hence the focus herein on compliance and privacy by design. However, this should not be seen as the two propositions being conflated with, or equated to, one another. Instead we contend that for a "by design" approach to succeed in tackling the challenges to privacy within consumer smart electronics, such an approach must address both regulatory compliance and privacy by design. Where the latter specifically includes issues of privacy which extend beyond the minimum regulatory requirements mandated by compliance regimes. Since legal requirements and other corporate, industry, or societal positions can make demands on how personal data[9] is treated, it is easy to see how issues beyond legal compliance remain relevant. It is therefore also necessary to define what is meant by the term privacy, since simply stating that the modern notion of privacy has developed over time and is now centred on control over one's own personal data [3], might be broadly correct it is definitely not functional. A more workable definition then, is offered by Ziegeldorf et al. who hold that privacy[10] is:

> *Information self-determination by enabling the subject (i) to assess his personal privacy risks, (ii) to take appropriate action to protect his privacy risks, and (iii) to be assured that it is enforced beyond his immediate control sphere.*

The third point above, relating to enforcement of the data subject's wishes beyond their immediate sphere of control, appears to speak to legislation. It is accordingly, no surprise that privacy is seen in many countries as a social good in need of protection by way of legislation [105]. Turning from this general discussion of privacy, to the IoT as a sociotechnical system with significant privacy implications, we naturally also expect that legislation should form part of the societal aspect of that system [30].

The obvious choice for regulation to review in this regard is the European Union's (EU) General Data Protection Regulation (GDPR)[11] which governs the rights and obligations related to the

[9] The term "personal data" is used in the EU and UK. Although it overlaps with the US term "personally identifiable data", the two are not equivalent.

[10] It would be tempting to hold that privacy threats are simply the absence or obstruction of the points proposed by Ziegeldorf. However, a more nuanced and context based approach is needed and for this we propose the LINDDUN threat modelling methodology introduced in Section 3.10.3.

[11] The ePrivacy Regulation is *lex specialis* to the GDPR and could have a significant future impact but is yet to be adopted. It has also seen so many revisions that no significant effort on including it herein could be justified at this time.

data of EU residents [125]. Barring certain exceptions, data subjects have the right to access, copy, amend, or delete any data held on them and be informed as to why, and what for, data was collected on them [12]. These requirements have clear and significant implications for consumer smart devices due to the preponderance of personal data they collect and process as well as the fact that data collection without consent remains rife [29]. This latter issue is also highlighted in our research presented in Chapter 4. Potentially more troubling though is that challenges in GDPR compliance do not just include non-compliant actions, but also core product functionality implemented in a non-compliant manner or even simply the current characteristics of consumer IoT which remains stubbornly heterogeneous with regards to standards, protocols, and its mixed use of edge-based, cloud-based and fog computing [97]. This heterogeneity also negatively impacts on data portability[13], which is a further right under the GDPR. This relates to subject data access requests typically being fulfilled by way of a CSV file, containing data in a layout and breakdown specific to the company providing it and without any assistance to the user as to what to do with this file. There is also no requirement on service providers to standardise the data collection and processing reflected in such a file, meaning that even if two service providers collect and process the same data, the fields in which the data are collected could be totally incompatible. Consequently, a CSV file retrieved from one service provider could not be directly added to the data pool of another [152]. This theme of core IoT functionality coming into conflict with GDPR requirements, continues with the use of M2M communications about data subjects and their data, but without the data subject's knowledge. A phenomenon also directly observed on our testbed.

The extensive list of failures, challenges and compliance issues introduced thus far might create the impression that consumer smart electronics are beyond redemption and an undertaking best avoided. Any such argument is moot though since these devices are not just very popular but are expected to become even more so going into the 2020's [8]. Due to this continued growth in the face of so many challenges, it is clear that PbD and compliance for smart consumer electronics, that is the IoT in the home, must be a high priority area for academic inquiry. Given the propensity for failures in consumer smart electronics to relate to designed functionality, such academic inquiry would then do well to not only investigate the current state of consumer IoT, but also to look at related work and propose design led solutions. It is our position that any such proposed solutions should be both academically sound and be focused on direct industry application. These points are of course not just based on the preceding discussion but also on a review of current literature presented in Chapter 3 and our testbed findings presented in Chapter 4.

[12] This is referred to as the purpose limitation.

[13] Please note that data access and data portability are related, but remain two separate rights under the GDPR. This is further discussed in Section 3.7

## 1.2 Framing the motivation

Although the introductory discussion outlines the problem space within which this research operates, a more distinct frame is needed to ensure a consistency of focus on a problem that can actually be addressed herein. In other words, describing or investigating the problem is in and of itself not good enough. Instead, a tightly scoped problem space must be investigated with the explicit aim of developing a solution with direct industry application. As previously stated, the problem space is consumer smart electronics with regards to privacy and legislative compliance where the primary failing is one of design and not of technological capability. This can be illustrated by way of example. As we will show in Chapter 4, failings such as routing traffic directly from a device to cloud storage, bypassing the data subject's control and scrutiny, is not a technological consideration but purely a design choice focused on maximum data extraction. As a direct result, any industry applicable solution derived from the research herein must be focused on design in the first instance and should naturally seek to leverage systems and procedures already widely in use and/or gaining traction.

System complexity has been on the increase over the last half century, and significantly so. This is a predictable result given the increased reliance on computerised systems and the heavy use of data driven analytics, which provides challenges and opportunities, both best met by an inherently interdisciplinary approach to systems design and management [83]. For the research presented herein, considering compliance and privacy in smart devices, this interdisciplinary approach is located at the intersection of the legal and the technical with a clear mandate to include stakeholder concerns.

The NASA Systems Engineering handbook describes systems engineering as "*a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system.*" [89]. Bearing in mind that systems engineering generally conceives of stakeholders as including regulatory bodies and legislation in addition to developers and end users, it is significant that Kapurch [89] refers to systems engineering as a means of achieving "*stakeholder functionality*". We consequently identify systems engineering as the context within which our design-based solution should be developed. This is, however, still too broad and in Chapter 5 we make the case that Model-based Systems Engineering (MBSE) is the appropriate form of systems engineering to engage with, which then further refines to the specific use of the Systems Modelling Language (SysML). The form of this engagement is a domain extension discussed in Chapter 7, which also develops the core of our design-based solution.

## 1.3 Research Questions

Thus far we have presented consumer IoT[14] as a large and continually growing market segment with significant potential for impacting on the data users generate in their homes and private

[14] Often referred to as smart consumer electronics, or simply smart devices.

lives. We further showed that this potential is often realised with negative consequences as the direct result of design choices made by service and device providers. As such, the problem space can be understood as; *systems design for smart consumer electronics that is disconnected from privacy and compliance requirements*. Therefore, any proposed remedy must take a "by design" approach and have direct industry application.

Our position, based on the above, is that there is a widespread and persistent failure to design consumer smart electronics that meet both privacy and compliance requirements. Included therein is the contention that these failures can be addressed, for industry application, by way of a methodology for both privacy and compliance by design. To investigate this hypothesis two distinct areas of inquiry are proposed. The first is a literature review to determine not just the current legislative position but also what other relevant research and possible remedies exist.Thereafter a testbed-based investigation of a typical consumer IoT setup can be conducted to not only target IoT devices, systems, and related terms and conditions[15], but also the manner in which these change over time. Once these areas of inquiry have been sufficiently examined they can then be used to inform a solution, with the literature review both providing a foundation for our work but also pointing to the gaps in the status quo. Working into these gaps the insight gained from the testbed can be used to ensure applicability to real-world consumer smart electronics.

A final factor informing our research is the following simple hypothesis, "*since these are design failings, system changes over time will not improve the situation but will in fact worsen outcomes*". Time is the independent variable here and compliance and privacy outcomes are dependent variables. Our research therefore draws on both experimental and observational data. This distinction is further explored in Chapter 2. To guide this undertaking and give rise to purposeful research [162], we propose the research questions presented below.

- Can a single SysML domain extension address both compliance and privacy by design in consumer IoT, and if so, what are its components[16]?

The following sub-research questions act as guideposts to direct our work towards the main research question presented above:

- RQ1: Which current compliance assessment, privacy enhancement or threat mitigation methodologies, strategies, and frameworks relate to consumer IoT implementations, or can be brought to bear on them?
- RQ2: Considering the defined problem space, what are the coverage gaps in prominent current frameworks and methodologies, including those not directly aimed at IoT?

[15] The terms and conditions presented to data subjects pay a vital role in compliance.

[16] Although we intend to develop a single coherent domain extension, this focus on components allows for a modular design. Furthermore, PbD in this sense is that which is drawn from threat modelling and is not the same as the data privacy measures dictated by the GDPR.

- RQ3: Which of these methodologies would best serve to assess the testbed results presented herein?
- RQ4: Do real-world observations of a consumer IoT implementation match up to the expected shortcomings, both as a snapshot and over time?
- RQ5: Can the lessons learned from the preceding investigations be translated into an actionable mitigation of the privacy and compliance related design challenges in consumer IoT?
- RQ6: What form could an approach to engineering compliance and privacy by design into IoT devices and systems take, if it were expressed in standardised systems engineering terms?

## 1.4 Contribution

In this chapter we have defined our problem space and drawn specific attention to some high profile failures for consumer smart electronics. In the following chapters we will significantly strengthen the case for academic research directed at this problem space both by way of a literature review and via primary research on a testbed. By doing so, we establish that consumer smart electronics present a significant and increasing risk to the privacy of data subjects who are often unaware of these risks. Since the data subjects in question constitute a significant and growing proportion of the population, any efforts to address these challenges can make a direct and clear contribution to strengthening data privacy for all.

The work presented herein provides targeted contributions not only in relation to directly addressing this larger challenge but also to the academic study of the field by employing an interdisciplinary approach which conceives of, and addresses the problem as, inherently sociotechnical. Addressing the first of these areas of contribution, which is the academic study of the problem space, we investigate and correlate a broad range of methodologies, frameworks, legal requirements, and industry practices, showing how each presents a crucial part of the privacy and compliance by design picture. This literature review also constitutes an independent contribution as it draws together and contrasts otherwise divergent work based on the privacy, compliance and IoT related focus therein. From this unique understanding of the problem space we then direct a series of tests at a number of consumer smart devices and do so over a period of three years.

In his forward to the book The Architecture of Privacy: On Engineering Technology that can Deliver Trustworthy Safeguards, Paul Ohm[17] observes that "*Data can either be perfectly anonymous or it can be private but never both*" [24]. We do not intend to argue this point but rather hold that any system dealing with personal data and issues such as anonymisation, must be built

[17] Paul Ohm is professor of law at Georgetown University and also holds degrees in computer science.

using tools that can engage with these problems in an adequately nuanced manner. Legal compliance in the design of networked electronics then, is not simply a matter of determining a set of legal requirements and directly implementing them. A detailed understanding of the technology is also needed, both to ensure that the eventual system meets other requirements but also to ensure that legal compliance is present at even the deepest levels as opposed to purely high level or tick box exercises [24]. As such, the research conducted in Chapter 4 represents a clear contribution in terms of primary research, with regards to both the method and length of data collection as well as the novel analysis thereof. Here we provide a second notable contribution to the academic debate by showing that significant shortcomings are not only located in the device functionality at a given point, but critically, is co-located in device documentation and has the potential to deteriorate over time. The importance of device documentation, including both manuals and "terms and conditions", is established by reference to the ways in which it can directly undermine key compliance requirements and actively impede the data subject's ability to fully understand the data collection and use associated with their devices. The knowledge gained from the testbed is then also used to adjust and reformulate some of the most important work identified in the literature review, contextualising that work specifically for application within consumer IoT design. This is also a timely contribution in its own right, as domain specific research for IoT segments with unique challenges is still a lagging field of inquiry with many researchers instead focusing on higher level or cross domain issues [88].

Moving on to the SysML domain extension in Chapter 7, we present a series of methods, frameworks and supporting documents unified under a governing method. Through the use of this governing method a single methodology is produced without the modular value proposition being lost. As such, our work presents a fully formed blueprint for systems engineers to integrate into their work, while more experienced practitioners can simply use those modules they need. Since this domain extension will also be made available online, practitioners will be able to directly import needed components into their system models.

Lastly, our new domain extension is not just presented as a finished product but is constituted of a series of self-contained elements. This modularity makes it easier for practitioners to apply our work to their specific needs and directly allows for application beyond the problem space of this work. This is also the logic for using a pre-existing framework to guide users in implementing our domain extension, as well as populating the "notes" fields when constructing requirements within SysML, since such notes can easily be used to provide free form instruction to model builders. The framework mentioned here is the MBSE Grid Framework and is introduced in Chapter 5 while the full list of requirements are presented in Appendix B.

## 1.5 Thesis structure

The chapters of this thesis can broadly be grouped into three distinct sections as follows:

1. Chapters 1 - 4 Ground work

2. Chapters 5 - 6 Structural work

3. Chapters 7 - 10 Domain extension and conclusion

In detail, this chapter provides an introduction to the work and delimits the area of focus while Chapter 2 describes the research conducted as well as the detailed rational there for. Chapter 3 presents the literature review while Chapter 4 presents the research conducted on the testbed. This is followed in Chapter 5 by a more detailed view of systems engineering and the choice for SysML. Directly building on the knowledge gained from the previous chapters, Chapter 6 contextualises the insights of the literature review within the realities of consumer IoT as investigated via the testbed. From there Chapter 7 formulates the domain extension, while Chapter 8 presents an example application thereof. Both the domain extension and the example implementation are then tested using standard verification and validation techniques as presented in Chapter 9. Finally, Chapter 10 presents the conclusion and also proposes future work. In each of these chapters the activity diagram presented in Figure 1.1 will be used to indicate progress as well as the primary activity and output of that chapter.

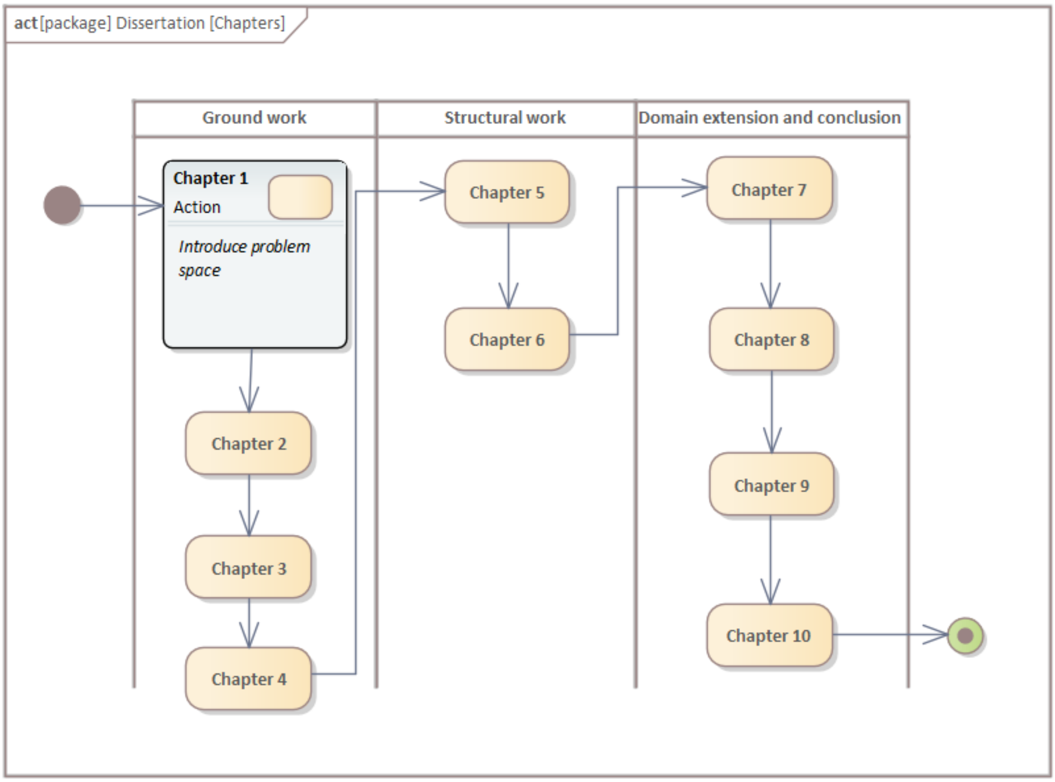



Figure 1.1: Chapter 1 research focus

### 1.5.1 Direct use of the domain extension

The domain extension developed herein is presented in Chapter 7. That chapter includes all the components of the domain extension as well as the rationale for each. If the reader however wanted to skip directly to the application of the domain extension they can view Subsection 7.2. Alternatively, to view just the domain extension without the supporting discussion, Appendix A presents the methods and other documentation while B contains the requirements.

# Chapter 2

# Research Methodology

## 2.1 From Methodology to method

In this chapter the “Onion model of research design” proposed by Saunders and Tosey will be used to position, and provide additional context to, the research presented herein [131]. This establishes the foundations on which both the literature review and testbed analysis will be built and is contextualised within the larger project by Figure 2.1. The onion model as presented in Figure 2.2, is a streamlined version of older and widely cited work by the same authors and seeks to classify research activities by segmented types. Working from the outside in, this model aids in directing the researcher’s thinking on their work by describing the applicable research philosophy, methodological choice, research strategy, time horizon and technical procedures. Although the classification and description of research methodologies constitute an active and varied field of inquiry in and of itself [32], we will not enter into an extended debate on the matter as that would constitute a detour from the work at hand.

A significant portion of the challenges in dealing with compliance and privacy issues for consumer IoT systems can be described as a threefold data access problem. Threefold since devices; 1) directly provide data to manufacturers; 2) expose data to ISPs, advertisers, and other snooping parties [9]; 3) and provide data to third-party applications and service providers operating across devices[1]. The data needed to investigate these challenges and eventually propose a remedy can be gathered by drawing on the work of others, by conducting direct primary research, or by a combination of the two approaches. We opt for the latter and will employ both a literature review and directly gather experimental data from consumer IoT devices over an extended period. In this we understand research as not only pertaining to the gathering of data, but also to the analysis and interpretation of that data [162] with these activities presented in chapters 3, 4 and 9. These issues are discussed in more detail below under Section 2.2.

[1]Such applications potentially include tracking apps, APIs, and others.

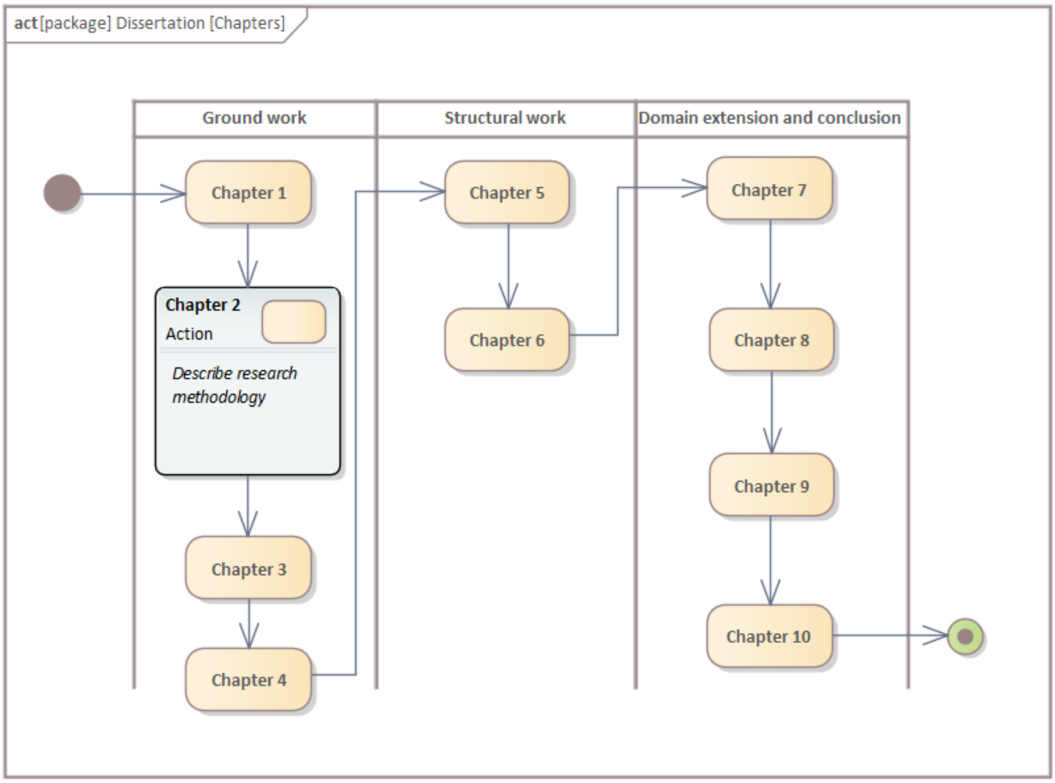



Figure 2.1: Chapter 2 research focus

Positioning this research within the work of Saunders and Tosey then, working from the outside in, we can describe the research conducted herein as a *Positivist*, *Multimethod Quantitative* approach in the form of *Experiments* conducted *Longitudinally* to gather and analyse data on the current state of affairs as well as the eventual functioning of the proposed remedy. This is presented with background reasoning in Table 2.1. A point to note on the testbed-based research being classified as experimental, is that we hypothesised in Chapter 1 that performance against privacy and compliance metrics (dependent variable) will not improve over any amount of time (independent variable) since under performance is due to design shortcomings. This fits squarely in the experimental camp of research design, but also only constitutes one aspect of the research conducted. Much of the research is instead concerned with gaining an understanding of the current state of affairs, which is commonly referred to as descriptive research [162] though Saunders and Tosey do not make a directly comparable distinction[2]. Lastly, the longitudinal approach taken herein is a less prominent, but still standard, approach within quantitative research [124] and is necessitated by the nature of the work.

[2] Saunders and Tosey do include case studies and these might be used to describe some of the testbed activities but delving in the minutia of classification does not add to the discussion in a useful manner.

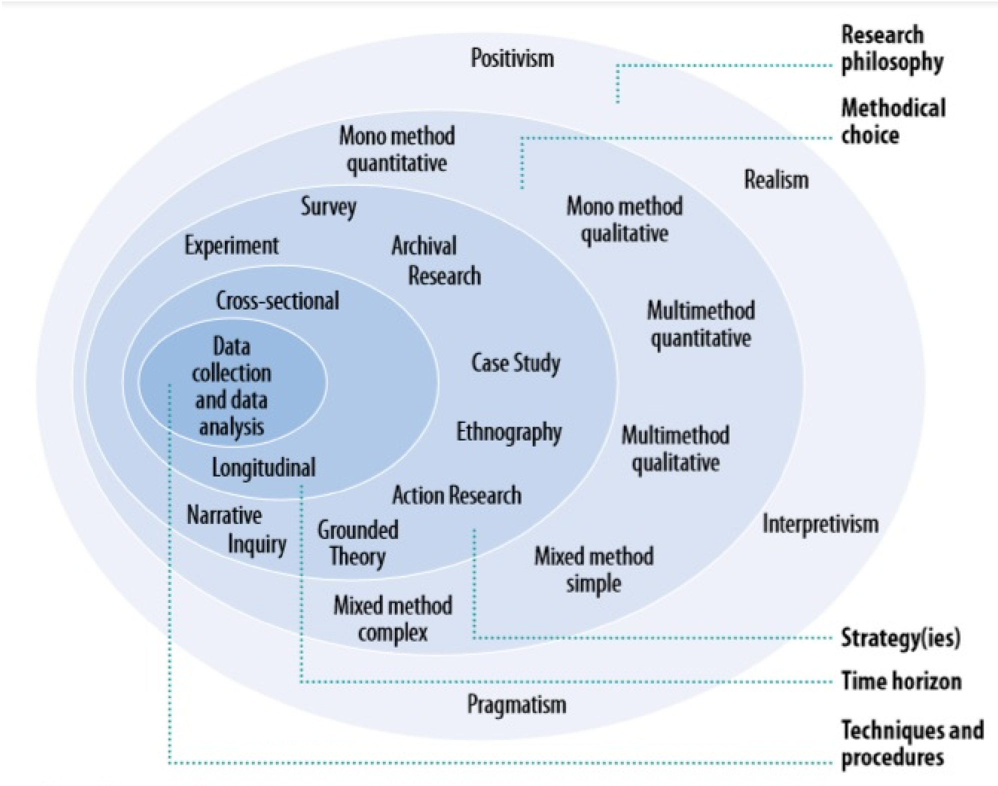



Figure 2.2: Saunders and Tosey's revised model of research design [131]

| Segment | Position | Reasoning |
|---|---|---|
| *Philosophy* | Positivism | The testing of structured data |
| *Method* | Multimethod Quantitative | Quantitative data subjected to two approaches |
| *Strategy* | Experiment | Experimental use of testbed |
| *Time* | Longitudinal | Data collected and tested continually |
| *Data Collection* | Testbed and SysML Model | Collecting input data and verifying output |

Table 2.1: Research approach contextualised

## 2.2 Research design

### 2.2.1 Primary research: The testbed

Research and discussion on related topics by other authors is the primary focus of Chapter 3, while Chapter 6 ties the literature review in with the testbed-based research from Chapter 4, with a view towards creating the proposed domain extension. Thereafter, Chapter 7 presents the domain extension itself, before it is implemented and then tested in Chapters 8 and 9.

This underlines the importance of the direct research presented in Chapter 4, as the insights gained jointly act as linchpin to the rest of this project. Without direct research it would not be possible to gain the needed confidence in the systems and procedures developed for inclusion in the domain extension. To accomplish this, a number of consumer IoT devices were installed on a dedicated Wi-Fi network with only two other devices present. The first is a mobile phone which hosts the control apps needed for interacting with most consumer IoT devices. The second is a single board computer (SBC) which replaces the typical all-in-one router setup normally found in homes and also allows for the interception and collection of network traffic. The data thus collected are added to by also inspecting the terms and conditions presented to end users and actively examining the functioning of control apps on the mobile phone. These data are then all compared to GDPR requirements by way of the CNIL's privacy impact assessment (PIA) for IoT devices [33]. A significant point of note here is that none of the devices or programs involved can distinguish between data relating to a real human and contrived data simply labelled as personal. As such, and since we are concerned exclusively with data flows, data storage, and data processing, there is no need to complicate matters by involving the data of a real person. Consequently, only artificial data are used and at no point will the private data of any natural person be involved.

The initial design called for the Wi-Fi network to be hosted on a Raspberry Pi 3B+ SBC and could have remained as such. However, during the course of the experiment, the hosting hardware was upgraded to a Raspberry Pi 4B. This has no material impact on the research and was purely a performance upgrade. The main function of the Raspberry Pi is to act as access node for the Wi-Fi network while also hosting software to intercept network communications. While the Raspberry Pi is itself connected via an Ethernet cable to a secondary network, the data collected are passed to a standard PC where software such as Wireshark is then used to perform further analysis. Since Raspberry Pi's have both wireless and wired network connections, it can pass data between the two, and perform Network Address Translation (NAT) as part of the process. This NAT-based option, as opposed to straight pass through, is needed for two reasons. First is the need to install software capable of performing a man-in-the-middle (MITM) attack[3]. An SBC that uses NAT makes for a significantly easier route to deploying a MITM attack. The second reason is the network settings on the second network the SBC connects to, since a straight pass through would expose the IoT devices to that network[4]. It is therefore also not surprising to see other researchers employing Raspberry Pi SBCs in this manner [101].

Network data captures will necessarily be in the form of PCAP files, though these will contain a fair amount of data that is not needed. In describing packet contents Miettinen et al. identifies 21 packet features which are presented in Table 2.2 [104]. These packet features are used by Miettinen et al. and others [19] as a prime component in automated device fingerprinting. Although we are not here concerned with the activity of device fingerprinting, there are some valuable lessons to learn from the work already cited. Specifically, that devices cycle through their most prominent actions in short order and that it is relatively easy to extract the data needed to perform the analysis. In short then, the most significant forms of data processing and collection by testbed devices, would be exposed early in the process. Furthermore, of the 21 packet features identified, we are only interested in IP, TCP, UDP, HTTP, HTTPS, and IP addresses.

[3]MITM attacks are discussed in more detail in Subsection 2.2.3

[4]This exact problem was encountered when testing a potential pass through setup.

| Type | Features |
|---|---|
| Link layer protocol (2) | ARP/LLC |
| Network layer protocol (4) | IP/ICMP/ICMPv6/EAPol |
| Transport layer protocol (2) | TCP/UDP |
| Application layer protocol (8) | HTTP/HTTPS/DHCP/BOOTP/SSDP/DNS/MDNS/NTP |
| IP option (2) | Padding/Router Alert |
| IP address (1) | Destination IP counter |
| Port class (2) | Source /Destination |

Table 2.2: Description of 21 packet features by Miettinen et al.

### 2.2.2 Device selection

Device selection would initially be limited to those already available via the Cyber Security Lab's selection of IoT devices[5], with the possibility of further changes as research progressed. As it turned out though, it was not only possible to get the full set of devices needed for the research, but this device selection yielded far more data than initially expected. The general criteria for selection was that the devices had to support Wi-Fi and/or Bluetooth communication and must have a related control app hosted on a smart phone. As discussed above, the testbed also has a smart phone to host those apps and a Raspberry Pi acting as router and point of capture for the data being transmitted to and received from, off-site sources. These off-site sources include cloud storage and other servers belonging to the data controllers or third-party processors. The devices were also chosen to reflect both the "starter kit" and mix-and-match approaches to consumer IoT. As such there are a group of devices from the same manufacturer which also share a common app[6], as well as other devices from other manufacturers each with their own control apps. The full list of devices are presented in Table 2.3 while Figure 2.3 shows them all in one picture for a better sense of scale and also includes a Raspberry Pi in the bottom left corner.

These devices cover popular categories in consumer IoT around health and well being, personal assistants, and data on the home environment. They all have associated control apps running on a smart phone where those apps are available for both Android and iOS. We opted for an Android device on the testbed since iOS devices are more difficult to compromise. On this account we did use a known weakness in Android 5 to advance the approach discussed in Subsection 2.2.3 and also tested using a "rooted" phone for a minimal OS installation aimed at reducing unrelated traffic on the testbed network. The former action proved far more useful than the latter as managing apps on the rooted phone was more cumbersome, while the reduction in superfluous data flows was too small to have a meaningful impact. One addition made though, was the introduction of a second fitness tracker. The Withings Go is from a European manufacturer and part of the "starter kit", the decision was made to introduce a second fitness tracker but from an unrelated US company. The device in question is the Shine 2 by Misfit, a brand owned by the US fashion and lifestyle company Fossil.

[5] At the University of Southampton.

[6] Three of the Withings devices share a common app but the fourth predates the current lineup and has its own app.


Figure 2.3: The testbed including Raspberry Pi

There are of course many other devices that might have been included herein, with one of the most prominent such options being so called “smart TVs”. We are however convinced that the testbed achieved a good spread of devices and related data points and that further additions would not have provided significant insight. Specifically on the point of “Smart TVs”, these devices do not make for a good fit with the testbed design and constraints. This is due to such TVs typically not having a related smart phone app from the same manufacturer as the TV, using exponentially more data than the rest of the testbed combined, and there being no clear indication that such an inclusion would yield significantly different insights to the rest of the testbed. On the point of data use, it should be noted that most modern “Smart TVs” host multiple streaming applications such as Netflix, Amazon Prime, and Disney Plus, implying that the use of private data on these apps, the interaction between them and the TV’s OS, and any associated third-party trackers would also become a necessary part of the research [155]. This would generate far more traffic than the Raspberry Pi could handle, would add a new but unique *type* of M2M communication to analyse, and present an environment where direct updates to the device are significantly more frequent that to other devices on the testbed. A full understanding of the data flows involved in the mentioned streaming services would also fall outside of the testbed’s proposed scope and require significant additional work to be performed [113]. In short then, we assert that including one or more “Smart TVs” would add significant complexity and additional work, necessitate changes to the testbed design to accommodate the additional data flows, and provide no clear advantage. This assertion was born out in the broad range of results eventually gained from the testbed, as presented in Chapter 4.

Although this chapter is concerned with the functionality and setup of a research testbed, a few higher level points regarding the testbed must be made. First, the unique nature of consumer IoT setups allows for small-scale testbeds to yield data on a wide range of issues. This relates

| Device | Name |
|---|---|
|  | Misfit Shine 2: Wearable fitness tracker |
|  | Withings Go: Wearable fitness tracker |
|  | Withings Thermo: Thermometer |
|  | Withings Body: Scale and health tracker |
|  | Withings Home: Camera, air quality sensor and baby monitor |
|  | TP Link Smart Plug: Remotely managed plug and energy sensor |
|  | Amazon Echo Plus: AI assistant and music player |

Table 2.3: Testbed devices list

to the mix-and-match, anything goes, nature of these setups. Lacking the rigorous vetting and uniformity one might expect from industrial applications, these setups allow for devices and systems from different suppliers, with divergent technologies and standards, to coexist on the same network. Effectively increasing the spread of data points that can be tested, as opposed to more homogeneous industrial applications. Second, what might be referred to as realism is not at issue. Placing the testbed in an actual home or having real-world data subjects use the devices and apps would not generate more or different data. This is due to the devices and apps treating all data as legitimate and lacking any ability to distinguish between real and artificial data points. Since we are interested in the manner in which these devices treat personal data and the devices accept all data entered as personal, simply because it was entered as such, no further steps aimed at "realism" are needed[7]. With the liberty afforded by such a system we can then proceed to record data on the nature of GDPR compliance and PbD in consumer IoT, both as a snapshot and over time.

As mentioned above, and discussed in Subsection 3.5.2, consumer IoT implementations are generally conducted in a haphazard manner by the data subject based on a range of personal preferences and including device pricing, advertising and linking to other devices. As such, having individual devices, a set of devices from one supplier, or a mix of devices and systems, are all outcomes which can be expected. We can therefore say with confidence that our chosen testbed

[7]None of the testbed devices are able to distinguish between synthetic data and authentically personal data.

devices fall squarely within the scope of functioning and composition for a consumer IoT implementation.

### 2.2.3 MITM testbed

All the devices on the testbed communicate with offsite servers to store data, process data, access updates, and register accounts. This M2M communication can go directly to the Raspberry Pi via Wi-Fi, or via Bluetooth to the smartphone and from there via Wi-Fi to the Pi, from where the connection to an offsite server is made. The testbed network is hosted via the Pi's Wi-Fi adaptor while its Ethernet port is used to connect to the Internet via the network access provided in the lab. Allowing data packets to flow over the Pi and out to the Internet is accomplished by way of network address translation (NAT), which essentially allows the Pi to act as middleman between the Internet and the testbed. The data flows on the testbed, device documentation, and any other relevant notes on device operation provide the data to be assessed using the CNIL's PIA. The use of NAT in and of itself does however not enable packet capture. To capture and subsequently analyse the data flows three widely used tools are employed, namely tcpdump for capture, MITMproxy for decryption and Wireshark for analysis.

Since all data communicated to, or received from, offsite servers must pass via the Pi a natural bottle neck forms. It is at this point that tcpdump is used to record all packets and write them to PCAP files for later export, with each file containing 6 hours worth of data. As a command line application, tcpdump is intended for manual launch with a large number of variables used to indicate operating parameters and the desired output. Since manual operation would be impractical, we used a purpose built script to launch tcpdump with our desired setting as a background process at startup. Due to variations between the Raspbian OS and other Debian distributions, this proved to be a better option as opposed to using systemd[8]. The script launches at startup and sets tcpdump up to capture Wi-Fi traffic (-i wlan0), to run continuously but report in six hour increments (-G 21600) and to save each increment in a specified directory with the time and date used as filename (-w /home/pi/caps/dump-%Y-%m-%d_%H:%M:%S). The full tcpdump command used in the script is:

```
tcpdump -i wlan0 -G 21600 -w /home/pi/caps/dump-%Y-%m-%d_%H:%M:%S.pcap
```

The four six-hour long PCAP files are then transferred via ssh[9] to a PC for analysis using Wireshark which is a purpose built packet analysis tool with a graphical interface. Each packet contains header data as well as the actual data being transferred and while the transferred data are often encrypted, the header data will not be as it is needed to route the packet. Inspecting these packets with Wireshark we can tell what type of data are being transmitted, if the data are encrypted, and who the recipients are. If the data were not encrypted, then Wireshark can also be

[8]A widely used but contentious Linux systems and services manager)

[9]Secure shell (ssh) allows for secure communications over unsecured connections.

used to gain even further insight into the packet contents. Naturally this provides a significant portion of the input needed to act against the CNIL's PIA as it enables us to inspect device data flows and contrast these to both documentation and GDPR[10] requirements.

For the initial run of the testbed only the packet meta data were used in analysis since almost all packets were encrypted. Though one notable exception is mentioned in the analysis[11]. To overcome this issue and decrypt the data, the second iteration of the testbed took a man-in-the-middle (MITM) approach.

A man-in-the-middle (MITM) attack refers to any number of exploits where an attacker inserts themselves between two or more legitimate parties, impersonates one or more of these parties and then intercepts and copies the data flows [37]. From there, depending on the aims of the attack, the data can be sent on altered or unaltered. To do so on the testbed, the MITMproxy software package was installed on the Raspberry Pi and used in conjunction with the smart phone. All HTTPS[12] packets coming from the mobile phone are routed via MITMproxy before being presented to NAT, or the reverse for incoming packets. Packets originating directly from the IoT devices bypass MITMproxy though since these devices were not compromised.

MITMproxy is a proxy server which in this case is set up to copy all the traffic routed to it before passing it on unaltered. To do this, MITMproxy must be able to unencrypt and re-encrypt data in real time. This is achieved by the proxy server appearing to one of the communicating parties as a valid certificate authority (CA). Doing so is easily accomplished by installing a MITMproxy issued certificate on a smart phone using Android 5 or older as OS. In these older versions of Android, user certificates enjoyed the same level of trust as system certificates, which is no longer the case from Android 6 on. For newer systems, the phone must be substantively compromised (rooted) after which Android debugging bridge can be used to add the needed certificate with root permissions. This process is rather tedious and varies from phone to phone, but can be even more involved and even prohibitively difficult for some devices. As such, only the phone was compromised and not the devices, leading to only HTTPS traffic coming from the phone apps being diverted to the proxy server. This setup allows MITMproxy to present itself to the phone as the off-site servers and *vice versa* for the actual off-site servers. This relationship is represented in Figure 2.4[13]. A final trick for MITMproxy to perform is that it not only has the ability to display the decrypted traffic in real time but can also capture SSL/TLS master key associated with the traffic. Importing this key into Wireshark also decrypts the PCAP data.

To further aid in our analysis, a smart phone based application was used. This app, Lumen[14], tracks data flows linked to the control apps on the smart phone with a specific focus on the leaking of private data and the presence of tracking software. An important aspect to the Lumen

[10]Naturally, the documentation will also be compared to GDPR requirements using the PIA to guide such comparison.
[11]Please see our notes on the Withings Thermo in Chapter 4
[12]HTTP traffic is already in the clear.
[13]Taken from: `https://docs.MITMproxy.org/stable/concepts-howMITMproxyworks/`
[14]The Lumen Privacy Monitor is available at: `https://play.google.com/store/apps/details?id=edu.berkeley.icsi.haystack&hl=en_US`

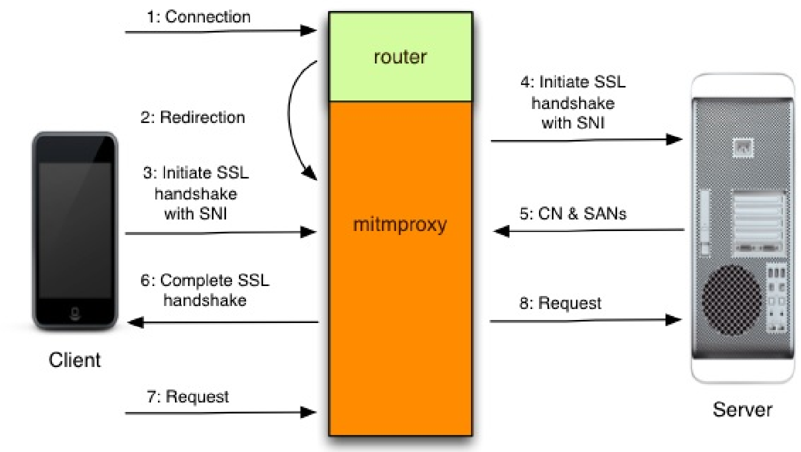



Figure 2.4: MITMproxy

app is that it routes traffic from the smart phone via a VPN to servers at UC Berkeley for analysis. Consequently, the Lumen app can not be used at the same time as the other analysis tools, since the VPN bedevils data capturing on the Pi, meaning that use must be alternated. The linking of tracking software to the IoT control apps and tracking the off-site servers the software connect to, which Lumen provides, is so valuable though that the inconvenience is entirely mitigated.

The steps involved in conducting the testbed-based research are as follows:

1. Set up Raspberry Pi including tcpdump script and MITMproxy integration
2. Link smart phone to testbed Wi-Fi
3. Install the Lumen app
4. Install control apps on smart phone
5. Initialise IoT devices on testbed
6. Conduct device setup on devices and control apps, agreeing to all terms and conditions[15]
7. Record all terms and conditions
8. Operate IoT devices and collect data
9. Export PCAP files to Wireshark
10. Include Wireshark analysis, terms and conditions, and Lumen results in CNIL PIA
11. Use insight gained from testbed and PIA as filter to produce our new methodology

[15] A fictitious persona is used.

### 2.2.4 Minor testbed and tool changes over time

Since the testbed was in use for more than three years some changes were inevitable. Fortunately none of these changes constitute a threat to the validity of the research and instead represent a refinement of tools and techniques over time, with the core testbed devices remaining unchanged.

The Raspberry Pi pictured in Figure 2.3 is a B 3+, which was used during the first year of the testbed's operation. It was, as previously mentioned, later replaced with a B 4, but since both functioned in exactly the same manner and ran the same software, there was functionally no difference other than speed and available memory. During this time the Raspbian OS was also renamed to Raspberry Pi OS[16] and upgraded to a new 64bit version. As an alternative to this setup, the OS was swapped to Kali Linux in order to take advantage of the diagnostic tools bundled with Kali. However, this was quickly reversed as Kali Linux, at least at the time, proved to be less stable and also yielded a number of issues for the implementation of network address translation.

## 2.3 Feasibility

The tools, both hardware and software, needed to conduct the work proposed herein are all well understood, well documented and for the most part, industry standard. The interaction between these tools is also well documented since the testbed deployed here is largely similar to that often used to investigate device and system security for consumer IoT implementations, where our focus only shifts from security to privacy and compliance.

The longitudinal nature of the testbed-based research is also not a particular problem since the devices used are all from major companies and could easily be replaced if they malfunctioned. The mobile phone, on the other hand, is only needed as a host device for the IoT device control apps and can consequently be substituted for any comparable device. The same logic also holds for the use of a Raspberry Pi as router and host device for packet interception.

Although the above addresses research feasibility, something must also be said about the overall feasibility of this project. In describing the extent of the challenge faced by system designers in the IoT, Alqassem and Svetinovic list the following three main points:

1. Information protection: Who has access, when to protect information, and what information to protect.

2. Integration risks: Integrating disparate technologies and standards leads to unforeseen interactions and risks.

[16]All official Raspberry operating systems (OS) are derivations of the Debian distribution of Linux, though the Raspberry Pi foundation take a "stability over innovation" approach to their releases.

3. Environmental issues: IoT implementations will, over their lifecycle, be subjected to environmental conditions other than those they were initially designed for.

Addressing these three challenges in turn and in reference to the project presented herein, we can conclude the following. First, when it comes to determining what is to be protected, when and from whom, our research will present two avenues for dealing with the information protection issue. These avenues are of course compliance and privacy-by-design. Second, using model-based systems engineering as the avenue for interpreting our research and formulating an eventual remedy, we are able to model, test and account for system wide actions and interactions making sure these adhere to the stated compliance and privacy requirements. Third, this modelling approach covers systems over their entire lifecycle and is able to change or update not just requirements but also related software systems or operator procedures.

## 2.4 Assessment of findings

Only tracking and investigating dataflows on the testbed, that is on and between the IoT devices and mobile phone, would not provide sufficient insight into compliance, privacy measures and the threats to both. Instead, dataflows to and from off-site servers must also be included, as well as the terms and conditions for device use presented to data subjects. Once all these data are captured and collated as described in Section 2.2.1 it can be assessed for GDPR compliance which forms a significant component of our work. To conduct this assessment reference is made to the framing of privacy issues provided by the SDM and LINDDUN as introduced in Section 3.8 and Subsection 3.10.3. To extract the needed data points though, we turn to the GDPR Privacy Impact Assessment (PIA) audit methodology for IoT devices and systems developed by the French data protection authority, the CNIL, which is introduced in Subsection 3.7.1.

The reasons for choosing LINDDUN and the CNIL's PIA are broadly similar. Both constitute excellent, well regarded, and well understood tools for niche tasks. In the case of LINDDUN, it was not only an early mover with regards to formalised privacy threat modelling, but has seen a number of revisions and remains relevant. The manner in which LINDDUN is structured, as discussed in the following chapter, also lends itself to inclusion in the domain extension. As such, using it in the assessment of the testbed findings is a logical step. With regards to the CNIL's PIA, it stands out from other options for being wholly in the public domain[17], directly covering the GDPR and including an IoT specific focus.

[17]Many other national regulators do not publish their audit methodologies.

# Chapter 3

# Literature Review

## 3.1 Chapter introduction

This chapter examines the Internet of Things and its application in the consumer electronics segment in greater detail than that presented in the introduction. The points developed here will also serve as input for Chapter 6 where a number of key building blocks for the new domain extension will be developed, before being implemented in SysML. This latter step can be found in Chapter 7. The primary research focus of this chapter then, is to answer the first and second sub-research questions which are as follows:

- RQ1: Which current compliance assessment, privacy enhancement or threat mitigation methodologies, strategies, and frameworks relate to consumer IoT implementations, or can be brought to bear on them?
- RQ2: Which of these methodologies would best serve to assess the testbed results presented herein?

Presented in Figure 3.1, this chapter is a review of prominent work and regulation dealing directly and indirectly with consumer IoT as it relates to privacy. Once these leading works are reviewed, a judgement can be made as to which provide the best tools for examining and assessing the results gained from the testbed, which is introduced in Chapter 4. Given that work presented in this chapter also directly impacts on the positions developed in Chapter 6, there is a need for a slightly more nuanced assessment of the actionable work presented below. By that we mean, any framework, methodology, regulation or other work that can direct action aimed at privacy or compliance in consumer IoT, will be graded on the metrics presented below, as this will be instrumental in shaping the domain extension developed later:

- A comprehensive approach covering both technical and legal considerations.

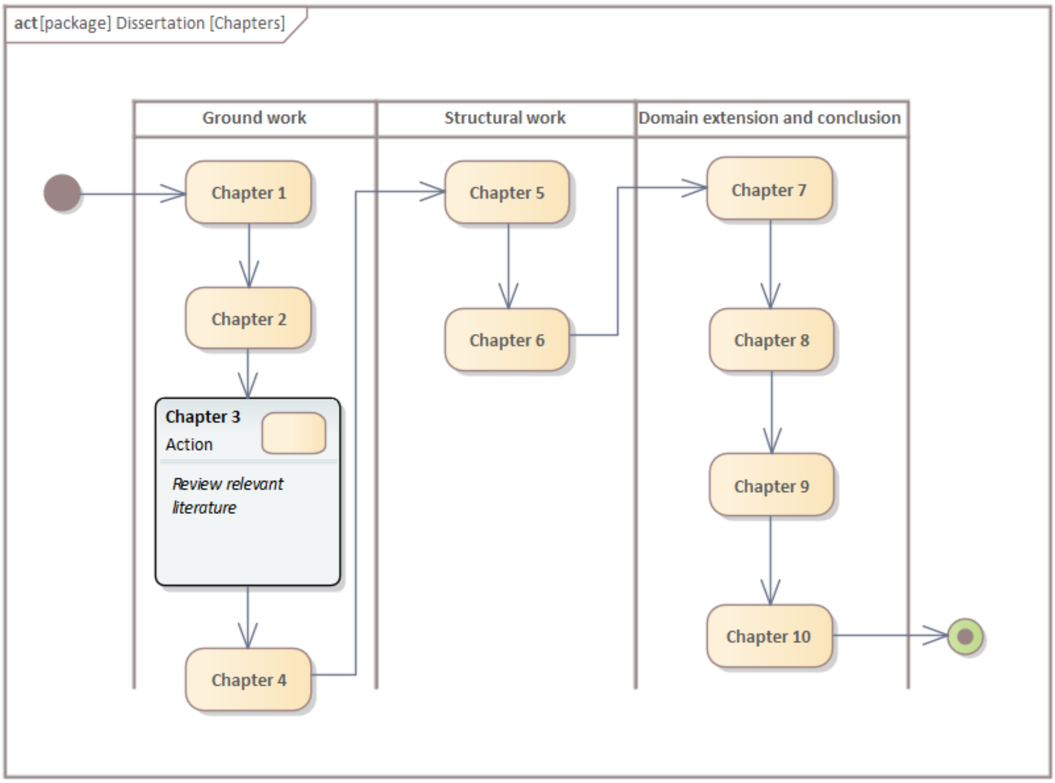



Figure 3.1: Chapter three research focus

- Focused on implementation as opposed to only assessment or suggested courses of action.
- Focused on auditing outcomes.
- An IoT specific focus.
- The use of data flow diagrams or other measures to track data flow.
- Privacy by design is the primary focus.
- Systems modelling using a language currently in use in industry[1].
- Can the specifics be represented in a standard systems modelling tool.

The eight metrics listed above are each intended to touch on a critical aspect of our proposed solution to the dual challenge of compliance and privacy by design in consumer IoT. This might seem to suggest that a certain answer is presupposed both in this discussion and in the testbed results presented in Chapter 4. Instead though, these metrics were determined before development of the testbed, since they directly elicit the details needed to assess the work reviewed below. This is not only for the purposes of the discussion in this chapter and Chapter 6 but also crucially as ground work for our proposed solution. We see this by taking a closer look at each of the metrics.

[1]One of the primary reasons we will opt for the use of SysML is its industry application and standardised tooling. Any system which adheres to these requirements will have an automatic advantage.

### 3.1.1 Comprehensive approaches

Through the investigation to follow, building on the work presented in Chapter 1, we will see that both compliance and privacy by design are woefully inadequate in consumer IoT. It will also be pointed out that although these are separate issues, compliance and privacy are related through both privacy preserving legislation and more focused data protection legislation. It is therefore clearly advantageous to address both issues, as opposed to opting for just one or the other.

### 3.1.2 A focus on implementation

There are, as one would expect, an extensive range of methodologies, frameworks and other approaches to dealing with both privacy by design and compliance. However, we are explicitly focused on developing a solution which not just frames or discusses the relevant issues, but which provides a directly implementable solution. As such, we must establish if the literature reviewed not only proposes a remedy but guides the implementation thereof. For instance, stating that an impact assessment must be performed and recommendations acted on, is a remedy. Going on to describe how to perform such an assessment, how to grade the results and how to act upon those, is implementation.

### 3.1.3 Auditing outcomes

Auditing is a key component of the work presented herein since it is the mechanism for demonstrating compliance. Literature which targets compliance, specifically in relation to privacy legislation or data protection, will be important. This not only includes work which directly deals with auditing, and audit best practice, but specifically also with audit assessment against the GDPR since it is the primary piece of legislation referred to.

### 3.1.4 A focus on the IoT

It might seem redundant to explicitly note IoT as a focus area given the topic of this project, but there is an important distinction to be made. This is between literature which explicitly focuses on the IoT and that which has a different focus but can be brought to bear on the IoT. A case in point is the audit methodology by the CNIL which is developed as a general privacy impact assessment against the requirements of the GDPR, but which is then extended to IoT applications by the CNIL itself.

### 3.1.5 Data flow tracking

There is a lively academic debate around the use of data flow metrics, specifically data flow diagrams [5]. We will not delve too deeply into that, since the interest here is a predetermined need to understand system data flows. Without that understanding GDPR compliance is simply not possible. As a result, even literature dealing with other topics, but which has some relevancy, must still be assessed against this metric. This concern is also not limited to the GDPR since the primary mechanism by which IoT devices impact on the privacy of data subjects is by collecting, processing and communicating data relating to natural persons. Data flows therefore remain critically important even if the GDPR is not being considered.

### 3.1.6 Privacy by design as primary focus

Although a significant portion of the literature reviewed below will focus on PbD, not all of it will. Also, within PbD there are a number of different focus areas and approaches. As such, PbD will be used to not only record if it is indeed an area of focus for the literature under review, but also to list any significant other approaches.

### 3.1.7 Modelling language use

As outlined in Chapter 1, we intend to fold the insights gained from this literature review, along with other work, into a new methodology for compliance and PbD within consumer IoT. This will take the form of a SysML domain extension. Although SysML is our chosen modelling language, any comparable work using other languages could provide valuable insights and should therefore be examined.

### 3.1.8 Standardised language use

The potential for our work to be deployed in industry, using standard tooling and without much additional training for practitioners, is one of the major contributions we target with the work presented in Chapter 7. Accordingly, specific note must be taken of any literature which deals with such industry application. This point goes beyond the use of any specific modelling language, since unhindered industry use rests not just on the use of a well understood and broadly used language, but also on the standard use of that language. In other words, we are specifically interested in introducing new capabilities in to the language used, without changing the functioning of the language itself.

To fully engage with the eight metrics introduced above, we will first have to establish additional background to both privacy by design (PbD) and the IoT in general. Once this baseline is established we can examine the interaction between the two and make specific reference to the eight metrics.

## 3.2 Knowledge representation and interpretation

Knowledge representation and the gaining of insight or value from that representation, is an important activity in many fields, with systems design being no different. A popular route in the development of new domain specific languages, for instance, is ontological domain analysis [141] and the use of ontology languages such as OWL [84] and the query language SPARQL. The World Wide Web Consortium (W3C) refers to these and other related elements as semantic technologies[2] In Figure 3.2 we present the semantic web layer cake[3]

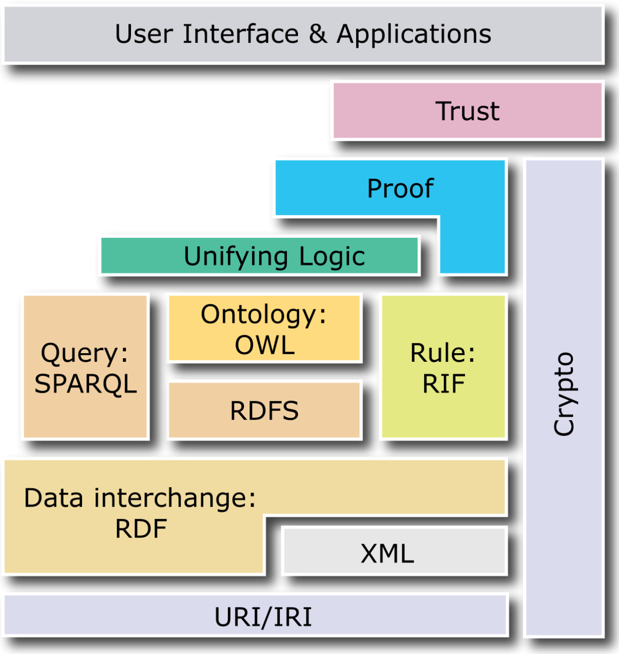



Figure 3.2: The Semantic Web Layer Cake

Although widely used across many fields, including domain modelling, these technologies are not an exact fit for our needs herein. This is in part due to MBSE already providing an avenue to interrogate models, as demonstrated in Chapter 9. It is therefore not needed to build new external ontologies or to use languages like OWL/2. An additional reason that this is not needed here, but is used in some domain specific applications, is the nature of the domain specific work. We are focused herein on producing a domain specific extension for SysML, as opposed to a new domain specific language. Accordingly we do not need to check the language itself for coherency, for which an externally constructed ontology would be a good option. We are therefore open to perform validation and verification tasks using SysML directly. These concepts are further discussed in subsections 3.2.1 and 3.2.4, as well as Chapter 9.

[2]A full listing is available at `https://www.w3.org/TR/?tag=data`, by selecting the tags options and leaving the other options blank.

[3]Image by the W3C, taken from `https://www.w3.org/2007/03/layerCake.svg`

### 3.2.1 Ontologies

The notion of an ontology is defined by Gruber as "*an explicit specification of a conceptualisation*" where such a conceptualisation is shared and formal[63]. In this we understand "formal" to mean machine readable, "explicit specification" indicates that all concepts are defined, and "conceptualisation" is an abstraction of that which is being represented. This constitutes the concept of an ontology favoured by the W3C and which forms part of its semantic web stack with a well known example being the Good Relations[4] web ontology for e-commerce.

As such, an ontology contains both a vocabulary and set assumptions about the meaning of that vocabulary, it allows for a shared understanding, meaningful querying of the ontology, and the reuse of elements or the whole. Similar points are made below and later on regarding the formalism with which SysML can convey meaning, including the direct use of ontologies. Please note the plural form here as these relate to the specific model described in SysML as opposed to SysML itself. As shown below, the systems engineering understanding of ontologies is also somewhat different to that in semantic web applications.

### 3.2.2 Formal model checking

The representation and interpretation of knowledge inherently implies that, where such knowledge is applied in a structured manner to reach a set goal, interpretation should include some sort of qualitative assessment. There are of course an extremely wide range of approaches available, but within software design, formal model checking is the preeminent approach. Though not the focus of the work presented herein, brief mention must be made of it if only to differentiate formal model checking from the model analysis discussed below.

Given the world-wide proliferation of computing systems, it is no wonder that the societal and financial cost of errors in these systems can be substantial. With the cost of computer errors in the US alone estimated to be in the high tens of billions it is imperative that such systems be assessed, with errors located and mitigated before system deployment, which is the aim of model checking [52]. Model checking is defined by Kim Guldstrand Larsen in the foreword to Principles of Model Checking by Baier and Katoen as *a formal verification technique which allows for desired behavioural properties of a given system to be verified on the basis of a suitable model of the system through systematic inspection of all states of the model.*

Where Baier and Katoen themselves go on to state that model checking is the applied mathematics used for the modelling and analysis of computing systems. Although their discussion further distinguishes prominent approaches within formal model checking, they are not further explored since, as mentioned earlier and again in Chapter 6, formal model checking will not be used in this work.

[4]See `http://wiki.goodrelations-vocabulary.org/Vocabularies/`

### 3.2.3 Model analysis

MBSE is primarily focused on providing to all stakeholders, a single version of the truth relating to the system or systems being modelled. This notion is further explored in Chapter 5. As such, the system model provides what Delligatti calls *a central repository for all design decisions*. This may indeed include the generation of software systems which are subject to the type of formal model checking described above. However, such checks are not conducted against the system model itself, but only one of its components. This then leaves open the question as to the metrics to use when checking the system model itself.

Unfortunately there is a tendency for terms to be used and reused to indicate different things. The first step to counteract this is use of "model analysis" to indicate tests against a systems model, as opposed to "model checking" which relates to formal modelling. The second is to pin down definitions for Verification and Validation, as these concepts are widely used but in terms of systems engineering there is a set and well established definition of each.

Going back to work done by Barry W. Boehm in the early 1980s we find Boehm writing on the topic of verification and validation for software[5] requirement and design specifications, and introducing two sets of definitions for each. The first set of definitions serve to clarify Boehm's reasoning on the topic, with the following quote taken from Boehm:

- **Verification.** The process of determining whether or not the products of a given phase of the software development cycle fulfil the requirements established during the previous phase.
- **Validation.** The process of evaluating software at the end of the software development process to ensure compliance with software requirements.

Though the area of focus here is software requirements, Boehm sets out both the purpose of, and the need for, verification and validation. Both are essential for ensuring that the system ultimately delivered is fit for purpose in every sense, with verification used to ensure that the system is built according to the build instructions, while validation ensures that the system finally delivered also adheres to the initial system requirements.

Given the clarity of this formulation it is not surprising to find it persisting in the field of requirements engineering [120]. This also explains the use of Boehm's formulation in other related fields including systems engineering generally and model-based systems engineering [50]. Though such acceptance does not mean that the topics of verification and validation are closed as areas of academic enquiry [160], it does mean that incorporating Boehm's work is safely within accepted practice. Doing so also allows for the use of Boehm's second and somewhat more elegant, definitions. These are again taken from Boehm and read as follows:

[5] Boehm has also published work explicitly dealing with systems engineering. Though the reference here is better suited to the discussion at hand.

- **Verification** "Am I building the product right?"
- **Validation** "Am I building the right product?"

### 3.2.4 Ontologies in modelling languages

Modelling languages can have a clear and well-formed ontology included or drawn in for domain specific application. The importance of this is stressed by Holt who refers to ontology as the cornerstone of MBSE, providing the following explanation in four parts:

- **Domain-specific language:** Systems engineering can not succeed if concepts and terms are not well defined.
- **Viewpoint contents and structural foundations:** Models are composed of views and the integrity of these views are derived from their associated viewpoints[6]
- **Model consistency:** Consistency is required for all model elements to jointly act to convey meaning. This is achieved through the joint use of a spoken language (SysML[7] in this case) and an ontology containing the domain-specific language.
- **Traceability:** A system model must not only present well defined and connected elements, but the artefacts produced by that system must also be traceable back to individual system elements. This is not only crucial for error tracing but also for deploying future changes.

With reference to the third point above, it should be noted that although the we present a single domain extension in Chapter 7, the underlying ontology is drawn from two sources namely, the GDPR plus the CNIL PIA, and the LINDDUN threat modelling methodology. These are all introduced in this chapter. Furthermore, the relationship between SysML as spoken language and the combined ontology presented in Chapter 7 is that SysML notation is used to structure, present, and guide the application of the ontology. This also ultimately allows for the resulting domain extension to be tested for coherency in general and more specifically for verification and validation tests to be carried out. In short, without an ontology we can not test if the domain extension functions correctly or is the right domain extension for its stated purpose, but without a language to provide access to it we can not test against the ontology. Hence the need for both.

## 3.3 Taxonomies of privacy

A single concise definition of privacy is frustratingly elusive [137], with positions readily opined ranging from "the right to be left alone" [24], to requiring that data subjects should have full and

[6] Please see the discussion of views and viewpoints in Section 5.2.
[7] SysML as a language for conducting MBSE is further explored in Chapter 5

sole control over their personal information, and even extending to issues of personal identity [3]. No wonder then that Daniel Solove describes it as "*a concept in disarray*" [138]. A significant part of the challenge is that notions of privacy are often culturally bound or deeply personal, further muddying the waters when searching for a common definition. These notions of privacy are of course also changeable, and rapidly so. One study found a rapid shift in what Facebook users were willing to share publicly, with those willing to share certain details decreasing from around 80% to just 40% over a period of six years [1], as presented in Figure 3.3.

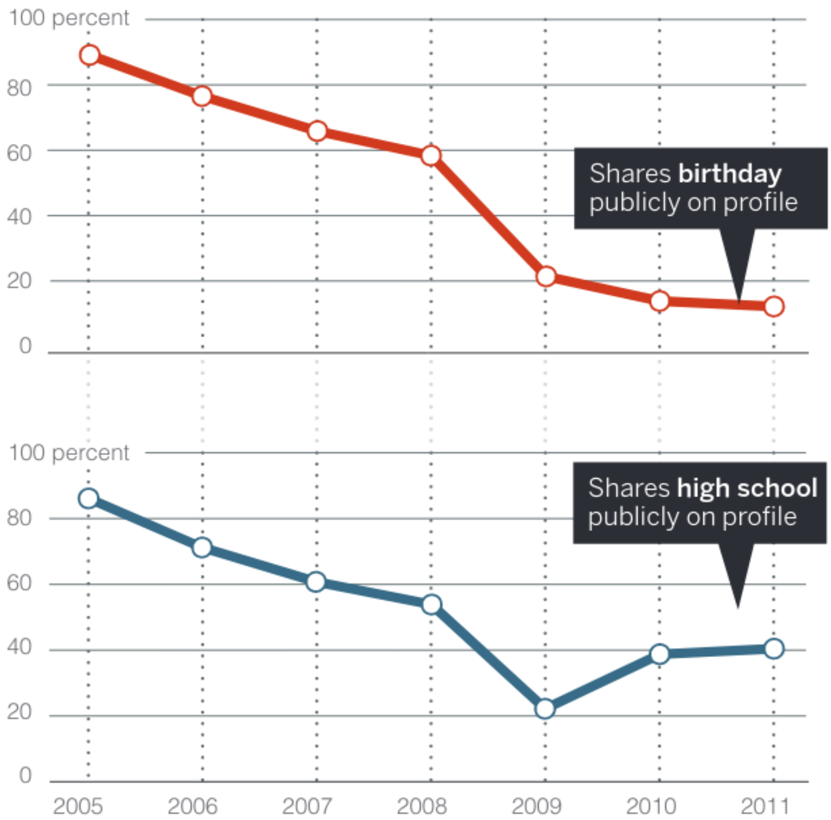



Figure 3.3: Disclosure behaviour on Facebook [1]

These challenges are, however, also not new. There is an extensive history to the legal consideration of technology related impacts on privacy with, for instance, the US Fair Information Practice Principles (FIPPs) dating back to 1973 [24]. The FIPPs are also clearly echoed in the later work of Ann Cavoukian and the GDPR, both of which are discussed in this chapter. The main tenets of the FIPPs are as follows:

- **Collection limitation:** no superfluous data collection
- **Data quality:** prohibition on the collection, processing, and storage of inaccurate data
- **Purpose specification:** obtain informed consent from data subjects
- **Use limitation:** obtain informed consent from data subjects for new processing purposes
- **Security:** protect data after collection

- **Openness:** transparency towards data subjects
- **Individual participation:** data subjects should have access to their data and the ability to rectify mistakes
- **Accountability:** the party collecting the data is responsible for the handling thereof

What is notable, and establishes some common ground, is that democratic governments generally strive to protect the privacy of the citizenry, with such protections aimed at the prevention of what is seen as clear harms. That is to say, privacy breaches are generally seen as harmful to the individual, even though the exact definition of privacy might be variable [116]. This does, however, open two avenues to approaching privacy, the first is to default to the position of direct stakeholders, which is an often used approach and will be further discussed in Subsection 3.11.1 and Section 7.7, with specific reference to Subsection 7.7.1. The second is to adhere to the legal requirements around privacy protection in the jurisdictions within which a product or service is provided. While the second avenue is of course a legal requirement and must be adhered to, this does not mean that the first is optional and we contend that following both avenues during the full lifecycle of any product or service will yield better results than just following one, which is a major tenet within the field of privacy engineering [69] and is further discussed in Section 3.6.

Understanding privacy as a social good, irrespective of the avenue taken, establishes the need for preemptive privacy protections in any system or device which can directly impact on the privacy of the citizenry. This takes the form of privacy by design (PbD) [94] and few sectors fit this bill for a broad privacy impact as clearly as consumer IoT. Before delving into PbD though, a lower level tool, the privacy taxonomy, should be covered. The utility of taxonomies in general is clear, but can be particularly useful in analysing developing fields such as the IoT [44]. In the following two subsections we will present a widely cited and well regarded general privacy threat taxonomy, followed by an IoT specific taxonomy.

### 3.3.1 The Solove privacy taxonomy

One of the most widely cited privacy taxonomies is that of Daniel J. Solove which, at its core, seeks to establish the dividing line between the private and the public [1]. With a foundation in American tort law, Solove's taxonomy classifies privacy threats by way of sixteen threats under four groups [137]. Although the taxonomy substantially predates the GDPR, there are ample points of overlap and some notable divergences. A detailed discussion of these will not be provided here as referencing the taxonomy's performance in describing consumer IoT devices is far more useful. The full taxonomy is as follows:

1. Information Collection
    (a) Surveillance

   (b) Interrogation

2. Information Processing
   (a) Aggregation
   (b) Identification
   (c) Insecurity
   (d) Secondary Use
   (e) Exclusion

3. Information Dissemination
   (a) Breach of Confidentiality
   (b) Disclosure
   (c) Exposure
   (d) Increased Accessibility
   (e) Blackmail
   (f) Appropriation
   (g) Distortion

4. Invasion
   (a) Intrusion
   (b) Decisional Interference

In using this taxonomy to assess the exact list of devices on our testbed[8], Thorburn et al. found that some of these threats, such as blackmail, simply did not apply to consumer IoT at all. Other threats, such as intrusion, yielded anomalous results. Since IoT devices perform a measure of surveillance on both the data subject and their environment as a matter of course [6], disabling such surveillance also renders the device inoperable, with some notable exceptions such as the smart plug in our test bed which remained functional, though pointless. For the other threats, the analysis did not bode well with only confidentiality breach, exposure, and increased accessibility not realised in any of the devices. However, the nature of these three threats make them highly unlikely or even impossible to detect on a testbed of the type used herein.

We can safely assume that most consumer IoT implementations would fair similarly badly when measured against the Solove taxonomy. Although this would appear to be congruent with the long list of privacy failings pointed out in chapters 1 and 4, it is equally, if not more so, a function of this taxonomy being badly suited to the task at hand. Since the collection, processing and dissemination of private data are foundational activities in consumer IoT implementations, a negative outcome under an unmodified Solove taxonomy is a foregone conclusion.

[8] Please see Chapter 4

### 3.3.2 IoT privacy taxonomies

A worrying number of papers on the topic of IoT privacy taxonomies seem to treat this as a subsection or junior partner to security in the IoT as opposed to a discipline in its own right. Per example, Alsamani and Lahza sets out to develop a "*taxonomy in order to categorize IoT's objects, so security and privacy issues would be fully addressed*" [7]. Not only is this goal highly questionable, but the subsection dealing with privacy is entirely constituted out of the four top level groupings in the Solove taxonomy. This latter point is particularly problematic, given the specifics of our discussion on Solove's taxonomy and its application to the IoT.

In a similar reference frame, but taking a different approach, Babar et al. develop their Security Model and Threat Taxonomy for the Internet of Things to be two separate documents. Based on an analysis of IoT properties, use cases[9], and high level security requirements, the authors formulate a security and threat taxonomy. From this they then develop their security model. The taxonomy holds that the IoT has a threat profile all its own and that this requires detailed assessment. Specifically, environmental factors should be assessed with an eye to the development of mitigation strategies.

The following threats are identified:

- Identification relates to the device, user, and session identifiers. Key factors include authentication, provisioning[10], and authorisation.
- Communication deals with direct attacks intended to interrupt communications.
- Physical threats relate to direct device access by attackers.
- Embedded security refers to threats such as data tampering, side channel attacks, etc.
- Storage management relates to confidentiality, integrity, and possible impacts on user identity.
- Dynamic binding deals with the naming of system entities and services, including the mechanisms for doing so.

The threats proposed above are presented by Babar et al. in Figure 3.4. From these threats and via a discussion of related work and current techniques for addressing some of the concerns highlighted, the authors then develop a security model for the IoT.

This model is presented in the form of a cube which interrelates three primary elements, namely trust, security, and privacy. This structure allows for the interrelation of all three primary elements to be displayed on a single diagram. This is also an ideal mechanism for dealing with the

[9]Referred to by the authors as objectives.
[10]The deployment, configuring, and management of systems.

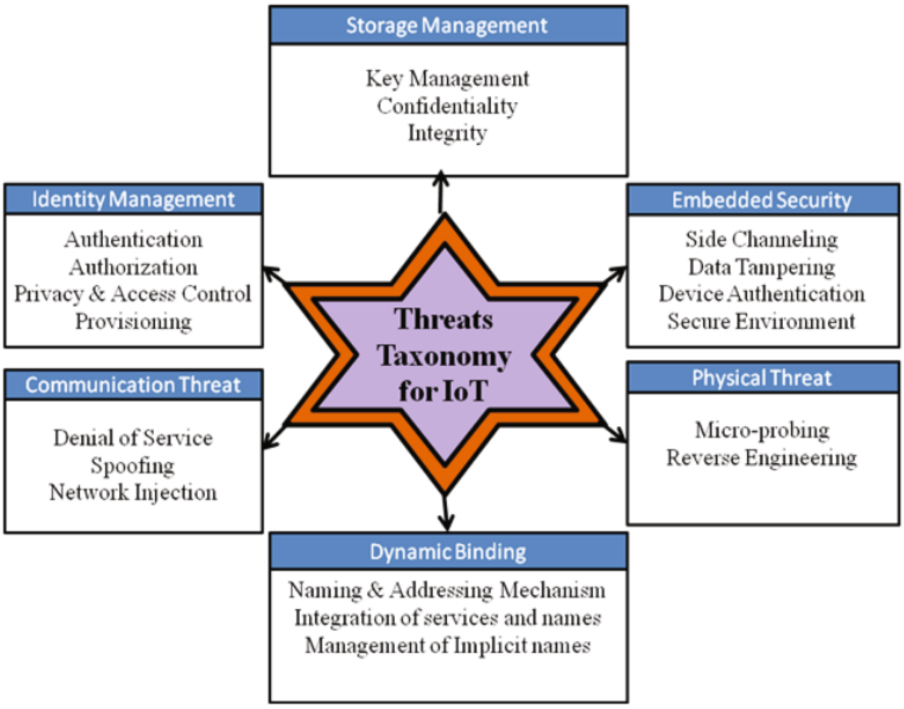



Figure 3.4: A threat taxonomy proposed by Babar et al.

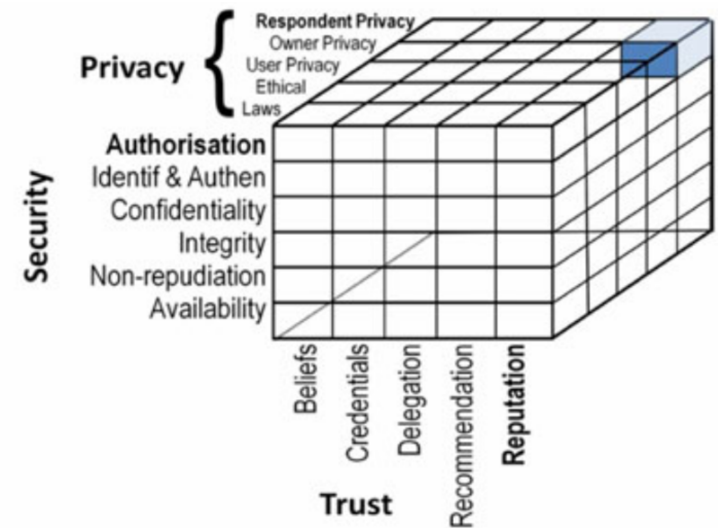



Figure 3.5: The IoT security model proposed by Babar et al.

high level of interconnectedness between people, services, and devices in the IoT[11]. Extending this notion further, it is noted that the data used for various actions and services in the IoT are often composite in nature, that is drawn from multiple sources and worked to perform certain functions. This is also reflected in the use of a cube structure to relate elements. Ultimately then, the type of amalgamated data needed to grant or reject access to other data or services will speak to security, trust, and privacy, which is the guiding notion expressed in Figure 3.5.

[11]This insight on the part of the authors has echos of the IEEE definition presented in Chapter 1, albeit not nearly as broad.

The threats identified in the taxonomy are security threats and relatively superficial at that. Although threats such as data tampering are identified, they are not explicitly formulated in terms of their privacy implications. Similarly, storage management is highlighted in terms of potential privacy impacts but only in as much as these are realised through security failings. Also, the authors explicitly describe their work as laying the foundations of "*an integrated systems approach for security and privacy in the Internet of Things*" [11].

In addition to the above issues, there is also a significant conceptual challenge in using the work described above to facilitate a "by design approach". That is the author's explicit insistence that the issues at play are best dealt with as a management issue, as opposed to taking a "by design" approach. In our estimation such an approach will inherently provide weaker results and is incompatible with the line taken herein. Ultimately then, a new taxonomy is needed. We present such in Section 6.3.

Comparing Security Model and Threat Taxonomy for the IoT proposed by Babar et al. to our defined metrics we find the following:

1. Not a comprehensive approach.
2. Does not implement remedies.
3. Auditing for compliance not mentioned.
4. IoT is the only focus area.
5. The use of DFD's is not specifically mandated, though not excluded.
6. PbD is not a focus area but IoT privacy threats are, though at a subservient level to security threats.
7. System modelling is not touched on.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.4 Introducing privacy by design

The development of PbD as a systems engineering approach can be traced to the work of Ann Cavoukian and others in 1995, entitled "Privacy-Enhancing Technologies: The Path to Anonymity" [77]. At its core, this report proposed that the integrity of digital communications could be preserved within a system which is designed to advance both privacy and anonymity [23]. Cavoukian, who previously served as the information and privacy commissioner of Canada, continued to develop these principles and subsequently proposed seven foundational elements to PbD [28]. These, with their main points in indent, are:

1. Proactive not Reactive; Preventative not Remedial

   *Clear commitment to privacy; methods to recognise poor designs and practices*

2. Privacy as the Default

   *Purpose specification; collection limitation; data minimisation; use, retention, and disclosure limitation*

3. Privacy Embedded into Design

   *Principled approach to embedding privacy; detailed privacy impact and risk assessments; privacy impacts demonstrably minimised*

4. Full Functionality – Positive-Sum, not Zero-Sum

   *Full functionality not impaired; innovative positive-sum; interests and objectives clearly documented*

5. End-to-End Security – Lifecycle Protection

   *Security; applied security*

6. Visibility and Transparency

   *Accountability; openness; compliance*

7. Respect for User Privacy

   *Consent; accuracy; access; compliance*

It would be difficult to overstate the measure of influence Cavoukian's work has had, and is still having, on the field. This can be seen when delving into details of her seven foundational principles as indicated by the indents above. These indented areas of focus read as a road map for subsequent work in PbD, not only for private sector and academic contributors, but also legislative efforts. This will become evident in the following overview of the pertinent literature but we will draw the readers attention specifically to the German SDM[12] and GDPR for their focus on collection and purpose limitations, privacy impact considerations, and recognition of the critical importance of a design led approach. This is similarly visible in other connected areas, such as threat modelling against privacy threats. On this topic, Wuyts et al. stated that PbD promotes "the introduction of privacy-focused activities in the early stages of the software development life-cycle" [167]. The work of Wuyts and other contributors will be examined in greater detail in subsections 3.10.3 and 6.5.

[12] Although the German Standard Data protection Model (SDM) is linked to the GDPR it is a significant and independent take on PbD and data protection. The SDM is discussed in Section 3.8.

## 3.5 Introducing the IoT

### 3.5.1 IoT reference models

The IEEE definition of the IoT presented in Chapter 1 is broad and sociotechnical in nature, which is significant since it not only points to the unprecedented scale of the IoT's impact on human endeavour and privacy, but also underscores the need for further work to delimit the scope of any project within the IoT. As a first step regarding this latter point, Chapter 1 also presented six elements of the IoT. To further refine our work and move towards the specific IoT implementation of interest here, we start by first considering a general reference model for the IoT. For this we turn to Ziegeldorf et al. [169] who's reference model, as presented in Figure 3.6, provides a basic and high level depiction of the IoT. Real-world implementations, such as that documented in Chapter 4 operates on a significantly more granular level and would require additional tools to describe accurately, though this reference model remains a good point of departure.

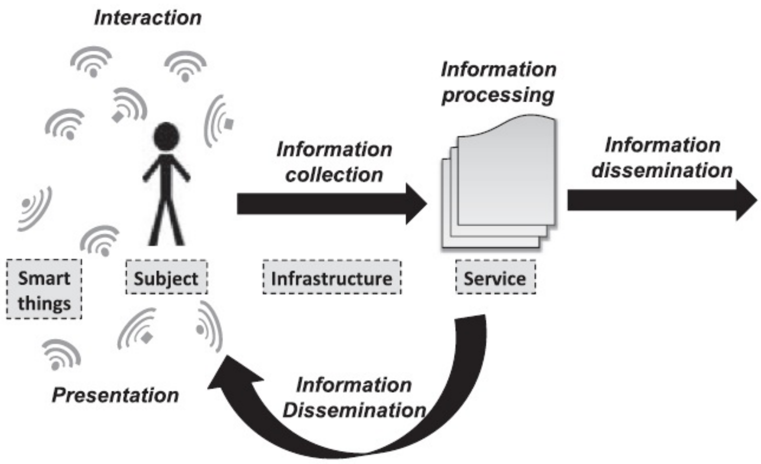



Figure 3.6: Ziegeldorf's IoT reference model [169]

Some takeaways from this reference model are 1) the positioning of the (data) subject within the IoT sphere, 2) the divide between smart things and other infrastructure and 3) the flow of data away from the subject. All three of these points are of critical concern for both privacy and compliance when dealing with the IoT in general, but even more so in the home. What is lacking though is the aforementioned granularity needed to fully unpack the challenges already introduced in Chapter 1, but also a focus on how these issues specifically impact on privacy and compliance. More often than not, challenges in the IoT are framed in terms of the related attack surface and security concerns, this is of course understandable given the fact that deeply personal data might be stolen [122], or in a worst case scenario with medical devices, lives might even be threatened [157]. What is needed therefore, is a more succinct focus on the principles and practice of data protection, starting with the former. For which a robust but focused reference model is needed as base.

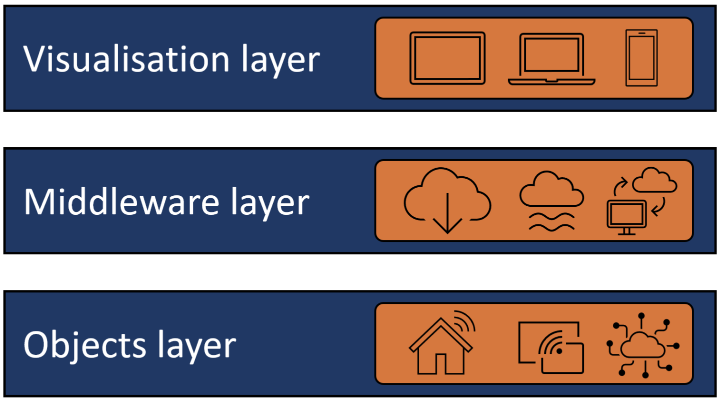



Figure 3.7: IoT system model

The model presented by Ziegeldorf et al. is of course just one of many, with competing offerings generally structured around a layered representation of the IoT stack, with reference to devices and functionality at each layer [31]. In this we opt for presenting the IoT system model via three top layers with a number of subdivisions. Doing so allows us, in Chapter 6, to build a new taxonomy which classifies by type, then function, and finally threat, whilst not encumbered by a near endless heterogeneous mix of devices and technologies present in the IoT [97]. These three layers are as follows:

1. **Visualisation layer:** The visualisation layer is tasked with presenting middleware layer functionality to the user via web, mobile or other appropriate applications. User functionality is realised at this level and includes system control, monitoring, and data visualisation.

2. **Middleware Layer:** This intermediate layer presents a service oriented architecture (SOA) which contains five linked but distinguishable services. These are applications, data abstraction, data storage, fog computing, and connectivity. Since cloud computing relates to both data storage and data abstraction, it would also be included.

3. **Objects Layer:** Edge devices such as sensors, actuators and controllers, literally the things in the IoT, are located at this layer. These objects collect data from the environment, communicate those data to higher layers and act on instruction from those layers.

### 3.5.2 Market segmentation for IoT applications

In Chapter 1 we addressed the measure to which the IoT is inherently heterogeneous. This is however not restricted to technologies and standards used, but also to the market segments targeted by IoT vendors. These notable segments include medical, industrial, and military applications. There is also the possibility that a device might find application in more than one of these segments or might purposefully operate across these segments. For the sake of clarity, we only regard IoT devices as being in the consumer segment if the data subject is in direct control

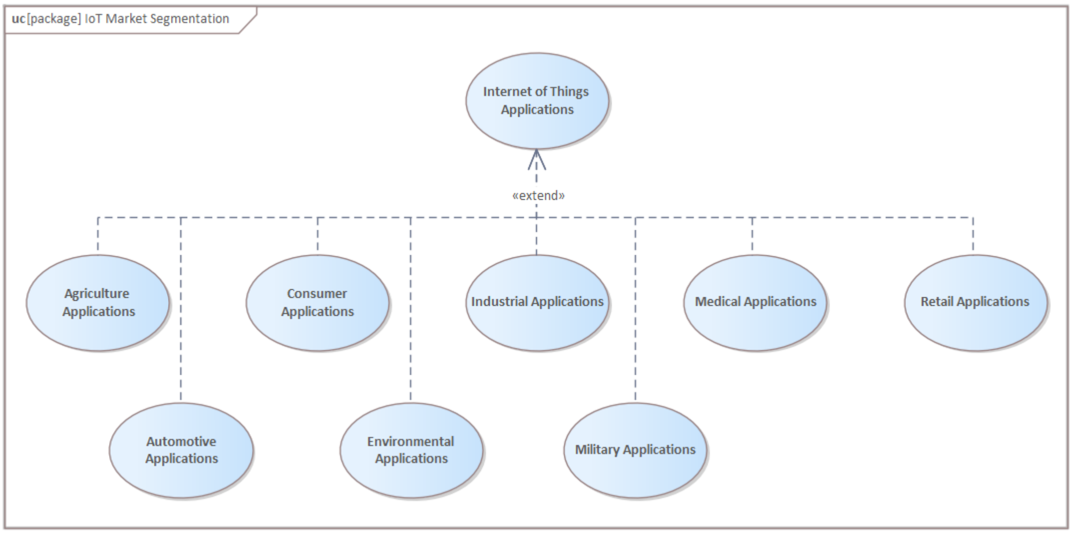



Figure 3.8: IoT market segmentation

of the device, including the decision to start and to stop using the device. As such, devices such as networked pacemakers, insulin pumps and the like are outside of our scope while devices such as fitness trackers are inside. This does not mean that practitioners in industry can not apply the work presented herein to these excluded devices, but rather that we are excluding such devices from our research as they fall outside our defined problem space. Although not using the same approach to device classification, Chanal and Kakkasageri provides a clear high-level overview of the different IoT market segments as presented in Figure 3.8.

These market segments and the prominence of certain devices within them is however, also transitory [98]. This is due to the IoT in general still growing, changing, and maturing. One result of this is the hodgepodge consumer offerings available, including standalone devices, starter kits [39], and add as you go options.

Although consumer, industrial [164], and medical [133] IoT devices all present with multiple failings, the nature of their deployment impacts privacy outcomes. This is to say, for industrial and medical applications it is typical to find a larger number of the same device(s) deployed by qualified system administrators. For consumer devices there is a mix of single devices deployed, often in an ad hoc manner, and by the general public. In amongst this mix of approaches, growing device popularity, and entrenched heterogeneity, it is also no surprise that consumer IoT is characterised in equal measures by its growth and its low quality privacy and security implementations [90]. These factors jointly, also support our previous choice for consumer IoT devices as focus area over industrial or medical applications.

### 3.5.3 Privacy threats in the IoT

As is evident throughout our discussion thus far, much of the IoT's value offering to consumers rests on the collection and processing of personal data. Which is one of the problems with applying a generic approach such as that taken by Solove, given their propensity to yield false positives. Instead, a bespoke approach is needed with Ziegeldorf et al. providing such through their seven privacy threat categories [169]:

1. *Identification*: Linking an individual to an identifier, such as de-anonymising a pseudonymous user account. This is often accomplished through the joint processing a number of smaller data sets about a subject. Individually such data sets do not expose the data subject, but jointly they can as too many unique data points are exposed or derived. This may also enable other threats such as profiling.

2. *Localisation and tracking*: Tracking the movements of a data subject over time via GPS data, Wi-Fi connections, geotagged photos and the like. Here too, data accumulating from multiple sources is a necessary condition for the threat to occur since a single data point would not be sufficient.

3. *Profiling*: A commonly used practice for social media and e-commerce sites, profiling involves the combination of multiple data pools to enable the drawing of inferences about the data subject.

4. *Privacy-violating interaction and presentation*: The sharing of personal data via publicly accessible channels may expose such data to third-parties. This includes the use of public touchscreens terminals, the use of voice assistants, public but targeted video adds[13]. Since the use of both voice and tracking IDs are entrenched, this type of public interaction based on personal data may yet become more prevalent.

5. *Lifecycle transitions*: Over the lifecycle of an IoT device many changes of significance may occur and some of these may even expose the personal data of users. Such changes include ownership, location, profile of use, linked systems and the like. In Chapter 4 we present our testbed findings which include a number of related findings. Specifically the level of access device control apps have to additional data on smart phones and some devices, such as the Withings Thermo passing additional data to the manufacturer allowing for device tracking even across successive owners. The lifecycle challenge is further negatively impacted by the fact that different systems, devices, and components might form a system of systems but still have independent and divergent lifecycles [76].

6. *Inventory attack*: The gathering of data on IoT devices and indexing of such devices in a set location presents a multilayered threat. Remote access to devices, devices gathering data on each other, and the ability to easily compile a device inventory remotely, all allow for data to either be directly compromised or done so via inference. For instance, setting the Wi-Fi module of a laptop to monitor mode might allow the user to harvest metadata on nearby IoT devices without the premises containing those devices being physically compromised.

[13] Such as the use of marketing IDs to track shoppers via their phones and then to display targeted on shop displays.

7. *Linkage*: The combination of data from different data sets can lead to the exposure of data that the data subject wished to keep private. This also raises the risk for the re-identification of formerly anonymised data, yet remains a separate risk to that of identification since identification is not a necessary condition for the risk of linkage to be realised.

The threats identified above should, however, not be seen as confined to IoT devices and systems only and can be realised by way of other factors such as operational procedures or terms and conditions. The impact of the latter is specifically highlighted in our testbed analysis. Other researchers have also picked up on this point with Paul et al. using Ziegeldorf et al. to scrutinise the terms and conditions presented to data subjects by manufacturers [115]. Ziegledorf's threat categories can also be brought into alignment with other work to classify and structure the IoT space, such as the six elements of IoT developed by Al-Fuqaha et al. even though there are some overlapping terminology between the two approaches. We present a mapping between the two systems in Table 3.1.

| **IoT threat** | **IoT element** |
|---|---|
| Identification | Semantics |
| Localisation and tracking | Sensing |
| | Communication |
| | Service |
| | Semantics |
| Profiling | Service |
| | Semantics |
| Privacy-violating interaction and presentation | Sensing |
| | Services |
| | Semantics |
| Lifecycle transitions | Identification |
| | Sensing |
| | Computation |
| | Semantics |
| Inventory attack | Identification |
| | Communication |
| Linkage | Semantics |

Table 3.1: Mapping privacy threats in the IoT to elements of the IoT

Mapping between the work of Ziegledorf and Al-Fuqaha covers all elements of both systems, implying threats across the board, but also shows the Semantics element is referenced most often. Based on the preceding discussion this is to be expected due to the central role of data processing in much of the IoT's value offering. This is also demonstrated in Chapter 4 where all the devices on the testbed, to varying degrees, are reliant on semantic[14] processes to deliver their service offerings. The implication then, for any new system wishing to address PbD in the IoT, is that it can not only focus on a single device but must remain cognisant of the larger system the device operates in and do so over the entire device lifecycle.

Comparing the six IoT privacy threats proposed by Ziegeldorf et al. to our defined metrics we find the following:

1. Not a comprehensive approach.
2. Does not implement remedies.
3. Auditing for compliance not mentioned.
4. IoT is the focus.
5. The use of DFD's is not specifically mandated, though data flow tracking will most likely be needed for a useful analysis.
6. PbD is not a focus area but IoT privacy threats are.
7. System modelling of some type could be used but a modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.6 Privacy engineering and data protection

Starting from the proposition that privacy engineering is a relatively new and evolving term, Bowman et al. go on to define privacy engineers as those "*responsible for ensuring that your product is developed, built, and used in a manner consistent with your company's privacy values*" [24]. In a broad sense then, most of what is proposed herein can be viewed as privacy engineering, provided that one understands "values" in this sense to extend to user and legislative requirements.

In order to initiate the privacy engineering process, Bowman et al. suggest using a series of questions to first understand the manner in which a certain device or system interacts with private data. This is intended to provide the privacy engineer with a better understanding of the issues at hand before starting any technical work on the system itself [24]. Although these questions cover privacy implications from multiple angles and is in line with the argument advanced herein, we

[14] As reminder, semantics here refer to the development of new insights based on extracted data.

will not pursue that approach as the LINDDUN methodology, as introduced in this chapter, does a stellar job of addressing these issues.

When dealing with privacy and PbD in the IoT, it is impossible to exclude the topics of data protection and data privacy since data mining and processing are core to the IoT value proposition. Such data in the home setting is, by definition, personal. A useful distinction though, is between data protection which deals with protection against unauthorised processing and data privacy which deals with who has authorised access to the data concerned [128]. This is of particular importance when dealing with legislation related to personal privacy since it is a common position for such legislation to focus on data protections. This is demonstrated in our discussion on the GDPR which we present below in Section 3.7 while Subsection 3.7.1 introduces auditing against GDPR requirements.

## 3.7 The GDPR and compliance

The European Union's General Data Protection Regulation (GDPR) devotes significant effort to formalising and regulating Data Protection by Design (DPbD) which includes a series of rights and obligations attaching to the personal data of data subjects. Notable in this regard is Article 25, which directly deals with data protection by design and by default [125] and requires that data controllers ensure compliance with all relevant requirements[15] [14]. Article 25 is also linked to Article 42 which addresses the certification of data protection measures. With Article 25 establishing data controllers as the party holding the primary responsibility for realising the rights of data subjects, while Article 5 establishes these rights starting with the following six requirements [82]:

1. *Lawfulness, Fairness and Transparency:* Directs the manner in which data are processed.
2. *Purpose Limitation:* Data collection and processing is only allowed for specified, explicit and legitimate purposes.
3. *Data Minimisation:* Only the data needed to fulfil the stated service, as permitted by the purpose limitation, may be collected.
4. *Accuracy:* Data must be accurate, up to date and either erased or rectified as soon as possible at the data subject's behest.
5. *Storage Limitation:* Data may only be retained in a form which links to a data subject, for the minimum period of time needed to fulfil agreed-upon services. Though there are some exceptions[16].

[15] As such, the GDPR requires a design lead approach to data privacy and also requires a demonstration of compliance which can be assessed by way of formal audit. This is of significant importance and is further explored in subsections 7.5 and 7.6.

[16] Such as archiving, statistical or regulatory uses.

6. *Integrity and Confidentiality:* Data processing must be conducted such that data loss, unauthorised access and unlawful processing or any other prohibited outcome, is guarded against.

In addition to the requirements listed above, which clearly echo the work of Ann Cavoukian, Article 5 further requires that data controllers must be able to demonstrate compliance to these requirements inline with the stipulations of Article 25. Moving on from Article 5, Articles 15 to 22 establish the following rights:

1. The right to transparent information from the controller and clear channels for exercising the rights listed here.
2. The right to be informed of not only data collection but also how to contact the data controller and its agents.
3. The right to be informed about the collection of personal data even if the collection thereof was not directly from the data subject.
4. The right to access their data and information pertaining to that data.
5. The right to rectify any errors in their data.
6. The right to be forgotten (the right to erasure).
7. The right to restrict processing if the data concerned are inaccurate, processing is unlawful, data are retained for legal reasons, or the data subject has objected to the processing.
8. The right to be notified of any actions taken by the data controller in relation to the rights to rectification, erasure and restriction[17].
9. The right to data portability.
10. The right to object to data processing.
11. The right not to be subject to any decision based solely on automated processing[18].

The GDPR is also intended to streamline and unify legislation across the European Union though arguably its greatest impact is to greatly increase and clarify the control the citizenry have over their own data. This is to a large part accomplished by way of the above mentioned data processing requirements [156]. The GDPR is of course not the only EU wide legislation which impacts on individual privacy. Delving deeper into legislation will not aid the discussion here given the GDPR's prominence on the issue. However, a brief mention of one additional piece of proposed legislation is warranted. That is the ePrivacy Directive (ePD), more formally known

[17] That is Article 16, 17 and 18.

[18] Article 22 requires that data subjects can request for any automated decision to be reviewed by a human, but not that a human must be in the loop if no request was made.

as the Privacy and Electronic Communications Directive. The ePD protects the confidentiality of electronic communications [53], but has been referred to as the cookie law due to its impact on that technology though, it also addresses unsolicited email [54].

Due both to a range of continuing issues with implementation and enforcement [53] and to the introduction of the GDPR, the ePD is set to be replaced. Taking its place will be the ePrivacy Regulation (ePR) which is aligned with the functioning of the GDPR but is in fact *Lex Specialis* to it [27]. In short, this means that for areas of overlap the ePR with its narrower and specific focus will override the GDPR and its more general scope. On this topic it should be noted that the ePR is intended to also deal with M2M communications in a more direct manner and would therefore, on the face of it, appear a prime candidate for inclusion herein. However, the ePR was set to be introduced alongside the GDPR but has instead been through numerous revisions and has still not been adopted[19].

### 3.7.1 Privacy Impact assessment under the GDPR

An important point to note before launching a discussion of the CNIL's audit methodology is that the term auditing is often used as singularly synonymous with financial auditing. This is incorrect as the formal auditing of performance against set targets enforced by law or industry standard is well established in a multitude of industries and business practices. Although the broader term compliance auditing is often used for auditing other than financial, this is not intended to diminish the thoroughness or importance of such audits, with the potential penalties of GDPR non-compliance boldly underlining this point. One of the most prominent areas of non-financial audit is those aiming to assess the privacy compliance of a system or device. To draw attention to this focus area, such audits are often referred to as privacy impact assessments (PIA), which is indeed the case with the CNIL's methodology, as introduced below.

The specific PIA we will be introducing here is that of the French Commission for Informatics and Liberty (CNIL). As is the case in general for PIAs, that of the CNIL can be described as assessing and mitigating specified privacy risks. Differentiating the CNIL's PIA though is the fact that it relates to GDPR compliance, meaning that the risks involved for those under audit go beyond the process of compliance but also include substantial financial penalties for failure to do so. A further key distinction is that CNIL's PIA includes an application of the EBIOS[20] methodology as developed by the French National Cybersecurity Agency (ANSSI) for the management of risk [35].

Under the GDPR the activities of a PIA are governed by Article 35 with data controllers being the primary actor concerned [151]. This focus on data controllers does of course not block any

[19] The current draft and assessment documentation is available at `https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-privacy-and-electronic-communications`, while procedural updates leading to ratification are available at `https://oeil.secure.europarl.europa.eu/oeil/popups/ficheprocedure.do?reference=2017/0003(COD)&l=en`.

[20] Expression des Besoins et Identification des Objectifs de Sécurité (Expression of Needs and Identification of Security Objectives)

other party from carrying out such assessments, either of their own volition or at the behest of data controllers. It is therefore also not surprising that one of the core tenets for the CNIL's work is the strengthening of the control data subjects have over their own data [35].

A significant related factor is that GDPR does not dictate the form and exact function of compliance auditing but is instead concerned with the outcomes reached. This leaves national regulators or auditing bodies the opportunity to innovate and bring best practice to the PIA space. An example of this is the traffic light system which the CNIL employs to assess the PIA outcome. The traffic light system subdivides areas of the PIA into nearly 50 distinct areas. For each area a colour code is assigned to indicate the state of the audit outcome. The codes are white for non-applicable, red for unsatisfactory, yellow for planned improvement and green for acceptable. The use of these colour codes is of course the last step in the PIA which is itself divided into four parts per the following [36]:

1. Determine the context for processing the activities being investigated
2. Determine proportionality of processing and compliance controls and reference both against the rights of data subjects
3. Identify and assess data security risks to ensure they are fully addressed
4. Validate the PIA itself using a traffic light based system

The CNIL's PIA methodology pays specific attention to the internal controls used by the party under audit, with key areas including the proportionality and necessity of processing and the manner in which the rights of data subject rights are protected. As would be expected, given that the PIA assess GDPR compliance, data processing is investigated in relation to the purpose limitation, lawfulness, minimisation, quality and retention. The protection of data subject rights are further assessed against the following seven principles [35]:

- Data subjects provided with easily accessible and transparent information
- Mechanism for consent to be withdrawn
- Data access for data subject provided in portable format
- Mechanism for the rectification or erasure of data at data subject request
- Mechanism for restrictions and objections to processing
- Identification of data processor(s)[21]
- The compliant handling of data transferred outside of the EU

[21] This is highlighted as a significant failure for one of the systems included in our testbed and discussed in Chapter 4

The CNIL further supports its PIA by providing standard documentation and templates to structure data collection and verification which, in conjunction with the traffic light system, aids in preparing a high quality audit file[22] [36]. Also included in the PIA is a direct measure to locate and mitigate threats to the personal data of data subjects. This is a two step process with threats first identified and remedial actions then determined [35]. There are three threat categories which are:

- Illegitimate data access
- Unwanted changes to data
- The disappearance of data

To attend to any of these threats the following four actions are to be taken in sequence:

1. Investigate the manner in which the data subject's privacy was impacted
2. Determine how severe the event is
3. Determine risks to any data stores and check for linked threats
4. Determine event likelihood

There are of course numerous other PIAs which could also have been investigated herein if our intention was solely to gain a general understanding of such systems. There are, for instance, well established systems in medical applications [43], insurance [55] and debt collection [42]. However, that of the CNIL stands out for its direct alignment to the GDPR, clarity of purpose, quality and being publicly available [23]. On the topic of varying approaches to PIAs, one study compared methodologies from Ireland, New Zealand, the United Kingdom, Australia, Canada, Hong Kong, and the United States [163]. From this review the authors derived best practice recommendations for conducting a PIA. These include, starting the process as early as possible, clarity on the position of involved parties, understanding the PIA as a process as opposed to a one off event and lastly, using questions to elicit both risks and potential solutions thereto. The authors also made the point that privacy can not be limited to data protection only. These points clearly support our contention that the CNIL's PIA is not only fit for purpose in general terms but is also of high quality. A final point of note is that the logical extreme of the best practices presented above would be to start the PIA during the product design phase and then carry it on throughout the lifecycle of the system under audit, which is indeed the position we take. Please see Subsection 7.5 for a detailed discussion.

The CNIL PIA's key indicators against our defined metrics are as follows:

[22] An audit file is a lose term covering all the documentation related to a single audit. For paper-based documentation this normally takes the form of one or more binders with an index page as cover. Colour coded tabs are also commonly used for cross-referencing in such paper-based systems.

[23] Not all regulators publish their audit methodologies and we believe that the CNIL's approach is to be lauded in this.

1. Not comprehensive approach, due to its legal focus.
2. Clear focused on implementation.
3. Auditing outcomes directly and clearly addressed.
4. IoT is a focus area by way of a connected device application specification.
5. The use of DFD's is not specifically mandated, though data flows will have to be tracked and explained to conduct the PIA.
6. PbD and related legislative requirements, including compliance thereto and remedial actions for non-compliance, are directly addressed in the PIA.
7. System modelling of some type should be used but a formal modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.8 The standard data protection model

The tenets of the GDPR are, unsurprisingly, also reflected in numerous other pieces of legislation and compliance requirements, both within the EU and without. One notable example within the EU is the Standard Data Protection Model (SDM) first published by the German Independent National Centre for Privacy in 2017 [62] and most recently revised in 2020 [150]. The SDM aligns preexisting German regulations with the GDPR and aims to create a level playing field both for compliance auditors and companies under audit by taking a goal oriented approach to achieving desired compliance outcomes. It lists the following six data protection goals which not only align to the GDPR but also echos the work of Ann Cavoukian discussed in Section 3.4:

- *Data Minimisation:* Requires a reduction in the collection and processing of data, as well as the introduction of systems to ensure certain data are either automatically blocked or deleted when no longer needed. One avenue for data minimisation is to ensure that only the data strictly needed for stated processing purposes are collected.
- *Availability:* Redundancy in software and hardware combined with other applicable systems to ensure that data are available when needed.
- *Integrity:* The use of techniques such as read/write restrictions, data timelines, and document access controls to ensure the integrity of data.
- *Confidentiality:* Mechanisms such as document access control and closing or barring potentially risky communication channels, aimed at enforcing secure authentication.
- *Unlinkability:* The prohibition of unsafe practices impacting on linking, such as backdoors, rights transfers, and open data flows between entities.

- *Transparency:* Keeping and making available logs and other documentation. These include policies and procedures, access logs, and client interaction logs.
- *Intervenability:* Data controllers must be able to intervene in processing at any time, while data subjects must be granted their rights to notification, information, rectification, blocking and erasure.

Using these goals as reference point the SDM aims to, in its own words, *systematise data protection requirements* [150]. The term systematise here refers to the grouping and structuring of data protection requirements by means of the listed goals. This allows for potential harm to data subjects to be reduced in a verifiable manner which rests on this simplified modelling of compliance requirements.

The SDM's key indicators against our defined metrics are as follows:

1. Not a fully comprehensive approach[24], due to its legal focus.
2. Focused on implementation through compliance goals.
3. Auditing outcomes directly addressed but not as a full audit methodology.
4. IoT is not a focus area.
5. The use of DFD's is not specifically mandated, though doing so would easily fit the with the operation of applying the SDM.
6. PbD and related legislative requirements, including compliance thereto, are the primary focus areas of the SDM.
7. System modelling of some type should be used but a formal modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.9 Privacy enhancing technologies

Privacy enhancing technologies (PETs) represent a key component in the privacy engineer's toolkit. Unsurprisingly, PETs are often deployed in systems where PbD is a key concern bringing about a positive privacy outcome aimed at preempting certain privacy threats [67]. The area of PETs is a broad and evolving one with examples of such technologies ranging from cutting edge development to those in standard use. To provide structure to the topic then, PETs can be grouped into seven distinct, but connected domains [29]:

[24] Although the SDM contains provisions and measures which extend beyond GDPR compliance, these do not constitute a fully formed elicitation and mitigation of privacy threats when compared, for instance to LINDDUN.

1. *Control Over Data:* PETs of this stripe require data subjects to gain a greater understanding of the collection and use of their personal data. This increase in knowledge is needed to facilitate the aim here which is then to provide data subjects with greater, and often more nuanced, direct control over their data.

2. *Enforcement:* Regulatory requirements, privacy policies and other controls can be strictly enforced through various means such as for instance XACML[25]. A headline advantage of such PETs is that they directly block the collection of data beyond what is needed and therefore act in service of data minimisation as discussed in Section 3.8.

3. *Anonymisation or Pseudonymisation:* Although in common use, these terms have a formal distinction in terms of the GDPR. The ICO[26] defines anonymised data as that which no longer relate to "*an identified or identifiable natural person* " and accordingly falls outside of the scope of the legislation, while anonymisation is the process[27] of rendering data as such. Pseudonymisation on the other hand is the removal or replacement of information linking a natural person their personal data. The attribution of such data to a natural person is therefore only possible through the use of additional information [81]. However, the subversion of anonymisation is well established [112] and is particularly worrying in the age of social networking [123].

4. *Personal Data Protection:* This goes beyond mere control over data and adds an element of enforcement. Such PETs are geared towards providing data subjects with the means to both determine and enforce policies. This also directly addresses concerns around consenting to processing and collection, since the data subject themselves determined the policy specifics.

5. *Anonymous Authorisation:* The concurrent use of both enforcement and anonymisation techniques, including pseudonymisation. This, for instance, allows authorised access to a system without the disclosure of personal details. A derivation of public key encryption called attribute-based encryption is a current example of anonymous authorisation [18].

6. *Partial Data Disclosure:* The preservation of privacy and a reduction in exposure to third-parties is targeted via control over data and anonymisation/pseudonymisation techniques. Differential privacy, which enables data privacy in aggregated disclosures, is a significant example of this class of PET [49].

7. *Holistic Privacy Preservation:* These PETs occupy the intersection between control over data, enforcement, and anonymisation or pseudonymisation. As such, these PETs provide the highest level of control and insight for data subjects and target the largest number of contemporary privacy threats. One example of this, Gray scale access control, is a hybridised access control scheme incorporating differential privacy [87].

[25] eXtensible Access Control Markup Language
[26] The Information Commissioner's Office is the UK regulator.
[27] This is also a processing activity in terms of the GDPR.

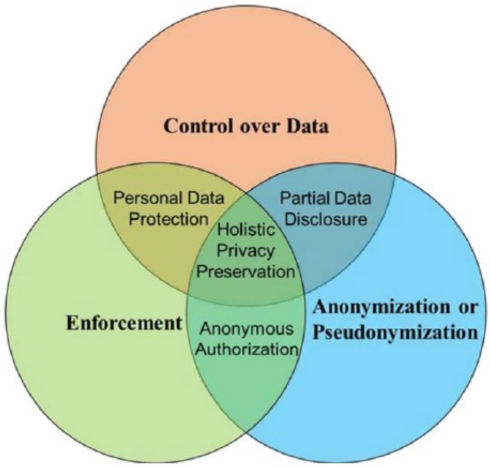



Figure 3.9: The PET classification by Cha et al.

The classification developed by Cha et al. is visually presented in Figure 3.9. This presentation clarifies the link between the different class of PETs but also creates the impression that instead of a bespoke approach based on system needs, the privacy engineer can simply opt for PETs in the "Holistic" grouping. Unfortunately this is not the case since PETs in this grouping are far from being mature technologies. In dealing with this, Cha et al. themselves found more than 20 studies on such PETs, but none had progressed further than prototyping and still had significant development work ahead. This is, for the time being, a deal breaker for this class of PETs but not for PETs in general. Instead, it is up to the systems designer to decide on the best fit technology for the task at hand, taking a "by design" approach to preempting privacy threats in the IoT system under development. Although this is clearly in line with DPbD, PbD and related compliance issues, PETs should not be seen as an all encompassing panacea for addressing privacy challenges. Instead, their use should be guided by the specific challenge being faced in each system under development, this also implies that PETs might be wholly undesirable for some circumstances. Where appropriate though, a data controller can use the classification presented above as a starting point for determining the correct technologies to deploy as part of their obligations under Article 25[28] of the GDPR. The inclusion of PETs in the systems design process is further addressed in Chapters 6 and 7.

Not addressed at any length in the above discussion, is the detailed functioning and merits of specific PETs. This is a purposeful omission as doing so would be wholly counter productive given the large and growing number of PETs available. This list is also continually expanding with new entrants often reflecting other technological trends such as the recent focus on distributed ledger technologies [143]. One example of this is the privacy verification chain (PVC) which takes a "by design" approach to privacy and security in the IoT by using a ledger to manage the data access rights of both data controllers and processors. The ledger also records data access by authorised parties and allows for the inspection thereof by the data subject. Adding further

[28] Data protection by design and by default.

functionality to the PVC, Foukia et al. built their Smart Data System (SDS) to act as means of access to the ledger and for the provision of some audit functionality [58].

The processing needs and general functionality of the SDS result in it running on more powerful hardware away from the edge. As such it also does not suffer from the power use, update and other constraints typical of edge devices. However, use-cases for PETs at or near the edge is easily imaginable, especially where communication to off-site control systems or ledgers might not be feasible. Similarly, it might be desirable to host processing intensive functionality in the cloud but retain user data on local devices. Doing so can provide the data subject greater direct control over their data and is particularly useful for dealing with sensitive data such as medical information [51]. Even more streamlined than this approach is the notion that a relatively simple PET based on established technology will be easier to deploy and yield a more reliable result. In this space we find Privacy Notices, which endeavour to provide data subjects with all needed information for the provision of consent and does away with overly complicated terms and conditions [132]. These notices are derived by taking existing terms and conditions, extracting their components via an automated system and presenting these to data subjects as colour coded graphs containing key words.

Lastly, PETs are a broad group of technologies and the deployment of such technology would be in aid of a solution determined by prior analysis. As such, the eight metrics used to analyse different approaches herein can not be utilised here.

## 3.10 Threat modelling

### 3.10.1 About threat modelling

Introducing threat modelling in his highly regarded and well cited book on the topic, Shostack lists four primary reasons for conducting threat modelling. These are, spotting problems earlier, understanding related requirements, designing superior products and services, and addressing problems that other approaches are not equipped to deal with. The activity of threat modelling is described as a process of abstraction through the use of models, aimed at finding problems in devices and systems that are yet to be built, as such it is inherently a by design approach.

There are no perfect solutions and every exercise in threat modelling is situated within a defined and limited problem space, as such, trade offs are part and parcel with the process [135]. This simple fact of threat modelling, does of course distinguish it from issues of legal compliance where the outcome is binary and the system under audit either is, or isn't, in compliance.

### 3.10.2 Microsoft STRIDE

STRIDE was initially developed to aid in software development, focusing on identifying potential security threats to that software [135]. Its application has broadened since [91], but the core principles remain the same with the name as an acronym for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. This core is also maintained by the integration of STRIDE into Microsoft's Security Development Lifecycle (SDL) where its use of DFDs to map systems and localise threats is key [86]. STRIDE is presented in full in Table 3.2.

The biggest point of impact for STRIDE on the rest of the work presented herein is its direct link to, and inclusion in, the LINDDUN methodology. We present that methodology below, but before doing so there is one further point of interest. In his work on threat modelling with STRIDE and UML, Johnstone found that more effective threat models can be generated by replacing standard DFDs with a UML model, specifically focusing on activity diagrams. This is an interesting proposition but outside of the current project. It is however revisited in Chapter 10.

STRIDE's key indicators against our defined metrics are as follows:

1. Not a comprehensive approach as it does not cover legal considerations.
2. Does provide explicit guidance on the implementation of any derived solutions.
3. Auditing outcomes not addressed.
4. IoT is not a focus area though the application of STRIDE to the field is entirely possible and in scope.
5. The system is explicitly modelled through data flow diagrams.
6. Privacy is not addressed directly.
7. The system is modelled but a formal modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

| Threat | Property violated | Threat definition | Typical victims | Examples |
|---|---|---|---|---|
| Spoofing | Authentication | Pretending to be somebody or something else | Processes, external entities, actors | Falsely claiming to be an OAP to get free travel |
| Tampering | Integrity | Modifying the contents of an electronic resource (disk, network, memory) | Data stores, data flows, processes | Changing the contents of a database, modifying a program binary, packet injection, packet altering |
| Repudiation | Non-repudiation | Claiming to have not done something. This can be honest or false but in both cases system evidence must be used to refute the claim | Process | Claiming to have not made an order, or not submitted a certain form |
| Information disclosure | Confidentiality | Information provided to unauthorised person | Processes, data stores, data flows | This can range from access to the communications of another to publicly visible meta data containing too much detail |
| Denial of service | Availability | Blocking resources needed to provide a service | Processes, data stores, data flows | Overloading network traffic, filling disks to capacity, using all available memory for one task |
| Elevation of privilege | Authorisation | Individual able to perform actions they are not authorised for | Process | Standard users executing code with admin privileges, allowing remote code execution that should be barred |

Table 3.2: STRIDE Threats [135]

### 3.10.3 LINDDUN

Like MPRAM, which is discussed in Subsection 3.11.2, LINDDUN makes extensive reference to system modelling and also takes a broad view on system participants, but beyond these foundational considerations the two systems have little in common. Instead LINDDUN presents as a threat modelling approach which is far more nuanced and extensive than STRIDE, even though the former heavily borrows from the core assumptions of the latter. This is immediately clear from the name which is an acronym for Linkability, Identifiability, Non-repudiation, Detectability, information Disclosure, content Unawareness and policy and consent Non-compliance. As central mechanism, LINDDUN uses data flow diagrams (DFD) to pinpoint instances of the seven privacy threats that make up the acronym and form the bases of its threat tree catalogue [165]. The authors describe LINDDUN as follows [166]:

> LINDDUN is a model-based approach that leverages a data flow diagram (DFD) as representation of the system to be analyzed.

Threat-modelling under LINDDUN is divided into three main steps with a number of subsections totalling seven actions. The three main areas with their sub steps in indent are as follows [45]:

1. Model the system
2. Elicit threats

   *Map DFD elements to threat categories*

   *Elicit threats*

   *Document threats*
3. Manage threats

   *Prioritise threats*

   *Select suitable mitigation strategy*

   *Select privacy enhancing solution*

However, these steps can also be split more directly in to just two phases, that is the problem phase and the solution phase. As this split matches directly with the MBSE Grid Framework used as procedural structure in Chapter 7, we will be opting for it. This layout is also more widely used and is presented in Figure 3.10.

By modelling the system and then immediately deriving a DFD which maps to threat categories, LINNDUN allows for the determination of remedial actions at the point of need and directs them accordingly. Doing so brings a further advantage in the form of ready made solutions to recurring threats [78]. An additional point of strength for LINDDUN is its continuous development which

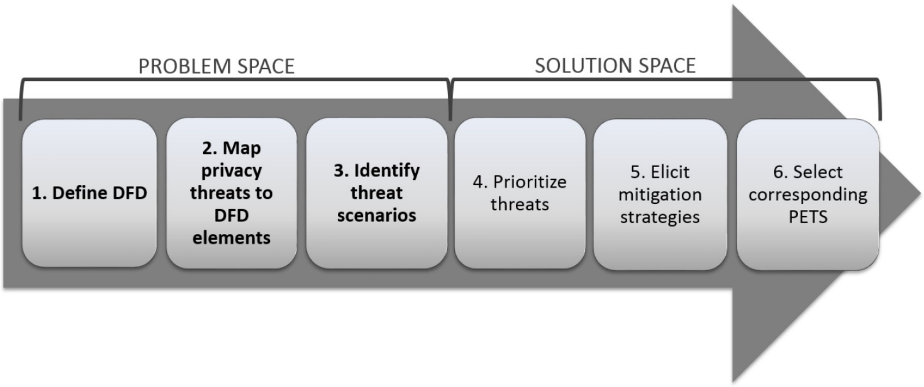



Figure 3.10: LINDDUN methodology steps

has brought about a number of revisions and improvements to the core system [167] and has also seen the launch of a stripped down version called LINDDUN GO [46] Table 3.3 presents an overview of the LINDDUN privacy threats taxonomy taken from the work of Hart et al..

| Threat category | Privacy property | Threat instance |
|---|---|---|
| Linkability | Unlinkability | Guess someone is dieting due to online searches |
| Identifiability | Anonymity, pseudonymity | Identify a user in a database |
| Non-repudiation | Plausible deniability | Determine who expresses a given online vote |
| Detectability | Undetectability, unobservability | Determine who accesses a web page |
| Disclosure of information | Confidentiality | Data breach |
| Unawareness | Awareness | Sharing pictures on Facebook with unintended audience |
| Non-compliance | Compliance | Disclosing data to third party without user's consent |

Table 3.3: The LINDDUN privacy threat taxonomy [70]

#### 3.10.3.1 LINDDUN step-by-step

Given LINDDUN's eventual inclusion in the DISCREET methodology proposed in Chapter 7, it is necessary to delve somewhat deeper into the step wise application of this methodology. For this we turn to the step-by-step tutorial provided by Wuyts and Joosen, which is summarised below [166].

**Step one: The DFD**

A system model in the form of a DFD must be created and will act as the basis for the analysis to follow. Such a DFD must represent the dataflows of the system involved and uses four building blocks namely: 1) Process (P) which is a work unit operating on data; 2) Data flow (DF) indicates a named flow of data through a process or system; 3) Data store (DS) is a logical data repository; 4) External entity (E) is a data endpoint. The abbreviations used for each of these four elements is of note as they are used to classify threats in latter steps. In Figure 3.11 a limited DFD with instances of each element is presented. To this, the modeller can also add trust boundaries, though these are optional.

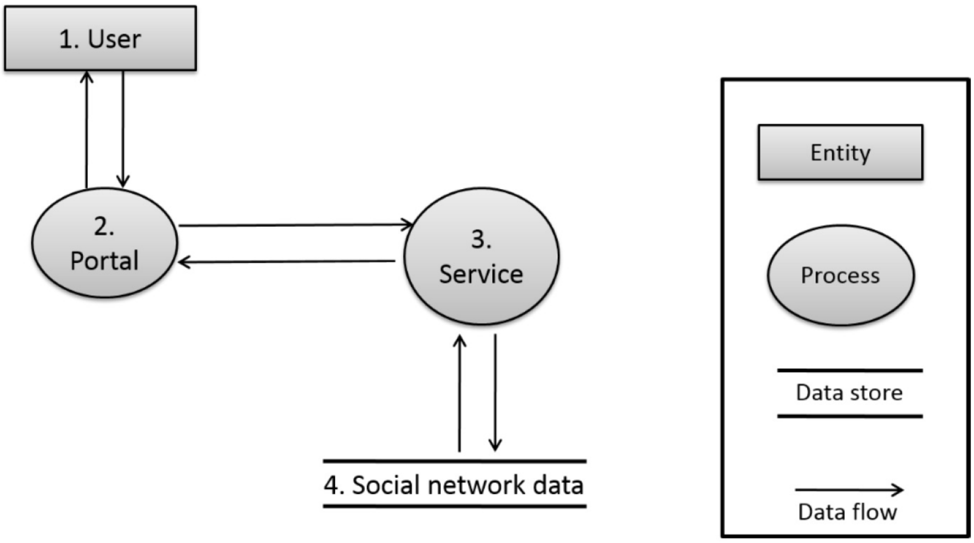



Figure 3.11: LINDDUN limited example DFD [166]

**Step two: Mapping DFD to LINDDUN**

Once the DFD is completed each element can be mapped to one or more of the seven LINDDUN threats, where this mapping is done by type. Such type mapping is the reason for DFD elements to be divided into the four types previously listed and also requires that fixed abbreviations be added to the seven LINDDUN elements to allow for the automatic generation of the threat modelling codes used later. The seven elements are briefly described below, including their abbreviations, while Figure 3.12 shows the mapping.

- Linkability (L) is the ability to determine if two items of interest (IOI) are related
- Identifiability (I) is the ability to identify a data subject
- Non-repudiation (Nr) is the irrefutable linking of a party and an action performed by that party
- Detectability (D) is the ability to determine whether an IOI exists
- Disclosure of information (Di) is data access by unauthorised parties
- Unawareness (U) relates to a data subject who is not aware of the information they are supplying to the system nor the consequences thereof
- Non-compliance (Nc) relates to a system that does not comply with data protection legislation, linked policies and the data subject permissions

| Threat categories | E | DF | DS | P |
|---|---|---|---|---|
| Linkability | X | X | X | X |
| Identifiability | X | X | X | X |
| Non-repudiation | | X | X | X |
| Detectability | | X | X | X |
| Disclosure of information | | X | X | X |
| Unawareness | X | | | |
| Non-compliance | | X | X | X |

Figure 3.12: The LINDDUN mapping template [166]

**Step three: Eliciting privacy threats**

For every element, the corresponding potential threat areas can be determined by looking at the mapping table in Figure 3.12, while multiple instances of the same threat can be combined. From there the presences of actual threats can be determined by a three step process which involves refining threats via a threat tree, documenting related assumptions, and documenting threats using a threat template.

For each possible threat type LINDDUN provides a list of threats organised on a tree structure. This contains the preconditions for specific threats falling under the type indicated by the mapping template, where these preconditions are vulnerabilities within the system. One of the shorter threat trees is that for non-compliance and it is presented in Figure 3.13. An integral part of this process is to determine if any parts of the LINDDUN threat tree is not applicable, typically due to the specifics of the system at issue. These decisions are still held as assumptions though and must as such, be documented. From there a final set of threat scenarios, called misuse cases, can be formulated. These must then also be documented explicitly.

**Step four: Prioritising risks**

Since the application of the above steps can yield a very large number of threats, there is a definite need to prioritise these. To do so a graded risk measure is assigned to each threat where risk is defined as the product of the risk's likelihood and impact. Although LINDDUN does not prescribe a set risk assessment to use, it does recommend the OWASP's Risk Rating Methodology and Microsoft's DREAD, but leaves the door open for the practitioner to use any applicable system[29].

**Step five: Mitigation strategies**

Resolving the system flaws which give rise to privacy threats is the central undertaking here, with the means of resolution referred to as mitigation strategies. With threats identified and then prioritised the systems engineer can proceed to reference the highest level threats first, to the LINDDUN mitigation strategies mapping[30]. In essence this is a taxonomy of solution strategies. This taxonomy is divided into two main elements, with the first dealing with the concealment of associations between a data subject and data about them. This can be seen as proactive since

[29] In the application of LINDDUN in Chapter 7 we also take this route.

[30] The mapping is available here: `https://www.linddun.org/downloads`

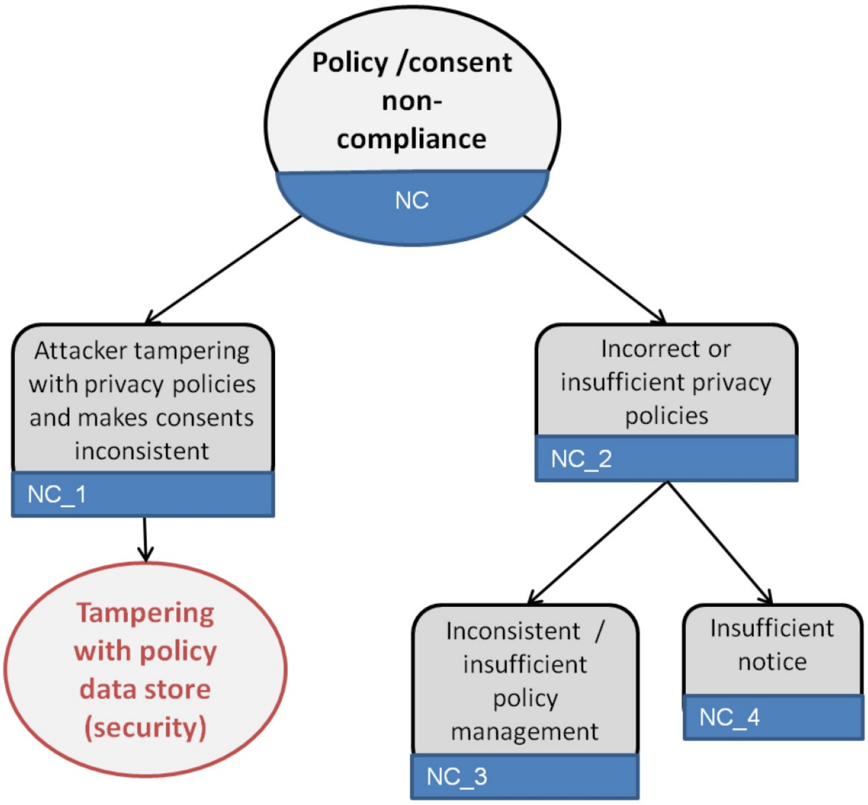



Figure 3.13: The LINDDUN threat tree for Non-compliance

such strategies are intended to preempt problems. The second is the guarding of associations which are used when data has already been collected or processed.

**Step six: Privacy enhancing solutions and requirements**

Once step five is complete and mitigation strategies have been determined, they need to be translated to actionable solutions or privacy requirements. This explicitly allows for LINDDUN to be incorporated across the system lifecycle. The LINDDUN solutions table, which is part of its privacy knowledge base, attaches an actionable PET to each mitigation strategy and also points to research on that PET. This allows the system engineer to directly incorporate the appropriate technology into the system being developed. If instead the requirement formulation route is followed, the system engineer follows the same process but can then work back up to the threat tree nodes and target system requirements which nullify those nodes, thereby preventing the threat and creating a privacy requirement.

LINDDUN's key indicators against our defined metrics are as follows:

1. Not a comprehensive approach as it does not cover legal considerations.
2. Focused on implementation through the identification of PETs.
3. Auditing outcomes not addressed.
4. IoT is not a focus area.

5. The system is explicitly modelled through data flow diagrams.

6. Privacy threat modelling is LINDDUN's primary function, though it takes a PbD view point.

7. The system is modelled but a formal modelling language is not employed.

8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.11 Privacy requirements

### 3.11.1 User requirement elicitation

As stated in the abstract to this research, and maintained throughout, we firmly believe that consumer IoT devices can be brought into alignment with privacy requirements, provided that a "by design" approach is taken not just to the products and services but to their entire lifecycle. An obvious next question to ask then is what exactly constitutes privacy requirements. As shown in the preceding discussion, there are legal requirements such as those from the GDPR which deal with privacy and PbD, but are often focused on data to the extent that the GDPR explicitly talks about DPbD. Most of the other approaches introduced in this chapter do however not take this view and deal with privacy and PbD in more general terms, often referring to the elicitation of privacy requirements or goals from various stakeholders. This notion of requirement elicitation is of significant importance to our work here since any take on PbD that was solely predicated on legal compliance would run significant risk of not meeting stakeholder requirements, with customers and clients prominently included in that list of stakeholders. The business justification for our position therefore seems clear, but acting on that position is less straightforward.

### 3.11.2 MPRAM

Seda Gürses, a leading researcher in the field of privacy by design takes an approach to the field that is tailor-made for a systems engineering audience when she describes PbD as "*the translation of complex social, legal and ethical concerns into systems requirements*"[31] [65]. Gürses further describes this understanding of PbD as "multilateral" given its multiple sources of origin which is also directly congruent with our motivation for focusing on both privacy and compliance by design. Although Gürses does not focus down on any specific domain of application, such as the IoT, she does distinguish between two significant aspects of PbD in general. The first of these is the use of target strategies to engineer the desired privacy results. The second is the deployment of privacy enhancing technologies (PETs), where such technologies are the tools used to realise the chosen design strategies [66]. Taking these various sources of requirements into

[31] A significant portion of the domain extension presented in Chapter 7 is explicitly concerned with this task.

account, Gürses proposes the Multilateral Privacy Requirements Analysis Method (MPRAM) which elicits the security goals of system participants and rolls these into system requirements.

MPRAM sets a definition of roles and groups within a system, including a broad system description, as its point of departure. From there the privacy and security goals of system participants are extracted by group, including commonalities between these. Using this information, system wide security goals can be established and from there incorporated into system requirements [64]. This process is presented in Figure 3.14.

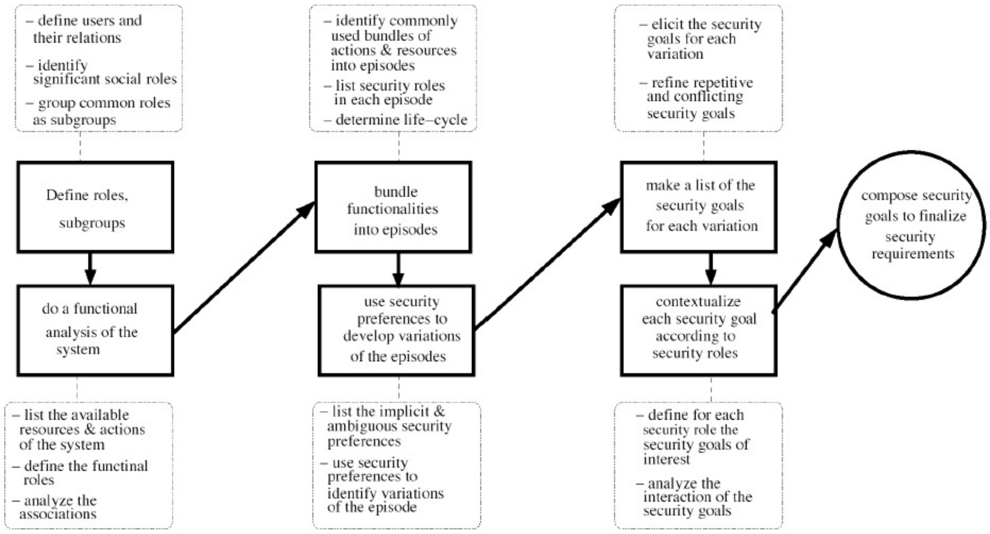



Figure 3.14: Multilateral Privacy Requirements Analysis Method (MPRAM) [64]

Although the components of MPRAM will not be incorporated in the work presented in Chapter 7, the new domain extension does share MPRAM's foundational understanding of the problem space. That is to say, viewing PbD as fundamentally a systems modelling activity within which stakeholder inclusion is a necessity for the design of systems that adhere to the principle of PbD.

Measuring MPRAM against our eight metrics we find the following:

1. A comprehensive approach including legal and technical considerations due to goal forming inputs including a broad base of stakeholders.
2. Not implementation focused.
3. Auditing outcomes not addressed.
4. IoT is not a focus area.
5. The system is modelled in relation to users, groups and their security and privacy needs, but not explicitly as data flow diagrams.
6. Security goals and requirements contain privacy requirements.
7. The system is modelled but a formal modelling language is not employed.

8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.12 The Kung IoT privacy engineering framework

Building on two EU projects Kung et al. developed a highly nuanced framework of their own aimed at privacy engineering in the IoT. The two building blocks provide a wide foundation to the new framework by melding real-world reliability measures with a research outlook. The first framework, RERUM, deals with application dependability in Smart Cities. The second is a methodological approach to PbD in information and communications research and is called PRIPARE. These two frameworks, plus an extensive list of supporting material are then used to formulate a new framework with four stages [96]. The first of these is the concept stage and is not subdivided while the following three, namely stakeholder, process, and organisational, are all divided into two segments focusing either on IoT systems, or subsystems.

The framework further distinguishes three areas of application which are subsystems engineering, use by data processors, and use by data controllers. This split focus is brought about, in part, by the framework's low-level of operation which precludes its universal application. A second reason for the split is that Kung et al. views the compliance and PbD needs of suppliers and their downstream clients (companies not data subjects) as not just separate but as hidden from the viewpoint of each company. This assertion, in our view, is tenuous at best. Although different companies might not have full access to each other's privacy and compliance threats and associated actions, they do not need to. Compliance metrics are known and even if their are not, downstream companies will buy products and services from suppliers which meet their requirements, including compliance and PbD issues. In bespoke services such as those between data controllers (client) and data processors (service provider) it is likely that the client will have a clearly detail set of requirements for the service provider to meet. Also, in terms of systems modelling, the use of the so called "black box" allows for ready made units to be imported into system models. As such, the need for a formally different approach for controllers and processors remains unclear.

## 3.13 DEFeND

Diverging from all the previously cited work in a significant way is DEFeND (**D**ata gov**E**rnance **F**or supporti**N**g g**D**pr) [119]. Although DEFeND[32] does not deal with IoT in a direct sense, it is intended for use across the board and should therefore be a valid option for GDPR compliance within consumer IoT systems, in a general sense. DEFeND takes a modular approach focusing on five central aspects of compliance to the GDPR, with each aspect further split between planning and operational elements. These components are then provided to users as fully functioning

[32] https://www.defendproject.eu/

software solutions, though they remain focused on the GDPR. This focus is enabled by way of a bespoke software platform including components focused on specific areas of the GDPR, but jointly covering the entire undertaking of compliance. This platform then expands the focus area of DEFeND to specifically include privacy and is called "Data Privacy Governance for Supporting GDPR".

The DEFeND platform is built on four procedural pillars, which are 1) User engagement[33] 2) Component integration, 3) Validation[34], 4) Training. These four pillars support the DEFeND architecture which in turn is constituted of five services, five back-end components, and a mapping for each to regulators, data subjects, and data users. The five services are 1) Data Scope Management Service, 2) Data Process Management Service, 3) Data Breach Management Service, 4) GDPR Planning Service, and 5) GDPR Reporting Service. The full architecture is presented in Figure 3.15[35].

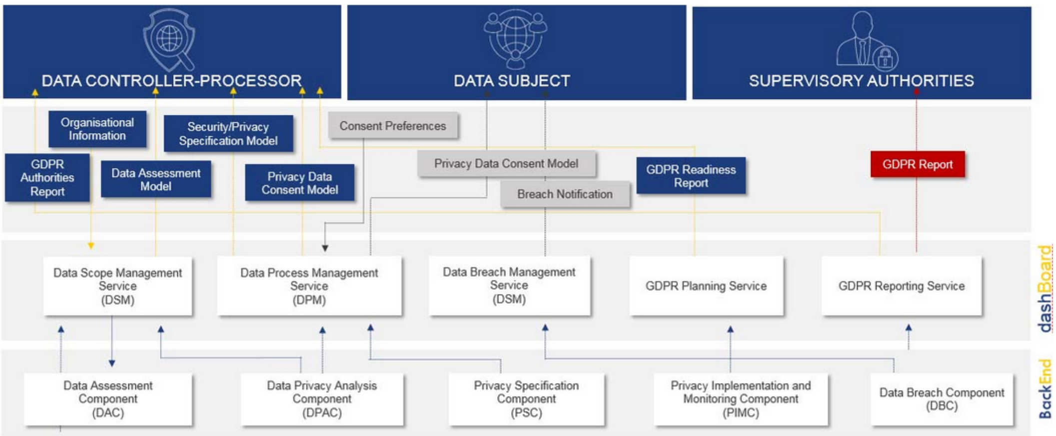



Figure 3.15: Architecture of the DEFeND Platform

The five back-end components are key to the functioning of the system and include software tools and various frameworks for accomplishing their stated aims. Although these components are intended to jointly cover the full spectrum of GDPR compliance, they are modular in nature, both in terms of their interaction with each other and their internal composition. Such modularity allows the user greater freedom in use, but still locks such use exclusively to GDPR compliance. It should also be noted that the tools and frameworks underlying these components are varied in scope, functioning, application and difficulty of use. They are drawn in and their outputs forged into a unified purpose through inclusion in DEFeND, but those seeking to use DEFeND will still have to engage with these divergent tools and frameworks.

Measuring DEFeND against our eight metrics we find the following:

[33] Including various stakeholders.
[34] Including the use of pilot projects.
[35] Although this image is present in the main source, the higher quality image used here is from https://www.defendproject.eu/.

1. Not a comprehensive approach targeting both legal and user privacy requirements as it is explicitly focused on GDPR compliance.
2. Through its various pieces of software and frameworks DEFeND seeks to manage compliance and the implementation of related measures.
3. Regulators are addressed in the system, but auditing is not internalised.
4. IoT is not the focus area.
5. Data flows must be understood to engage with the system.
6. Privacy by design is a central concern, but only in terms of GDPR compliance.
7. The system's data flows are modelled but a formal modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.14 IoT privacy compliance through provenance

Proposing that an audit of data flows can be conducted to show compliance to both regulatory and user privacy requirements, Pasquier et al. take a provenance-based approach to the issue. Fundamental to this approach are transparency and accountability where the former is the ability to gain an unobstructed view of system behaviour, while the latter is the assessment of that system behaviour against regulatory and user requirements. The presence of both must also be evidenced by way of related documentation, that is audit documentation. In this, provenance is the mechanism enabling transparency over information flows, but also the means by which auditing of those information flows is enabled. The authors also hold that system audits and the tools used to enable such audits can be enabling factors for users wishing to exercise their rights.

The approach presented here has strong merit in that provenance, by definition, tracks data origins and changes within a systems, while this can be presented using a clear acyclic graph. Such a graph, referred to as a provenance graph, shows not only the data and the actions on that data, but also who or what acted on it. More formally, these graph elements are referred to as entities (the data), activities (data transformation), and agents. As such, Pasquier et al. views regulatory and user requirements as expected system behaviour while provenance data constitutes the actual systems behaviour, with a comparison of the two allowing for auditing to be conducted. The actual process of using provenance in this manner, is divided into three main components which are identifying compliance violations, audit record legibility, and structuring the audit graph.

Compliance violations are identified by way of 1) drawing the graph 2) comparing data flows to that which is required[36] 3) checking for the effects of context[37] changes on data. There after,

[36]This includes data collection, processing, storage, and the parties (agents) involved.

[37]Per example, a fitness tracker sending an urgent notification to emergency services changes the data collection and processing context and accordingly also the applicable requirements.

care must be taken to present the provenance record in audit findings in a manner which is clear and easily accessible to the intended audience. Although Pasquier et al. grapple with ways of relaying this data to system users, we include this work in Chapter 7 where the intended audience is system engineers already using this work within SysML, thereby automatically attending to the presentation question. It should also be noted that non-SysML elements can easily be referenced within a SysML system model and the use of provenance does therefore not present an obstacle. Lastly, the provenance graph should be split by context, with separate segments for each context but also for data flows between contexts. This compartmentalisation is aimed at ease of use.

Measuring Pasquier et al. against our eight metrics we find the following:

1. A comprehensive approach targeting both legal and user privacy requirements though not explicitly taking a "by design" approach.
2. Not implementation focused, although it describes what analysis to conduct it does not provide a step wise method for doing so.
3. Auditing is addressed, but as a creation of this system and not as a formal audit by external regulators.
4. IoT is the focus area.
5. Data flows form the primary mechanism through which provenance is determined.
6. Privacy by design is not an explicit focus as the the system is concerned with meeting requirements via a snapshot analysis.
7. The system's data flows are modelled but a formal modelling language is not employed.
8. Since a modelling language is not used, a related modelling tool can not be used.

## 3.15 Model-driven IoT risk control

In their paper entitled "Model-driven Evidence-based Privacy Risk Control in Trustworthy Smart IoT Systems" Muntés-Mulero et al. presents their work on preempting privacy related risk in Trustworthy Smart IoT Systems (TSIS). To do so the authors develop a bespoke risk management methodology which they then connect to preexisting components.

The first component used is GeneSIS, an extension to the domain specific modelling language ThingML, with tool support, for the continuous orchestration and deployment of IoT systems. The strengths of GeneSIS include that it directly addresses heterogeneity in IoT applications from the edge to the cloud, and also its ability to deal with system components that have intermittent Internet access [57]. The type of system-wide control proposed by GeneSIS places it at odds with the mix-and-match approach often found in consumer IoT. This is also why DISCREET functions at both the device and system level ensuring that compliance and PbD can be localised in each component without the need for system wide control[38].

The second component is DFDs, for which they briefly mention STRIDE but more strongly lean on LINDDUN, specifically since the latter directly references the former [109]. With the DFDs providing the functional element and GeneSIS the technical element, the new risk management methodology can then guide the process of TSIS design. The methodology runs over six phases with different roles assigned to each stage. The relevant roles are Risk Management Owner(O); Product Owner (P); Architect(A); Developer (V); Risk Analyst (R). Although we will not be looking at the methodology in any further detail, the six steps and their related roles are included below Figure 3.16 as taken from Muntés-Mulero et al..

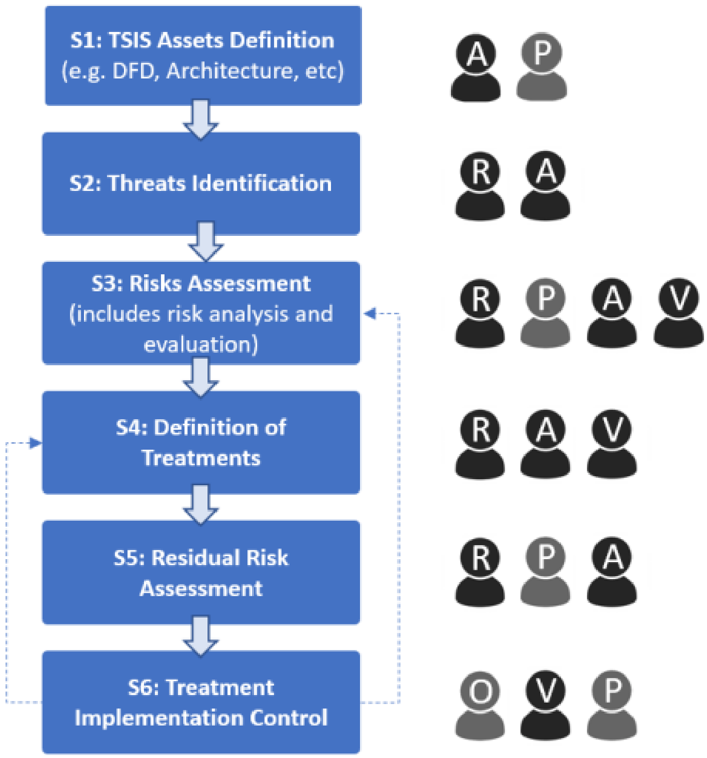



Figure 3.16: Six methodology steps by Muntés-Mulero et al.

The sixth step in this methodology is of specific significance since it requires that previously derived mitigations not only be implemented but continually monitored and improved as needed. This lifecycle-based approach is also highlighted by the authors as a key a strength of their methodology. These and other similarities between this methodology and our domain extension are however, not as deep as they might at first appear. To the extent that deeper fundamental differences preclude the incorporation of this work into DISCREET.

In addition to the point made earlier about GeneSIS taking a system-wide approach, there is also one other fundamental reason why GeneSIS and any system incorporating it could not be

[38] This is simply a point about the applicability of GeneSIS to our needs and not a criticism.

included in DISCREET. That is its position as extension to the domain specific modelling language ThingML. ThingML[39] is intended for use in IoT, cyber-physical systems, and embedded systems applications, and includes both tooling and a methodology. This is a significant difference since ThingML is not a general purpose language, while SysML is. Coupled with the inclusion of a methodology and the result gains focus but at the expense of the freedom offered by SysML. DISCREET, as a modular domain specific extension to a general purpose language not only addresses the needs of its focus domain, but also gives the modeller far greater freedom in deriving a solution fit for their individual needs.

Measuring Muntés-Mulero et al. against our eight metrics we find the following:

1. Not a comprehensive approach targeting both legal and user privacy requirements as it is focused on privacy.
2. Implementation is addressed both as result of design and the development of mitigation strategies and also on a lifecycle basis.
3. No formal auditing is included.
4. IoT is the explicit area of focus.
5. Data flows are extensively tracked and included in the methodology.
6. Privacy by design is a central concern though this term is never mentioned explicitly.
7. The system's data flows are modelled and a modelling language is also used.
8. The associated tool is available as plugin for the Eclipse IDE.

## 3.16 IoT modelling in SysML

As SysML is a general purpose modelling language there is no intrinsic reason it should not be applicable to the design and development of systems in any field. This does however also open the door to domain specific extensions intended to enhance SysML's ability to generate models in a specific domain. Given the prominence and continued growth of the IoT in general it is therefore not surprising to find domain extensions targeting this field. It is against this backdrop that Costa et al. also identifies the interdisciplinarity and heterogeneity of the IoT as a textbook case for the use of systems engineering and propose SysML4IoT.

Despite these opening remarks seemingly indicating a point of connection there are, however, significant points of difference at a fundamental level between DISCREET and SysML4IoT. In the first instance, the latter is focused on the project design phase as opposed to dealing with the entire system lifecycle. More significant though is that DISCREET is a domain extension

[39] More information about ThingML can be found at: `https://github.com/TelluIoT/ThingML`.

of the type *«modelLibrary»*, while SysML4IoT is a domain extension of the type profile[40]. In short this means that DISCREET is language compliant with SysML and can therefore be used in SysML tools as is, the same is not true for SysML4IoT and it accordingly introduces a tool. Both DISCREET and SysML4IoT do however take up the point that SysML is a language only, by introducing methods to guide application of their respective systems.

A further divergence between the two domain extensions is that SysML4IoT is not a standalone entity, but functions within a larger methodology aimed at addressing issues of heterogeneity, divergent stakeholders, and IoT application design. This methodology is a model-based systems engineering approach called IDeA – IoT DevProcess & AppFramework. In Figure 3.17, taken from Costa et al., we can see how IDeA draws in various components including SysML and then splits its functioning into a development process (IoT DevProcess) and a tool (IoT AppFramework[41]). From there, the totality of IDeA, including the SysML4IoT profile, can be used during the design of new IoT systems by means of the previously mentioned tool.

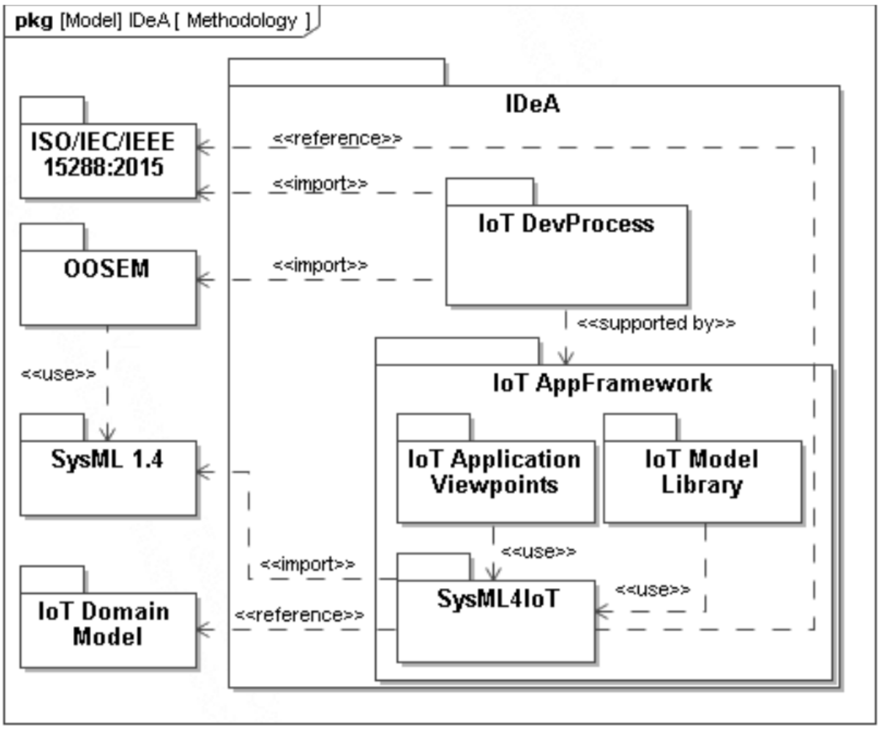



Figure 3.17: The IDeA methodology meta-model

Of the four elements drawn in to IDeA, the first three are established industry standards. The forth, the IoT domain model, is the IoT-A reference model which aims to standardise IoT architectural referencing[42]. The next significant element to SysML4IoT is the profile itself. A partial representation thereof, taken from Costa et al. is included in Figure 3.18. A bulk of the included

[40]A profile changes one or more of the base language's components by way of *«stereotype»* additions and/or alterations.

[41]The IoT AppFramework is a plugin for the No Magic Cameo Systems Modeler. As such it will not work with other tools, which is unfortunate given recent difficulties with the tool vendor. The future impact of SysML v2 on SysML4IoT is also a potential cause for concern.

[42]IoT-A is available here: `http://www.iot-a.eu/public`.

stereotypes are taken from the IoT Domain Model while blocks and stakeholders are standard SysML elements. System, system-of-interest and enabling system, on the other hand are taken from ISO/IEC/IEEE 15288. Additionally, services are defined as having four attributes which are: serviceType, technology used; in and out, data types used for input and output; serviceArea, the area affected by the service.

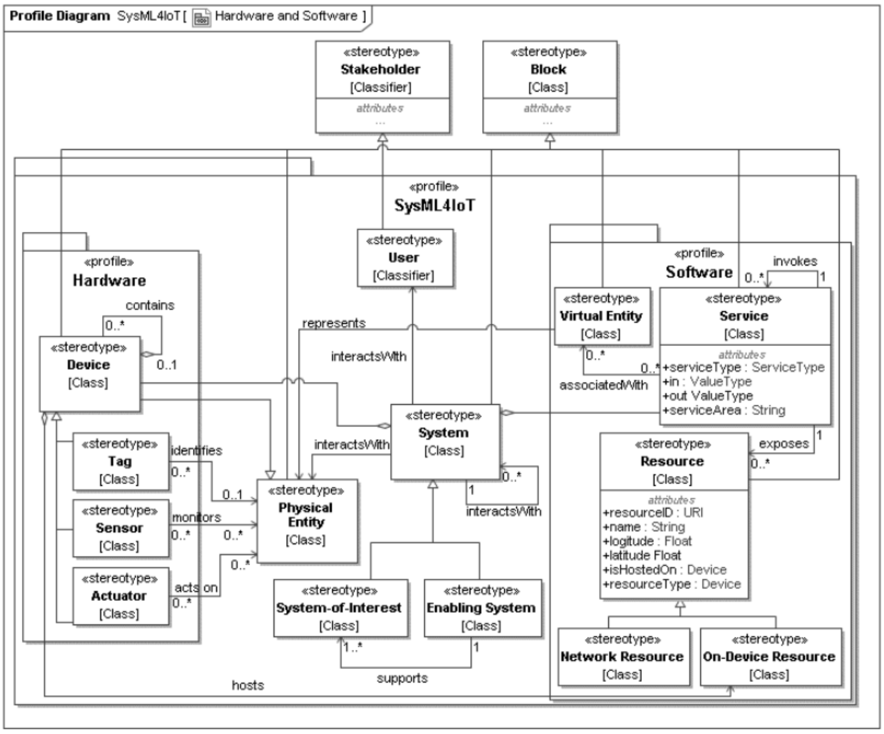



Figure 3.18: Partial SysML4IoT profile

Measuring IDeA, including SysML4IoT, against our eight metrics we find the following:

1. Not a comprehensive approach targeting both legal and user privacy requirements.
2. Implementation is extensively addressed.
3. No formal auditing is included.
4. IoT is the explicit area of focus.
5. Data flows are extensively tracked and included in the methodology.
6. Privacy by design is not an area of focus.
7. SysML is used and an extension thereto is developed.
8. A dedicated modelling tool is used.

## 3.17 The Need for a new approach

The preceding discussion explored compliance and PbD for consumer IoT, our chosen problem space, by focusing on the first two sub research questions. What we found was a wide ranging set of approaches to various aspects of the problem space, but none that addressed all the issues we are concerned with. We have also shown that even systems which target a large number of our areas of concern still do not cover all of them sufficiently. Furthermore, delving deeper into these systems we found that many of the similarities between them and the work proposed in Chapter 7 are superficial. The work of Muntés-Mulero et al., as discussed in Section 3.15, is a good example of this. These findings serve to answer the second research question [43].

Only the CNIL's PIA directly addresses the IoT in reference to GDPR compliance and related auditing. Although none of the work examined presented a comprehensive "by design" approach to both PbD and compliance for consumer IoT, the effort is far from wasted. We gained valuable insight into major topics related to our problem space and also explored the functioning and limitations of methodologies, frameworks and other prominent approaches to that space. The finding is two-fold, in that not only does none of the work reviewed meet all criteria, but the simultaneous use of several of these approaches might be functional, but far from optimal. This latter concern is located in incompatibilities relating to differing scope, operation and intended outcomes. For example, the metric of comprehensiveness requires that both PbD and compliance be addressed and since the CNIL's PIA squarely deals with compliance and makes inroads into privacy beyond mandated compliance, using it in combination with LINDDUN would seem a good option. However, doing so does not yet address all the metrics and creates a whole slew of new challenges around tracking and managing two different systems with divergent goals and procedures. This leaves the system engineer with a great deal of clutter to clear up all in aid of an unsatisfactory result.

Including the final two metrics, that is systems modelling and the use of a standardised modelling language, we can begin to see a solution to the challenges introduced above. This however also underlines a well established process within interdisciplinary research, which is to establish points of commonality between varying fields and to use these points as a foundation from which to build out new, but linked, work [127]. Similarly we intend to use points of commonality, as established by reference to our eight metrics, as foundation for our answer to the challenge of compliance and privacy by design in consumer IoT devices and systems. This implies using portions of the work introduced in this chapter to build out the domain extension presented in Chapter 7. To give a clearer image of the measure to which each of the works discussed performs against the metrics proposed, we present Table 3.4.

From the results presented in Table 3.4 we can also provide an answer to our third research question[44], which is clearly the CNIL's PIA. Although primarily aimed at legal compliance, it

[43] RQ2: Considering the defined problem space, what are the coverage gaps in prominent current frameworks and methodologies, including those not directly aimed at IoT?

[44] RQ3: Which of these methodologies would best serve to assess the testbed results presented herein?

| Feature | Babar | Ziegledorf | CNIL | SDM | LINDDUN |
|---|---|---|---|---|---|
| **Comprehensiveness** | No | No | No | No | No |
| **Implementation** | No | No | Yes | Yes | Yes |
| **Auditing** | No | No | Yes (Full) | Yes (Limited) | No |
| **IoT Focus** | Yes | Yes | Yes | No | No |
| **Data flows** | No | No | No | No | Yes |
| **Privacy focus** | IoT threats | IoT threats | PbD and DPbD | PbD and DPbD | Threat modelling (PbD) |
| **Modelling** | No | No | No | No | No |
| **Standardisation** | No | No | No | No | No |
| | | | | | |
| **Feature** | **MPRAM** | **DEFeND** | **Pasquier** | **Muntés** | **SysML4IoT** |
| **Comprehensiveness** | Yes (Limited) | No | Yes (Limited) | No | No |
| **Implementation** | No | Yes | No | Yes | Yes |
| **Auditing** | No | No | Yes (Limited) | No | No |
| **IoT Focus** | No | No | Yes | Yes | Yes |
| **Data flows** | No | Yes | Yes | Yes | Yes |
| **Privacy focus** | Security Goals | PbD and DPbD | Requirements | IoT threats | No |
| **Modelling** | No | No | No | Yes (Limited) | Yes (Limited) |
| **Standardisation** | No | No | No | Yes (Limited) | Yes (Limited) |

Table 3.4: Eight metrics used to assess various approaches

does not integrate into an established modelling language and does not address privacy beyond compliance at sufficient depth, this PIA remains an excellent tool for not only assessing legal compliance but interrogating the entire system under audit. It therefore answers compliance questions directly and provides a significant portion[45] of the information needed to investigate those issues not directly covered by legal compliance.

Furthermore, Table 3.4 shows that none of the current approaches fully overlap or sufficiently address the eight assessment metrics, yet we can make a clear case for the need of such work. There are of course more than one possible explanation for this, including the scope and complexity of the problem. A further significant factor is also the continued development and growth in the IoT implying that it is still far from being a mature technology, and accordingly there is much work to be done on the topic [95]. This of course precludes any academic consensus being reached on which segments of the problem space to address first, or indeed how to address them. This is, however, not the case in industry where there is a clear and pressing need for a new approach which, at the very least, addresses both compliance and PbD over the entirety of the product lifecycle.

The need referred to above was of course illustrated by way of numerous examples in Chapter 1, though for that need to be fully met, any proposed solution must clearly be both deeper and broader than merely indexing existing solutions for compliance and PbD. Additionally, since both broader privacy issues and more clear cut compliance issues are dealt with, a guidelines or lose framework approach favoured by some, can not be taken [118]. The proposed solution must therefore be well structured and methodological in nature, directly address both compliance by design and privacy by design for consumer IoT, be standardised and have wide applicability. These last two requirements specifically speak to the ever broadening scope of consumer IoT applications as well as the proliferation thereof [40]. Simply put, a bespoke solution for every subsection of consumer IoT will not work.

[45] Aspects such as the elicitation of stakeholder needs are of course not addressed.

From the preceding we can now conclude that a structured and comprehensive new methodology for engineering privacy by design into IoT devices and systems is needed. Presenting it in an industry standard modelling language will further aid utility and reusability. We justify our position on the following grounds. First, the eight metrics introduced in this chapter are not adequately addressed by any single one of the approaches we covered. Second, numerous other authors have called for all or some of these metrics to be addressed with regards to consumer IoT. Third, any project that fails to address these issues runs the risk of serious repercussions ranging form substantial fines to lost revenue due to negative customer reaction[46].

In the next chapter we will present a testbed populated with consumer IoT devices and use the CNIL's PIA to assess these devices over an extended period.

[46] The "My friend Kayla" case, for instance, presents a good example of just this.

# Chapter 4

# Testbed Findings

## 4.1 The testbed

Thus far, we have highlighted numerous failings in both PbD and compliance for consumer IoT, but also a lack of any single approach aimed at addressing this over the entirety of the system lifecycle. Establishing these positions though, has been done by way of reference to the work of others and that includes work indicating the need for the research conducted herein. As part of the discussion we also established that the CNIL's PIA should be used to assess the results of our primary research. How to obtain those results was discussed in Chapter 2, which dealt with setting up the testbed and related systems.

In this chapter we present the findings gained from our testbed as the final segment of our ground work, presented in Figure 4.1. This includes running the testbed over a period of more than three years, using various tools to gather data from the testbed and finally interrogating the data gathered using the CNIL's connected devices PIA audit methodology for GDPR compliance. The use of this specific PIA was discussed in the previous chapter and is the answer to our third research question[1]. Implementing that decision in this chapter not only provides us with significant insight into the current state of compliance and PbD in consumer IoT, but also describes the shift in that position over time thereby providing an answer to our fourth research question.

- RQ4: Do real-world observations of a consumer IoT implementation match up to the expected shortcomings, both as a snapshot and over time?

Although the work presented here is of course shaped by the discussion in preceding chapters, the analysis presented is entirely based on primary research using the testbed devices, their smart phone based control apps, the associated data handling[2], and the accompanying terms and conditions presented to end users. However, We did not reverse engineer any software nor extract any

[1]RQ3: Which of these methodologies would best serve to assess the testbed results presented herein?

[2]This includes local storage and processing as well as M2M communications with off-site resources.

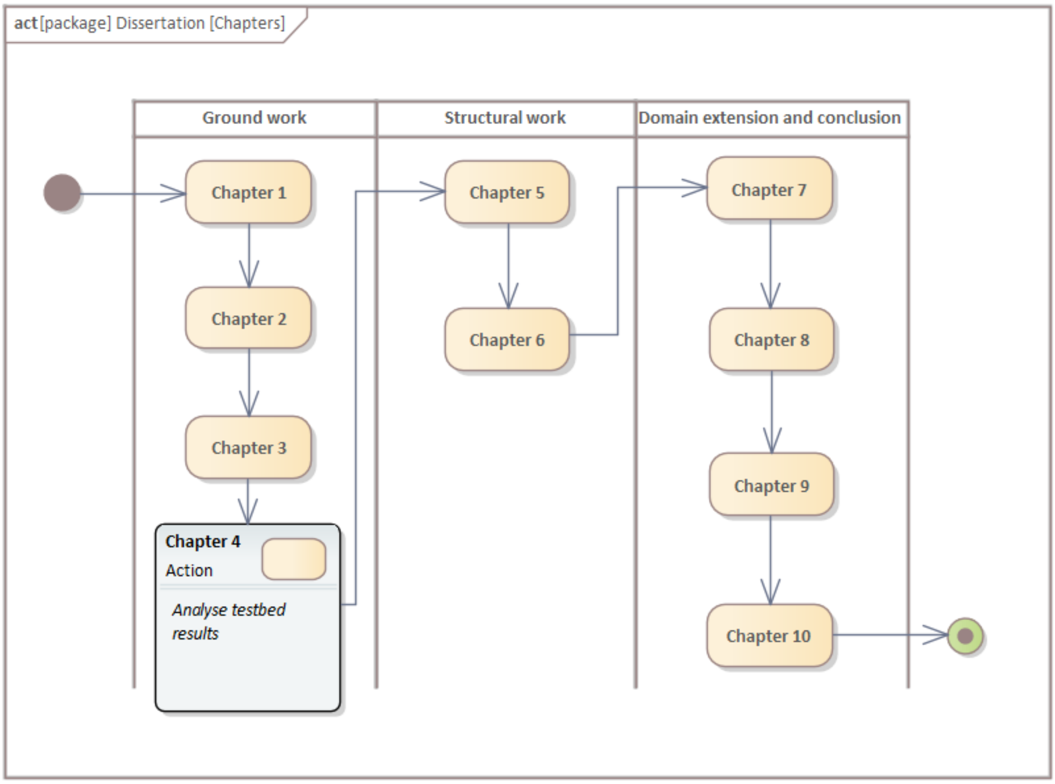



Figure 4.1: Chapter 4 research focus

logs or other data directly from the devices or the related control apps. Taking these additional steps would not serve any clear purpose as the testbed already yields far more data then needed to conduct the research outlined here and in Chapter 2 and the actual format and contents of device logs are also liable to be widely different [157] thereby increasing the associated effort without providing any defined gain.

## 4.2 Findings and assessment

The results presented in the following sections and subsection include the most significant findings from our analysis covering data generated by all the tools used, the testbed data flows, and the device terms and conditions. These constitute self-evident issues as well as the high level results from the privacy impact assessment (PIA) performed to gauge GDPR compliance. Due to the number of devices, the activity of their control apps, and the use of tracking software by some apps, the resulting data flows are far too involved to be presented here in detailed diagram format. One high level overview, focused on data flows away from the data subject's control, will still be included below though. A further reason for the specific choice of the CNIL's PIA, apart from its direct focus on the GDPR and reference to the IoT, is that it can be used in a modular fashion, applying those components covering the data we have access to. Specific data that could be covered but which we did not have access to are those data relating to the internal functioning of the data controllers and processors which we could not track via the testbed. Examples of the latter include internal controls and procedures within those companies. Even so, the results here are based on the PIA's full knowledge base [34], methodology [35], templates [36], and IoT implementation [33].

These results are further enriched by way of reference to the SDM's data protection principles. As such, the privacy issues presented below contain two sets of results. The first are privacy violations as contraventions of the GDPR, identified by way of the PIA. The second are SDM data protection principles which have been breached. What is not included though is the measure of scale or severity for each infraction. The CNIL PIA facilitates such assessment, but to do so one would have to be an internal auditor for the party under audit. As external users of the PIA we do not have sufficient data, but this does not have a meaningful impact on the assessment since the primary goal here is to identify the infractions present, not grading them.

### 4.2.1 General findings

To introduce, and provide context to, the formalised discussion of testbed findings presented below, we present the following general findings. These are also graphically presented in Figure 4.2, which uses a STRIDE DFD model[3] to present data flows and associated trust boundaries. This is a high level and simplified view though, since a truly detailed representation of all the data flows would contain so much detail as to undermine legibility and functionality. Specifically, we treat the IoT devices and the off-site servers as black boxes and do not provide an overly detailed view of the smart phone's internal operations.

The first area of significance is the IoT device itself, which can be interacted with by the data subject and at a minimum has a Bluetooth connection to the associated smart phone. It is also common for IoT devices to have Wi-Fi modules and connect to the testbed network via those. Unfortunately, it was also found that IoT devices used these connections to directly communicate with off-site servers, with such communication sometimes including personal data. This is noteworthy since it bypasses the data subject's locus of control, which is the control app.

The second area of significance is the smart phone. It hosts the control app, other system apps, non-volatile storage, and as we found, additional tracking software installed by the control apps. This tracking software can take the form of fully independent applications, or code embedded inside the control app. For the sake of clarity Figure 4.2 shows this tracking software as a standalone app to demonstrate that it has access to system storage independent of the control app and also receives additional data flows from the control apps. Lastly, where control apps send data to both known and unknown off-site servers, those that the tracking apps connect to are by definition unknown and yet again underscore the position of these trackers as uninvited guests on the testbed.

[3] As with all model elements in this work, Sparx Enterprise Architect (EA) was used to generate this figure. Although EA supports a full suite of STRIDE tools including colour coding for threat classification, these tools were not used and the DFD only represents a single IoT device and its associated data flows.

The third and final area is the Internet-based services involved. Typically these are presented to consumers as “cloud storage” and must be clearly visible to the data subjects concerned. We did, however, not find this to be the case for the testbed devices.

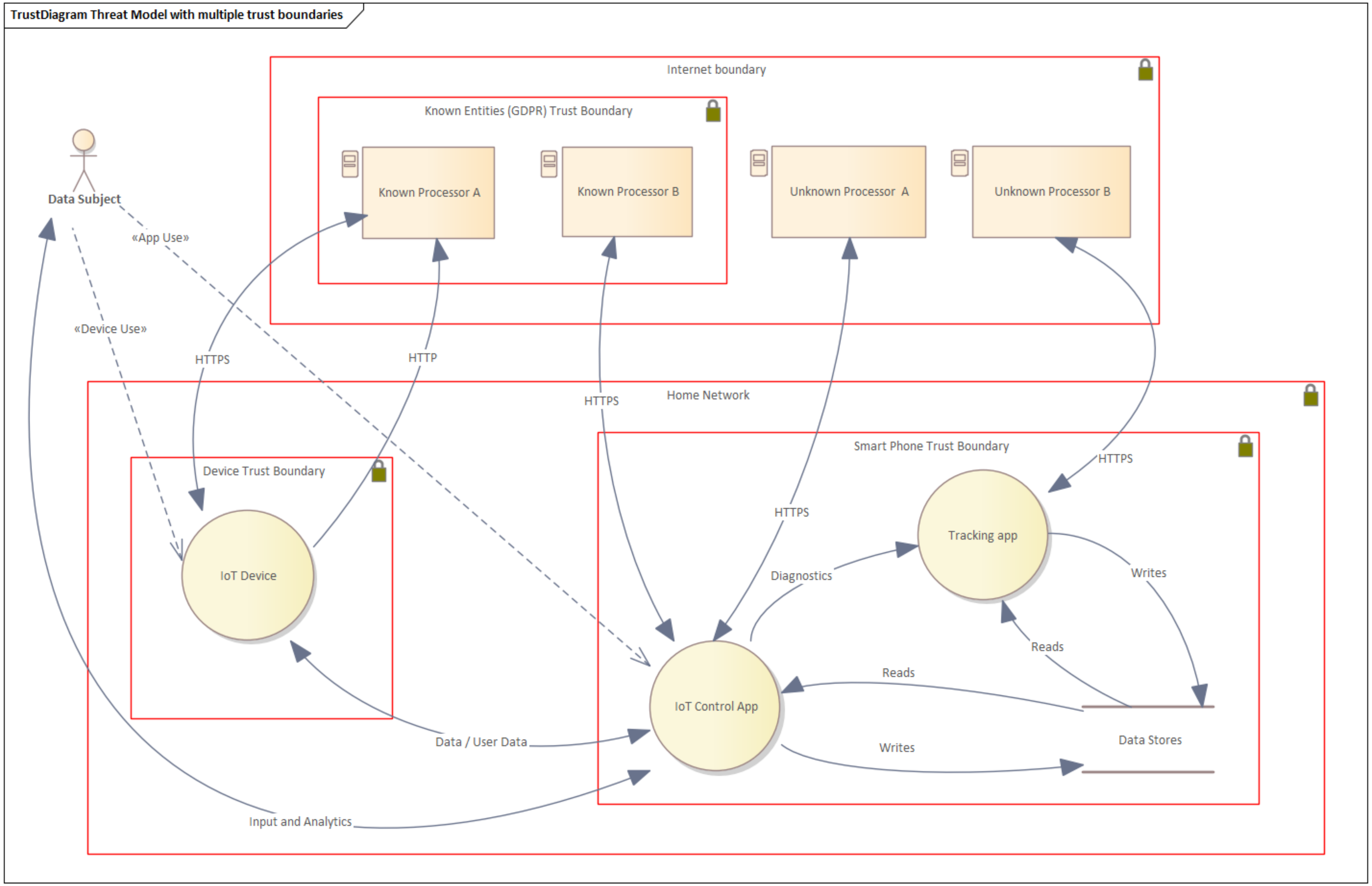



Figure 4.2: Testbed trust boundaries in a simplified view

### 4.2.2 Purpose limitation violations

As already discussed, all the testbed IoT devices have associated smart phone control apps, which is a standard approach within consumer IoT [164]. This is done to shift some of the processing and interface needs from the IoT device to the phone, thereby reducing the specification and costs of the devices. The apps are also easier to update and manage over time. Accordingly, there is no inherent reason for the use of such apps to present a challenge to either compliance or PbD, unless they are purposefully designed to contravene either or both. Unfortunately the latter proved to be the case as the Lumen privacy monitor located third-party tracking software connected to all the control apps except one. The stand out case was the TPLink Kasa app, with no additional tracking software included in the version tested. Such tracking software is supplied by, or even managed by, third-party providers which are never declared to data subjects. The obvious opportunities for such declarations would have been in the formal terms and conditions or even in the prompts given during app and device setup on the phone. Making this all the more egregious is the finding that none of the tracking software perform functions that were aligned to the stated purpose of data collection for the device.

Although there is a wide range of such tracking software available on the market, only two types were detected on the testbed. The first, which was used by all the involved control apps, is called Crashlytics and is used for crash reporting and associated diagnostics. The second piece of software is Amplitude which is a full blown behavioural[4] analytics suite. Amplitude is deployed by the Health Mate app which controls the Withings Thermo, Body and Go.

The undisclosed presence of tracking software on devices which hold a range of personal and potentially sensitive data is deeply concerning in and of itself. Worse still, both Amplitude and Crashlytics communicate with servers in the US, irrespective of any limitations on processing. However, neither were found to directly access or compromise the initial personal data that data subjects provided to the control apps. This might then seem to be the proverbial get out of jail free card for the data controllers, but it is not the case. The tracking software is generating new data relating to data subjects, including the other apps on their smart phones, how they use the individual control apps and an extensive list of personal preferences. Such a data set can easily be used to gain significant insight into the personal life of the data subject. These data clearly fall under the GDPR's purview and is not automatically exempt as some might be tempted to argue. This is also a clear case of the modern trend to the over collection of data, spurred on by the value of personal data and in contravention of data minimisation and proportionality[5] principles [24].

In the first instance, this new data about the data subjects are pseudonymised in the sense that third-party providers only have access to user IDs but not the natural names or other details of the individuals concerned[6] This would accordingly indicate that the data can be treated as

[4] In their more recent marketing, Amplitude opt for the term "product analytics" though the functionality is still focused on the behaviour of users.

[5] That data collection is proportional to the data needs for stated purposes.

[6] We deduced this by inspecting the data flowing across the testbed.

anonymised [108] and that the provisions of Article 11 apply which hold that once a data subject can no longer be identified in the data, then the other provision of the GDPR no longer apply. However, there is a significant sting in the tail, as Paragraph 2 of the same article holds that, should a data subject be able to provide additional information that links them to the data concerned, that data is clearly deanonymised and the GDPR again applies. This is a significant problem since our MITM setup allows for the additional information to be collected. Not only can this process be replicated by determined users, but the Withings Thermo was also found to communicate some data in plain text over an HTTP connection. Although this latter example relates to data being passed to a Withings server, the issue still stands.

Violation of the purpose limitation:

- Factor: Data collection is conducted beyond declared purpose.
- Factor: Data controllers do not inform data subjects of this data collection or the data processors involved.
- Risk: Data relating to data subjects are collected and processed without data subjects being properly informed, and consequently without due consent. This includes data that can be used in profiling the data subject. Any use, sale or leak of such data could impact on the data subject without their knowledge and consequently without them being able to counteract such an outcome .

Data protection principles violated

- The additional data collection counteracts the principle of data minimisation
- Since data subjects can not interact with data they are not aware of, availability and intervenability are violated
- The lack of information on data collection contravenes the principle of transparency

### 4.2.3 Lack of consent

In addition to reducing the cost of consumer IoT devices by shifting some processing, functionality, and input to the control app located on a smart phone the user already owns, doing so is also intended to streamline the process of presenting terms and conditions and gaining user consent. Streamline in this sense refers to providing all relevant documentation in electronic form during the device setup process. Unfortunately, we found that only two of the testbed devices approached this in a compliant manner.

Per example, the Misfit Shine 2 presented the data subject with a confusing array of policies and procedures spread over several web pages. These were a “General Terms of Use” policy, an “EU

Privacy Policy", and an "App Privacy Policy". Worse still, is that these policies were not even presented to the data subject during device setup. To find them the data subject would have had to independently visit the brand's website and search for the documents. This is also not the end of the problem since subsequent revisions to the website have removed the policies listed above and replaced them with a "Privacy Policy", "Cookie Policy", and "Terms of Use" but at the time of writing all three links were dead and simply displayed an error message. There is however also a link to what appears to be a form requesting that the data subject's data not be sold by the Fossil Group. This is also dead, and while the Fossil Group owns Misfit, its involvement was never disclosed to data subjects. Although other researchers have already demonstrated a deterioration of security for IoT devices over time, we found this to extend to compliance and PbD. The problem of PbD and compliance deteriorating over time is further discussed in Section 6.1.

Presenting a whole different set of challenges, the Withings devices use a shared app which directly prompts the user to agree to the applicable terms and conditions including a privacy policy. Clicking into these terms and conditions though, takes the data subject to a website hosting the applicable documentation. Troublingly, for several months during the initial year of the testbed operation, these were not Withings documents but in fact Nokia documents. Though similarly titled, these documents detailed an agreement with Nokia. The cause of this is however simple to explain. Withings was originally an independent company, which was then bought by Nokia, became Nokia Health, under performed and was sold back to one of the original Withings founders [25]. Although this explanation is clear, it does not constitute a valid excuse since Nokia and Withings were by this time separate legal entities. As such, there was no valid agreement with the company the data subject perceived as the data controller (Withings). This is further aggravated by the fact that Nokia at this time continued to act as data processor with Withings devices and apps still sending data to Nokia owned servers. One standout device in the Withings lineup is the Home camera and air quality monitor which has a standalone app. During setup the app made no mention of any relevant terms and conditions but presented a link to the aforementioned Nokia documentation via the app's settings tab.

Privacy risk brought about by a lack of consent:

- Factor: The provision of informed consent can not be demonstrated by the data controller.
- Factor: The controller does not provide accessible and understandable documentation to the data subject.
- Factor: Data subjects are unable to exercise their right to withdraw consent since it was not properly and transparently obtained.
- Risk: The data subject's personal data are collected, processed, and stored without a fully formed agreement obtaining consent reflecting the reality of the situation. As such, data subjects risk losing the full protection they are entitled to.

Breached data protection principles:

- A third-party with no explicit link to the data subject is conducting the storage and processing of personal data, thereby breaching the principle of confidentiality.
- Since the data flows to Nokia servers was not explained, the requirement for transparency was breached.
- Given the lack of a formal relationship it is unclear how data subjects would exercise their rights against Nokia and as such the principle of intervenability is breached.

### 4.2.4 Inaccessible data

Although the failure to manage and explain data collection and processing directly impacts the obtaining of consent and leads to a violation of the purpose limitation, it also has a secondary effect. This effect is the undermining of a data subject's ability to access their data due to a lack of knowledge. In the most direct sense the data subject is simply not aware that the data exists and can therefore not exercise their rights against it. This specifically relates to the fact that a large proportion of these data are not presented inside the associated control apps. This also implies that the data subject will not be able to enact an inclusive data subject access request (DSAR) since they might not even know the correct party against which such a request is to be filed.

This problem is exasperated by devices and apps either generating or obtaining data without input from the data subject and then communicating those data to off-site servers. This includes the collecting of user and environmental profiling data and communicating that back to device manufacturers. Per example, the Misfit app collects data on the type of smart phone and operating system used, while the Withings Thermo communicates its MAC address and user credentials over HTTP to a server. A clear example of the problems this can cause comes from the GetHealth data breach which exposed a set of health and personal records with more than 61m entries. This included names, email addresses and locations, as well as which IoT devices each data subject used [59]. GetHealth is a New York based firm providing datasets sourced from health and fitness wearables and although no mention of it was made to Misfit or Withings customers in the documentation examined, data from both companies were included in the leak. This was also not just for US-based customers but globally, including continental Europe and the UK.

The associated privacy risks to the data subject is also much broader than it might seem at first glance since the risk is located in the cumulative effect of such data [126]. Consider for instance that the Amazon Alexa requires access to a record of all the Wi-Fi networks the associated smart phone connects to as well as the contacts stored on the phone. This can easily be used to deduce any number of additional data points about the subject and if linked to external identifiers such as an advertising ID, the potential for abuse becomes even greater.

Privacy risk derived from lacking data access

- Factor: Since data subjects are not able to fully exercise their data access rights against Article 15, it follows that their Article 16 to 20 rights are also infringed.

  Article 16: Right to rectification

  Article 17: Right to erasure

  Article 18: Right to restriction of processing

  Article 19: Notification obligation regarding rectification or erasure of personal data or restriction of processing

  Article 20: Right to data portability

- Risk: Third-party processors are involved without the data subject's knowledge, or any clear means to obtain such knowledge. This often includes the processing of data the subject is also unaware of. Cumulatively this represents a broad infringement of the data subject's rights and a significant risk to privacy, exasperated by the accumulation of seemingly innocuous data into ever larger subject profiles.

Breached data protection principles:

- The collection of data not needed for core functionality of the service offered breaches the principle of data minimisation.
- The accumulation of data on the subject and their environment breaches the principle of unlinkability.
- The undisclosed data collection and accumulation breaches the principle of transparency.
- Since the data subjects are not aware of the data in the first instance, they can not alter or interact with that data and the principle of intervenability is breached.

## 4.3 LINDDUN-based assessment

The CNIL PIA is of course targeted at assessing compliance on multiple levels including the business process level and accordingly, only a portion of the full methodology could be applied herein[7]. Even so, the process of going through the compliance assessment was significant. For LINDDUN though there is no such limitation as the full methodology can be brought to bear on any data which are "in scope". Although this makes for a more complete picture it also presents the risk of analysis overload due to the extensive detail such an examination can go in to. To counteract these concerns, the following discussion is constrained to the compliance failures identified in the previous section and specifically the manner in which LINDDUN approaches these, as opposed to targeting the entire testbed and deploying the full methodology.

[7]This is reflected in DISCREET itself as only the applicable subset of the CNIL PIA is imported.

Focusing on compliance failures not only allows for a discussion of a manageable size but also highlights the way in which LINDDUN's guarding associations can be dealt with. Here, because an existing system is being analysed, guarding associations are addressed and specific PETs are proposed. This contrasts with the application of LINDDUN presented in Chapter 8 where concealing associations are identified and addressed by way of requirement generation for a system still under design. In both cases the use of LINDDUN directly addresses shortcomings identified by the PIA. In the guarding association case, existing privacy threats are addressed through PET implementation while in the concealing association case, potential shortcomings identified via the PIA are mitigated by way of new requirements.

### 4.3.1 Discussion

The three headline failures introduced previously are the purpose limitation violation, a lack of consent, and inaccessible data. Although these are all compliance failures and therefore relate to the same LINDDUN threat tree, there is some additional nuance here. In the first instance, much of what is wrong here relates to the involvement of third-party tracking apps. The simplest solution to this issue would be to remove these apps as they do not add direct value to the data subject, are not needed for the provision of any user oriented functionality, and do not operate as GDPR compliant data processors. From the data subject's perspective (view) such removal would have no impact on system functionality since these third party trackers do not influence customer facing system functions directly[8]. As opposed to that of the data subjects, any system view relating to regulators would include these third-party trackers. Though their removal would still not hinder over all system functionality and would instead have an inherently positive impact on privacy outcomes. This being said, it would still be possible for the service provider to include these trackers and move from a state of non-compliance to one of compliance. This will be included in the discussion below which considers each leaf on the non-compliance threat tree individually.

### 4.3.2 NC_1: Tampering by an attacker

The threat of potential tampering by an attacker generally relates to instances where privacy policies and consent integrate with access controls. Since the testbed-based research does not provide any insight into the functioning of off-site systems the opportunity to assess this issue is limited. For all the devices on the testbed, with regards to the data subject, consent and access control are integrated while privacy policies are not.

[8]Analytics gained from behavioural trackers could be used to change functionality over time, though this is still not a direct impact or even one that will necessarily take place.

Typically, a privacy policy is hosted externally and only linked to during device setup. Consent and access are, however, generally linked via the control apps hosted on the smart phone. Accordingly, no instances of such a threat were detected on the testbed while even the likelihood of this type of attack must be viewed as slim to none.

### 4.3.3 NC_2: Incorrect or insufficient policies

Incorrectness and insufficiency played significant roles in the "lack of consent" discussion in Subsection 4.2.3. This includes the Withings devices not declaring Nokia as a processor, or in one case, simply not linking to privacy policies at all.

The LINDDUN Privacy Threat Tree catalogue [165] directly states that if "*the privacy policies are incorrectly or insufficiently implemented (NC_2), the system will not be compliant.*". This is indeed also what was found in the compliance audit.

### 4.3.4 NC_3: Insufficient policy management

Similarly to NC_2, NC_3 also featured heavily in the "lack of consent" discussion. Insufficient policy management within LINDDUN refers to the absence of well-formed, or user friendly, policies. A major culprit on this count is the Misfit Shine 2 with its initial jumble of compliance requirements, which have subsequently been replaced by almost none at all.

Interestingly, LINDDUN holds that failures of this type inherently leads to non-compliance with relation to user-based consent. On the face of it, this might seem an obvious proposition. However, going beyond the establishment of a link and positioning this relationship as being inherently the case is the greater insight here as nothing short of clear, precise and actively managed policies would have yielded a compliant outcome for the testbed devices.

### 4.3.5 LINDDUN mapping

Insufficient notice (NC_4) is not discussed here as it relates to corporate level actions. We can therefore take the above threats, that is NC_2 and NC_3 and move to the taxonomy and mapping. From the LINDDUN taxonomy we find that compliance relating to policies and procedures is located under guarding association and as expected would yield a PET-based remedy.

The associated mitigation mapping is fairly simple as all NC results are condensed to a single outcome set with three recommended PETs. These are The Platform for Privacy Preferences Project (P3P), eXtensible Access Control Markup Language (XACML), and Enterprise Privacy Authorization Language (EPAL). As the first is aimed at websites and the last at data collection and use within an enterprise, the middle option holds the most promise. XACML not only deals with access control directly and on a low level, but allows for communication and interoperability between multiple parties. Deploying XACML would necessarily require the alignment and reworking of policies and procedures presented to clients, both in static and in-app forms and would thereby be a significant step in addressing the listed compliance failures. Furthermore, the interoperability feature could be used to bring third-party processors into the fold, provided that their activities are also clarified to data subjects. It is also clear from these caveats that although there is definite value in using LINDDUN in this manner, clear external structure is needed to guide the process. This again confirms the need for coherent systems modelling which can provide a single source of authority for these activities. Expanding on this point is the subject of the next chapter.

# Chapter 5

# Model-based Systems Engineering

## 5.1 Introduction

Modern systems are increasingly complex, costly to develop, difficult to manage with a document centric approach, and place greater emphasis than ever before on meeting stakeholder needs. It is exactly this challenge which model-based systems engineering (MBSE) is uniquely suited to address and which is covered in this chapter. This development process does not just speak to the increasing adoption of MBSE but also the amount of work, especially around frameworks, methods, and domain specific additions that remains needed [130]. As such, adopting MBSE allows us to draw on an extensive body of knowledge, frame our contribution in a manner that aids utility and allows for industry adoption, but also opens the door for others to reuse and modify our contribution to fit their needs. This level of pragmatism is also generally taken to be a core value of systems engineering [72].

As the first chapter of the “Structural work” section, as shown in Figure 5.1, this chapter introduces MBSE and SysML in answer to the challenges researched and discussed in the “Ground work” section. This answer is then connected to the preceding work via Chapter 6 in preparation for the new work developed in the “Domain extension and conclusion” section. As a result, Chapter 5 does not present new research but instead draws in background material focused on SysML. As discussed here and in Chapter 7, we contend that the multiple failures detected on the testbed were failures of design and as such a design-focused solution is needed. This holds true even-though there was a significant bespoke element to the testbed. This bespoke element relates to the packet capture setup and the selection of devices. However, as discussed in Subsection 2.2.2, the actual devices used are off-the-shelf consumer products selected to cover both the mix-and-match and starter kit approaches taken by consumers when buying home IoT devices. To address the design failings of these devices then, a design-lead approach is needed, for which systems engineering lays the groundwork.

The underpinnings of systems engineering can be found in systems theory. Although systems theory will not be presented as a topic here, it is important to note its position on system structure

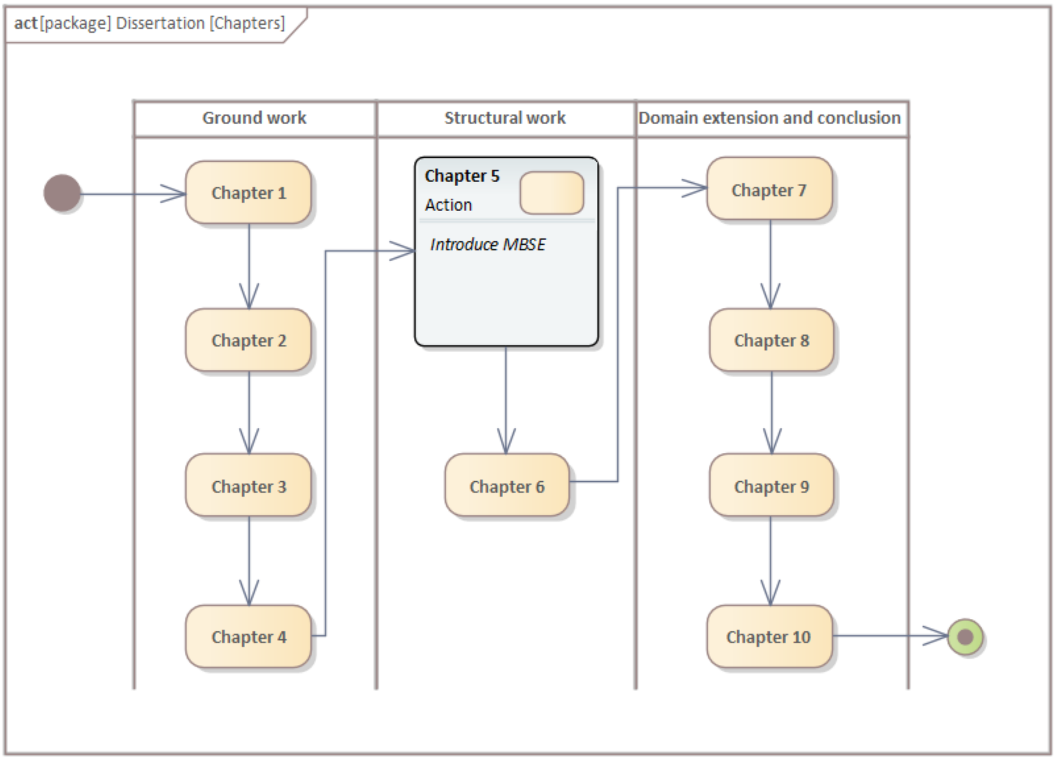



Figure 5.1: Chapter 5 research focus

and understanding. This is, the components of a system are best understood in relation to each other within the wider system context, as opposed to as individual unlinked entities. [76] This naturally implies that the nature of the system involved will also have an impact on how its constituent parts are understood. On this count Peter Checkland provides a popular systems classification with five categories namely; natural systems, designed physical systems, designed abstract systems[1], human activity systems, and transcendental systems[2] [168]. As Holt points out, not only can a system fit into more than one of these classifications at a time, but systems engineering can be used to address all five classifications.

One of the key drivers of systems engineering is the realisation that modern systems are increasingly complex and interlinked, which drives up the potential for misunderstanding and establishes the need for a single and accessible version of the "truth" to be maintained. Systems engineering deals with this need by addressing what Holt refers to as the three evils of systems engineering. These are:

- **Complexity:** undefined complexity going unchecked.
- **Communication:** a breakdown in communication or a lack of clarity.
- **Understanding:** all viewpoints[3] are not accounted for and assumptions are not checked.

[1] Such as hypothetical scenarios.
[2] Issues not known or beyond understanding.
[3] Please see discussion of viewpoints in Section 5.2.

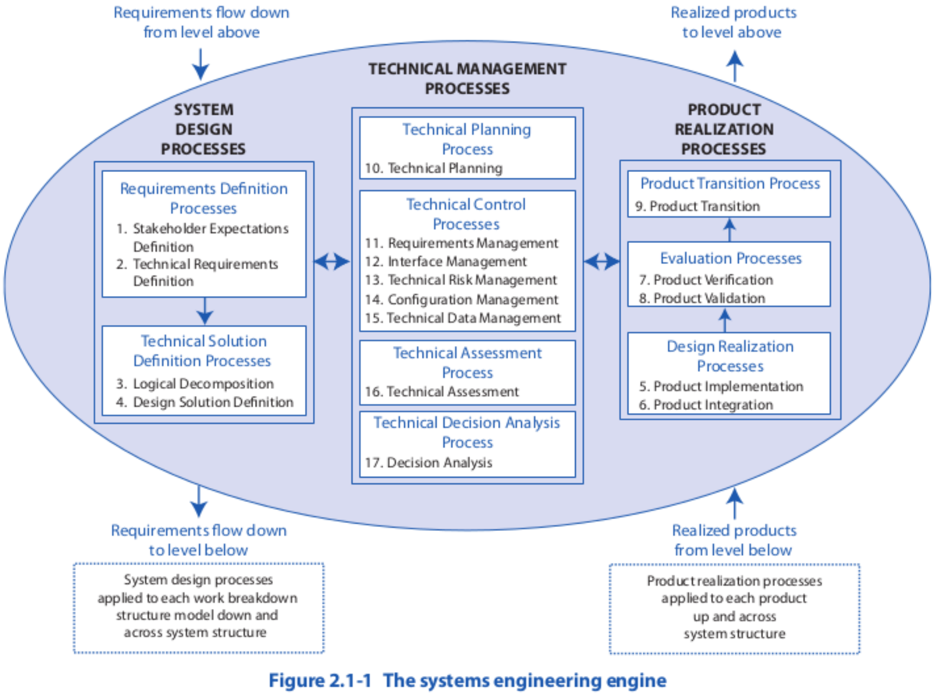



Figure 5.2: NASA's Systems Engineering Engine ([89])

One key way to explain the difference between a document-based approach and MBSE is in how they treat knowledge about a system. With the traditional approach all knowledge about the system is presented and stored in documents (paper or electronic), while MBSE presents all such knowledge as an abstraction in a system model [76]. A significant part of the advantage therein is the measure to which MBSE directly accommodates complex systems such as the IoT, referring back to our definitions of the IoT presented in Chapter 1. MBSE conceives of systems as a set of varied components which jointly achieve results not obtainable by any single one, where such components include people, devices, software, other systems and so forth [89]. This understanding of systems engineering is the basis of what NASA calls the Systems Engineering Engine, which is shown in Figure 5.2.

Due to its focus on the design of large scale system of systems and space flight systems, we will not be adopting NASA's approach here[4], though its general take on MBSE and specifically the use of the design phase as starting point, is of note. This is further explored in sections 5.4.3.1 and 5.4.3.2, but the key point is that the design phase is initiated by defining stakeholder expectations which are presented as requirements in the system model. This establishes stakeholder requirement definition not just as foundational to systems design [41], but also as the vehicle for meeting stakeholder needs [80].

From this stakeholder-based[5] point of departure, MBSE seeks to ensure that all needed system

[4] It extends significantly beyond what is shown here.

[5] With stakeholder understood in the broadest possible sense as all parties with an involvement with the system under consideration.

components are included and functioning currently [61]. This also allows for an MBSE model to not only be used in the design of a new system but also to describe an existing system. Accordingly, both design and lifecycle management can be conducted and system assessments can be carried out with reference to external measures [83]. Major strengths of MBSE is described by Ryan et al. as "*flexible architecture definition, program documentation, requirements traceability and system engineering reuse*". It is specifically these strengths which we wish to capitalise on in Chapter 7.

MBSE in general, and SysML specifically, is exceptionally good at the early identification of misconceptions, conflicting goals, and fundamental mistakes. Of course the early detection of such is ideal given that early detection allows for significant savings in both time and money.

One of the standout advantages of using MBSE, as introduced above, is the ability to deal with the entire system lifecycle. This is specifically due to the potential for requirement volatility over the system lifecycle to have negative procedural and financial impacts [117]. Any system which can address these issues, especially from a single system model, has clear utility. MBSE can deal with this issue since it adds rigour and precision to the engineering process, but also provides the systems engineer with the ability to rapidly change a model and easily gain insight into the system-wide implication of those changes [121]. These changes can then also be presented in preliminary designs (black and white box phases), or a fully fledged system (system solution) [107]. Lastly, there are certain issues which are dealt with preemptively simply by taking an MBSE approach, such as the persistent problem of natural language texts being understood differently by different individuals [61].

Taking all the above into consideration, one can define MBSE as an interdisciplinary object oriented design process which addresses system architecture, requirements, constraints, and other relationships, while replacing a document centric approach with traceable models, user viewpoints, and a reusable repository of design information [130].

The process of systems design and specification involves the determination of system requirements to meet the needs of stakeholders and the subsequent mapping of these requirements to system components [61]. Given the importance of stakeholders and their requirements to MBSE then, the following deeper discussion is needed. A final caveat in the discussion of MBSE is that we are only dealing with what is needed to facilitate the development and eventual testing of the DISCREET domain extension. The topic is invariably both deeper and wider if approached without these constraints.

## 5.2 Stakeholders

The notion of stakeholders in a system is well established [48] and is, at its most basic level, defined in the corporate context by Freeman as those who impact upon or are impacted upon by the operations of a company. From this early development, the inclusion of stakeholders in many fields has become standard practice, including those such as management science where this inclusion has often been framed in terms of potential financial benefit [56]. MBSE in general, and SysML specifically, takes an even broader view on stakeholders and includes all those with the possibility to require something from, or interact with, a system model.

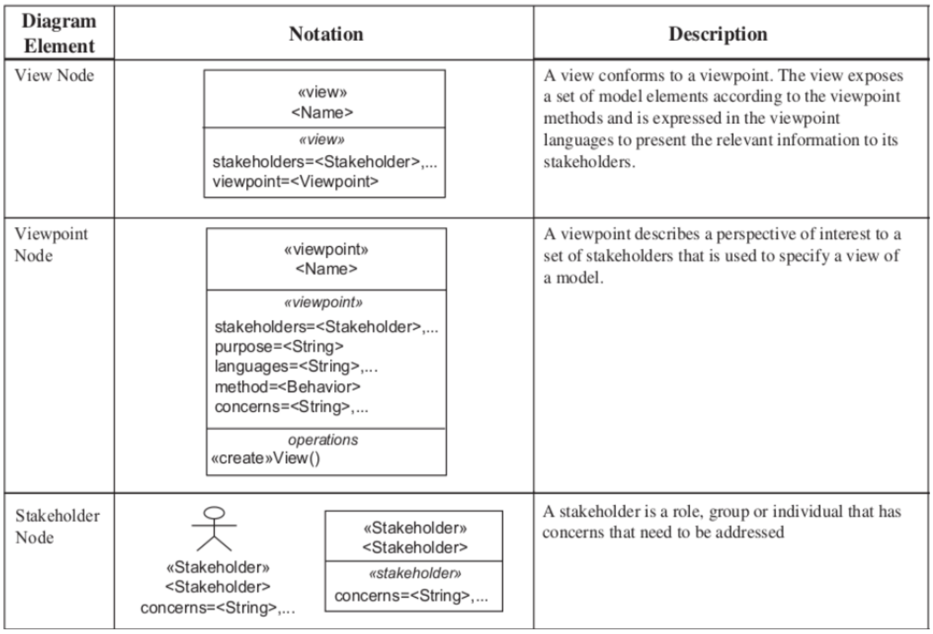
Looking specifically at SysML, stakeholders interacting[6] with a system model each spawns a related viewpoint package, which in turn produces a view thereby allowing for all such interactions to be modelled in a consistent manner [146]. The viewpoint for each stakeholder declares the language, outputs, and methods needed for addressing their requirements, while accessing a system model from that viewpoint provides the related view. That is to say that the viewpoint package describes the actions needed to meet the needs of a given stakeholder and also determines which elements of the model lifecycle relates to the stakeholder, the latter is the view [61]. The scope of what is included in a viewpoint therefore also varies depending on the specifics of the stakeholder.

The contents of a viewpoint package, taken from Friedenthal et al., are presented below. This is however not a mandatory layout and a real-world example may well only include a subset of these. The primary concern of the viewpoint here, that is the functionality that is not optional, is to establish which elements of an associated ontology are included, excluded, or optional [76], thereby establishing what is expected from the related view.

- **Stakeholders:** The one or more stakeholders using this viewpoint.
- **Concerns** Listing the issues this viewpoint addresses.
- **Purpose** Reasoning for generating the viewpoint, which may include how to address stakeholder concerns.
- **Language** The language to be used, such as SysML.
- **Method** Guides the creation of viewpoint artefacts.
- **Presentation** Includes constraints or directives relating to presenting generated artefacts.

Of specific concern for the work presented in Chapter 7 is the functioning of methods. These viewpoint elements are used to create viewpoint artefacts and spawn related views [41]. However, it should be noted that although the viewpoint and associated view exist within a given system model, the generated artefacts might be external, such as the generation of a compliance report. Such artefacts are still represented in the model and accessible, but are products of the model as opposed to functional elements of the language [146]. The system elements of view, viewpoint, and stakeholder, are all elements of the SysML language and present in the language specification. Taken from [61] this position is presented in Figure 5.3 while Figure 5.4 shows the use thereof.

[6]This implies that not just parties such as customers and regulators are stakeholders but also system designers, auditors and a range of other actors.



| Diagram Element | Notation | Description |
|---|---|---|
| View Node | «view» <Name> / *«view»* stakeholders=<Stakeholder>,... viewpoint=<Viewpoint> | A view conforms to a viewpoint. The view exposes a set of model elements according to the viewpoint methods and is expressed in the viewpoint languages to present the relevant information to its stakeholders. |
| Viewpoint Node | «viewpoint» <Name> / *«viewpoint»* stakeholders=<Stakeholder>,... purpose=<String> languages=<String>,... method=<Behavior> concerns=<String>,... / *operations* «create»View() | A viewpoint describes a perspective of interest to a set of stakeholders that is used to specify a view of a model. |
| Stakeholder Node | «Stakeholder» <Stakeholder> concerns=<String>,... ; «Stakeholder» <Stakeholder> / *«stakeholder»* concerns=<String>,... | A stakeholder is a role, group or individual that has concerns that need to be addressed |

Figure 5.3: View and viewpoint nodes

([61])

An interesting additional outflow of this approach to stakeholder management is that although stakeholders and their related requirements are integral to the the system of interest, they are actually external to that system. In other words, although the views and viewpoints relating to stakeholders are represented in SysML, this representation is only to provide structure in certain contexts and should not be viewed as these stakeholders actually forming part of the system being modelled [76]. The link from the stakeholders to the internal functioning of the system comes instead, by way of the requirements used to represent the needs and wants of the stakeholders. This is the topic of the next section.

## 5.3 Requirements

According to the NASA systems engineering handbook, the definition of stakeholder requirements is a foundational step for systems design and product realisation [89]. The act of definition is a two fold process including identifying stakeholders and determining their range of interactions with the system. This also includes the ability of some stakeholders, such as regulators, to establish system constraints. However, this is not a one off process and can be revisited throughout the system lifecycle which allows for stakeholder requirements to be clarified, incorporated and acted upon within the systems model on an ongoing basis. Doing so is only possible if a dynamic and system wide model, such as those created in SysML, is used [41].

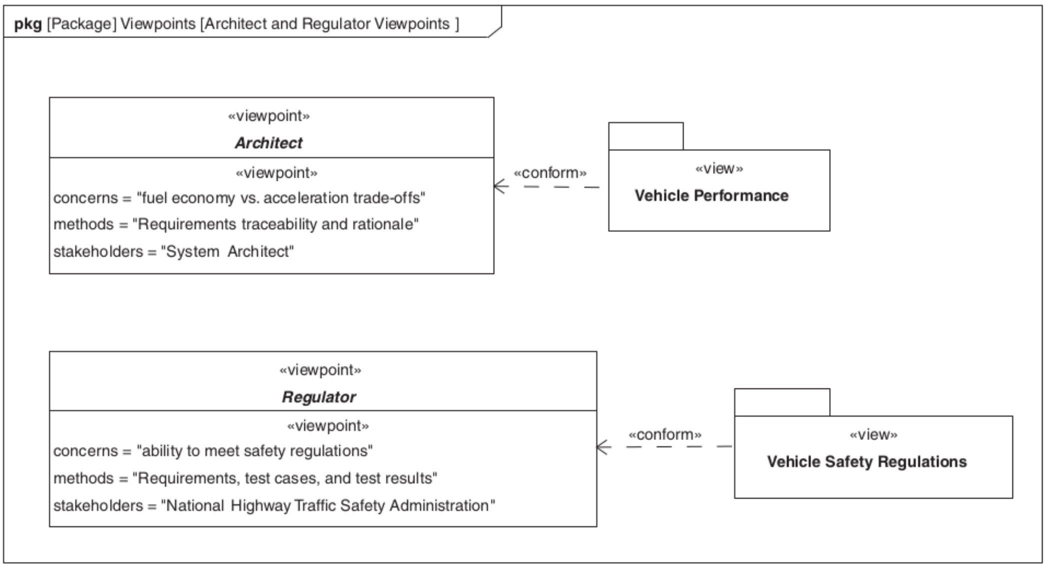



Figure 5.4: Viewpoints and domain
([61])

### 5.3.1 Requirement elicitation

Broadly speaking, the DISCREET domain extension has two main areas of interest, that dealing with auditable compliance, and that dealing with privacy by design (without reference to specific legislation). For the former, the requirements and associated stakeholders needs, are predetermined. This comes in the form of GDPR requirements and CNIL audit requirements. For the second, we use the LINDDUN methodology to guide the process of determining privacy threats and formulating remedies. However, this is not typical but rather a consequence of the specific aims of this domain extension where systems engineering in practice, as is to be expected, expends significant effort in eliciting requirements in service of stakeholder needs. INCOSE, the International Council on Systems Engineering, holds that the goal of stakeholder needs and requirement definition is to deliver the capabilities needed by said stakeholders within the context at hand[7] [80]. Moving from elicitation to the definition of stakeholder needs and requirements is then described as follows:

- **Preparing for needs and requirement definition:** Determine the stakeholders and/or classes of stakeholders involved, taking a cross-lifecycle view to incorporating said needs and requirements into a system model. This extends to determining the need for enabling systems[8]
- **Definition of stakeholder needs:** Elicitation, prioritisation, and specification of stakeholder needs.

[7] This is in compliance with ISO 15288 which addressed systems and software engineering with a view to system life cycle processes.

[8] As distinct from the system of interest, which is the primary system the model describes. This is also discussed in Section 8.2.

- **Operational concept development:** Determine the various capabilities and behaviours across the system lifecycle. This has a direct impact on the future validation of the system model.
- **Develop stakeholder requirements from the elicited needs:** Determine the applicable system constraints. From there, list stakeholder requirements as they pertain to core system functionality and under various states of operation.
- **Analyse stakeholder requirements:** Definition of validation criteria for stakeholder requirements, including measures of effectiveness (MOEs) and measures of sustainability (MOSs), as applicable.
- **Management of stakeholder needs and requirement definition:** Confirm with stakeholders that requirements are correctly expressed, then record, manage, and revisit these requirements over the system lifecycle as needed. This also directly speaks to traceability throughout the system by requirements being met, inherited, and linked back to stakeholder viewpoints.

Dealing with the same topic Holt provides the graphical representation in Figure 5.5.

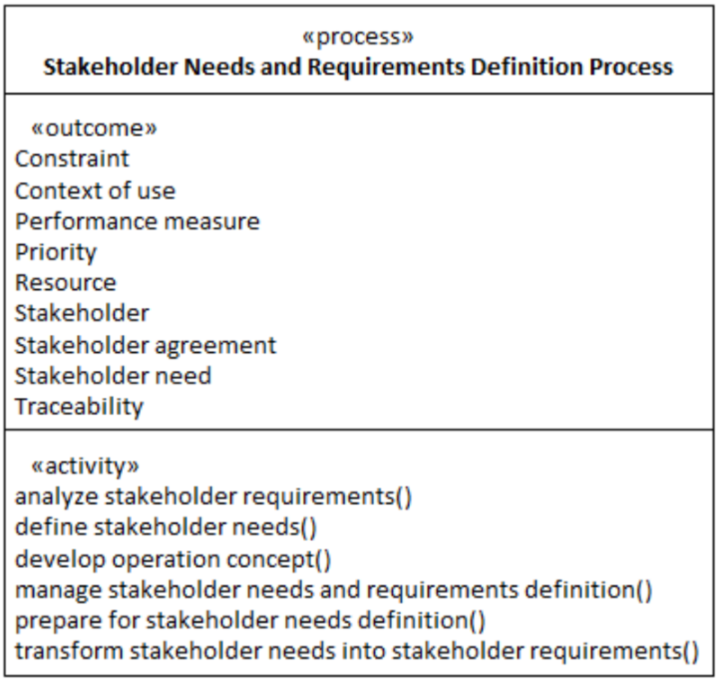



Figure 5.5: The ISO 15288 process context view for stakeholder needs and requirements definition

([76])

DISCREET does not replicate this process given the nature of the requirements it includes. Instead it is formulated in a manner cognisant of this process and its outcomes, so as to seamlessly integrate with other stakeholders and their requirements once implemented in a full system model. The application thereof is of course up to the system modeller. The one caveat here of course is that the GDPR requirements are legal requirements and take precedence over other considerations such as general stakeholder needs.

Stakeholder needs can be split into smaller actionable compartments formalised as individual requirements. Each requirement has a name, an identifier, and written specification. Such stakeholder needs can be presented in a requirements table or requirements diagram, wherein hierarchical relationships between requirements can also be displayed [107].

## 5.4 The choice for SysML

### 5.4.1 Introducing SysML

As an extension to a subset of UML 2.0, the systems modelling language (SysML) focuses on systems engineering across the system lifecycle. This is accomplished through a combination of changes to inherited UML structures and the addition of new diagrams and capabilities [146] as presented in Figure 5.6

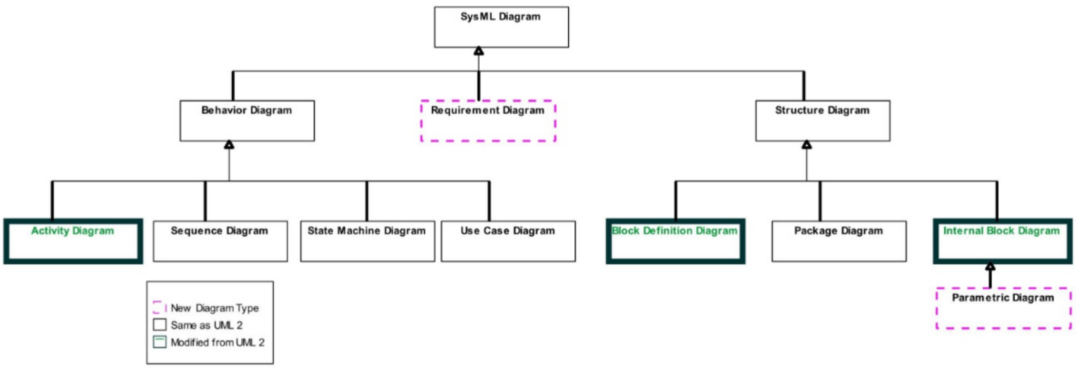



Figure 5.6: SysML's extension of UML
([146])

The first departure form UML is the replacement of classes with blocks, which can represent both the structure and behaviour of a system in a modular and divisible manner [146]. Further aiding SysML's utility is the incorporation of parametric and requirements diagrams, and features such as inheritance[9] [61]. These system elements of SysML can be divided into four groups, commonly referred to as the four pillars of SysML. They are structure, behaviour, requirements, and parametrics [61] and are presented in Table 5.1. Thinking of MBSE as an activity then, SysML provides the lexicon needed to conduct that activity [41], without prescribing the exact steps to follow[10]. As such, SysML can be described as a primary enabler for MBSE in industry [107].

[9]Lower order systems can inherit traits for higher order ones.
[10]This is explained in Section 5.4.2.

Our new domain extension is intended for use with any suitable modelling language, using any language compliant tool. However, a choice of language and tool had to be made for expressing the work. Picking a standard from the *Object Modeling Group* (OMG), such as the *Unified Modeling Language* (UML) or the *Systems Modeling Language* (SysML) would seem the obvious choice and given the focus here on systems design, the choice was made for SysML. Although a full investigation of the history and functioning and future of SysML is well outside of the scope of the work presented here. However, a couple of key points should be made to support the choice for SysML, not only on historical and practical grounds, but also looking towards the future.

The Object Management Group (OMG), which owns SysML, UML, and a host of other languages, standards, and frameworks, notes that the development of SysML can be traced to a decision to adapt the Unified Modelling Language (UML) for systems engineering, which occurred at an INCOSE workshop in 2001 [145]. Also specifically noted by the OMG is that SysML is intended to enhance or enable the specification, analysis, design, and verification and validation of a broad range of complex systems. These may include hardware, software, information, processes, personnel, and facilities.

A further point to support our chosen path is that a focus on privacy engineering as opposed to higher level strategic issues will substantially enhance the direct utility of our work [96]. At the same time though, using SysML means that such higher level issues are not left by the wayside but are incorporated into the system model, typically by way of requirements[11]. Ultimately, it is SysML's core competency as a language for modelling systems, which puts it in a unique position to address the challenges faced by systems engineers working with the IoT in general and consumer IoT specifically. This includes the difficulty in specifying privacy requirements for a system that is not only dynamic and evolving in its deployment but is truly heterogeneous with regards to components, standards, and functional ability [6].

### 5.4.2 The lack of a guiding methodology in SysML

SysML gives the systems engineer the ability to describe, alter, and mange a system over its entire lifecycle but does not dictate the exact procedure to follow. This provides a significant measure of freedom but also then requires an additional input in the form of a guiding methodology or method to make SysML actionable. Within MBSE the term method has a specific function which Friedenthal et al. describes as: "*a set of related activities, techniques, conventions, representations, and artefacts that implement one or more processes and is generally supported by a set of tools*" This can be contrasted with the position for SysML itself which Morkevicius et al. explains as follows:

[11] This is not reflected in the domain extension but will form part of any model which incorporates the domain extension.

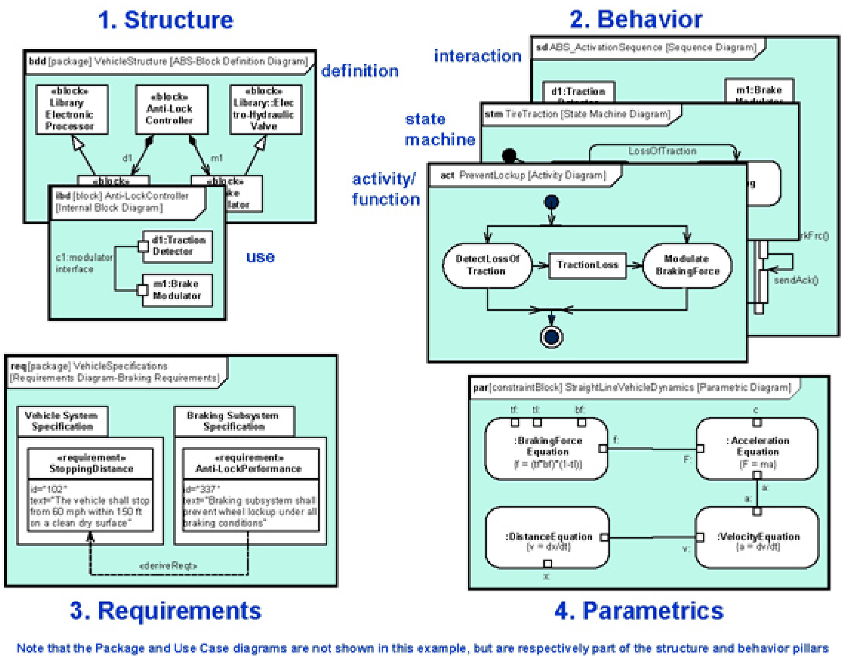



Table 5.1: The four pillars of SysML

> *SysML is neither a framework nor a method: it provides no information about the modelling process and thus must be combined with some methodology to become truly applicable.*

There have been a number of high profile initiatives to develop such guidance for SysML, though none has gained outright prominence, which is in part due to many of these initiatives being industry specific. This presents a clear opportunity to learn from the *status quo*, aligning with the best suited options[12], and building accordingly. We do this by directly integrating the MBSE Grid Framework, but to do so, some of the alternatives must first be introduced. OOSEM and SYSMOD are two well known methods applicable to SysML [41] and could have been considered for our endeavour.

The Object-oriented Systems Engineering Method (OOSEM) is a scenario-based method that approaches the full spectrum of systems design activities top down using SysML [61]. A key strength for this method is that its integration of an object-oriented approach with modelling and more traditional systems engineering concepts, allows for the creation of more flexible architecture which can more easily deal with significant changes over the project lifecycle [80]. OOSEM also specifically targets ease of use with object-oriented software design [61].

The second is the Systems Modelling Toolbox (SYSMOD) which, in keeping with its name, presents the system engineer with a "*toolbox of methods*" [158]. SYSMOD further divides

[12] As described here there is a clear winner for our purposes meaning that developing our own alternative would be a counter productive activity.

projects into the following main phases: project context description, requirement collection, system context modelling, structure and state modelling, and the collection of domain knowledge [107].

Considering the above, as well as other methods and methodologies, Morkevicius et al. conclude that these systems generally have issues with one or more of sequential artefact generation/collection, dealing with information complexity, dealing with different levels of abstraction, or the iterative collection of data. These issues can lead to errors in a model and difficulty in assigning responsibilities to various stakeholders. The authors further acknowledge that formal architectural frameworks such as DoDAF[13], MODAF[14], TOGAF[15], and Zachman[16] can be used to alleviate some of these concerns, especially around differing layers of abstraction. This also implies that a systems engineer might be forced to use an architectural framework even for more lightweight models [106]. However, these frameworks all consist of multiple *views*, without any simplified route to addressing view subsets. This is a significant concern given that SysML not only includes the "usual suspects" as stakeholders but also all users of the system model. For each stakeholder there is an associated *viewpoint*, which sets out all the specifics needed to generate a *view* of the model specific to that stakeholder [61]. Given that the DISCREET domain extension specifically deals with subset[17] *views*, this point is also of central concern to us. To address these concerns Morkevicius et al. proposes a more streamlined and general[18] framework for conducting the business of systems modelling. This is the MBSE Grid Framework.

### 5.4.3 The MBSE Grid framework

The MBSE Grid Framework (MGF) is based on the four pillars of SysML discussed previously and viewpoints taken from DoDAF and other sources [106]. The four pillars are presented as columns and the viewpoints as rows, thereby forming a grid structure. The rationale for this structure is that the four pillars describe the primary areas of the system model while the rows address different levels of abstraction. The first of these, in line with best practice [134], describes the problem space and thereafter the second develops the solution there to. This was however not sufficient to fully attend to the problem space and as a result the top row is subdivided into black box and white box rows [107]. The black/white box approach is also a standard concept within systems engineering with the black box representing an external view of the system as a whole, while the white box deals with the system internals [80]. This also implies that the interlinking between the black box and white box is key to effectively modelling the system.

In the MGF the black box explicitly includes the whole System of Interest (SoI) including stakeholder needs, functional expectations, and Measures of Effectiveness (MoE). The white box on

[13] Department of Defense Architecture Framework

[14] Ministry of Defense Architecture Framework

[15] The Open Group Architecture Framework

[16] An early and well regarded enterprise wide architecture.

[17] In dealing with compliance and privacy, the issues addressed will inherently be subsets. Such as the subset of all compliance auditors who are GDPR compliance auditors.

[18] General in terms of broad applicability as opposed to being limited to a specified use.

the other hand addresses subsystem functionality, such as the SoI's subsystem input and outputs and also presents system requirement specifications as derived from stakeholder needs. This interaction between black and white box is not only central to the MGF but also one of the primary reasons for it being incorporated into the DISCREET methodology presented in Chapter 7. In Table 5.2 the MGF is presented with each cell as a view of the system model. Here the authors specifically reference the ISO definition of a view which describes it as a "*work product expressing the architecture of a system from the perspective of specific system concerns*" [107], which is of course consistent with the SysML position of *viewpoints* spawning *views* into a related part of the system model [61]. A definition of each *view* is presented below.

| | | | Pillar | | | |
|---|---|---|---|---|---|---|
| Layer of Abstraction | | | Requirements | Behavior | Structure | Parametrics |
| | Problem | Black Box | Stakeholder Needs | Use Cases | System Context | Measurements of Effectiveness |
| | | White Box | System Requirements | Functional Analysis | Logical Subsystems Communication | MoEs for Subsystems |
| | Solution | | Component Requirements | Component Behavior | Component Assembly | Component Parameters |

Table 5.2: Elements of the MBSE Grid Framework [107]

- Stakeholder needs: This includes a broad stakeholder base and extends to users, regulators, policies, internal guidance, etc. Measures of elicitation include questionnaires, focus groups, and existing documentation, with no need for input in any set format since the use of SysML will attend to such.

- Use cases: Refined functional stakeholder needs are captured via SysML use case diagrams which include actions, constraints and prerequisites.

- System context: Interaction between the environment and the SoI is captured here, including the interfaces needed to facilitate that interaction. As a result, SysML internal block diagrams (ibd) are used in this *view*.

- Measurements of effectiveness: Non-functional system goals or requirements are captured via MoEs and displayed on a SysML block definition diagram (bdd). The corresponding calculations however, are presented on parametric diagrams.

- System requirements: Requirements derived from stakeholder needs are captured via SysML requirements diagrams.

- Functional analysis: Deeper functional use case analysis dealing with internal systems and represented via activity diagrams.
- Logical subsystems communication: Control and resource flows derived from functional analysis is used to map logical subsystems. Both bdd's and ibd's are jointly used to capture this view.
- MoEs for subsystems: Measures of performance and MoEs identified for each logical subsystem using bdd's and parametric diagrams.
- Component requirements: Formal design constraints gathered from system requirements and presented in requirements diagrams.
- Component behaviour: Precise behaviour for components is presented in terms of states and actions and consequently uses SysML state machines, activity diagrams and sequence diagrams.
- Component structure: Logical subsystems created in the problem *viewpoint* are implemented via physical components and interfaces and presented on bdd's and ibd's.
- Component parameters: The characteristics of components and how these perform against the MoE's and measures of performance defined in the problem space, are presented on bbd's and parametric diagrams.

Transposing the specific SysML elements onto the grid provides each *view* with a set of diagrams through which they can be presented to observers and also serves to enforce the functional boundaries of each. This is presented in Table 5.3. What is less obvious from the table though, is that this grid represents all the elements of the framework, but not the elements that a systems engineer must use in a set implementation. The latter is a subset of the former, with the specific elements to use depending on the model being generated. This does also not exclude the use of any other methods, tools, or techniques which might be appropriate. Accordingly, the addition of other methods, requirement repositories, or tools, is not excluded but in fact enabled by the MGF, as long as said additions are valid for MBSE application. On a side note, not too much should be read into the presence of the block definition diagram (bdd) in multiple views. The bdd is the *de facto* blueprint for other diagrams and also plays a foundational role in establishing any model.

A key strength of SysML is its treatment of traceability, which is spread over four specifications. These are direct relationships[19], subject/owner relationships, composition[20], and derived computation [107]. In Figure 5.7 the MBSE Grid Framework's authors present trace relationships between *view* specifications. Although this presentation of traceability might seem a jumble, especially to those new to MBSE, it is important to remember that this complexity is significantly reduced through the use of an appropriate modelling tool.

[19]Satisfy, allocation, derive, refine.
[20]Not only part properties but also activity call action behaviours.

| | | Pillar | | | |
|---|---|---|---|---|---|
| Layer of Abstraction | | Requirements | Behavior | Structure | Parametrics |
| Problem | Black box | **Stakeholder Needs:**<br>• Requirements diagram<br>• Requirements table | **Use Cases:**<br>1. Use Case diagram<br>2. Activity diagram | **System Context:**<br>• Internal block diagram | **Measurements of Effectiveness:**<br>• Block definition diagram |
| Problem | White box | **System Requirements:**<br>• Requirements diagram<br>• Requirements table | **Functional Analysis:**<br>• Activity diagram | **Logical Subsystems Communication:**<br>1. Block definition diagram<br>2. Internal block diagram | **MoEs for Subsystems:**<br>• Block definition diagram |
| Solution | | **Component Requirements:**<br>• Requirements diagram<br>• Requirements table | **Component Behavior:**<br>• State machine diagram<br>• Activity diagram<br>• Sequence diagram | **Component Structure:**<br>1. Block definition diagram<br>2. Internal block diagram | **Component Parameters:**<br>• Parametric diagram |

Table 5.3: The MBSE Grid Framwork mapped to SysML [107]

To implement the MGF then, one would work from the top down and going left to right, establishing the problem space, determining the internal workings thereof and finally, developing the solution based on the above. This is the approach employed within DISCREET, with the addition of the new trace methods and system artefacts developed in Chapter 7. In short, the newly developed components deal with what needs to be done and how to do it, while MGF is incorporated to address issues of "when and in which order" to perform these actions. This is the same challenge MGF solves for MBSE as a whole.

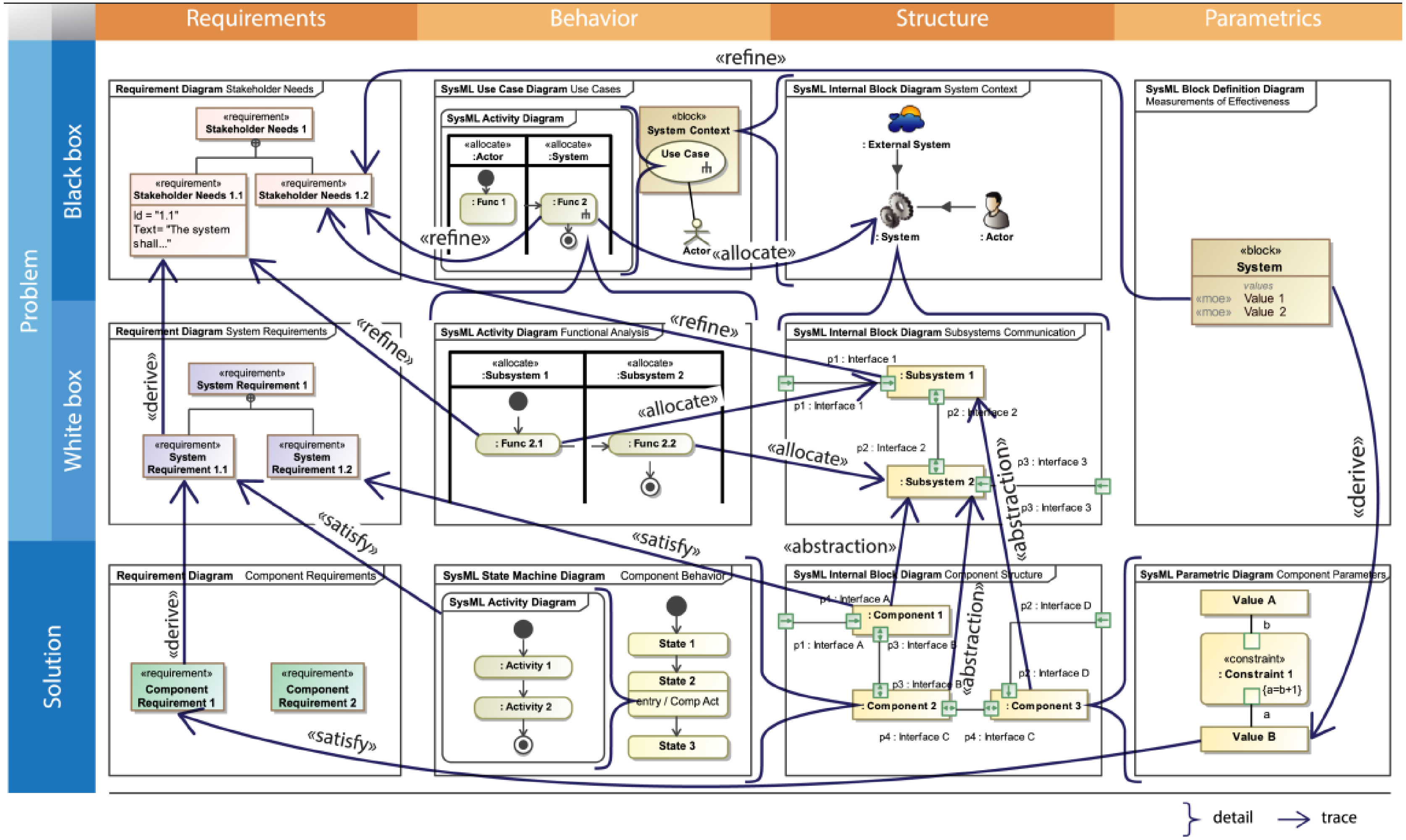



Figure 5.7: Traceability in the MBSE Grid Framework [107]

#### 5.4.3.1 MGF black box

To provide some greater clarity on the application of the MBSE Grid Framework, this and the following two subsubsections will follow the discussion Morkevicius et al. provides on implementation. Here, as in the MGF itself, we start with the problem domain described via a black box approach, then move on to the problem domain described via a white box approach and finally, the development of a solution.

Starting with the top left square of the MGF and working to the right, we start by presenting stakeholder needs as top level SysML system requirements. This includes a mission needs statement which outlines the broader project including the need for compliance and privacy requirements. In terms of DISCREET, this already acts as the linchpin for drawing in all other methods and generating the needed system artefacts. These stakeholder needs are then decomposed into actionable requirements[21]. The system model is populated with a full set of all requirements including their interrelationship, while a requirements diagram can be used to present those requirements relating to the system *view* that spawned the diagram[22].

These requirements are subsequently reflected throughout the black box phase as they describe system use and related actions, giving rise to use case diagrams and activity diagrams. As models of any size will invariably have multiple use cases and associated activities, the MGF groups these by context. This is also expressed further in activity diagrams through the use of swimlanes while the link back to actionable requirements is established via *«refine»* relationships.

The system of interest (SoI), that is the main system dealt with in specific model[23], can be presented on an internal block diagram (ibd) showing not only the specifics of the SoI but also how it interacts with its environment. Such an ibd-based view can be spawned for each context within which the SoI might be situated. Per example, an all weather camera (SoI), might be used on a clear day on a snowy mountain top (cold but high light levels), or a misty early morning in the countryside (low light levels and high moisture). The SoI can then be presented on ibd's for each of these contexts. This also directly leads to the formulation of measures of effectiveness (MoEs)[24], which can then be presented on bdd's.

#### 5.4.3.2 MGF white box

Moving to the next row down on the MGF, as shown on Table 5.3, we define the problem space taking a white box approach. This follows directly from the work of SoI definition conducted in during the black box phase, starting back with requirements. Here, system requirements are connected to view specifications while the relationship between row one (black box) relationships and row two relationships is one of derivation, indicated on the model as *«deriveReqt»*.

[21] Individual "shall statements".

[22] A *view* relating to a vehicle's propulsion might show requirements on engine capacity but not requirements relating to cabin fit and finish.

[23] System models can, and often must, include broader elements located outside of the SoI.

[24] Such as the camera must be able to handle exposure values (EV) ranging from very low to very high.

This functional analysis in the MGF is further standardised by set rules starting with the use of swimlanes in activity diagrams. The reasoning behind the swimlane-based approach is its simple utility. Since the activity diagram is generated in accordance with a requirement, and swimlanes partition the activity diagram into logical contexts, this automatically addresses the creation of logical subsystems. The rules flowing from this simple principle are as follows:

1. From the black box row's activity diagrams, the swimlanes will spawn the logical subsystems to include in the white box phase.
2. There must be a SysML activity diagram for every function of the SoI.
3. Each function included in such a diagram can be refined in turn, by an additional activity diagram.
4. Each such function can also have a *«refine»* relationship with functional system requirements.
5. Connections and interfaces between logical subsystems must be defined.
6. A SysML bdd must be created for the SoI and each logical subsystem.
7. A SysML ibd must be created for the SoI capturing the interfaces between logical subsystems.
8. Define the MoEs for subsystems using bdd's, including evaluation methods, with the latter presented on constraint blocks.
9. Specify the link between MoEs and system requirements using a *«refine»* relationship.

Lastly, for the problem space as a whole, it should be noted that the primary activity is one of dealing with requirements and what they entail. As a result of this, horizontal relationships between requirements and other system elements are represented using a *«refine»* relationship.

#### 5.4.3.3 MGF solution

With the problem defined in terms of both black and white box specifications, we can move on to developing one or more solutions. The solutions thus developed are presented over four component levels, these are requirements, behaviour, structure and parameters. Although the word component is used here, it should be understood as any part of the systems, but including all the parts, as the solution addresses the entirety of the architecture.

Starting again with requirements, these are narrowed down and directly refer to physical components and their characteristics. From there component behaviour can be modelled, with the logical subsystems of the white box phase being replaced by physical components connected to one another by way of physical connections. Components and interfaces can also be grouped by type, such as electrical, mechanical, software, etc. As the behaviour for each component must also be defined, state machines, activity diagrams and sequence diagrams are used as appropriate. The structure for the related physical components can then be presented using both bdd's and ibd's. Finally, parametric models are developed to reflect MoEs in accordance with the physical properties of the components. Behavioural simulations and trade-studies are also typically run at this level, to decide between competing designs.

### 5.4.4 Incorporation into DISCREET

As the user moves down the rows of the MGF, each lower row is more precise and more ingrained into the specific model than the preceding one. This holds for the entire system model being developed and therefore also DISCREET if it is used. To align each part of DISCREET to the MGF a mapping based on each DISCREET method is used. This mapping is presented by way of activity diagrams and introduced in Chapter 7. Before that can be done though, the set of resources to include in DISCREET must be determined. This is the project of the next chapter.

# Chapter 6

# Systematic Overview

This chapter not only analyses what was previously presented, but also determines the project scope for the chapters to follow. This is done by taking a standard interdisciplinary approach, which involves two distinct steps. First one compares divergent fields to determine areas of overlap or agreement, which form the foundation for new work. Second, one determines open areas which are in need of development and can be built from the previously determined foundation[127]. This chapter revisits the work introduced in Chapter 3, draws in the elements which will serve as foundation for the domain extension which presents the new and interdisciplinary outcome, as positioned in Figure 6.1.

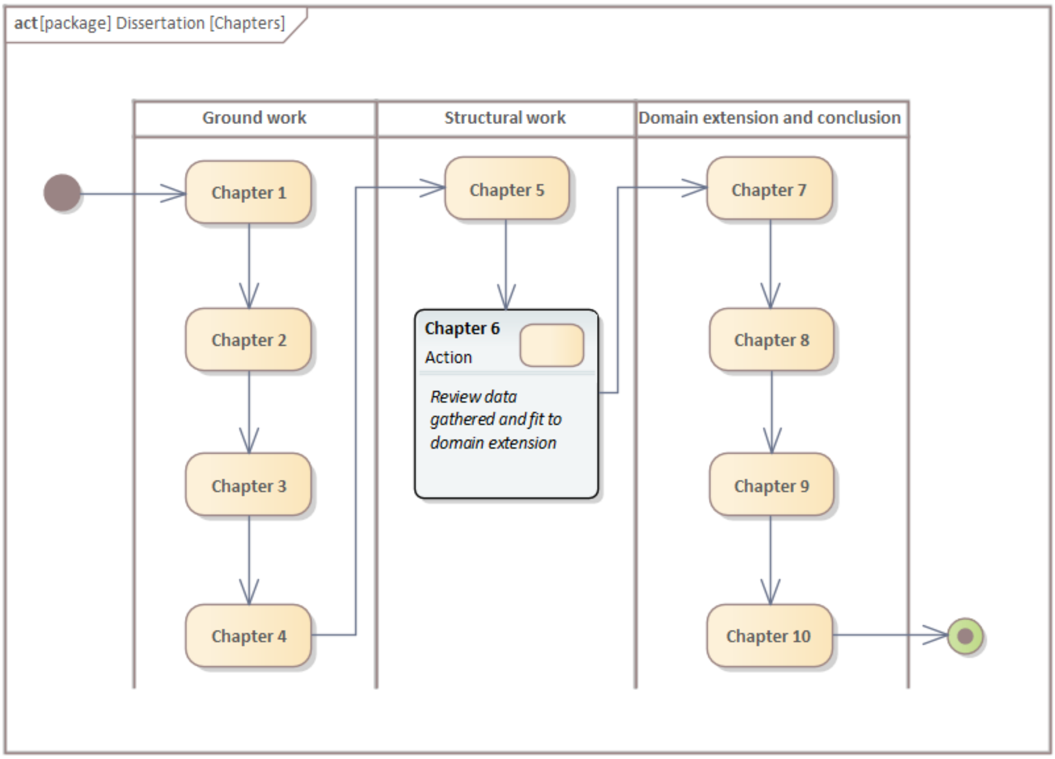



Figure 6.1: Chapter 6 research focus

## 6.1 Lifecycle failures and PbD in review

As mentioned in Chapter 4, we reasonably expected to gain useful information from running our test bed, but were surprised by the amount of data gathered and both the extent and egregiousness of the failures found. This not only included devices, services, and documentation coming up short, but also an utter failure to address these issues over the device or system lifecycle. As the testbed was rerun over several years, none of the failings we spotted were addressed in any meaningful manner. Instead we saw cosmetic changes or even an outright worsening of the situation. Two prime examples are the Misfit Shine device and Withings terms and conditions. Misfit have changed their product strategy from traditional manufacturing and long term device support to a “drop” based approach. In the latter a lot of marketing hype is generated for a new device which will “drop” with limited availability. This enhances the device’s status due to limited availability and makes planning on the manufacturing side much less speculative. In this change, the company has removed all previous devices from its website and streamlined its terms and conditions, managing to make it even less useful than before. Withings on the other hand have updated their privacy policies several times and to their credit have now produced a more accessible document which also includes direct links to older versions although this only goes back to June 2017. On the plus side though, they have now reintroduced Nokia as responsible party.

In other words, the changes over time were either neutral, negative, or targeted at minimal compliance. When it comes to device functionality then the status quo has been maintained, which is of course to be expected since we identified the primary challenge as one of design, hence the absence of PbD being evident in the testbed devices. Positioning PbD at the core of the design process implies that privacy measures form part of the core system functionality. This also implies that users are less likely to circumvent privacy measures for the sake of expedience since they are now tied into usages goals [24]. Conversely, if PbD was not a focus area during the early stages of the product lifecycle, a higher level of performance against privacy metrics should not be expected later on.

## 6.2 Domain extensions in review

Since one of the primary goals of systems engineering is to establish a single version of the truth, it implies the need for language with agreed formalism. Such a language can then be further advanced by adding in new capabilities targeted at the needs and functioning of a specific domain. In our case we have opted to take a systems engineering approach to these needs, specifically looking at MBSE[1] for the needed capabilities. Specifically, we chose SysML as the general language of operation and within that we intend to develop a domain extension to address needs relating to compliance and privacy by design for consumer IoT.

[1] Although we will not spend more time on systems engineering specifically, please note that MBSE is a form of systems engineering and not a subset thereof. Accordingly, MBSE can conduct all systems engineering activities.

As shown in Chapter 3, there are a large number of other systems, methodologies, frameworks and even a SysML domain extension that touch on some of the issues we have identified. However, none of these fulfil all the needs identified and the one domain extension included radically departs from the approach taken herein. Although this opens up some new research options, discussed in Chapter 10, it also shows the need for a novel domain extension.

## 6.3 A taxonomy of IoT privacy threats

In order to develop a new taxonomy three foundational components are needed. These are 1) an IoT system model 2) a model of IoT functionality, and 3) a practicable privacy threat model. The system model draws in all the relevant IoT systems, technologies, and components while the second model describes the functionality and interaction of these elements. Lastly, the privacy threat model draws in the tools needed to analyse the threats occurring within the context described by the first two components of the taxonomy.

The work of Solove and Ziegeldorf et al. would seem a clear fit for this project. However, as shown in Chapter 3, their work does not score high enough with regards to the eight analysis metrics used and especially the Solove taxonomy is ill suited to direct application in the IoT. Given its propensity to take a dim view of most consumer IoT services, a more nuanced tool is needed to locate threats and go further by suggesting applicable remedies. Given our previous discussion and its existing general use LINDDUN proves the ideal candidate for our third element. For the second, that is a model of functionality in the IoT, we turn to the work of Al-Fuqaha et al., while the IoT systems model is that introduced in Subsection 3.5.1. Lastly, although the work of Ziegeldorf et al. is not incorporated here, the manner on which it was mapped to that of Al-Fuqaha et al. will be, but over two stages. First we will map LINDDUN to the functionality model developed by Al-Fuqaha et al. and then both will be mapped to the IoT system model to provide the full taxonomy. Doing so provides clarity on the construction of the final taxonomy but also allows for users to take a functionality only approach, which could be useful during early design and planning.

In Table 6.2 we present the mapping between Al-Fuqaha et al. and LINDDUN. Key takeaways from this are that 1) there are multiple possible privacy threats at each level of functionality, 2) the threat profile at each level varies with semantics yet again being worst off, and 3) the ease with which this can be brought to bear on the design process. Although these functional elements were introduced in Chapter 1, it is useful to briefly mention some examples here as this aids in understanding the mapping to follow. In Table 6.1 Al-Fuqaha et al. provides the functional elements with examples of the hardware, standards, and software found at each.

Moving on from the functionality only mapping in Table 6.2, Table 6.3 presents the full taxonomy by mapping the elements of IoT functionality to the three main IoT layers, which in turn brings in the LINDDUN threat categories. The taxonomy has three columns which, from left to right, are the three IoT layers, the six IoT functional elements, and the seven threat categories. The

| IoT Elements | | Samples |
|---|---|---|
| Identification | Naming | EPC, uCode |
| | Addressing | IPv4, IPv6 |
| Sensing | | Smart Sensors, Wearable sensing devices, Embedded sensors, Actuators, RFID tag |
| Communication | | RFID, NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-Wave, WiFi, WiFiDirect, , LTE-A |
| Computation | Hardware | SmartThings, Arduino, Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, BeagleBone, Cubieboard, Smart Phones |
| | Software | OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, Hadoop, etc.) |
| Service | | Identity-related (shipping), Information Aggregation (smart grid), Collaborative-Aware (smart home), Ubiquitous (smart city) |
| Semantic | | RDF, OWL, EXI |

Table 6.1: Elements of IoT functionality and examples [2]

taxonomy can be used left to right or right to left as needed. This allows users to determine the likely areas of threat for a proposed IoT project, localise a single component within a larger project, or to trace a privacy failure back to impacted systems.

## 6.4 The CNIL's IoT PIA in review

Thanks to its focus on GDPR compliance, with extension to the IoT, the CNIL's privacy impact assessment makes for a great inclusion herein. This is further strengthened by the ready made model artefacts in the form of reporting templates and freedom of use which allows for the party under audit to use the PIA as preparation for future audit. We take this significantly further by adding the PIA directives as system requirements within a SysML model, linking them to GDPR requirements and tracking action against these. Our DISCREET methodology, which incorporates use of the CNIL's PIA also provides a method to guide the conduct of the above, however, the PIA should be read with the IoT extension and then applied in Chapter 7.

## 6.5 LINDDUN in review

As far as legal compliance and related auditing are concerned, there is a measure of a natural fit with the work conducted herein, in as much as it is given that not all legal provisions are applicable in all situations. It is therefore standard practice to determine what applies and how

| IoT Functionality | Threat Categories |
|---|---|
| Identification | Detectability |
| Sensing | Identifiability |
| | Detectability |
| | Non-compliance |
| Communication | Linkability |
| | Detectability |
| Computation | Linkability |
| | Non-repudiation |
| | Unawareness |
| | Non-compliance |
| Services | Non-repudiation |
| | information Disclosure |
| | Unawareness |
| | Non-compliance |
| Semantics | Linkability |
| | Identifiability |
| | Non-repudiation |
| | information Disclosure |
| | Unawareness |
| | Non-compliance |

Table 6.2: Mapping privacy threat categories to IoT functionality

to apply it, which is done in this case via the domain extension introduced in the next chapter. For a standalone methodology such as LINDDUN which includes the steps needed to determine which provisions are applicable, the same logic does not hold and as such, LINDDUN is incorporated in its entirety, with the application of LINDDUN structured as a method within DISCREET. Furthermore, there is nothing specific to the nature and structure of LINDDUN that would exclude its use in this manner, while the originators of the methodology themselves have also contemplated domain specific applications [167].

LINDDUN was not only chosen for its ability to dynamically address a wide range of privacy threats, but also because it is actively maintained. As such, users are advised to regularly check the official LINDDUN site[2] for new releases and updates. A further equally significant reason for the inclusion is that LINDDUN directly addresses a by design approach to privacy engineering, but explicitly caters to use across the system lifecycle, this makes its operations fundamentally congruent with the use of SysML proposed herein. Furthermore, when dealing with the issue of domain specific application, LINDDUN is described as a "*generic privacy threat modelling*

[2] https://www.linddun.org/downloads At the time of writing, the LINDDUN threat tree catalogue is at V2.0.

| **IoT Layers** | **Functional Elements** | **Threat Categories** |
|---|---|---|
| Visualisation Layer | Services | Non-repudiation |
| | | information Disclosure |
| | | Unawareness |
| | | Non-compliance |
| Middleware Layer | Identification | Linkability |
| | Sensing | Identifiability |
| | Communication | Non-repudiation |
| | Computation | Detectability |
| | Services | information Disclosure |
| | Semantics | Unawareness |
| | | Non-compliance |
| Objects Layer | Identification | Linkability |
| | Sensing | Identifiability |
| | Communication | Detectability |
| | | Non-compliance |

Table 6.3: IoT privacy threats taxonomy

*technique that does not include any domain-specific knowledge*" but which "*can however be easily extended with domain- or application specific knowledge when required*" [166]. This yet again underlines its applicability to the work developed in Chapter 7.

It is also noteworthy that the utility of LINDDUN is improved by a deeper understanding of the system under consideration, which implies that the integration of a DFD into a fully fledged SysML systems model will yield results which are superior to less detailed approaches. This last point coupled with the LINDDUN methodology steps already being subdivided into problem and solution spaces also support the decision for inclusion.

A final point of note is that although LINDDUN guides the user to form a detailed and actionable response to privacy threats, these responses[3] are not the final word in privacy solutions for the system being developed. The door remains open for other steps to be taken and LINDDUN does not exclude the use of additional privacy measures. For instance, it is always advisable for an organisation to have certain minimum privacy measures in place, an obvious case in point being access controls [24].

[3] Either in the form of PETs or privacy requirements.

## 6.6 Model analysis in review

Thus far, this chapter has dealt with the building blocks needed to construct the proposed domain extension. Still lacking though is a means of assessing the domain extension once completed. For this purpose, Chapter 3 introduced and contrasted model checking and model analysis in subsections 3.2.2 and 3.2.3 respectively. As stated there, formal model checking is not directly used in this project but was discussed to set the stage and will also be referenced again in the conclusion to this work. Model analysis, on the other hand, includes the use of verification and validation against system requirements.

In the most general sense, verification and validation allows for the testing of system models against functional requirements and stakeholder requirements. In the case of the DISCREET domain extension, verification plays a significant role. This is due to the domain extension being open to use by any stakeholder in the domain in a number of different ways[4]. Accordingly, validation against stakeholder requirements is less insightful since the stakeholders directly referenced are generic constructs such as system engineers, auditors, and the like, as opposed to individuals with unique requirements. This does not, however, undermine the prospect of testing or analysing the domain extension for correctness and functioning, but does place a greater emphasis on verification.

For an example of testing the functionality of a domain extension, SysML4IoT again provides an interesting example with Costa et al. conducting what they refer to as a "*a proof of concept implementation*[5]". Broadly speaking, this is the approach adopted herein with the following three chapters presenting the domain extension, implementing it by way of case study, and finally addressing verification and validation with a focus on verification.

Some statements about validation can be made though, for example, whatever an individual stakeholder's requirements might be, these must include the ability to use the DISCREET domain extension in their system model. Demonstrating that DISCREET can be imported into an existing system model will therefore speak to validation.

## 6.7 Chapter conclusion

In this chapter we revisited a number of major points established in the preceding chapters, casting these as the basis for the interdisciplinary work to follow. The need for this work was in large part established by our testbed which not only found an extensive range of failures but also determined that these are failures of design. Consequently, the proposed solution must be

[4] Importing the full domain extension as model library into another model, importing part of the model library, or using a portion of the domain extension as blueprint for a different approach.

[5] SysML4IoT has as one of its headline features the facilitation of formal model checking for IoT systems. This is however not an avenue for assessing SysML4IoT itself.

design-based, must attend to both compliance and privacy by design, must take a lifecycle-based view of design, and must at least have the potential for industry application.

From the above base, we then propose that this area of need is best met by a domain extension to SysML which is focused on designing compliant and private systems and devices for consumer IoT. Such a domain extension must then also be a model library and not a new profile, to allow for easier implementation using existing tooling. In the following chapter we present this domain extension.

# Chapter 7

# The DISCREET Domain Extension

## 7.1 The fundamentals of DISCREET

### 7.1.1 Introducing DISCREET

In this chapter, presented in Figure 7.1, we introduce and develop the DISCREET domain extension by building on the work presented in the previous chapter and thereafter setting the stage for use and testing in the following two chapters. However, an implementable version without additional discussion and rationale is presented in Appendix A. We also include the full list of DISCREET requirements separately in Appendix B. The name is derived from its core functionality, which is **D**oma**I**n exten**S**ion for **C**ompliance and p**R**ivacy by d**E**sign in consum**E**r io**T**. The omission of SysML from the name and core functionality is not an oversight but an acknowledgement of the fact that although the domain extension is presented in SysML, it is not ultimately limited to SysML applications only. Depending on the domain and language used, there are multiple possibilities for using the DISCREET methodology outside of SysML applications. Such applications will of course be up to the system engineer using it and will not be discussed further. Also not dealt with in this project are issues relating to the enterprise level use of SysML applications. There are a number of different options here, from those focused on reporting and collaborations, such as Sparx systems Prolaborate[1] to those focused on dealing with massive data pools, such as IBM's requirements management platform Doors[2].

### 7.1.2 Positioning DISCREET as a methodology and domain extension

SysML is a general purpose modelling language and as such is applicable across the piece. However, additional resources or capabilities may well be needed in certain domains and SysML is specifically tailored for the easy incorporation thereof. There are two avenues for rendering such

[1] https://prolaborate.sparxsystems.com/
[2] https://www.ibm.com/uk-en/products/requirements-management

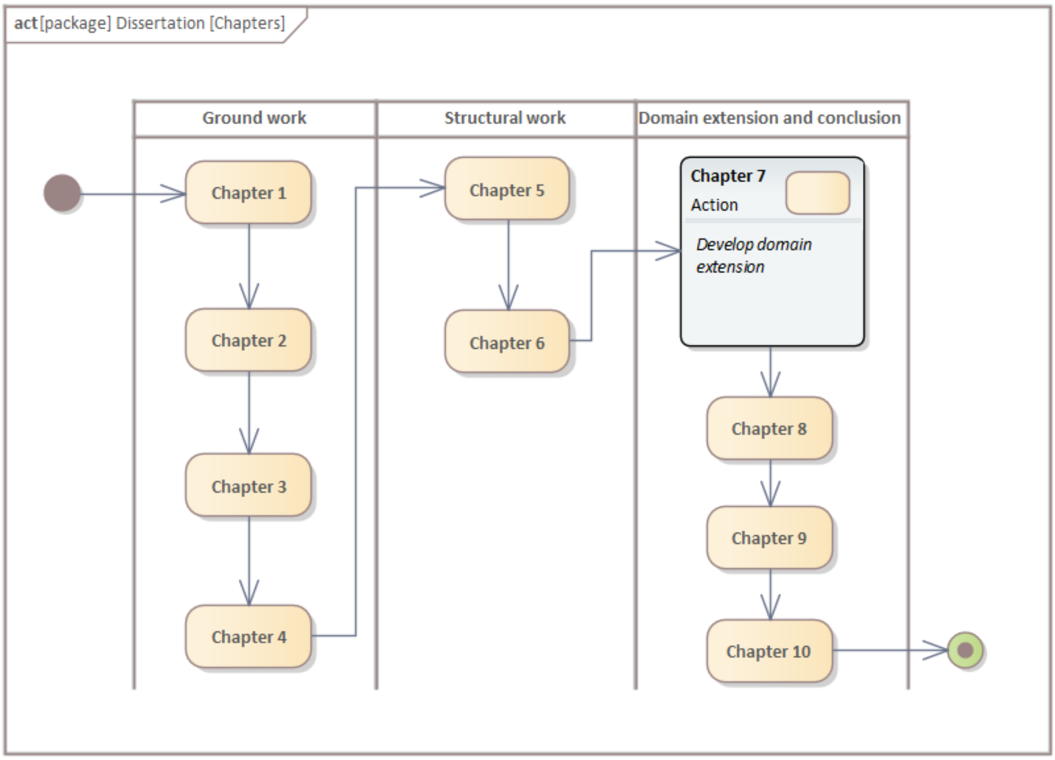



Figure 7.1: Chapter 7 research focus

domain specific extensions [61]. The first is by way of stereotypes, grouped into packages as profiles. Such profiles extend or alter the language itself with a prime example being SysML, which is a profile of UML. The second form of domain extension is a set of reusable and predetermined model elements which are packaged as model libraries. This latter type of extension adds domain specific capabilities without changing the language.

DISCREET contains a series of methods, requirements, and resources which are presented to the user as reusable model elements. These are presented as a model library and as such meets the definition of the second type of domain extension. However, the methods contained within DISCREET direct its application, and lay out the process of conducting compliance and privacy by design for consumer IoT, while using SysML tooling. On this, Costa et al. describe methodologies as collections of related processes, methods, and tools. As such, DISCREET can also be read as a methodology in the generic sense. Although we will not delve into DISCREET's use as methodology any further, it is important to note this as it may open up further avenues of use for third parties.

### 7.1.3 Compliance and privacy by design

In Chapter 1 we presented a number of high profile consumer IoT failures, which were contextualised within PbD in general. However, in Chapter 4 we directly observed such failings on our testbed and found them to be both clear compliance contraventions for the GDPR and failings for PbD when taking a threat modelling approach. In both cases the failings could have been prevented taking a “by design” approach. Although this is an established approach for privacy, as evidenced by the referenced work on PbD, this is not the case for compliance and there remains significant need for such work in general but also specifically for GDPR compliance [119].

Addressing tools aimed at GDPR compliance [119], identifies the following five characteristics of a successful tool:

- Providing snapshot analysis of current compliance performance
- The ability to split out requirements that do not apply to the system
- Determining the actions needed to comply to the GDPR
- Ensuring end users that the system empowers them to exercise their rights[3]
- Providing authorities with documentary evidence of compliance

DISCREET not only meets, but surpasses these criteria.

### 7.1.4 Domain extension presentation, layout, and application

As discussed above, domain extensions can either extend the concepts and capabilities of a language by way of new or changed stereotypes, or they can go the route of reusable model libraries. We have opted for the latter since it situates DISCREET within the constraints of language compliance and therefore allows for any SysML compliant tool to implement DISCREET. System engineers already familiar with SysML should also have a minimal learning curve. As such, DISCREET can be presented as a reusable package of the type *«modelLibrary»*. Included in that package, and displayed below in Figure 7.2[4], we have the DISCREET meta-model, a view of the model library, all included viewpoints and views, methods, the mission needs statement, and there after sections for each of the three requirement sets and finally the use cases including use contexts.

The application of DISCREET starts with the user importing the model library into their existing or new model. Thereafter the user can take a position on using the entire domain extension as is or in a selective manner. This is not covered here as doing so would be speculative. For the use of the entire domain extension the user would start by following the governing method as introduced below.

[3]This point seems somewhat superfluous since compliance with the GDPR will naturally bring this about.

[4]Each element marked with a triangle contains one or more sub-element, the list of which covers several pages, hence the collapsed view in this figure.

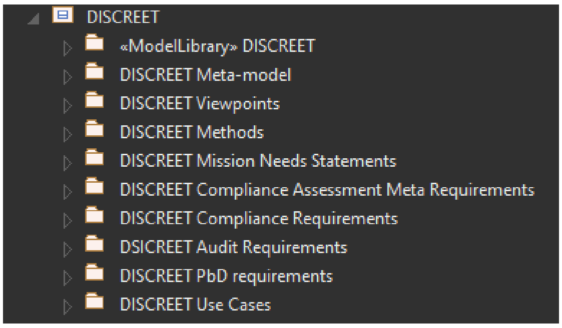



Figure 7.2: DISCREET elements

### 7.1.5 Tool selection

Unfortunately, tool selection can have a significant impact on the success of any SysML modelling exercise [107]. Although there is competition in the market, not all the available tools are up to date or have fully compliant implementations of the SysML standard. Additionally, even if a tool meets all these needs, it might still not be appropriate depending on the ancillary functions and support structures available for the tool. This is especially the case for larger enterprise wide solutions. Although these issues can only be fully addressed by the party wishing to use SysML, we can make sure that our domain extension does not add to the challenge. This is done by providing a fully language compliant domain extension, developed on a tool which is similarly compliant and which supports XML Metadata Interchange (XMI)[5].

Colour coding for traceability can be included in tools but is not a language specification. Although this functionality is not shown here, the practitioner is free to use it as it does not impact on our work or language compliance in as much as it is only a colour variation showing block status, which can include proposed, under review, accepted, mandatory, or even bespoke settings.

We do not advocate for any specific tool though Sparx Enterprise Architect is used throughout. This choice was based on cost, and quality, with the tool being one of the most capable yet still remaining affordable and having a significant discount for academic users.

Matrix views are not specified in SysML, but they are included in high profile tools. Accordingly these matrix views are widely used and even referenced in leading texts [61] on the topic of SysML but can vary significantly due to not being a language specification. Even so, their clear utility see them included.

### 7.1.6 Methods, relationships, lifecycles, matrix and traceability views

Also of significant importance for any systems engineering project, is proper lifecycle management including a well developed understanding of, and plan for, the lifecycle management of each entity within a system [76]. This however, is a primary task for the systems engineer and

[5]XMI is an OMG standard for exporting and importing models and model elements between models and tools.

must be cognisant of the specifics and context of a given system being designed or managed. It is not something which should, or even could, be dictated in a domain-specific extension such as the one developed here. It is therefore imperative that DISCREET does not interfere with the ability of the system modeller to develop the needed lifecycle management protocols. More to the point, it would be foolish to try and predict such needs given the heterogeneous and fast changing nature of consumer IoT. This position also informs the manner in which the MGF is used to structure the rest of DISCREET[6] in Section 7.9.

### 7.1.7 The DISCREET meta-model

The DISCREET meta-model has at its heart the DISCREET domain extension itself, which imports a subset each of the GDPR requirements and CNIL PIA requirements to form the core DISCREET requirements. The domain extension also directs the use of the LINDDUN threat modelling methodology but does not import the provisions directly as requirements. For the sake of completeness SysML 1.5 is also referenced by way of a *«trace»* relationship. This again points to the different approaches to domain extension taken by DISCREET and SysML4IoT as the latter references SysML by way of an *«import»* relationship. This distinction is also evident in our decision not to include a domain model, given the formalism it introduces into a problem space that is inherently dynamic. Options around this issue are, however, discussed in Chapter 10. The final external component is the MBSE Grid Framework which is imported into our methods to provide additional structure.

Moving from external references, DISCREET Viewpoints inform use cases, methods, and requirements as these all serve to populate related views. Lastly, the DISCREET domain extension has four primary outcomes. These are as follows:

- **Primary change:** Model elements are changed due to new compliance, auditing, and PbD requirements
- **Secondary change:** The primary change is not limited to the design phase only but is extended across the system lifecycle
- **Compliance reports:** Through the joint use of the compliance and audit trace methods, audit reporting becomes a persistent requirement
- **LINDDUN threat modelling:** Structured threat modelling becomes a persistent requirement

[6] It should also be reaffirmed that DISCREET is purposefully modular with practitioners free to make changes as needed. Specifically, MGF might be swapped out for OOSEM in larger projects or for reasons of procedure.

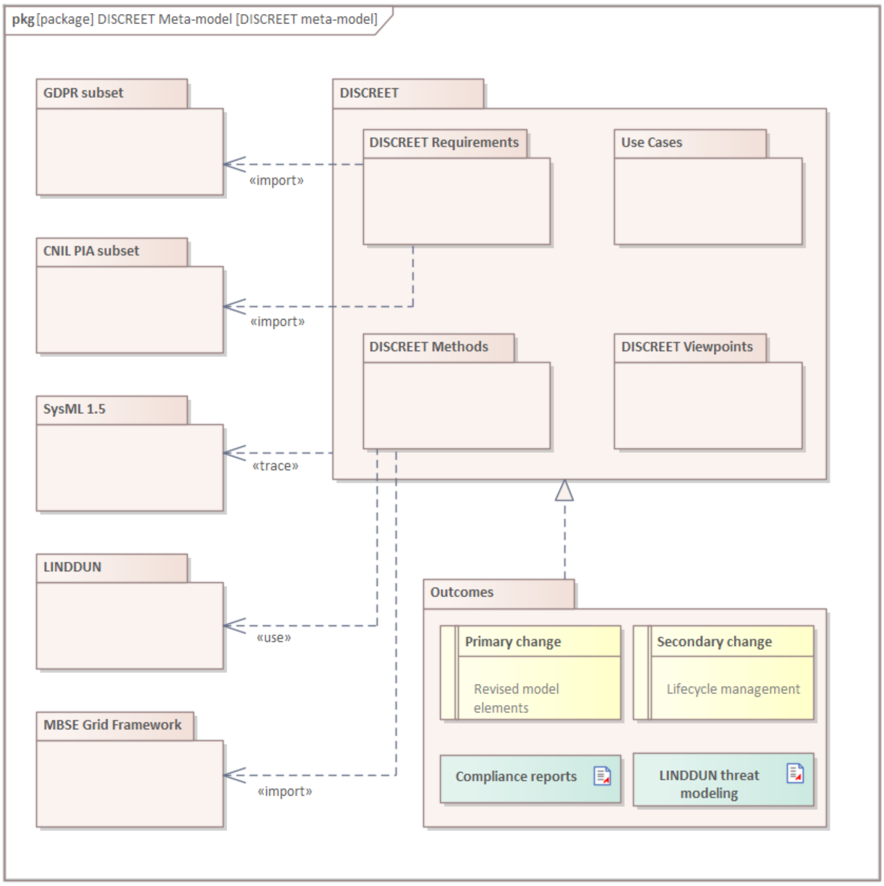



Figure 7.3: DISCREET meta-model

## 7.2 Governing method

The following method describes and defines the components included in this domain extension, allowing for end users to select those components they directly need, using them in a modular fashion. Thereafter, this method describes the systematic use of the entire domain extension as a methodology for compliance and privacy by design in consumer IoT.

This method therefore guides the use of the domain extension in three distinct ways:

1. As a collection of system design resources which practitioners can include in their models in an *ad hoc* manner.
2. A structured methodology for conducting compliance and privacy by design for consumer smart electronics, using SysML.
3. A ready made *«modelLibrary»* for inclusion in other system models.

The method sequence is as follows:

- Derive system model, including stakeholders, viewpoints, and requirements not related to DISCREET[7]
- Determine which elements of DISCREET are needed
- Clarify if MGF or another methodology is used to structure overall execution
- Import the related DISCREET[8] meta requirements to the system model
- Follow the DISCREET methods for each meta requirement
- Maintain and produce DISCREET outcomes
- Maintain continuous reference to DISCREET in the system model to allow for changes over the system's lifecycle

In Figure 7.4 we present an activity diagram detailing the above method. This will be the case for all following methods too, with one notable variation. As part of the governing method the user will have to choose a way of structuring the process of their modelling activities, for us that is the MBSE Grid Framework (MGF) and as such all following activity diagrams for methods will have swimlanes included to show which of the three main MGF phases are involved. Since the governing method is implemented before the choice for MGF, that is not the case here, though the user can of course position the governing method within the black box phase if they wished.

Regarding the progression of the activity diagram, it follows the method steps but provides some more detail. Once the user has reviewed DISCREET they must choose to use it or select certain parts to use in a bespoke application. Choosing the latter option effectively ends DISCREET's management of the activity and it is up to the user to determine their own path. If DISCREET is used as is then the user will still have the opportunity to swap out MGF for other options, this is a realistic prospect as we chose MGF for its ease of implementation and lightweight approach, implying that it is also easier to replace with a different and likely more intensive, approach. If the user does swap out MGF then they will have to restructure the application of DISCREET as needed before proceeding. Thereafter the governing method enters into a loop where DISCREET methods are followed to produce the desired outcomes[9] and DISCREET elements are linked to any new or changed model elements thereby triggering the appropriate methods at the appropriate level. Doing so ensures the application of DISCREET across the system lifecycle.

[7]DISCREET only directs design and corrections relating to compliance and PbD for the system concerned. All other model elements, including business process issues relating to compliance and PbD still need to be drawn in as per the user's needs.

[8]Each element of DISCREET, as discussed below, is hooked into a system model by the introduction of a single meta requirement.

[9]Primary change, secondary change, compliance documentation, and privacy threat modelling.

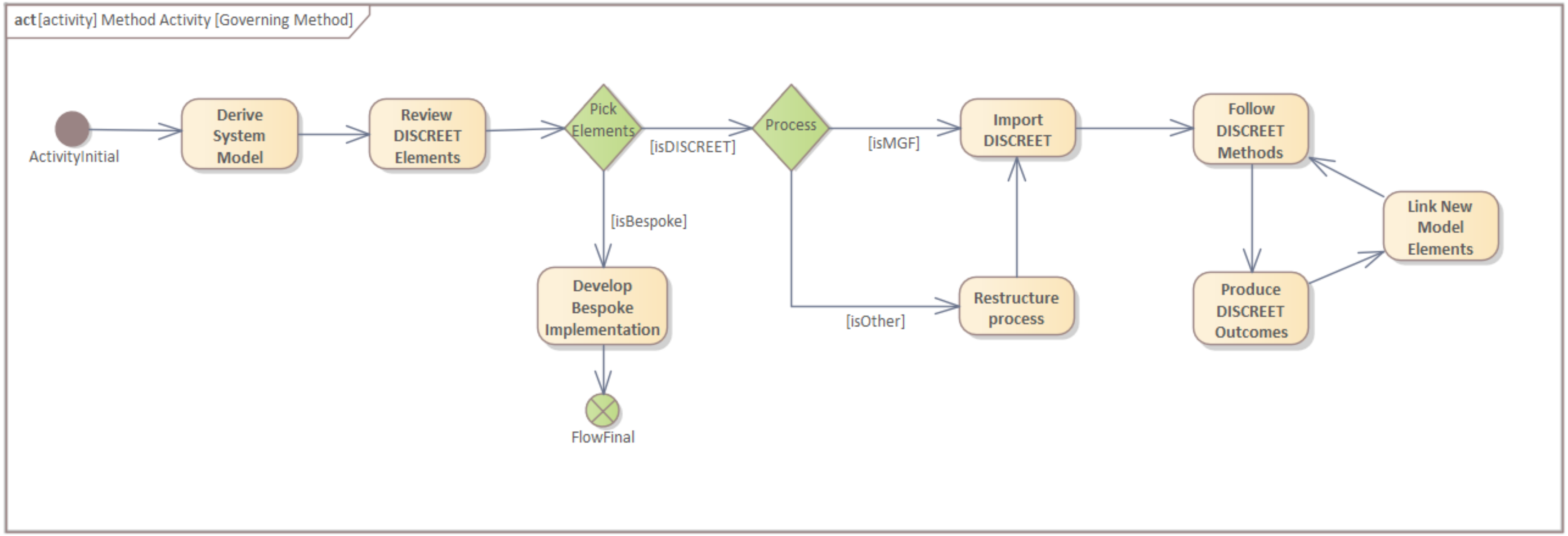



Figure 7.4: Governing method activity diagram

### 7.2.1 Governing method viewpoint

In this subsection, as well as those following each of the subsequent methods, we will present the relevant viewpoints and views. In Figure 7.5 we show the systems engineering viewpoint, which utilises the governing method to draw in most of the DISCREET model library for systems engineers to use. The only exclusion from the model library inside this viewpoint diagram is other viewpoints. Also of note is that one of the use cases included under "DISCREET Use Cases" will of course be the systems engineering use case which has two *«extend»* relationships covering the use of either the entirety of DISCREET or a subset thereof.

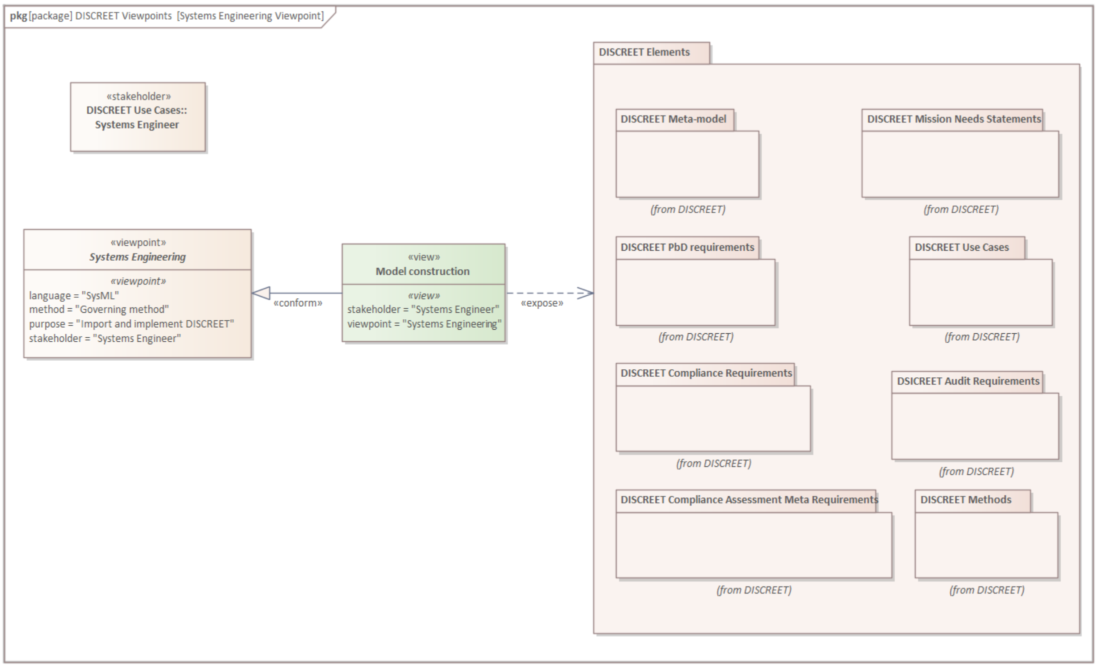



Figure 7.5: DISCREET systems engineering viewpoint

As is evident in Figure 7.5, viewpoints contain a significant amount of information and play a key role in realising the ultimate utility of the modelling project. On this point Holt suggests four questions which must be answered to confirm that a set of data about a model constitutes a well formed view. The first is who the stakeholders are that would be interested in this view. This is essential as there must be a link between the stakeholder and the model presented in the view. Second, why are these stakeholders interested in this view. This relates to the value a stakeholder derives from observing the view. Third, what is the informational content of the view. Informational content in this sense includes the model elements and other artefacts contained in the view. Fourth and finally, what language does the stakeholder expect the view to be presented in. On this final question it might seem obvious that SysML is the language used, however this is not the whole story. As will be highlighted in the following sections, auditors have very specific requirements including knowledge representation. In short, auditors are not interested in SysML models unless they are specifically dealing with a related industry standard. Instead, auditors expect specific items of proof to be presented in specific ways, which is why DISCREET also includes the generation of artefacts specifically containing this evidence using the correct language and format.

## 7.3 Compliance by design

Many regulatory regimes[10] come with associated compliance requirements and ultimately auditing or some form of formal assessment [102]. Given this fact and the avoidable nature of many audit failures [16], as discussed below, we propose that the audit function and how to address it as the party under audit, should not be viewed as something that can only be accessed in an *a posteriori* sense. Rather, with audit guidelines and procedures already published, parties under audit have *a priori* knowledge of the audit process and related requirements. Consequently, this knowledge can be integrated into the systems design process to take a “by design” approach to compliance which is cognisant of audit requirements and automatically produces system artefacts to demonstrate compliance.

Our contribution, as opposed to other approaches such as those discussed in Section 7.3.1, allows systems engineering practitioners to directly integrate the external audit function as a new stakeholder with related requirements. These requirements are traced through systems components to regulatory requirements, as normally included, with any completed trace producing a systems artefact in the form of audit documentation. Traces which do not complete this link also produce system artefacts in the form of a non-compliance reports and remedial actions. All of this is achieved through the use of standard model-based system engineering (MBSE) features. Though the examples presented here are expressed in the Systems Modelling Language (SysML), the work is portable to other prominent modelling languages and architectural frameworks[11]. As such, practitioners will not need any new tooling or have to resort to non-standard tool use to deploy this method, or indeed any other part of DISCREET. It should also be noted that the method proposed in Subsection 7.5, as is the case for all the work presented in this domain extension, provides a system for framing the elements within a given problem space using SysML but does not presuppose the eventual answers to those questions as doing so would run counter to standard practice and undermine the utility of the method over the system lifecycle[12].

### 7.3.1 Alternate approaches

Although there have, unsurprisingly, been many attempts at dealing with regulatory compliance in a more formal manner, some times including a “by design” approach, these approaches all aim

[10] Not just those concerned with financial regulation.

[11] Such as using the United Profile for DoDaf/MODAF (UPDM) to bring SysML into alignment with the defence contract requirements.

[12] The system lifecycle runs from design, through implementation, to decommissioning and may stretch over many years. Any number of external changes can impact on the system and its domain meaning that solutions to design challenges may well change over time.

at dealing with the same challenges. Legislation can be densely written, contain potentially ambiguous phrasing, and be heavily dependent on additional resources including other legislation [92].

The notion of using a model-based approach to attach requirements to the functional elements of a system, is well established in model-based Systems Engineering (MBSE) [61]. This approach does, however, not require a specific modelling language to be used and can indeed be applied to other solutions, such as the pattern-based analysis conducted by Beckers et al. [17].

Researchers dealing with these challenges in a more general sense, may take the option of developing an automated system for requirement extraction. This is the route taken by Kiyavitskaya et al. through their adaptation of the Cerno framework[13] for application to US legislation [92]. Although the authors report promising results there are a number of significant factors which disqualifies this approach from being considered. First, this approach does not identify and transform legal textual requirements into the nuclear "shall statement" form needed for use in an MBSE approach [14]. Second, the possibility of both false positives and false negatives is a serious consideration when dealing with regulatory compliance given the penalties that may be applied by regulators. Third such an approach does not provide any structure for linking the extracted requirements to the engineering and auditing processes. Fourth, and potentially most significant, MBSE explicitly supports the reuse of model elements from other sources. As such, a fully extracted set of legal requirements can be imported from publicly available sources [15]. We therefore concluded that although there is nothing inherently wrong with taking the automated extraction route, it is ill suited to the specific application presented here and does not address the requirements of auditors.

A further significant area of study is the derivation of privacy requirements from the needs and wants of device users. The results of such an exercise can by definition not speak to the legal compliance of a system *vis-à-vis* a formal compliance audit [147]. These approaches do, however, hold substantial value as far as customer expectation and the provision of privacy assurances beyond the legally compelled minimum. Accordingly, the topic is further discussed in Section 7.7.1.

Yet other systems seek to take a pattern-based approach which is compatible with systems modelling, such as that developed by Beckers et al. [17]. As opposed to their system, the one proposed here does not explicitly rest on UML, is not inherently linked to an application domain[16], and is not geared towards use with a specific tool.

The last significant set of alternate approaches are those, such as Hassan and Logrippo [71] who take a purely logic-based approach to assessing legal compliance. Within model-based systems,

[13] The Cerno framework is used generating semiautomatic text annotations

[14] Why we view an MBSE-based approach superior is covered in Chapter 5 and Section 5.4.

[15] Setting up such a repository containing a full set of GDPR and CNIL auditing requirements for consumer IoT devices is currently underway and will incorporate the method presented herein.

[16] DISCREET functions as a SysML domain extension but remains broadly modular to the extent that users can still port it to other application domains.

with their recourse to state machines and other system representations, such an approach is definitely feasible, though it does present some challenges. Chief amongst these is the notion that the "map is not the territory" [75]. This implies that proving model compliance is not yet an airtight guarantee that the real world system is 100% compliant since there may well be implementation errors or other discrepancies. This is the exact reason for audit regimes to include a specific focus on implementation when assessing compliance[17]. In other words, a compliance check is performed against the "territory" and not the "map". This brings us to the same set of objections previously listed, in as much as the output from a formal model does not correspond to the documentation required by auditors.

### 7.3.2 Requirement engineering for compliance

Stakeholder engagement is a well established concept in general terms, how that concept is realised though remains highly context bound and is still often seen in terms of the direct, and positive, financial implications [56]. However, modern systems engineering practice takes a much broader and more inclusive view, holding that all parties who could interact with, or impact on, a system model, are stakeholders. This is not just aimed at including all relevant parties at a superficial level, but to constitute the entire system lifecycle as a stakeholder-centric model, a point underscored by Kapurch who refers to systems engineering as a means of achieving "*stakeholder functionality*" [89][18]. This stakeholder functionality is realised by casting stakeholder needs as requirements which attach to functional elements within a model, as part of the design and management process within MBSE [61]. These foundational elements are then also present in the *Systems modelling Language* (SysML)[71], within which the compliance trace method is described in Subsection 7.6[19].

Although standard practice within MBSE would suggest that the inclusion of audit requirements in system models would be a given, this conclusion does not entirely follow in practice since models are ultimately built by practitioners with their own goals in mind, while auditing is often viewed as an activity to be conducted after the fact. In the compliance trace method presented in Subsection 7.6, and as alluded to earlier, we contend that viewing auditing as something which purely takes place once a system is in place, is particularly risky and can lead to unnecessary audit failures. We therefore advocate a "by design" approach which integrates positive audit outcomes into the system design process and generates audit documentation procedurally, not only during the design and development phases but throughout the system lifecycle. A lifecycle-based approach such as this can generally be viewed as superior to static approaches such as those only focusing on the design phase or post-implementation auditing [136]. The need for an integrated system which generates audit documentation and spots non-compliance before the fact

[17] This notion is explored in detail with reference to the CNIL's PIA audit methodology in Chapter 3 and in this Subsection 7.5.

[18] As previously also mentioned in Section 1.2

[19] Using a standard from the *Object modeling Group* (OMG), such as the *Unified modelling Language* (UML) or SysML not only opens up possible interoperability with other languages and schemes but also removes barriers to industry application since no new tooling and very little new training is needed.

is also clearly supported by the fact that non-compliance due to human factors, system failings, and a lack of systematic attention to compliance, constitutes a set of decades-old [142], persistent [129], and avoidable reasons for compliance failures. A case in point, is a study which found that more than 80% of compliance failures related to an inability to gather sufficient proof[16][20].

## 7.4 Traceability via package diagrams

As already stated, we propose to address this lack of documentation through the automatic generation thereof in the systems engineering process and over the entire lifecycle of the system. To accomplish this we not only use MBSE in general, but specifically focus on SysML, where SysML can be understood as the language within which we conduct the activity of MBSE [41]. As opposed to UML classes, which some might be more familiar with, the primary structure in SysML is modular and divisible blocks [146]. Coupled with features such as inheritance, package diagrams, and requirement modelling, SysML provides a rich and nuanced language within which system models can be expressed [61].

From the long list of SysML capabilities, both unique and incorporated from UML, one of the most prominent for our purposes is the use of package diagrams to display model elements and represent the relationships between these elements. These relationships can be expressed in a number of ways and include significant flexibility in dealing with unique and changeable situations. Combined with permissive naming conventions for both system elements and package diagrams, this flexibility allows for the compliance trace method[21], introduced in Subsection 7.6, to be used in any standard SysML tool. Using an established modelling language and introducing new capabilities without the need for new tooling is essential since it enables privacy engineering and compliant systems design in a context familiar to systems engineers [99]. This not only enhances the system engineer's ability to deliver on stakeholder needs but also enables the creation of a model that is significantly better at representing the full scope of a truly compliant[22] system. Figure 7.6 presents the traditional view where requirements are only drawn from the related legislation without the inclusion of the audit function. Lastly, it is important to note that the proposed compliance package diagram is an element within the model being used and not an artefact of that model. The compliance package does, however, contain the means by which adherence to requirements can be traced and related proof artefacts produced.

[20] This specific issue was one of the main motivating factors for including an audit method in the larger domain extension, due to first hand experience in compliance audit and dealing with consistent audit failure due to lacking documentation.

[21] In SysML the term "method" is used to describe linked activities, artefacts, and options, and the system for representing or executing these.

[22] Although we use the GDPR and the CNIL's PIA for IoT, any other compliance regime with associated audit requirements could benefit from this approach.

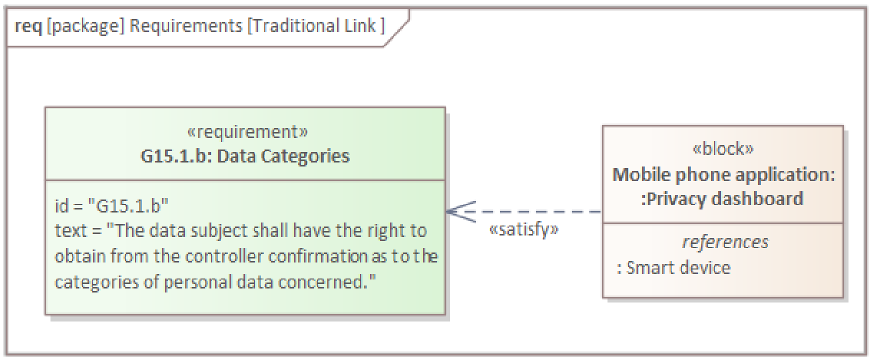



Figure 7.6: Traditional requirements treatment

## 7.5 Audit trace method

The compliance trace method introduced in Subsection 7.6 heavily relies on the inclusion of audit requirements in addition to the standard use of compliance requirements. This functions in line with standard SysML practice with auditors treated as stakeholders, leading to the generation of the associated views and viewpoints. However, factors such as secrecy from auditors, lagging or outdated audit practice[23], or a total lack of an audit authority[24] could bedevil the compliance trace method. To structure the inclusion of existing audit requirements and also guide the generation of audit requirements where none are readily available, we present this Audit trace method.

The audit trace method aims to provide stand in audit requirements which are traceable with regards to their origin and formulation, but can also easily be swapped out if better suited or official audit requirements become available. The core principles of this framework are as follows:

1. The audit trace method is an element of the larger domain extension
2. Requirements generated as part of the framework are labelled as such within a model to allow for easy future replacement
3. Requirements thus generated will be subservient to any compliance requirements generated for published legislation, in the case of a conflict
4. The principles contained in the audit trace method are based on established best practice for both auditing and requirement formulation

In Figure 7.7 we present an activity diagram describing the process of the audit trace method, with swim lanes used to differentiate between the black box, white box, and solution phases. Per definition, the inclusion of an audit methodology does not yet constitute a solution to the modelled problem space and accordingly, the solution space swim lane will remain empty at this

[23] In reference to new compliance regimes.

[24] If this is expected to change over the lifecycle of the system being designed then future audit requirements should be accounted for as far as possible.

level. There is also a three way fork in the activity diagram which indicates a choice between an official audit methodology, such as CNIL for French GDPR compliance, an adjacent methodology such as CNIL for UK Data Protection Act[25] compliance, and a placeholder methodology where no official or adjacent methodology exists (yet). For this latter position we draw on the work of Pasquier et al. and their approach to auditing compliance within the IoT. This work was selected not only for its direct focus on the IoT and reference to privacy auditing, but also the inclusion of a system wide focus.

Regarding the CNIL's PIA, this is the audit methodology which we will be using in DISCREET and as such, we will follow fork two of the audit trace method, as outlined below, when presenting a case study in Chapter 8. It is of course possible that a user of DISCREET might opt for something wholly unrelated to the CNIL PIA, which is why this answer is not prescribed. We have, however, ensured that the four fundamental requirements of conducting the CNIL PIA are inherently met by our methods, as these are best practice for any compliance audit. The four fundamental requirements [35] are listed below:

1. Define and describe the context under consideration
2. Analyse the controls guaranteeing compliance to required principles
3. Assess risks and ensure they are properly treated
4. Formally document the audit outcome and revise system as needed based on audit outcomes

The first point above is directly addressed in the audit trace method, while points two and three are realised through the interaction between the audit trace method and compliance trace method. Concrete examples of context determination include, understanding the type of processing involved[26], identifying the controller and all processors, and collecting the documentation needed to evidence the preceding.

The key to understanding the fourth point above is that the CNIL PIA is explicitly intended as an iterative process, at least on the part of the data controller. Meaning it facilitates improvement over time, up to and including conducting an external audit and is expressly focused on a "by design" approach. Where the work presented herein diverges from both the CNIL PIA and the GDPR is of course that DISCREET is aimed at device and systems design and not at business process management. As a result, those audit and compliance requirements aimed at the organisational level, or the overall management of the project, will not be included.

**Fork 1: An official audit methodology**

- Determine context

[25] The compliance requirements (in theory) match between the GDPR and the Data Protection Act, but the UK does not provide an audit methodology comparable to that of the CNIL.

[26] This includes the data, data subjects, processing purposes, processing scope, and duration.

- Determine if there is an official audit methodology
- Include a requirement that all audit requirements should be drawn into the model as appropriately formulated “shall statements”
- Determine that all audit requirements are included and up to date, if not, then return to the previous step[27]
- Record all requirements to the requirements matrix for later use in the compliance trace method

**Fork 2: An adjacent audit methodology**

- Determine context
- Determine if there is an official audit methodology
- Determine if there is an applicable adjacent audit methodology
- Include a requirement that all audit requirements should be drawn into the model as appropriately formulated “shall statements”
- Determine that all audit requirements are included and up to date, if not, then return to the previous step[28]
- Determine that no new official audit methodology has been released, if so, return to step two[29]
- Record all requirements to the requirements matrix for later use in the compliance trace method

**Fork 3: Stand in audit via provenance graphs**

- Determine context
- Determine if there is an official audit methodology
- Determine if there is an applicable adjacent audit methodology
- Develop a system wide DFD-based provenance graph
- Split the graph into inter and intra-context segments, per IoT context[30]
- For each compliance requirement, formulate an audit requirement requiring proof of compliance

[27] This is both a check for the correctness of the work done and a lifecycle change check.
[28] This is both a check for the correctness of the work done and a lifecycle change check.
[29] This is a lifecycle change check.
[30] Processing context can change the regulations at play and therefore the compliance requirements.

- To each newly formulated audit requirement, attach a provenance graph describing the associated data actions
- Develop standard documentation to capture compliance outcome and associated graph
- Determine that no new official, or adjacent, audit methodology has been released, if so, return to step two[31]
- Record all requirements to the requirements matrix for later use in the compliance trace method

Regarding fork three, it is important to recall the work of Beasley et al. who found that the most prominent reason for audit failure was an inability to provide documentary evidence of compliance. This is a failure of provenance.

[31] This is a lifecycle change check.

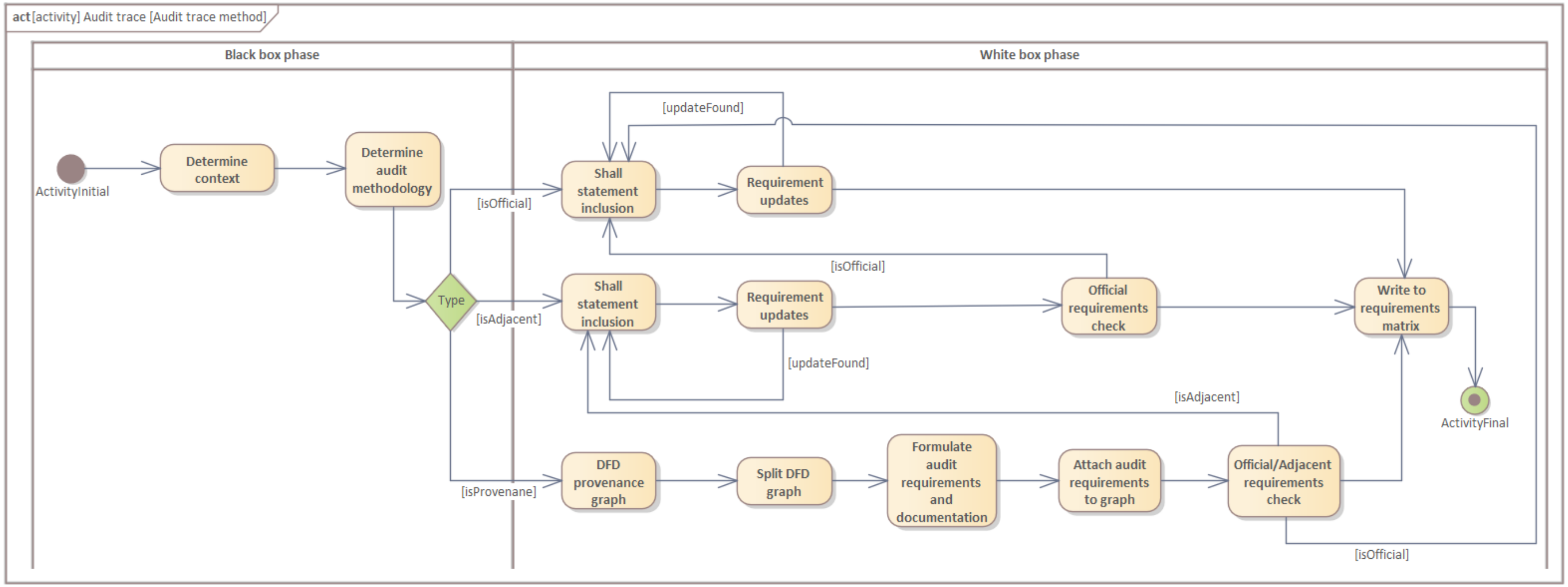



Figure 7.7: Audit trace method activity diagram

### 7.5.1 Audit trace method viewpoints

Although all the DISCREET methods are intended for use by systems engineers, there may still be other associated stakeholders and consequently more than one viewpoint. That is the case here as we have the external compliance auditor as a stakeholder. This stakeholder has an associated viewpoint reflecting their need to perform a compliance audit using the CNIL PIA, with CNIL templates as output. The associated view includes the DISCREET audit requirements, compliance reports[32], data subject, use case for the data subject and of course the DISCREET meta requirements.

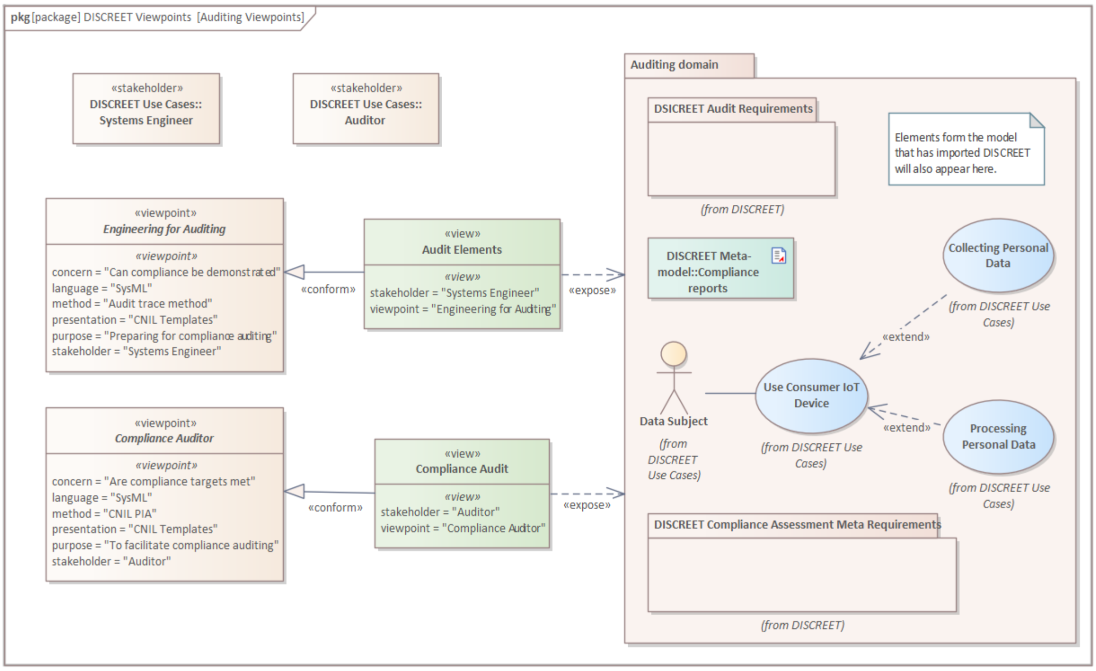



Figure 7.8: DISCREET auditing viewpoints

The audit trace method is not explicitly shown in the view as it is the method referenced in the viewpoint. This holds for all the viewpoints and views other than that of the governing method given that the governing method refers to all of DISCREET. Here it is also important to note that there are of course a number of other stakeholders that could be involved and accordingly more viewpoints and views to generate. These, however, are dependent on the system and model that DISCREET is ultimately drawn in to.

## 7.6 Compliance trace method

Once the relevant compliance and audit regimes have been identified, the systems engineering practitioner can proceed to first elicit the appropriate requirements in the form of "shall statements", then capture them in a compliance matrix and finally include these in the system model

[32]Here we use the "Compliance reports" element from the meta-model to indicate the notion of a compliance report as opposed to including a specific report type or instance of an actual report.

being developed. This inclusion takes the form of connecting requirements of all types to other system elements. This act of connection also introduces further crucial details to the requirement matrix and allows for it to be used to quickly trace system connections and gain an overview of various current system states[33].

A key component of the first step, that is eliciting requirements from compliance and audit regimes, is ensuring that this refinement includes all possible requirements which could be placed on the system under development [71]. However, we propose going one step further and capturing all provisions and requirements which could impact on systems of the type being developed[34]. This implies that the practitioner does not prune out requirements that might currently appear to be outside of the scope of the system, which is necessitated by taking a lifecycle-based approach to system's design. In short, a multitude of changes might impact on the functionality of a system over its entire lifecycle [117] and the requirements included in the system's requirement matrix should be as broad as possible to cover such changes. The activity of requirement elicitation and casting as "shall statements" is standard practice in MBSE [89]. The result then, of this broader inclusion of possible requirements is that the only change in either system or environment that would cause requirements to be revisited or altered is a legislative change, which is regarded by systems engineering practitioners as the least likely event to cause model change [117].

The compliance trace method proposed here is divided into two sections, namely a method sequence and the generation of reporting artefacts. The method sequence formalises the foundational ideas expressed above while the generation of reporting artefacts is highlighted due to its key role in the method's value proposition to practitioners.

As discussed in Sections 5.3 and 7.2, the structure for executing the methods contained in DISCREET is provided by the MGF. This structure requires that design commences with a black box phase and then a white box phase to round out the problem space, from there, including all relevant specifics of the system under design, the solution model can be proposed.

### 7.6.1 Compliance trace method requirement inclusion and viewpoints

The compliance requirements presented herein are drawn from the GDPR and include all articles which directly address system or device functioning. The intent is to only exclude those articles relating to issues of context, scope, or other factors outside of the direct system being modelled, leaving the *«modelLibrary»* as detailed as feasible. Accordingly, articles such as Article 1 which deals with the "subject-matter and objectives" of the GDPR, are excluded. Similarly, Article 14 which deals with personal data not sourced from data subjects is also excluded. DISCREET deals with IoT devices and systems as they collect and process the data of data subjects who

[33] Such as how many system elements connect to a certain compliance requirement. The compliance trace method extends this to also include a percentage overview of full, partial and no compliance.

[34] Bearing in mind that we are operating within the context of legal compliance. This should not be read as an instruction to include general requirements outside of that context.

are users of those devices and systems, and not with information brokerage. Furthermore, the wording of the English GDPR text is used as far as possible. To limit requirement length some might be split into more than one requirement and in such cases the article number, which we also use as requirement ID, is appended with a Roman numeral. In most cases though, the core of the requirement from the GDPR is used to formulate the shall statements, with additional information included in the “notes” field for each requirement. This allows for clearer requirements without the loss of potentially critical information. We also use the “notes” fields to include links to external resources which systems engineers or other internal stakeholders might find useful when dealing with the GDPR. However, as will be clear from the discussion in Subsection 7.6.3 simply including compliance requirements and audit requirements in the same model does not yet enable design for auditable compliance. What is needed is a series of meta requirements to guide this process and direct the generation of audit documentation. As such, these meta requirements also form part of the related view of DISCREET or a future system model it is imported in to.

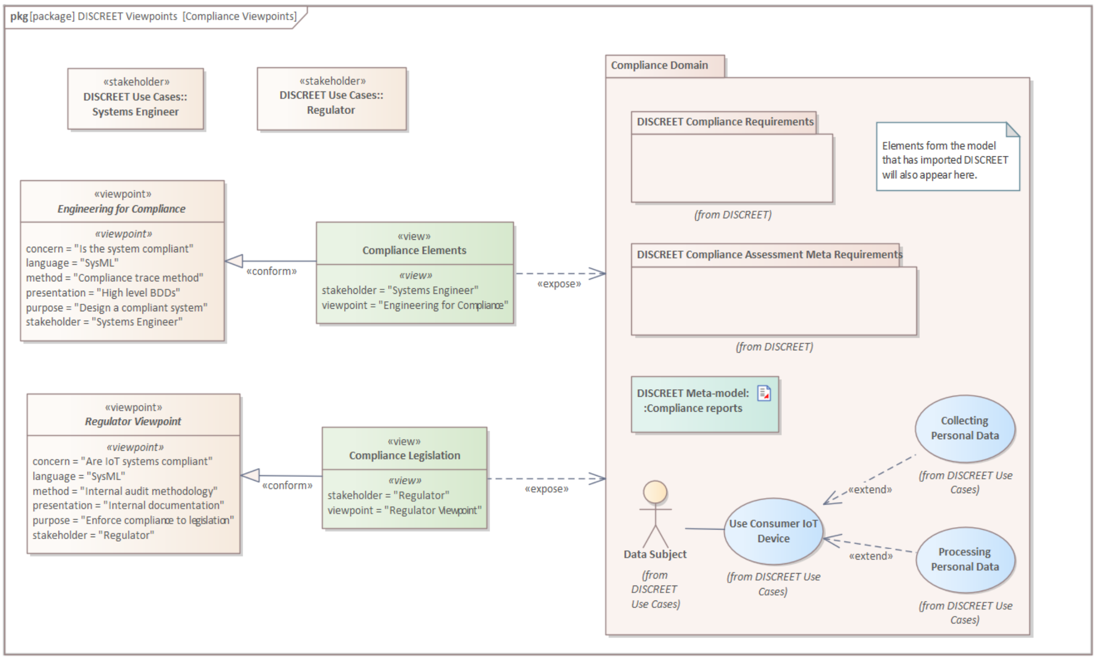



Figure 7.9: DISCREET compliance viewpoints

In Figure 7.9 we present DISCREET’s compliance related viewpoints. As was the case with auditing, there are two viewpoints which both spawn the same view. This is due to the system engineer’s interest in generating a model (and system) that can answer the needs of an external stakeholder, in this case the regulator. For example, the regulator might not be interested in the DISCREET meta requirements, but they will certainly be interested in the contents of the internally generated compliance reports, which are reliant on those meta requirements. As before, we present the notion of a compliance report as taken from the meta-model, a high level view of a data subject using some consumer IoT device which deals with personal data. Explicit allowance is also made for the future inclusion of elements drawn from models created by the system engineers using DISCREET.

### 7.6.2 Method Sequence

The compliance trace method consists of the following discrete steps, with the audit trace method subsumed therein:

- Determine applicable legislation
- Cast the compliance requirements as "shall statements" and include in requirements database[35]
- Complete the audit trace method
- Include meta requirements for reporting artefacts generation as per Subsection 7.6.3.
- Build out the system model in SysML
- Attach the applicable requirements to model elements using the *«satisfy»* relationship
- Generate reporting artefacts where requirements and model elements intersect or should intersect.
- Link meta requirements to reporting artefacts using the *«realization»* relationship
- Link compliance requirements, audit requirements, and system components to the newly generated reporting artefact using the *«trace»* relationship

An important point of note here is that a system model can distinguish between proposed and implemented architecture. This also means that the above method allows the systems engineer to monitor compliance performance in real time and even against logical components in the white box phase, thereby increasing the chances that realised architecture in the solution phase will be compliant. Therefore, the same method can be deployed across the system lifecycle and is represented by the final steps of the method in both the white box and solution phases. The activity diagram presented in Figure 7.10 presents the steps involved in completing the compliance trace method including the entirety of the audit trace method as a single action. Since that action breaks out into an activity diagram of its own, an upside down rake symbol is displayed. This single action is also included in the white box swim lane as that is where the bulk of its workings fall, which is also why the solution swim lane is not present for the audit trace method.

[35] In this work the requirement database and requirement matrix are one and the same. However, practitioners may separate these out and reflect all captured requirements in the database and only those in effect in the matrix.

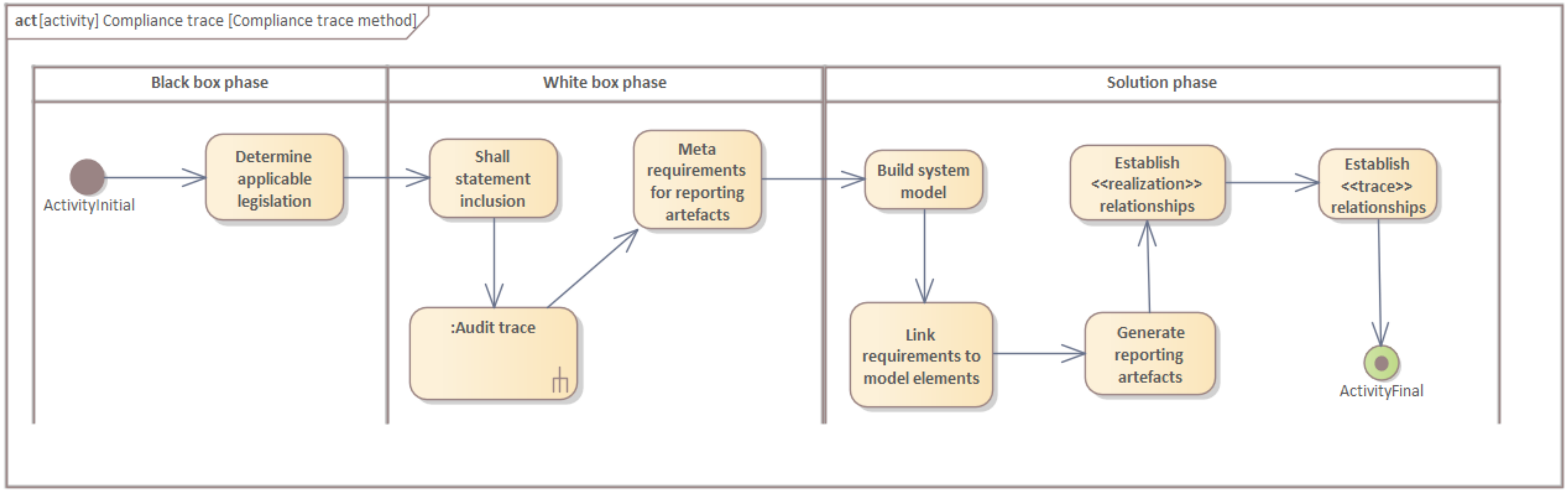



Figure 7.10: Compliance trace method activity diagram

The introduction of the aforementioned meta requirements into the system model is crucial as this is the linchpin which makes the entire trace method actionable, including the production of reporting artefacts as discussed in Subsection 7.6.3. The generation of these reporting artefacts can then be presented in SysML using a dependency relationship of the *«realization»* type, between meta requirements (supplier) and report artefacts (client). The connection is shown using a dotted line and arrow from the client to the supplier[36]. These reporting artefacts must of course also link to the related compliance requirements, audit requirements, and system components to enable the "trace" functionality of the audit trace method. For this the aptly named, but semantically weaker[37], *«trace»* dependency type is used. This trace not only shows which model elements impacted on given reporting artefact but also allows for cross-referencing in a matrix view of system elements which is generally supported by most modelling tools.

### 7.6.3 Generating reporting artefacts

Reporting artefacts are the compliance reports produced either in accordance with audit requirements or to explain, and propose remedies to, non-compliance. The process of generating reporting artefacts is crucial since these reports must be consistent across the system lifecycle, easy to interpret, and clearly linked to the components and requirements they report on. Achieving consistency and ease of interpretation is attended to by the rules presented below while dependency relationships were discussed in Subsection 7.6.2.

The starting point for ensuring consistency is to class the reports generated by type and to formalise this structure within the system model by way of five meta requirements. One of the results of importing these meta requirements is that all DISCREET compliance and audit requirements active in the system model must link to at least one of the four meta requirements via a *«satisfy»* relationship. This not only allows for the easy inclusion of the compliance trace method within standard SysML practice but also allows for additional analytics to be drawn. For example, the establishment of these *«satisfy»* relationships will mean that practitioners can draw compliance snapshots from the requirement matrix to see what proportion of system elements are non-compliant. The four meta requirements each correspond to one of the classes listed below.

1. Class 1: Full intersection (Compliance requirement, audit requirement, and model element.). Generates class one output: Compliance documentation, as per the linked audit requirement.

2. Class 2: No audit intersection (Compliance requirement and model element). Generates class two output: provisional documentation and proposed remedy.

[36] This is standard SysML notation.

[37] This is weakness in the sense of less associated formalism and therefore greater flexibility in application.

3. Class 3: No compliance intersection (Audit requirement and model element). Generates class three output: Non-compliance documentation[38].

4. Class 4: No intersection. A model element should, on the face of it, be under the requirements of the applicable regulation but has no compliance or audit intersections. Generates class four output: Non-compliance documentation and proposed action to determine cause of the issue.

The reporting artefacts, generated whenever any other system elements link one of the class meta requirements, must state the class that spawned it, all dependencies or other system links associated with it, the package diagram within which it is represented, a unique identifier, and of course the proof mandated by the associated audit requirement. If the reporting artefact was not generated in accordance with a fully compliant setup, that is if Class 2, 3 or 4, were realised, then the report must also explain why this is the case and what remedial steps are to be taken. A number of factors can bring about this outcome, including a modelling error, compliance and audit requirements being out of sync due to lag from legislators or regulators, or technological development outpacing regulations. Finally, for Class 3 non-compliance reports the relationship between the model element and the audit requirement should be *«trace»* instead of *«satisfy»* to indicate a relationship but a failure to satisfy the requirement.

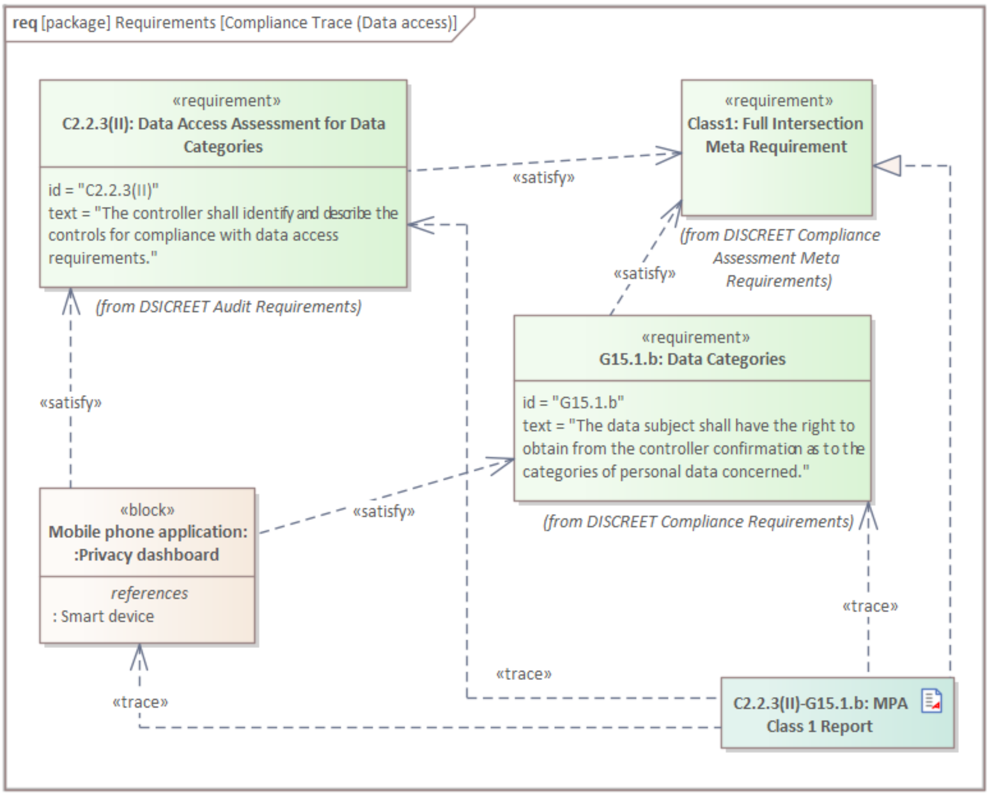



Figure 7.11: Data access compliance trace

38This indicates non-compliance since a model element does not have a *«satisfy»* relationship with a compliance requirement, while an audit requirement seeks proof of compliance.

In Fig 7.11 we present a privacy dashboard, meta requirement Class 1, GDPR Article 15 data access requirement [151], CNIL privacy impact assessment Section 2 data access requirement [35], and the reporting artefacts generated [129]. The fictional system presented here is a privacy dashboard which is one element of a mobile phone application, which in turn controls a linked consumer Internet of Things device[39]. The reporting artefact is named to reflect the GDPR and CNIL requirements it complies with, while the contents of the report will be a compliance information sheet [36] following the CNIL's audit implementation for IoT devices [33], and other supporting documents which are referenced in the information sheet.

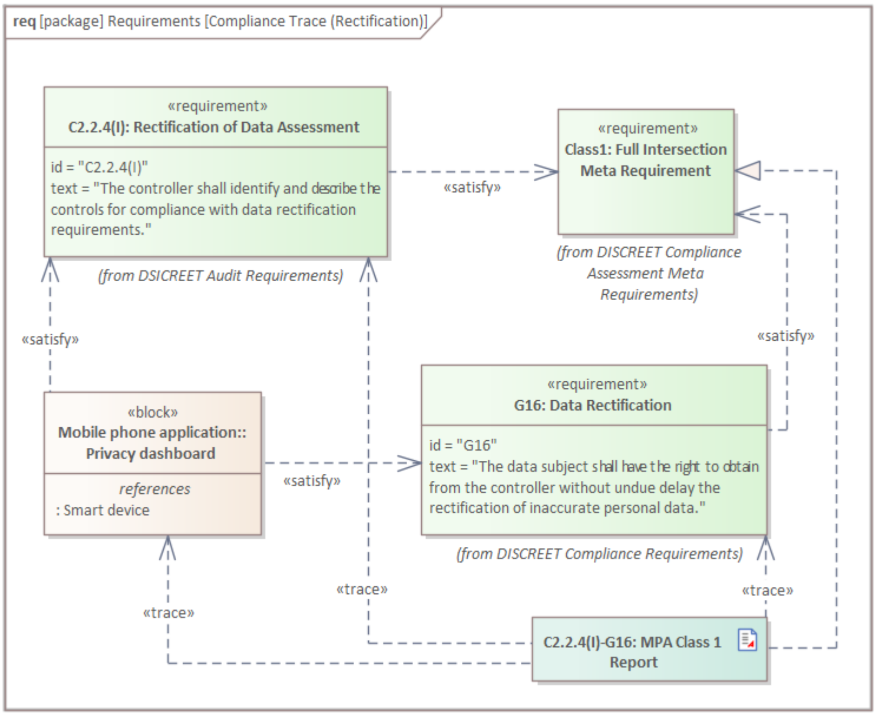



Figure 7.12: Data rectification compliance trace

The privacy dashboard in our fictional example is still being developed and at present does allow users to edit (rectify) their data, but does not yet have any functionality for data deletion. Consequently, a further Class 1 compliance report is generated and one Class 3 non-compliance report as per Figures 7.12 and 7.13 respectively.

[39]This is a simplified model used as an illustration only.

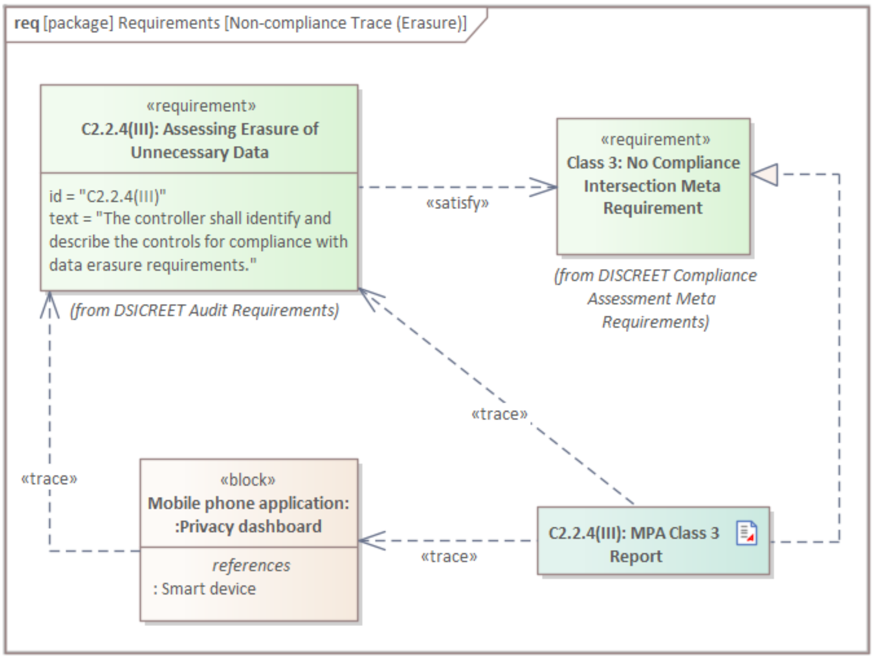



Figure 7.13: Data erasure non-compliance trace

The compliance trace method, as demonstrated through the above diagrams, can address regulatory compliance, related auditing and the associated issue of traceability between the various model elements. As opposed to other approaches to these issues, the one proposed here does not neglect the audit function, is not limited to use with a specific tool, and remains functional over the entire system lifecycle. The final advantage offered by this method is its ability to provide system engineers, or other stakeholders, with easily understandable reporting on system compliance and related issues. This is achieved by way of matrix analysis and although the exact layout of such will be tool dependent, the outcome will remain the same.[40] This can be seen in Figures 7.14 and 7.15 which not only indicate if legal and audit requirements are satisfied, and which compliance/non-compliance documents have been generated, but also provides a visual snapshot of these states changing in real time.

| | Class | Class1 | Class2 | Class3 | Class4 |
|---|---|---|---|---|---|
| **Compliance and Audit Requirements** | | | | | |
| G16: Data Rectification | | ^ | | | |
| G15.1.b: Data Categories | | ^ | | | |
| C2.2.4(I): Rectification of Data Assessment | | ^ | | | |
| C2.2.4(III): Assessing Erasure of Unnecessary Data | | | | ^ | |
| C2.2.3(II): Data Access Assessment for Data Categories | | ^ | | | |

Figure 7.14: Requirements satisfied (*«satisfy»*)

[40] This matrix analysis is not part of the SysML language but due to the obvious utility of being able to plot system elements against one another, most tool vendors include it in their offerings.

| | Class | Class1 | Class2 | Class3 | Class4 |
|---|---|---|---|---|---|
| **Reports** | | | | | |
| C2.2.4(I)-G16: MPA Class 1 Report | | ^ | | | |
| C2.2.4(III): MPA Class 3 Report | | | | ^ | |
| C2.2.3(II)-G15.1.b: MPA Class 1 Report | | ^ | | | |

Figure 7.15: Compliance report generation (*«realization»*)

## 7.7 Privacy by design

### 7.7.1 Non-regulatory privacy requirement elicitation

There are of course a vast array of privacy requirement work which is not centred on compliance. This comes with both advantages and challenges, as is evident from work such as the Privacy requirement Distillation method developed by Thomas et al. [147]. In our literature review presented in Chapter 3 we also explored a number of different takes on privacy in general and privacy linked to set domains. One of the most telling examples was that of the Solove privacy taxonomy which has gained wide acceptance but which also has the potential to cause a significant number of problems for consumer IoT devices and systems with their heavy reliance on personal data, both in collection and processing. As such, an approach is needed which not only takes the real world specifics of the system under consideration into account, but that produces actionable results. For this and other advantages we turn to threat modelling.

On the topic of defining threat modelling, Shostack in his well regarded book on the topic, at first sidesteps the issue of precise definition by stating that pinning down definitions "*is a strange game, and the only way to win is not to play*". He comes back to it though and finally states that threat modelling is "*the key to a focused defence*" and "*the use of abstraction to aid in thinking about risks*". Focusing on privacy by design in consumer IoT then, we did not find a suitable domain specific threat modelling methodology or tool, but did find that the LINDDUN threat modelling methodology is entirely applicable as is.

This use of LINDDUN is, however, not an exclusive option. There is nothing preventing practitioners from swapping out LINDDUN in this application. This is one of the strengths of DISCREET as domain extension in the form of a model library, as ease of use is greatly advanced, opposed to a profile-based domain extension. Furthermore, there are also other sources of user requirements, including privacy related requirements, which are not addressed here. These relate to focus groups, trend analysis, and other metrics focused on consumers. Trying to address these would be counter productive but there is need for one clarification. Although such requirements will make up a considerable portion of the requirement set for any consumer IoT system, these requirements can not over ride audit and compliance requirements as this would result in non-compliance. We recommend that the same approach be taken with PbD requirements (or the mitigation suggested by LINDDUN) in that these should override requirements derived from non-regulatory sources, if there is a conflict.

## 7.8 Privacy by design trace method

LINDDUN is intended for iterative application across the system lifecycle [166], which is not only congruent with the aims of DISCREET but also enables our use thereof. Referring back to the DISCREET meta-model presented in Figure 7.3, we see that there is a direct *«use»* relationship between DISCREET methods and LINDDUN. This reflects the structured use of the LINDDUN methodology. However, as is the case with the compliance trace method, the initial impetus needed to set the system in motion comes from the inclusion of a specific requirement in the system model. In this case though it is two requirements where the first is the requirement to conduct privacy threat assessment throughout the system lifecycle using LINDDUN and the second, connected to the first by way of a *«deriveReqt»* relationship, is a requirement that the LINDDUN generated requirements or PETs should be appropriately introduced into the system model.

The flow therefore is as follows, the first requirement kicks off the threat analysis, the result of which is the generation of a threat tree and the proposal of one or more PETs. If more than one PET is proposed a trade study can be used to decide on the correct one to use. Once one PET is chosen, a new requirement is spawned directing the implementation of that PET into the system. Recasting LINDDUN mitigation strategies as requirements is also explicitly supported in LINDDUN [166]. But instead of the inclusion of a PET the goal is to prevent the privacy threat from occurring by re-framing it as a privacy requirement. The PbD trace method follows the steps outlined in Subsection 3.10.3 and is as follows:

- Derive a DFD diagram for the system at issue
- Type all items on the DFD as P, DF, DS, or E[41]
- Use the LINDDUN mapping template to map threat areas on to the DFD components as L, I, Nr, D, Di, U, or Nc[42]
- Determine if any portions of the LINDDUN threat tree are not applicable and create a system artefact to document these assumptions
- Refer mapped threats to the threat tree to locate detailed privacy threats as misuse cases

[41] Process, data flow, data store. or entity.

[42] Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance.

- Create model artefacts to document misuse cases
- Prioritise risks using preferred method[43]
- Determine if a concealing association or guarding association is at play[44]
- Consult LINDDUN mitigation strategy taxonomy based on the previous step
- Determine if the system is under design or the threat is realised
- For realised threats determine one or more PETs and run trade studies if a choice is to be made
- Derive a requirement to implement the selected PET
- Implement the PET and link it to the requirement above with a *«satisfy»* relationship
- For systems under design, work the results back to the privacy threats found and develop privacy requirements
- Link both the requirement above and the PET implementation requirement via *«trace»* relationships to the meta requirement used to initiate the PbD trace method

[43] Risk is seen as the function of likelihood and severity, but the exact method of calculation is left to the systems engineer.

[44] Is the exposure of data being prevented before the fact or managed after the fact?

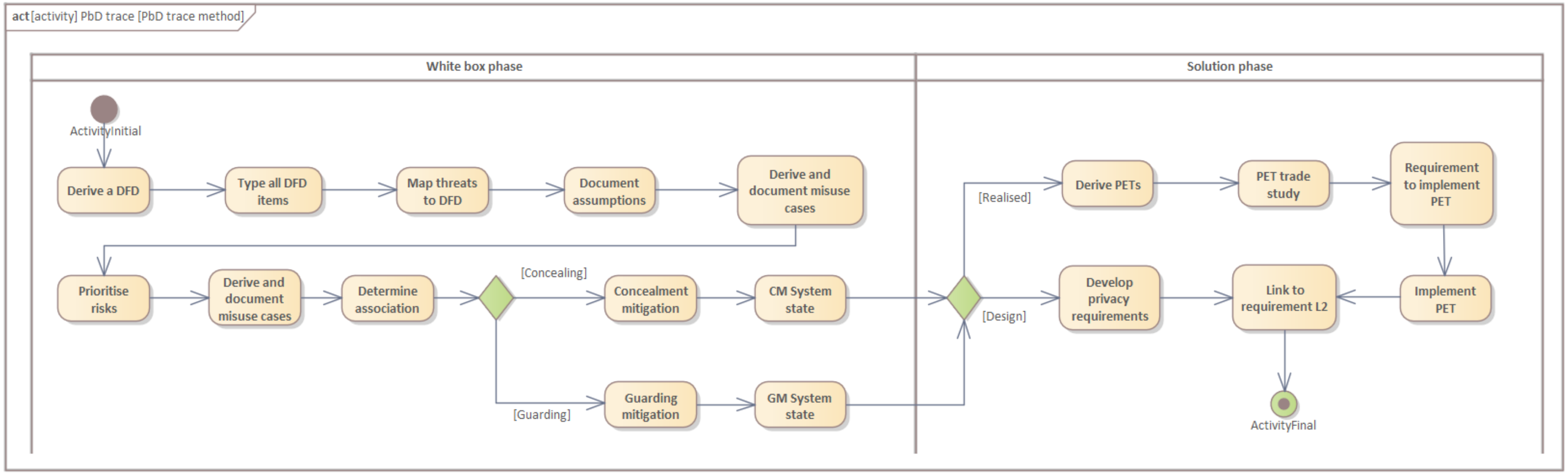



Figure 7.16: Privacy by Design trace method activity diagram

### 7.8.1 Privacy by design trace method viewpoint

As was the case with the governing method introduced in Section 7.2 and its related viewpoint shown in Figure 7.5, the systems engineer is the only stakeholder present for the PbD trace method viewpoint. Accordingly, Figure 7.17 includes a single view. However, the key points included under the viewpoint significantly diverge from those under the previous sections. This includes the use of LINDDUN threat modelling, the formulation of new requirements or the inclusion of PETs.

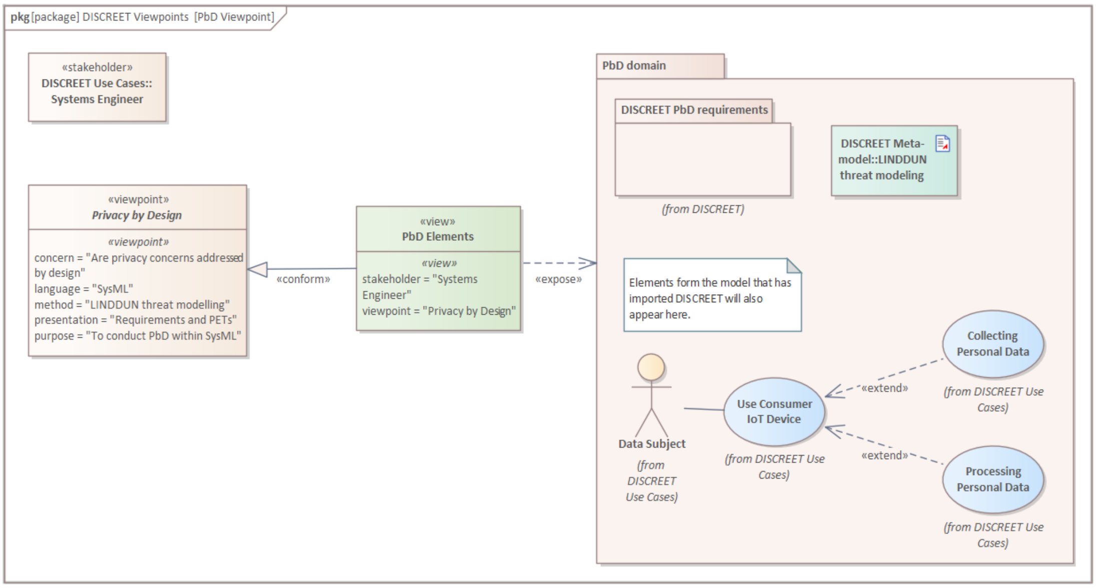



Figure 7.17: DISCREET systems PbD viewpoint

## 7.9 MGF-based execution of DISCREET

Lifecycle as a term is somewhat deceptive, and doubly so in the consumer IoT space, due to the fact that there most likely isn't a single lifecycle at issue but multiple different lifecycles each corresponding to a different entity within a given system [76]. Also, as previously mentioned, MFG does not itself account for lifecycle changes but is presented as a purely linear approach to systems design going from initial concept to first deployment. This however, is not the proverbial "*deal breaker*". Our choice for MFG is based on its ease of use and the light touch needed to integrate it with the rest of DISCREET. This latter point specifically means that it is easy enough for a user of DISCREET with specific requirements to use a different method. A further reason for not including a more formalised approach to lifecycle management in DISCREET is that, in addition to the changeable and heterogeneous nature of consumer IoT, there are also different approaches to lifecycle management. Consequently, these decisions must be left to the system engineer to make.

Since MGF describes building a system model from black box to solution generation, any changes to the model brought on by new requirements, external changes, changes in linked systems, and the like, can be mapped to MGF actions at one of its three phases. Accordingly, this change will have the potential to affect all other elements in its phase and subsequent phases. DISCREET itself purposefully initiates this by way of its governing method ending in a loop. The ability for DISCREET to be brought in at different stages of the system lifecycle is demonstrated in the following chapter by way of a case study. There, an existing device is getting enhanced features and DISCREET is used to manage compliance and PbD for these features.

# Chapter 8

# Case Study

## 8.1 Rationale and use

In this chapter and the next, we present a case study based analysis of DISCREET's functioning. Starting in this chapter with building the model, we import DISCREET into a subset of a third-party model as shown in Figure 8.1. Then in Chapter 9 we analyse the results and assess the functioning of DISCREET. This is not only an obvious avenue for testing a domain extension but is indeed also the path taken by Costa et al. in testing SysML4IoT. Although there are some differences, as is to be expected given the different natures of the two domain extensions, SysML4IoT was not only tested by way of a case study but also used a subset of the external model involved. Doing so limits the test scope to a manageable size whilst still having a test environment that is independent from our own work. The procedure for selecting and using this third-party model is explained below in Section 8.3.

Although testing such as that performed here is a reasonable conclusion to the work of developing a domain extension, the motivations for these tests should still be stated explicitly so as to guide the process and establish how the results should be interpreted. This is a two part problem, split between conducting the test and then assessing the results. Here we will conduct that test, but both this chapter and the next directly relate to the primary and sixth sub-research questions. Although Chapter 7 seemingly already provided an answer to both these research questions by developing the DISCREET domain extension, that answer still needs to be strengthened by way of functional testing. By this we mean, confirm that the domain extension can indeed be imported into another model and then used to alter that model as intended. This "as intended" qualifier implies testing both the domain extension in its own right and also as part of the case study it is imported into. Naturally the case study will therefore also have to be presented in SysML. For reference, both the primary research question and RQ6 are repeated below.

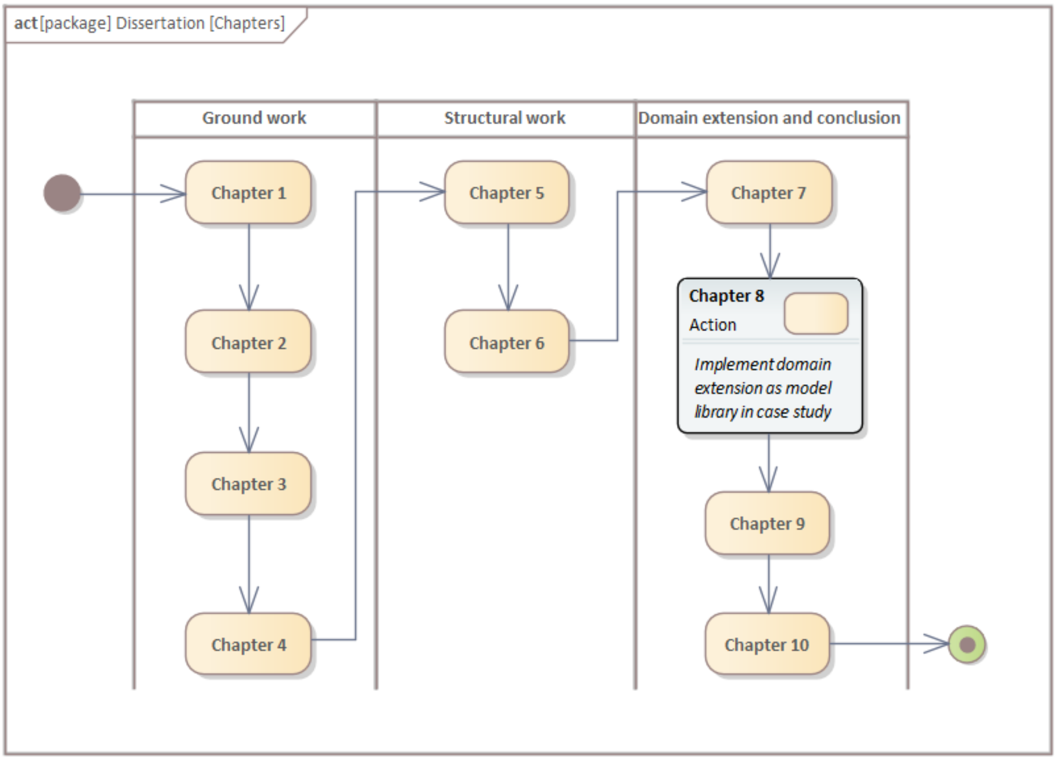



Figure 8.1: Chapter 8 research focus

The primary research question:

- Can a single SysML domain extension address both compliance and privacy by design in consumer IoT, and if so, what are its components?

Sub-research question six:

- What form could an approach to engineering compliance and privacy by design into IoT devices and systems take, if it were expressed in standardised systems engineering terms?

## 8.2 Case study choice

The case study is defined in Section 8.3, before getting started though, some ground rules must be established. This specifically relates to where to begin and how to proceed. In short, the DISCREET domain extension will not only be imported in the case study system model, but will also provide the starting position and sequence of events for modelling as we will be using each DISCREET method in sequence. However, two additional concepts must be noted. These are the **system of interest** and the **enabling system**. Where the system of interest is the system being designed and enabling systems are those interacting with the system of interest. This also implies that enabling systems relate to the system of interest as stakeholders [76].

Drawing in an extensive model with an overly complicated system of interest and many enabling systems could easily become so involved that clarity is lost and the quality of the analysis undermined. Further complicating these considerations was the need for a case study that would not only involve consumer IoT but also make some narrative sense with regards to the need for using DISCREET. It was therefore decided to find a case study that had the best narrative fit and then to use only a subset of its system of interest. This limiting of scope is aimed at producing a manageable but also fully functional example.

We considered the "Residential Security System" example used by Friedenthal et al.. As a case study that could be a good example of a consumer device now subsumed into the IoT sphere. However, we found the narrative link a little tenuous while the example itself was intended for use with OOSEM and therefore also not a good fit for our purposes, since we wish to test DISCREET as is. Next we considered the examples used by Hetherington in his introduction to SysML. Although the examples in that text are of high quality and already in the Sparx EA native format, they are unfortunately far too small and often simple to make for good test vehicles. The third option considered, and the one chosen, was the "Portable Audio Player" tutorial example provided by Sparx Systems itself. This too, can fit the bill as a consumer electronics device now networked and offering advance services.

## 8.3 Introducing the case study

Figure 8.2 shows the model index for the portable audio player and also lays out the intent Sparx Systems had with this case study. We will immediately discount the second part of this case study as we are not interested in UML integration or the generation of source code. Next, we considered the model elements and decided to focus on the system of interest as subset of the larger model. The system of interest subset model is shown in Figure 8.3 as model elements and presented on a bdd in Figure 8.4. Unfortunately this still provided too large an example. The scope was then further narrowed to focus exclusively on the introduction of a single new element to the system. By making this an "in process" addition the scope is sufficiently limited without hindering the opportunity to showcase different parts of DISCREET functioning as intended.

In Figure 8.4 we present the portable audio player device's system of interest block definition diagram (bdd). This is taken directly from the Sparx case study with two amendments. The first is changing compartment inclusion to not display all part and flow properties. This saves on space with no effect on the model. The second was to migrate the system of interest to SysML 1.5. This is the version of SysML worked for DISCREET and was also needed in a more general sense since the Sparx case study is in need of an update as it is provided in SysML 1.2. The update here was simple since we are only dealing with a single diagram and its elements, though migrating the entire case study would have been more involved[1].

[1] Although not needed here, Sparx does provide model migration advice which can be found at https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/migrate_sysml_1_1_model_to_sys.html

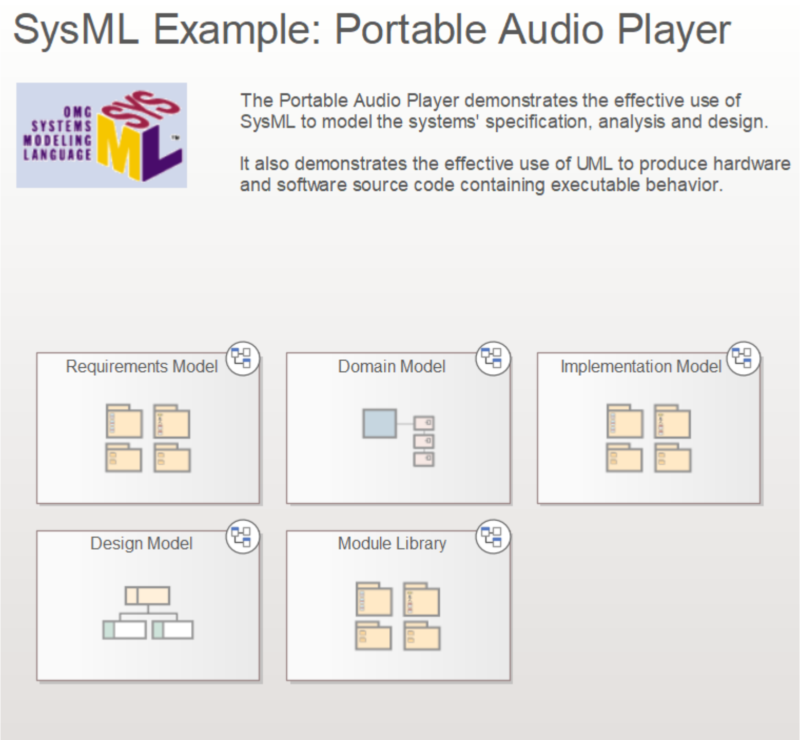



Figure 8.2: Portable audio player case study

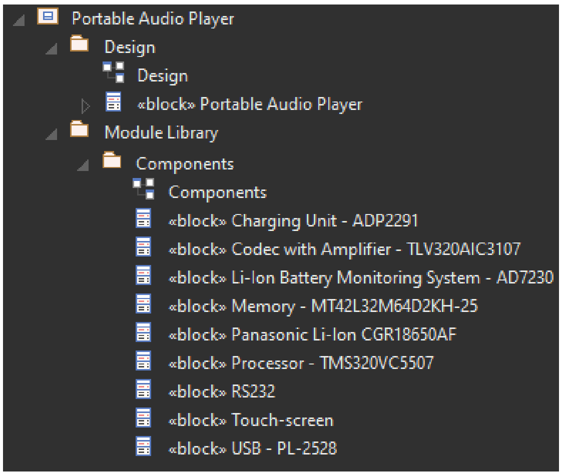



Figure 8.3: Portable audio player system of interest model subset

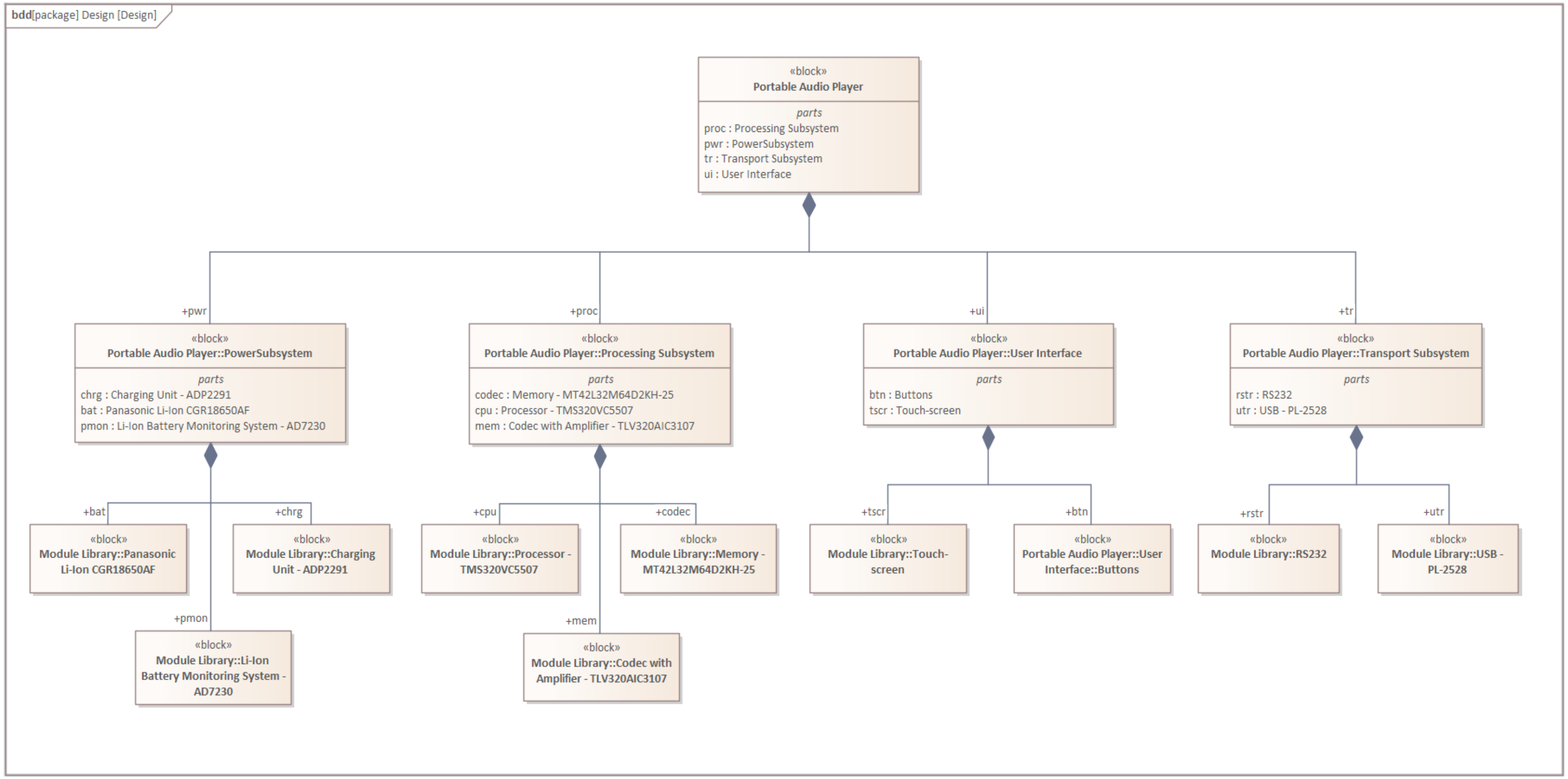



Figure 8.4: Portable audio player bdd

### 8.3.1 Case study scenario

The hypothetical scenario involving the Sparx portable music player example is as follows. A team of engineers have been tasked with taking an existing product and enhancing its functionality for a new edition. Specifically, an existing music player will be provided with new software and hardware to enable it to connect to the Internet and interface with streaming services such as Spotify. The manufacturer has decided that standalone music players can not survive without the extensive music library and recommender systems of a service such as Spotify. This decision was reached after reading relevant research [85] and assessing the capabilities of their current hardware. The company then tasked a systems engineer with the redesign effort.

After reviewing the hardware and new targets, the engineer decided to start by adding some existing data input software to the device. This software can be called via the existing UI and displayed on the device's touch screen. At present all it can do is have a user input some data and change that data. The intention is to expand this basic functionality to a fully integrated system for the user to manage not just their data but the entire system. At this point, the engineer realises that there is a potentially significant problem as their system is collecting personal data and providing it to third-party processors such as Spotify. To get ahead of the problem at this early stage, the engineer decides to import DISCREET into their system model and act accordingly.

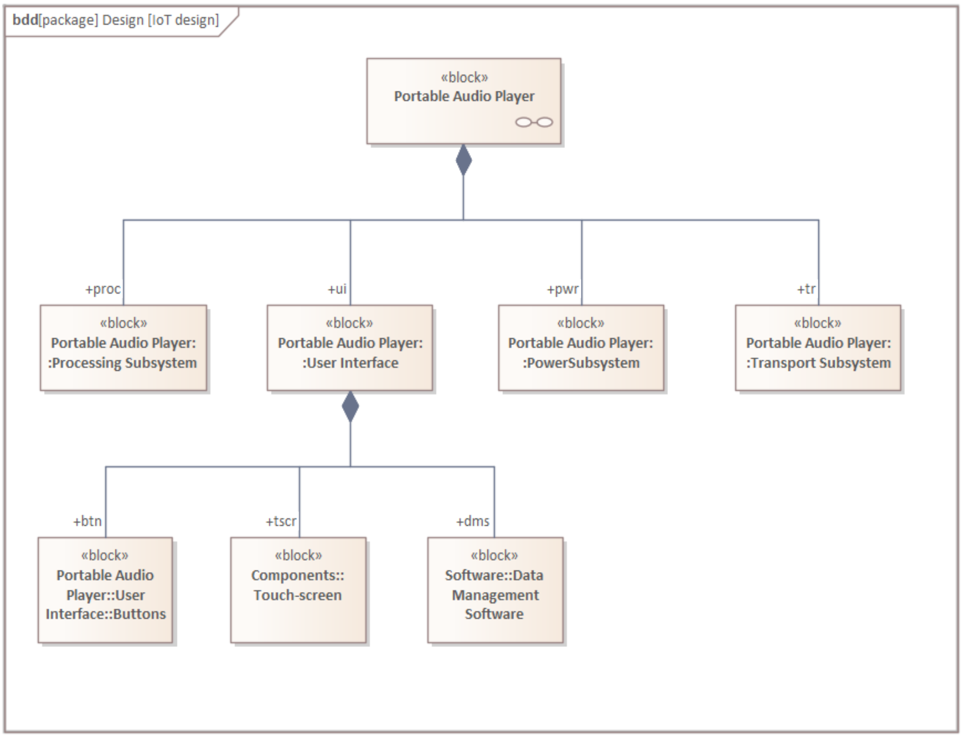



Figure 8.5: Portable IoT audio player bdd

In Figure 8.5, the updated system of interest is presented. The data management module, now deployed to the music player, still only has the basic front-end functionality of inputting data, including personal data such as that for account setup, and amending data. At the back-end though the software module is setup to pass that data on to the servers of third party processors via the POST and PUT methods, with PATCH and DELETE not implemented. It is at this stage that the engineer responsible decides to bring in DISCREET.

## 8.4 Model building

### 8.4.1 Governing method execution

The systems engineer reviews DISCREET and, following the process of the governing method[2], decides to import DISCREET as is, with components used as and when they become relevant to the system under development. At present, only the data management module deals with personal data and this will accordingly be the area of focus[3]. This process of drawing in DISCREET and choosing which portions to follow are all structured via the governing method. The governing method activity diagram, introduced in Section 7.2, is included below as Figure 8.6 and amended to highlight those portions of the method followed by the systems engineer in our case study. The full sequence followed by the systems engineer is as follows:

- Derive system model: This is already done as the engineer is working with an existing product, the system model at this level includes the system of interest as per Figure 8.5
- Determine which elements of DISCREET are needed: The engineer decides to follow the "isDISCREET" guard and use all of DISCREET in their system model
- Clarify if MGF or another methodology is used: They follow the "isMGF" guard and use MGF to structure their activities
- Import the related DISCREET meta requirements to the system model: Since all of DISCREET is used a full import is performed as shown in Figure 8.7
- Follow the DISCREET methods for each meta requirement: This is discussed in more detail in the following subsections
- Maintain and produce DISCREET outcomes: This is discussed in more detail in the following subsections
- Maintain continuous reference to DISCREET in the system model to allow for changes over the system's lifecycle: The system engineer's choice for the full import of DISCREET ensures that any future changes to the system will also have to pass the checks imposed by DISCREET

[2]All the specifics of each DISCREET method are not repeated here, but for ease of reference they are all listed in Appendix A. Some of the diagrams that are included here are also edited with parts faded to draw attention to the Case Study implementation of DISCREET.

[3]This is only the current case, but as the system model develops, more components that explicitly deal with personal data and the opportunity for data leakage will be introduced. The use of DISCREET would attend to both occurrences.

Next the systems engineer starts to follow the MBSE Grid Framework (MGF)[4], but since they are working with a pre-existing system, much of that system is in a state of full development. As such, their first task is to map the existing system to the MGF. A full and mature system model will populate all four pillars of the MGF. However, the targeted implementation presented here initially presents a complete solution model (the existing audio player) to which a new logical component is introduced (the data management software), where after DISCREET is imported.

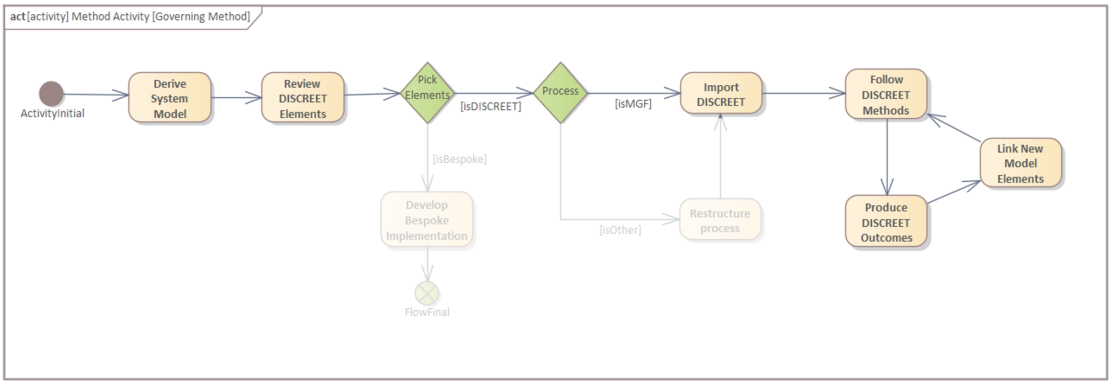



Figure 8.6: Governing method for the case study

In other words, the initial problem space for a portable audio player was defined and worked until an eventual solution could be developed. This places the audio player model in the solution phase, and since we are only viewing the bdd for the system of interest here, all components reside under the structure pillar[5]. The subsequent introduction of the data management software, sees this block placed under the structure pillar but at the white box level. This is because this new block addresses subsystem functionality and is not yet fully implemented into the audio player model. Therefore, as the data management software is developed from a logical subsystem to a realised subsystem, it will move down to the solution phase in the same pillar. As this move can not happen until the previously mentioned privacy and compliance failings are addressed, the new component remains in the white box phase for the time being, as is shown in Table 8.1.

As the systems engineer proceeds to the following methods, they can now be guided by the mapping for MGF which is already included in DISCREET. For ease of use, this is displayed on the activity diagrams for each method by way of black box, white box, and solution swim lanes. These are however, only shown if they are appropriate, which is why none of them are included for the governing method since it precedes other modelling activities. Similarly, the audit trace method does not include a solution phase swim lane since required solutions can only be developed once the compliance trace method is enacted.

[4]The MGF was introduced in Subsection 5.4.3.
[5]Please see Table 5.3 for a recap on the positioning of SysML model elements in the MGF.

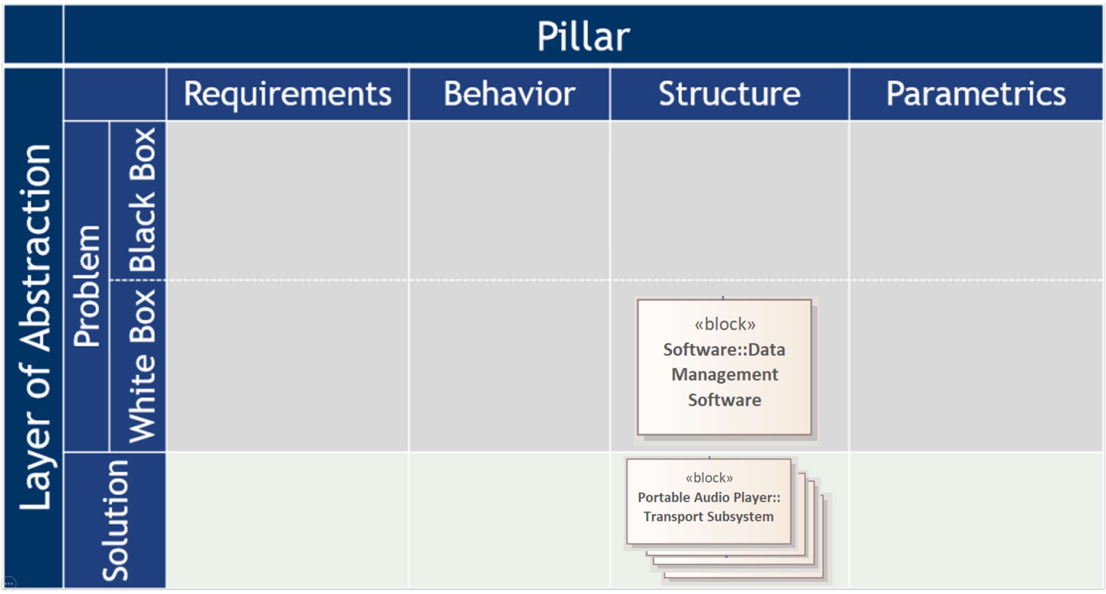



Table 8.1: MGF state after governing method execution

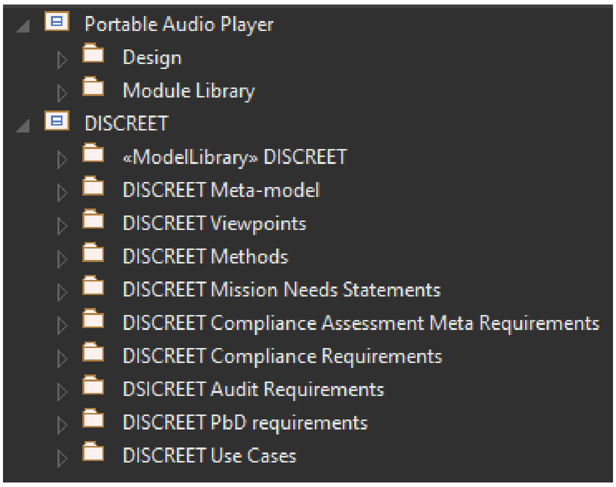



Figure 8.7: Case study subset with DISCREET imported

### 8.4.2 Audit trace method execution

Next the engineer follows the audit trace method and determines that since they will have to comply with the GDPR, a linked audit methodology is needed. Their local regulator does however not publish such. They therefore follow fork two of the audit trace method and select the CNIL's PIA as applicable adjacent methodology. The next steps would be to draw in the needed audit requirements and link them as appropriate, but since DISCREET is used in full, the CNIL requirements are already present. The full activity sequence for the audit trace method is described in Section 7.5, while the activity sequence for fork two is as follows and is also presented in Figure 8.8:

- Determine context: An existing audio player is being revised to include networking functionality to capitalise on the popularity of consumer IoT devices

- Determine if there is an official audit methodology: None publicly available
- Determine if there is an applicable adjacent audit methodology: Yes, that provided by the CNIL
- Include a requirement that all audit requirements should be drawn into the model as appropriately formulated “shall statements”: White box requirement inclusion
- Determine that all audit requirements are included and up to date: Those included with DISCREET are the newest available
- Determine that no new official audit methodology has been released: None released[6]
- Record all requirements to the requirements matrix for later use in the compliance trace method: White box inclusion

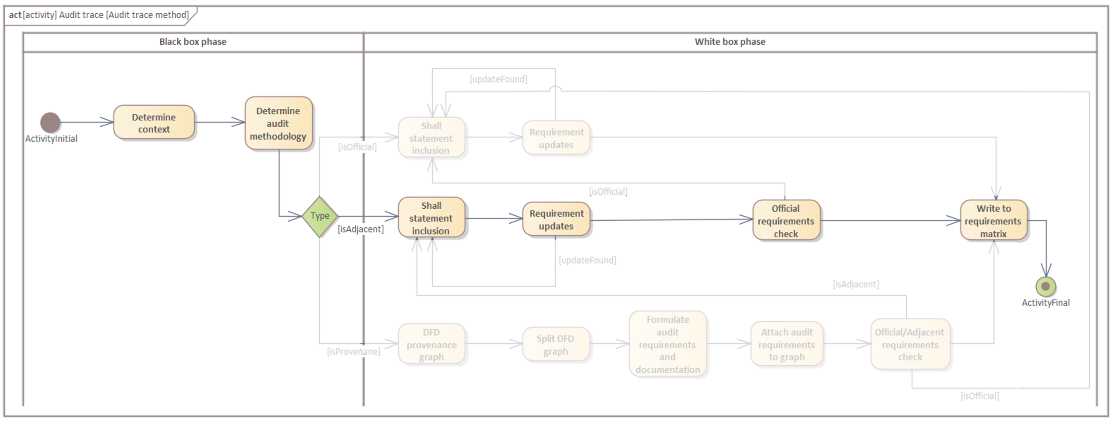



Figure 8.8: Audit trace method for the case study

With audit requirements now drawn in, as shown in Table 8.2[7] the systems engineer can start to assess the data management module and immediately sees a problem. From the audit requirements that could be linked, C2.2.4(I): Rectification of Data Assessment is a clear fit. It requires that the data controller, the company the engineer works for, must describe the controls deployed to allow data subjects to delete or change their data, including how third-party processors are informed of such changes. This is not done and will have to be attended to, however, no action is taken as yet since the compliance trace method must first be executed.

### 8.4.3 Compliance trace method execution

The engineer proceeds to the compliance trace method and finds that the initial steps are already taken care of. The applicable legislation is known, requirements formulated, and the audit trace

[6]This may seem a repetition of step two, but is intended as a recurring check given that DISCREET is applied across the system lifecycle. Although this activity diagram ends, the governing method does not and as such elements of all other methods are called throughout the system lifecycle.

[7]Only requirement ARNS is shown as all the other audit requirements are derived from it and there are 55 of them. The full list is available in Section B.4.

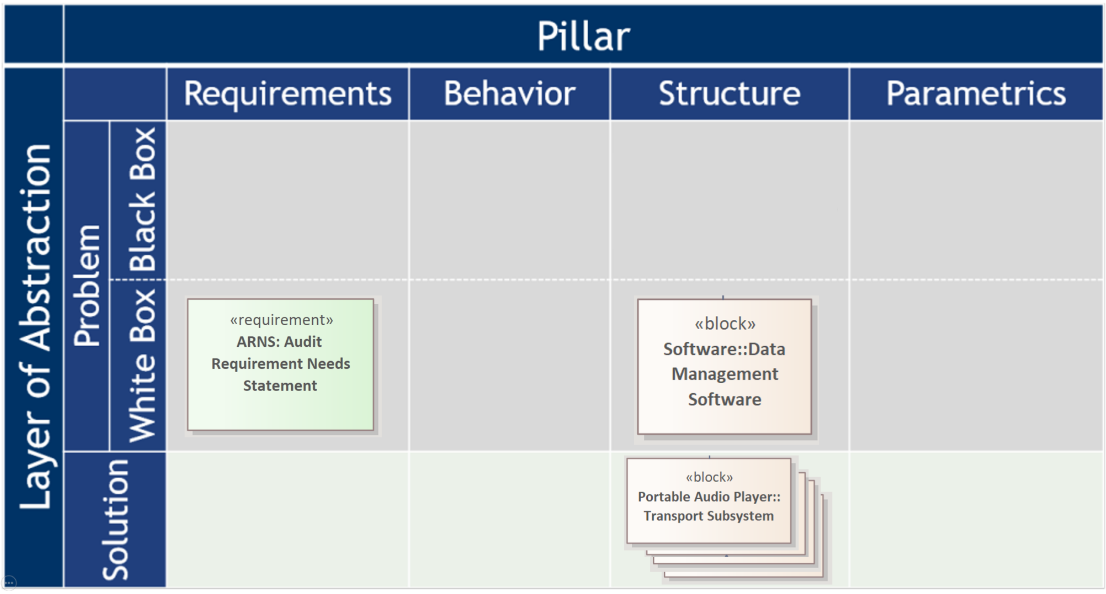



Table 8.2: MGF state after audit trace method execution

method completed. Similarly, the meta-requirements for document formulation are present and the system model is complete, though new additions must now be made to connect DISCREET requirements to model elements. The full compliance trace method is discussed in Section 7.6 wile the steps followed here are:

- Determine applicable legislation: The GDPR
- Cast the compliance requirements as "shall statements" and include in requirements database: DISCREET provides the applicable GDPR requirements[8]
- Complete the audit trace method: Already done
- Include meta requirements for reporting artefact generation:[9] Done
- Build out the system model in SysML: Done[10]
- Attach the applicable requirements to model elements using the *«satisfy»* relationship: Discussed below
- Generate reporting artefacts where requirements and model elements intersect or should intersect: Discussed below
- Link meta requirements to reporting artefacts using the *«realization»* relationship: Discussed below

[8] All requirements are traced to GDPR articles via the numbering scheme used and also includes cross referencing to linked audit requirements and in some cases, outside resources.

[9] Since DISCREET meta requirements are linked to stakeholder needs they slot into the black box phase, while the requirements and reporting artefacts produced are in the white box phase.

[10] For the case study elements.

- Link compliance requirements, audit requirements, and system components to the newly generated reporting artefact using the *«trace»* relationship: Discussed below

After the inclusion of DISCREET compliance requirements the MGF state has an additional 57 requirements added to the white box phase, here represented by their common top level requirement, CRNS. Table 8.3 presents this state, while Section B.3 lists all the requirements. Also shown is the inclusion of meta requirements in the black box phase. As the systems engineer now proceeds to link requirements, meta requirements and other system components, reporting artefacts are produced and system components altered. These changes are reflected in the white box and solution phases. However, at this stage the first issues with the data management software are only located and the software itself can not yet be changed, which is why the logical component in the white box phase can not yet be replaced by an implemented component in the solution phase.

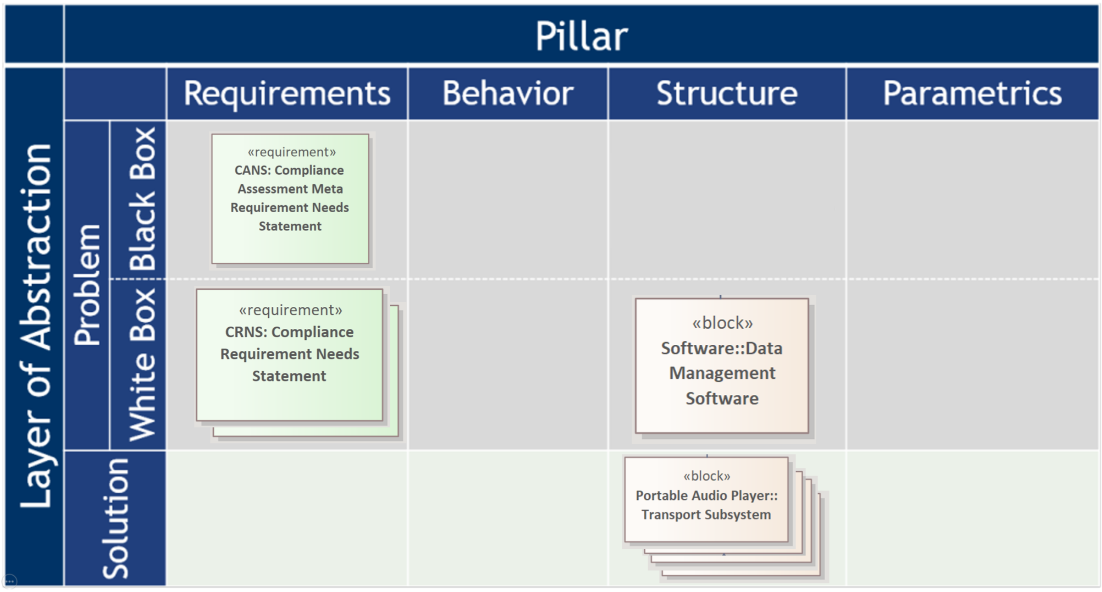



Table 8.3: MGF state after compliance trace method requirement inclusion

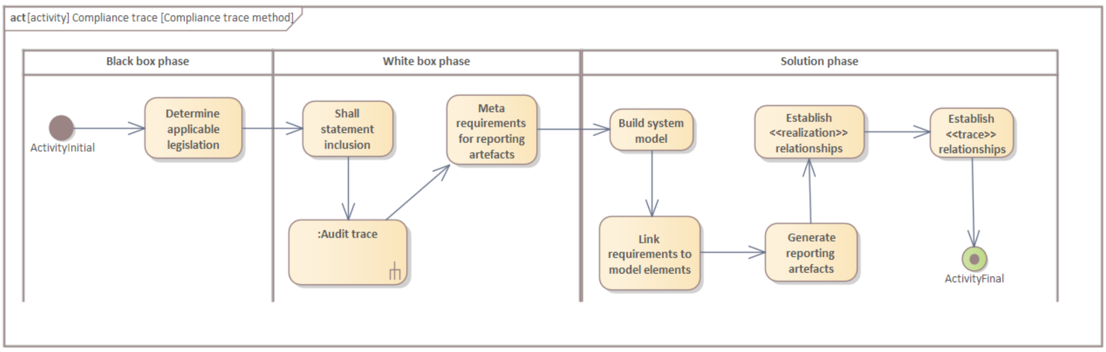



Figure 8.9: Compliance trace method for the case study

Acting on the DISCREET meta requirements and related compliance and audit requirements, the systems engineer creates two new diagrams to cover the two different issues relating to personal data, namely changing data and erasing data. Since this is simple data changes and deletion, the relevant audit requirements are C2.2.4(I)[11] and C2.2.4(IV)[12]. The notes section for each requirement indicates the relevant compliance requirements which are, respectively, G16[13] and G17.1.b[14]. Here the engineer finds that G16: Data Rectification is currently in compliance since the data subject is able to edit their local data freely and can also change off-site data. However, data deletion only holds for local data with the data subject not able to delete all off-site data held on them. This is because requirement G17.1.b has not yet been implemented with respect to the data management software. As a result audit requirement C2.2.4(IV) is in a state of non-compliance and a Class 3 report will be generated.

### 8.4.4 Meta requirements met

While executing the last steps of the compliance trace method the engineer creates a new diagram entitled Compliance Outcome G16 dms[15], as shown in Figure 8.10. Here we see that both the audit requirement C2.2.4(I) and the compliance requirement G16 are met. As a result this satisfies meta requirement Class1 and generates a related compliance report. This report is entitled "C2.2.4(I)-G16: DMS Class1 Report" and linked by a trace relationship to all other elements other than the Class1 requirement since it realises that requirement's specifications. The report contains any free form information the engineer might deem relevant, the information of the system elements linked to and finally the compliance information itself which is filled in using form "2.2.4 Exercising the rights to rectification and erasure" from the CNIL PIA knowledge base addendum "Application to IoT Devices".

Thereafter the engineer creates a second diagram entitled Compliance Outcome G17.1.b dms. Although G17.1.b does not appear on the diagram itself, the title is used since the main point here is the non-compliance to G17.1.b. For the elements that are present the connections are made as before but this time the dms to C2.2.4(IV) link is a *«trace»* which reflects the fact that G17.1.b is not satisfied in terms of C2.2.4(IV) though there is a link which is needed for reporting purposes. Doing so allows for *«trace»* relationships to be used for reporting in matrix views. The engineer then produces a report as before selecting the elements and the CNIL form needed. However, a mitigation for the non-compliance must now be included. This mitigation can be formulated now or drawn from the results of the PbD method, which comes next. With regards to matrix views, Figure 8.12 shows the utility of using these with a somewhat unorthodox link. The *«trace»* relationship between a model element and audit requirement sticks out like the proverbial sore thumb. A similar approach can be used to locate links between Class 1, 2, and 4 meta requirements and the reports documenting them. A further point on reporting is that quick matrix reports like that presented here are helpful for locating specific issues, but reporting to

[11] Rectification of Data Assessment
[12] Data Erasure Assessment for Consent Withdrawal
[13] Data Rectification
[14] Erasure After Withdrawal of Consent
[15] The compliance outcome in relation to G16: Data Rectification for the data management software (dms).

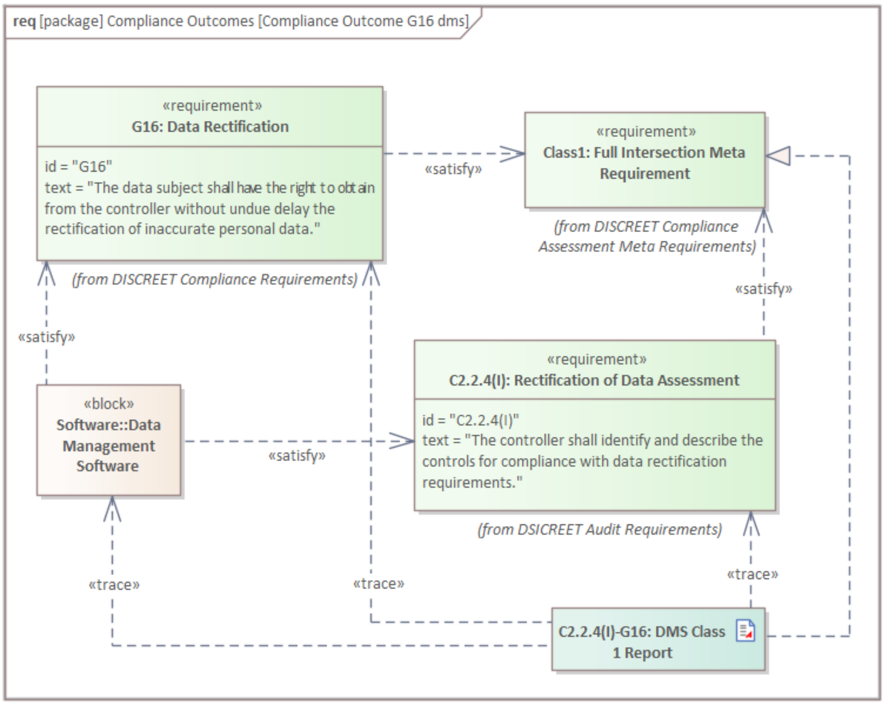



Figure 8.10: First compliance outcome for the dms

stakeholders would benefit from exporting the tool generated matrix to something more capable such as Excel, which was the case with Figure 7.14.

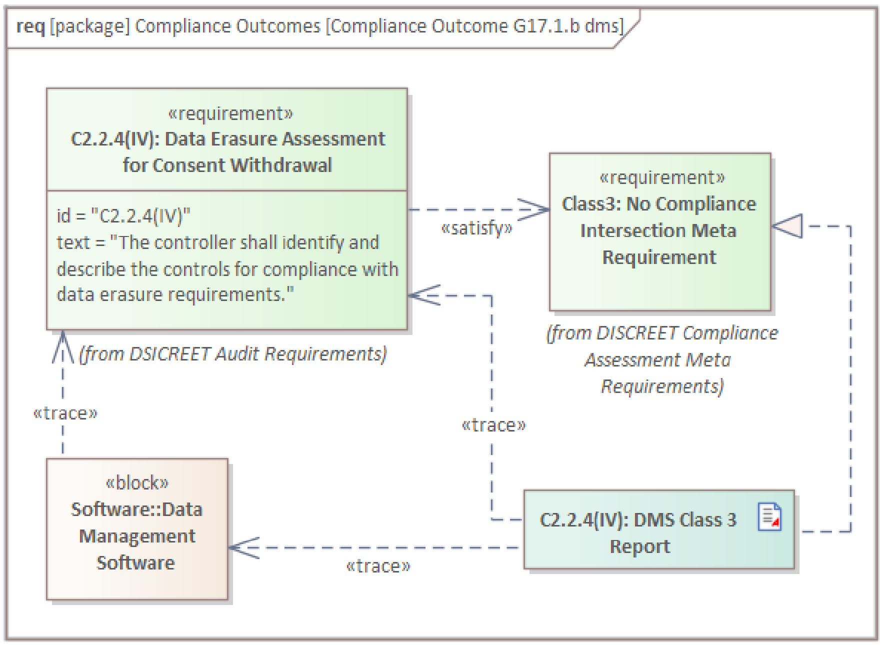



Figure 8.11: Second compliance outcome for the dms

The new reporting artefacts and the the requirement diagrams they are included in, now reside under the solutions phase for the requirements pillar on the MGF. This is shown in Table 8.4, though only the reporting artefacts themselves are shown to reduce clutter.

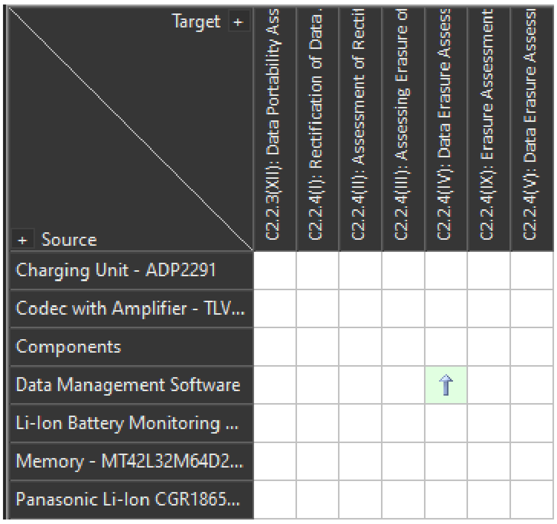



Figure 8.12: Audit issues matrix for the dms

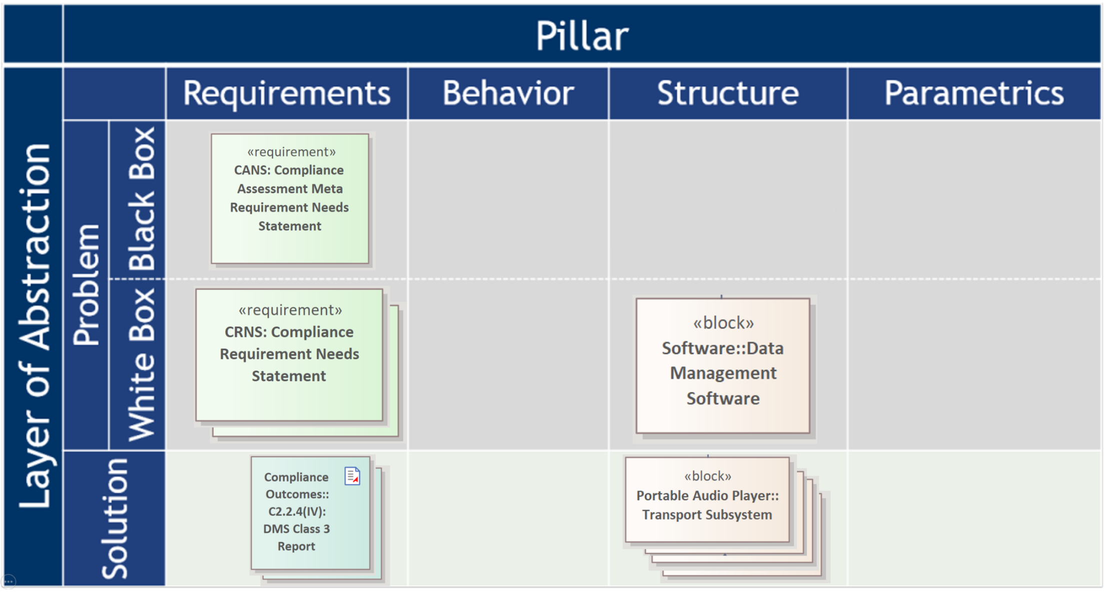



Table 8.4: MGF state after compliance trace method application

### 8.4.5 Privacy trace method execution

After completing all the steps relating to the first three methods the systems engineer now turns to the fourth and final method, which governs privacy by design by way of the LINDDUN threat modelling methodology. Following the steps of the privacy trace method, as introduced in Section 7.8, the engineer is able to clearly identify and describe the privacy threat at issue and also develop additional requirements. These steps are outlined in Figure 8.13, which shows the portion of the privacy trace method deployed in the case study.

The systems engineer starts by determining the DFD, typing all elements and determining that the problem here is located on the data flow (DF) between the device and the third-party. To recap, this is due to the data subject's ability to edit and delete local data about themselves, but

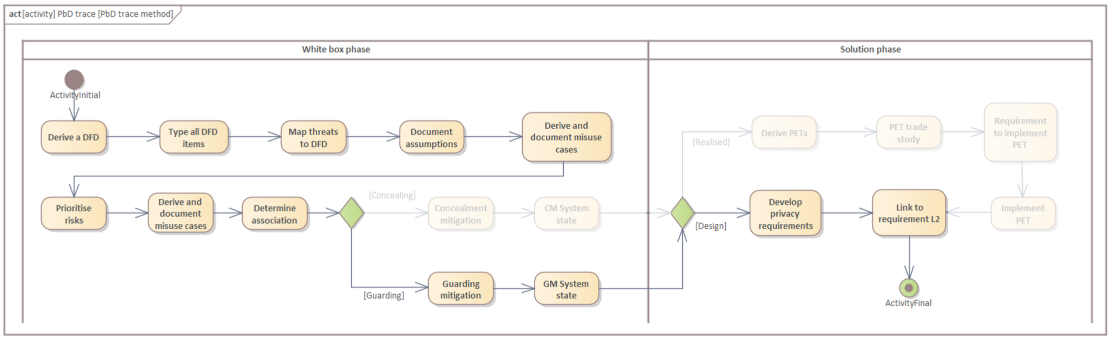



Figure 8.13: Privacy trace method as applied to the case study

not that stored in the cloud. This is presented on the system DFD in Figure 8.14 and shows that the data subject is able to interact with the data management software, though no analytics or other returns are offered yet. By using this software on the audio player, the data subject can view and change any data about themselves as long as it is stored on the audio player itself. For third-party services, there is a similar flow of data between the third-party service and its own data store, but the data subject is not able to access that data in the same manner. This is due to the limited functionality of the data management software on the audio player, which in its current state only allows for the POST and PUT methods to be used, while empty data fields are also not allowed. As a result, the ability of data subjects to manage their own data is severely limited and clearly not in compliance with GDPR requirements.

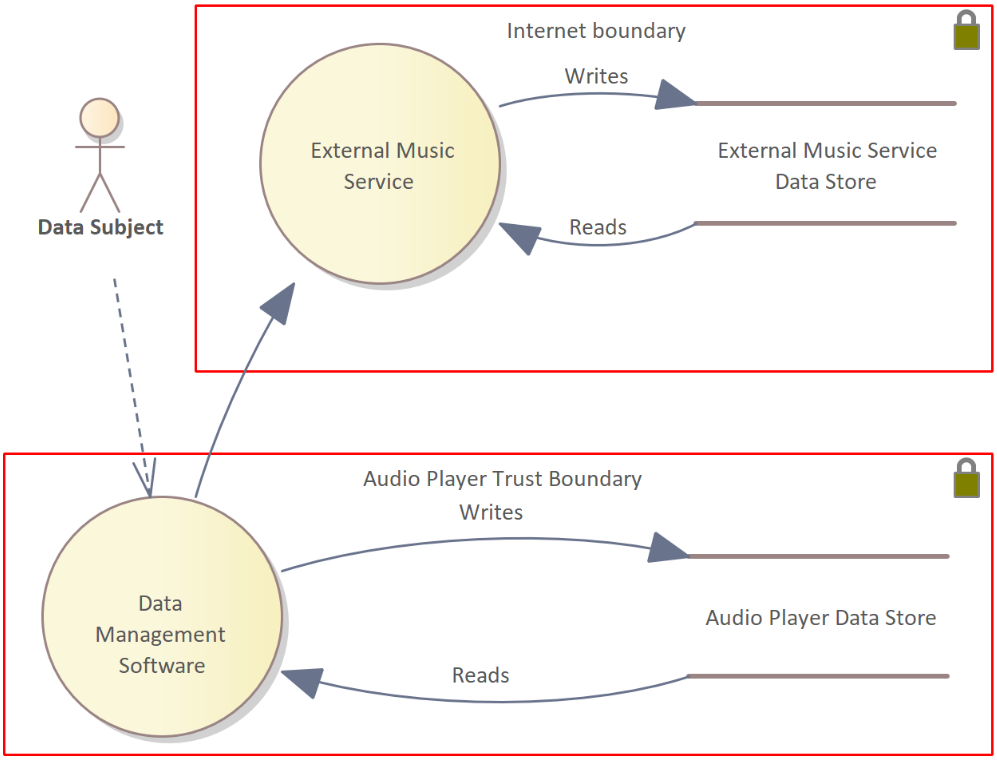



Figure 8.14: DFD for the proposed IoT audio player at present state of development

It is further determined that the threat type is that of non-compliance (Nc). The engineer can now map the threat to the LINDDUN threat tree, as per Figure 8.15, and determines it to be "Nc_2 Incorrect or insufficient privacy policies". No prioritisation of threats is needed thereafter as there is only the one at present and documenting the threat was already performed under the previous method, though the engineer is free to add the LINDDUN outcomes to those reporting artefacts. Finally, the threat is found to link to a guarding association since, although the threat isn't realised due to the system being under development, the threat could only occur if data is shared with a third party. From there it is found that new system requirements can be generated with the following proposed:

- LR1: Third-party data provision; The portable audio player shall only provide data to third parties if the API provided by those parties supports, at a minimum, the amendment and deletion of data directly by the data subject.
- LR2: Third-party data management; The portable audio player shall implement all data deletion and amendment capabilities relating to third-party services.

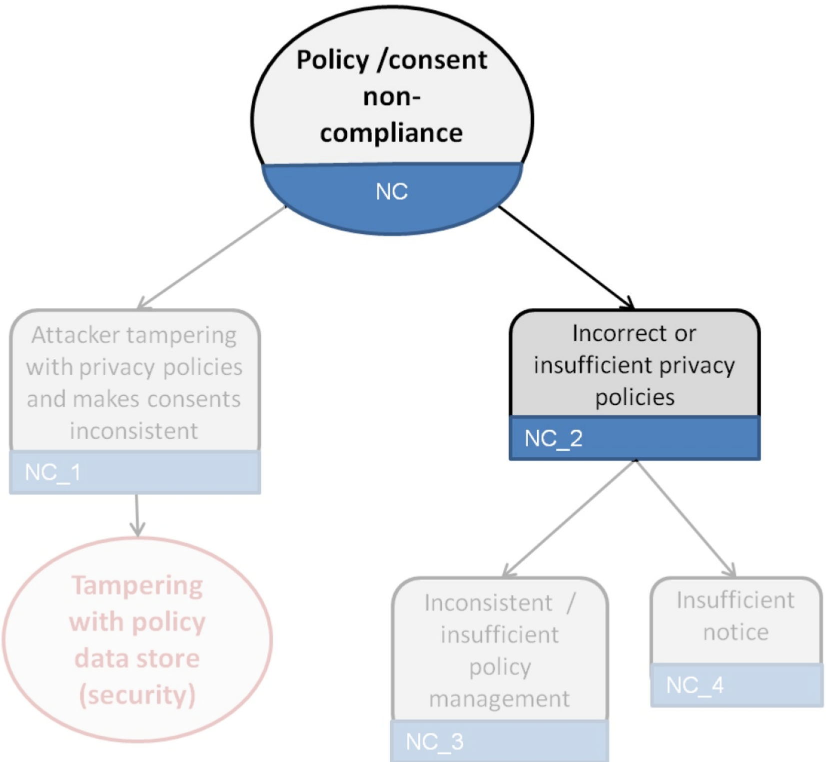



Figure 8.15: Case study non-compliance threat tree

The newly generated requirements are then linked to DISCREET requirement L2[16] and included in the solution phase of the MGF. However, if the threat had already been realised then LINDDUN would have guided the engineer to its solutions mapping which, for compliance, suggests

[16]L2: LINDDUN Results Implementation. Results from the LINDDUN threat modelling methodology, including requirement formulation and PET recommendations, shall be imported into the system model as appropriate.

a number of PETs including IBM's Enterprise Privacy Authorisation Language[17]. Again, the requirement option to head off threats before they are realised is the cheaper and less cumbersome outcome.

With DISCREET now integrated into the model and the previous issues with the dms addressed, the engineer can continue with their project. Since they now have access to the compliance and audit requirements before the fact, they can test new systems as development proceeds according to each of the methods. They are also free to use the fourth method, PbD, in a preemptive manner.

Returning to the MGF, the systems engineer can now use it to guide their actions, specifically with regards to when and in which order actions are to be taken. To recap, this works from left to right and top to bottom, with newly added elements slotting in at their top most spot. This also links to DISCREET's governing method which purposefully ends in a loop and thereby ensures use across the system lifecycle even though the MGF itself is used in a linear fashion.

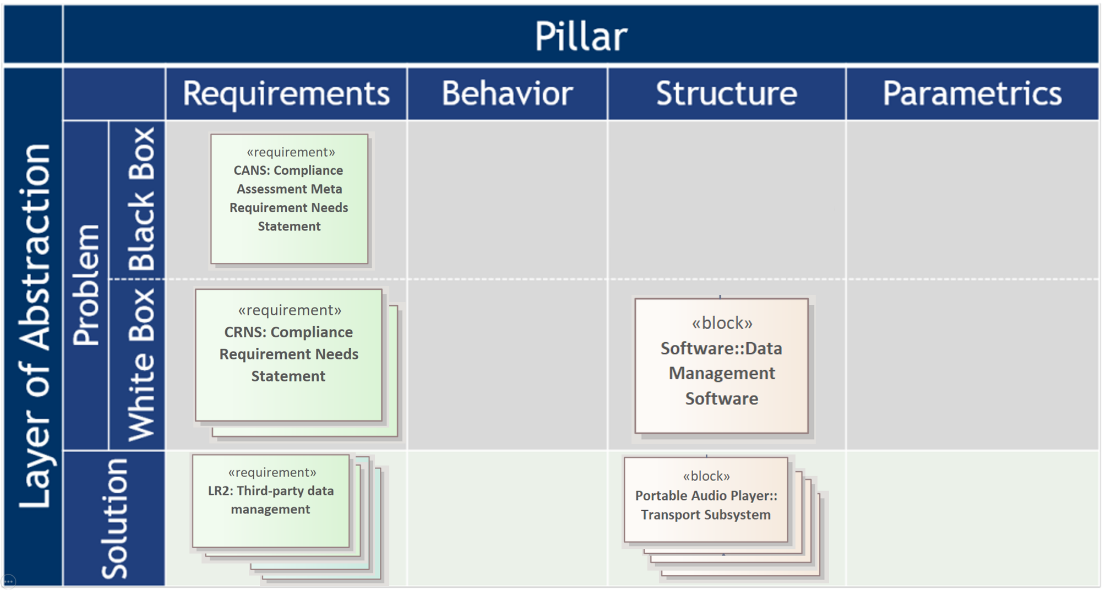



Table 8.5: MGF state after privacy trace method application

It is also important to reaffirm that in building a full system model, the use of the MGF will play a much greater role than in the case study discussed here. That is because the case study starts with an existing system model, only covers the system of interest for that model, and DISCREET already plots its functioning to the MGF. When creating a system model from scratch then, the MGF will function as primary guide for the system engineer's efforts, whilst the DISCREET mapping ensures that process related conflicts are avoided.

The system engineer can now proceed to change, or request changes to, the data management software in accordance with DISCREET requirements. This may involve the use of parametric diagrams to describe the new functionality and guide any needed testing and will ultimately lead to the implementation of the updated software as a new subsystem in the solution phase of the

[17] More information at https://www.w3.org/2003/p3p-ws/pp/ibm3.html

MGF. At that point the engineer will be able to proceed with further development of the IoT audio player, starting with black box requirements. As this process must therefore also touch on the already included DISCREET requirements, these will activate again and ensure that any new components are GDPR compliant and that PbD is continually assessed. This also implies the continued use of DFDs. The first of these future DFDs though, would be the redrawing of the data flows related to the data management software once it has moved to the solution phase on the MGF.

## 8.5 Moving forward

The case study presented in this chapter purposefully focused on a single issue as this allowed for DISCREET to be tested in a constrained manner[18]. Doing so also allowed for DISCREET to be imported and linked to an independent system model. As a result, we are in a position to not only test DISCREET on its own, but also test its functioning in connection with the external model. Such testing includes the measure to which the previously determined viewpoints and their requirements are met, traceability, and a tool specific test for language adherence. These tests are presented in the following chapter.

[18]A full implementation would have generated hundreds of pages worth of reports alone.

# Chapter 9

# Verification and Validation

## 9.1 Aims

The primary tasks of the systems engineer include developing the specifications for designing, validating and verifying all aspects of a system. Using SysML the engineer has access to clearly defined system tolerances, measurements and other parametrics [139]. However, it should be remembered that MBSE provides an abstraction of the system under design, in the form of a system model. This model, being an abstraction is not a one to one copy of the system, but instead contains only the elements needed to successfully realise the system [76]. Accordingly, both verification and validation are focused on successful outcomes for the system as modelled, but in different areas. Where verification is concerned with demonstrating that the system works, validation is focused on determining if the system works as intended. Returning to the abridged definitions provided by Boehm this can be formulated as:

- **Verification** “Am I building the product right?”
- **Validation** “Am I building the right product?”

Broadly speaking then, for DISCREET the question of verification is if it can demonstrate internal coherence, if its elements are well formed, and if it is able to maintain these traits even after being imported into other models. Validation on the other hand will focus on the extent to which DISCREET addresses compliance, auditing, and privacy concerns accurately and fully, that is to say, does it do what it set out to do. DISCREET is, however, not a fully fledged system but a model library and accordingly can not fully demonstrate the above without application to a system model. That was the task of the previous chapter, while this chapter will analyse those results, as shown in Figure 9.1.

There is typically a test case, with set performance metrics involved in system testing [76]. However, the testing presented here is focused on DISCREET and as such does not involve the type

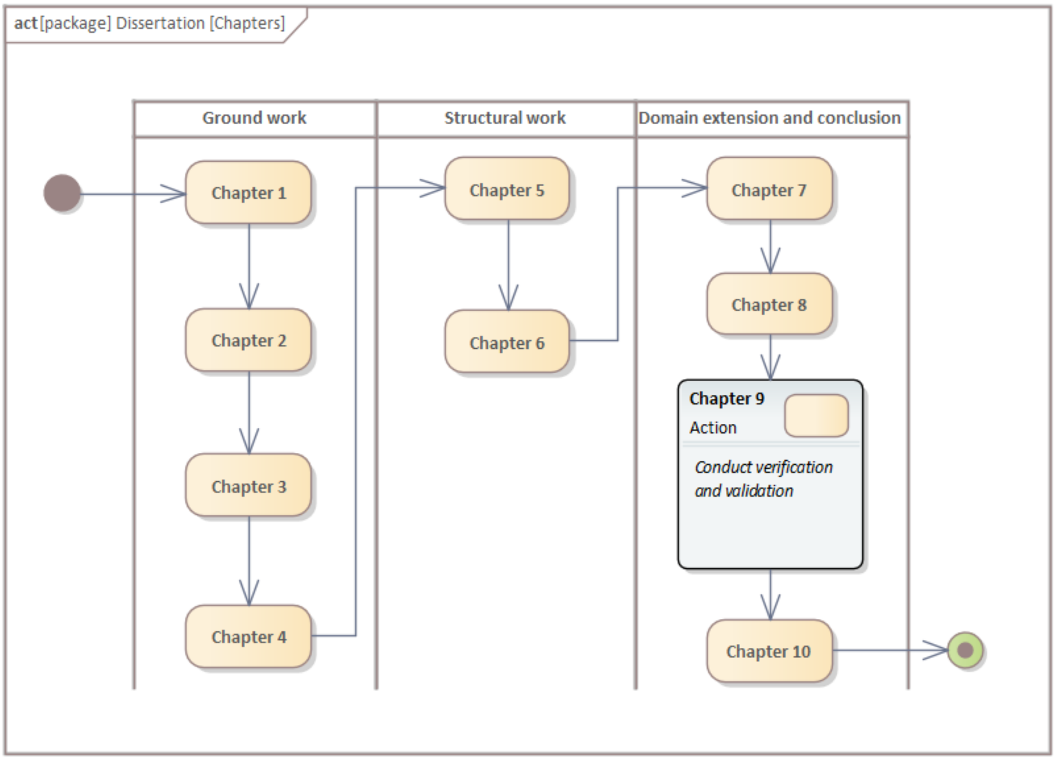
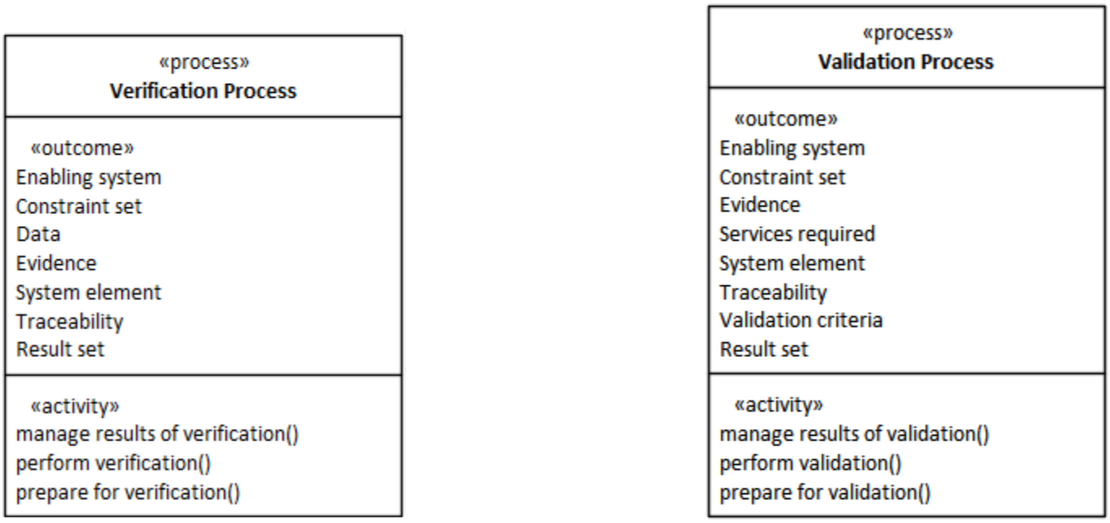



Figure 9.1: Chapter 9 research focus

of metrics that might be associated with requirements around engine performance, power use, up time, or a myriad of other system dependent engineering requirements. This purposeful omission is aimed at DISCREET only and not at the case study used in the previous chapter. In a real-world scenario then, the systems engineer would perform a full set of model-driven and defined tests for both validation and verification across the model lifecycle. In the context of testing DISCREET though, doing so would be counterproductive and involve a lot of conjecture. Instead, when conducting verification and validation we will endeavour to follow best practice, which in this case comes in the form of ISO 15288. Figure 9.2 presents the activities and outcomes for each as process blocks taken from [76]. Verification is problematic though due to the stakeholders involved being archetypes of potential users. This is further discussed in Section 9.3.

Casting our verification activities in the light of Figure 9.2 we can lay out the following project. To verify our work we will use one internal (model analysis) and one external (traceability) tool, both introduced below. This constitutes the «activity» portion of the process, where the use of each tool is doubled. Meaning each is applied to DISCREET on its own and then to the combination of DISCREET and the case study model. The «outcome» of this process then will be evidence that DISCREET is a well formed SysML model library, can be imported successfully into other models and allows for its elements to connect to the elements of those other models and act upon them as directed by each of DISCREET's methods. The latter point also addresses the issue of traceability.




(a) The verification process

(b) The validation process

Figure 9.2: Verification and validation best practice
([76])

## 9.2 Verification

### 9.2.1 Model analysis options

In the practice of design science there are two often used avenues for assessing MBSE models [161]. The first is model analysis by way of an external ontology[1] and the second is a case study approach. Using a standalone ontology is especially useful in testing new domain specific languages or domain extensions extending an existing language via new or altered stereotypes. However, for a domain extension of the type «modelLibrary» within SysML, we can fall back on SysML's own formalism to check for coherency and as such make a meaningful statement regarding verification[2]. This still leaves validation to be addressed though, for which we have used a case study. However, due to DISCREET needing to not only function in its own right but also once imported into another model, model assessment is also conducted by way of the case study, with this work done in Sparx Enterprise Architect.

Sparx Enterprise Architect comes as standard with a "Model Validation" tool which can be launched as part of the system output window. In Figure 9.3 we see a snapshot of this tool running and producing a long list of warnings. These are "unrealised" warnings related to each of the DISCREET requirements which is unavoidable since DISCREET is a model library and not a fully fledged system model[3]. Given the nature of these warnings it would be safe to ignore them and still use the tool, as we do in Subsection 9.2.2. A further point of note is that although

[1] Not to be confused with the ontology or ontologies included in a model and typically drawn in via viewpoints.

[2] The first option is useful for domain specific languages since it can be used to confirm language coherence. This is not useful in the case of a model library though since those language checks would be running against SysML and not DISCREET.

[3] The requirements will only ever have a «deriveReqt» relationship until DISCREET is imported into a system model and used.

Sparx has chosen to call this a “Model Validation” tool, what it is actually doing is model analysis in terms of verification. That is to say, the tool determines if the model is built correctly.

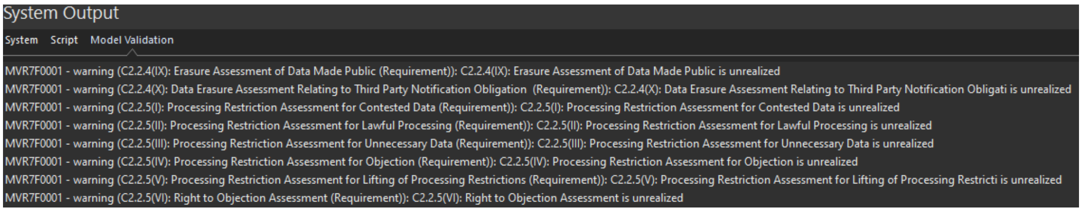



Figure 9.3: Model verification in Sparx EA

In addition to the warnings generated in Figure 9.3 the Sparx EA tool also currently, and in all builds available at time of writing, incorrectly identifies all «includes» and «extends» relationships for use cases as erroneous. The issue has been raised with Sparx and a bug report entered but a resolution is not expected imminently. Thankfully this has a minimal impact on the work conducted here and the related test can be disabled.

In addition to the Sparx tool we will also use the Model Expert “Snapshot meta-model” to specifically assess traceability. The tool creates an abstraction of all model elements by type with all explicit links between each type. This is often used by consultants when assessing modelling done by clients as it not only provides a high level model overview but immediately shows any missing traces.

### 9.2.2 Model analysis

The Sparx EA tool performs checks (verification) against the metrics listed below. Since each of these are optional, and given the aforementioned issues with requirements, we have deselected requirement checking as an option[4] and then ran the validation check against all remaining options. As shown in Figure 9.4, DISCREET passes all checks with no errors and no warnings.

Sparx EA model analysis metrics:

- Element: Well-Formedness
- Element: Composition
- Element: Property Validity
- Element: OCL Conformance
- Relationship: Well-Formedness
- Relationship: Property Validity
- Relationship: OCL Conformance

[4] After confirming no true errors were present.

- Feature: Well-Formedness
- Feature: Property Validity
- Feature: OCL Conformance
- Diagram: Well-Formedness
- Requirements: Management
- Systems Modelling Language (SysML) Rules

The primary areas checked then are well-formedness, composition, property validity and conformance. Well-formedness rules determine if an element, relationship, feature or diagram is well-formed. For instance, checking if all the elements on a diagram are valid. Composition rules are used to assess the containment of children for each element, with regards to validity, number of, and missing children. Property validity, on the other hand, refers to checks gauging if elements possess their defined properties and if any values are incorrect. The last of the four main checks is that for conformance to the Object Constraint Language (OCL), which is checked with regards to elements, relationships and features. Two additional checks are also included, namely requirements management and SysML rules check. The latter is needed since the main checks run against UML rules. Since SysML is a profile of UML, the UML checks are functional but do not cover SysML specific requirements, hence the inclusion of a SysML check. The remaining check is that for requirements management which ensures that all requirements are realised. Though this can be very useful in a fully developed system model, it is problematic for a model library as already discussed in Subsection 9.2.1.


Figure 9.4: DISCREET model verification with requirements deselected

After running these checks on DISCREET, the next step was to run the tool against the case study on its own. This was done in the first instance by targeting the full case study instead of just the subsection used to assess DISCREET. In Figure 9.5 we present the rather disastrous results of this exercise. We also confirmed that this was not due to error on our side by running the model analysis directly upon opening the case study for the first time. A closer look revealed that although the individual elements of the model are listed as being of the stereotype "SysML 1.5", the actual model was created in SysML 1.2 and the author recommends setting Sparx EA to use 1.2. Alternatively there is the conversion script provided by Sparx but neither of these options appear all that promising. It is for this reason, and given our need for only a subset of the case study, that we opted for recreating that subset from scratch in SysML 1.5 to ensure compatibility with DISCREET for testing purposes. As shown in the DISCREET meta-model in Figure 7.3, DISCREET is written in SysML 1.5, though this does not imply incompatibility with older versions of the language. The case study could have been used without recreating it, and would have served perfectly fine if the ability to execute the model was the only test being performed. However, since the model analysis used specifically assesses full adherence to up to date language standards, we have to either abandon the model analysis or update the case study and opted for the latter.

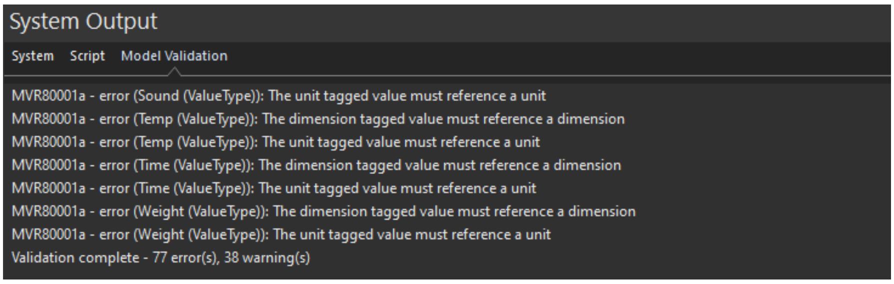



Figure 9.5: Full case study errors and warnings

After importing DISCREET into the updated case study we reran the same verification test using the built in tool and found that all tests were successfully passed. This is shown in Figure 9.6 and confirms that DISCREET is well-formed, adheres to language rules, and maintains its integrity even after being imported in to the case study model.

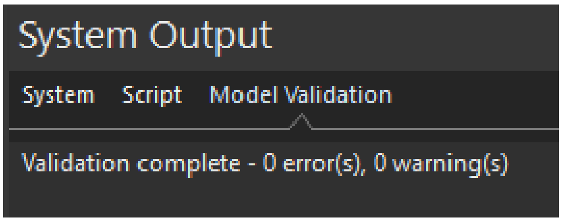



Figure 9.6: Case study verification with DISCREET imported

### 9.2.3 Traceability checking

Traceability is an essential component to a well formed model, not only in the general sense of having a connected and functioning model but also specifically in ensuring that both system requirements and stakeholder requirements are met [68]. This includes requirements drawn into a model but not applied at present. There must be a higher order requirement justifying their inclusion, which either directly or through its derivations, link to other model elements. The easiest way to establish such linking is via a graphical meta-model representation showing no unexplained unconnected elements. In Figure 9.7 we see Model Expert's "Snapshot meta-model tool" performing such an analysis with the results in Figure 9.8.

Since DISCREET is in the first instance a domain extension and model library and not a fully fledged system model, it should be expected that some of the elements it contains would cause problems for this type of analysis. Specifically, change elements which are not SysML elements and example artefacts contained in model views. With this accounted for, DISCREET still contains three distinct and separated sets of elements. The first and largest is the directly functional

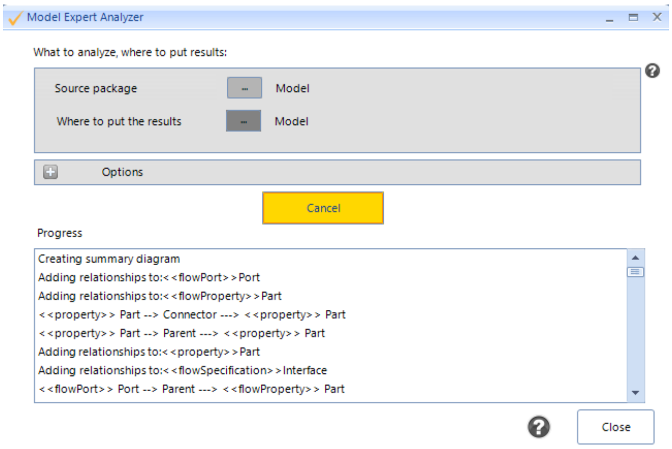



Figure 9.7: Model expert determining model relationships

elements including requirements and activity diagrams (methods). The second is the viewpoint related elements which represent the various applications of DISCREET and the third is use cases[5]. We would therefore expect a snapshot meta-model to present three islands, one for each set of elements. This is indeed also what we find when we run the test, as shown in Figure 9.8.

With traceability for DISCREET itself established, the next step is to repeat the test with the case study model once DISCREET has been imported. Here we again expect to see three islands form for the same reasons. The larger left island must grow larger due to the case study model elements now being connected to the functional elements of our domain extension. At the same time the right hand side islands must remain the same. This again is exactly what we found and is presented in Figure 9.9. Incidentally, when looking at the left hand island in the new snapshot, it is clear that there is a single element type which connects otherwise two distinct models. This element type is of course the requirement.

## 9.3 Validation

Although verification questions could be settled by way of automated testing, validation can not directly be addressed in the same manner. Instead, we need to determine if we have built the right model. This can be done by way of reference to the various viewpoints introduced for each of the four DISCREET methods. In the following discussion we will briefly revisit each included viewpoint and then determine of the concerns of their stakeholders have been addressed.

A caveat is needed here as testing a model against elements contained therein, including viewpoints, is more in line with verification than validation. The latter should focus on stakeholder requirements, however, since DISCREET is a domain extension in the form of a model library, the stakeholders involved here are archetypes of potential users and interested parties, as opposed to actual individuals. This closes the door to much of what could otherwise have been

[5]Although all the use cases are represented in the related viewpoints, those connections are not picked up by the snapshot tool, which results in viewpoints and use cases forming separate islands.

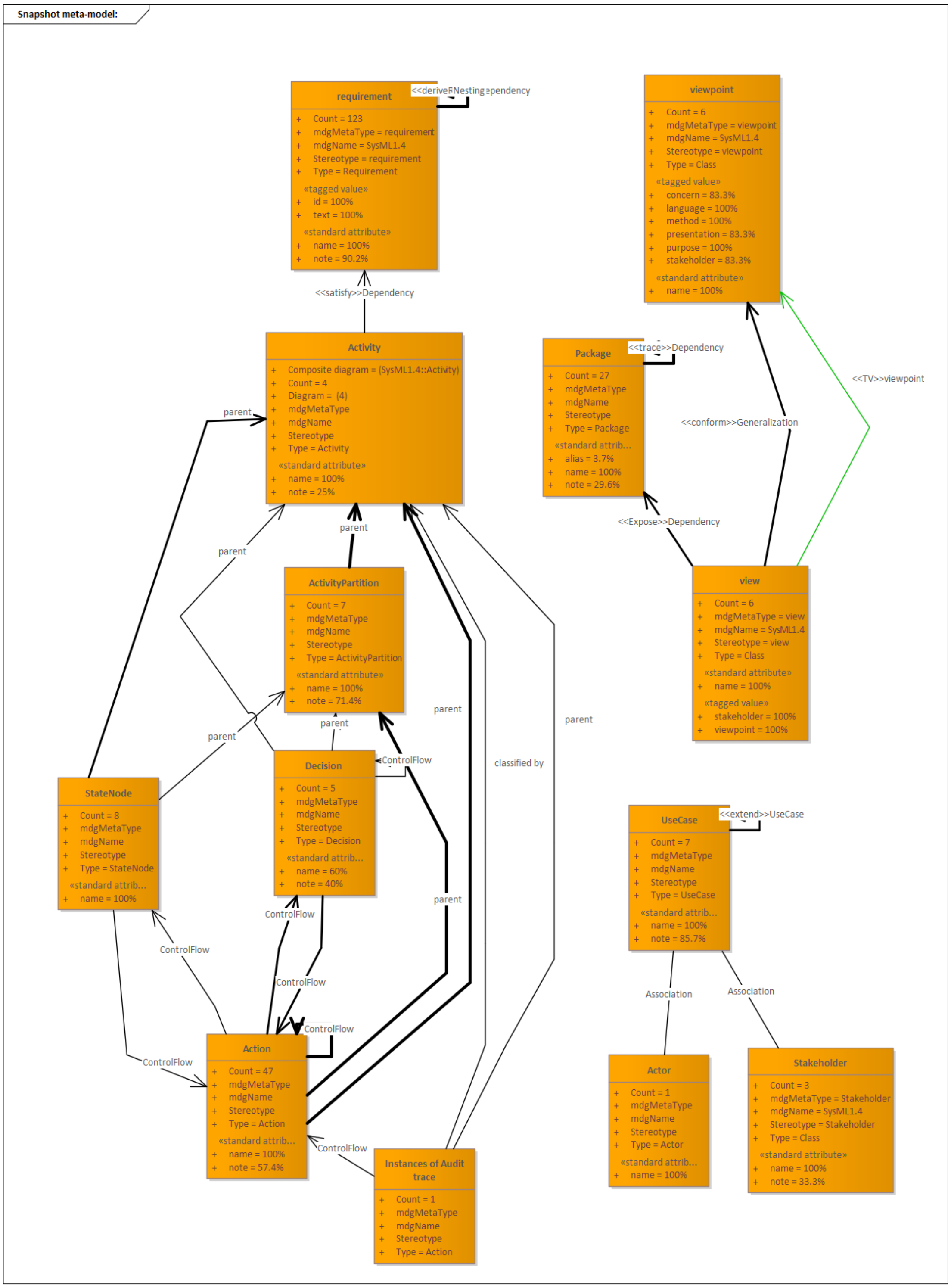



Figure 9.8: DISCREET Snapshot meta-model

done to assess a model in terms of validation. However, since stakeholder needs at the minimum must include that DISCREET is usable for their purposes and these purposes are contained in the viewpoints developed earlier, reference to the viewpoints can be used to make validation

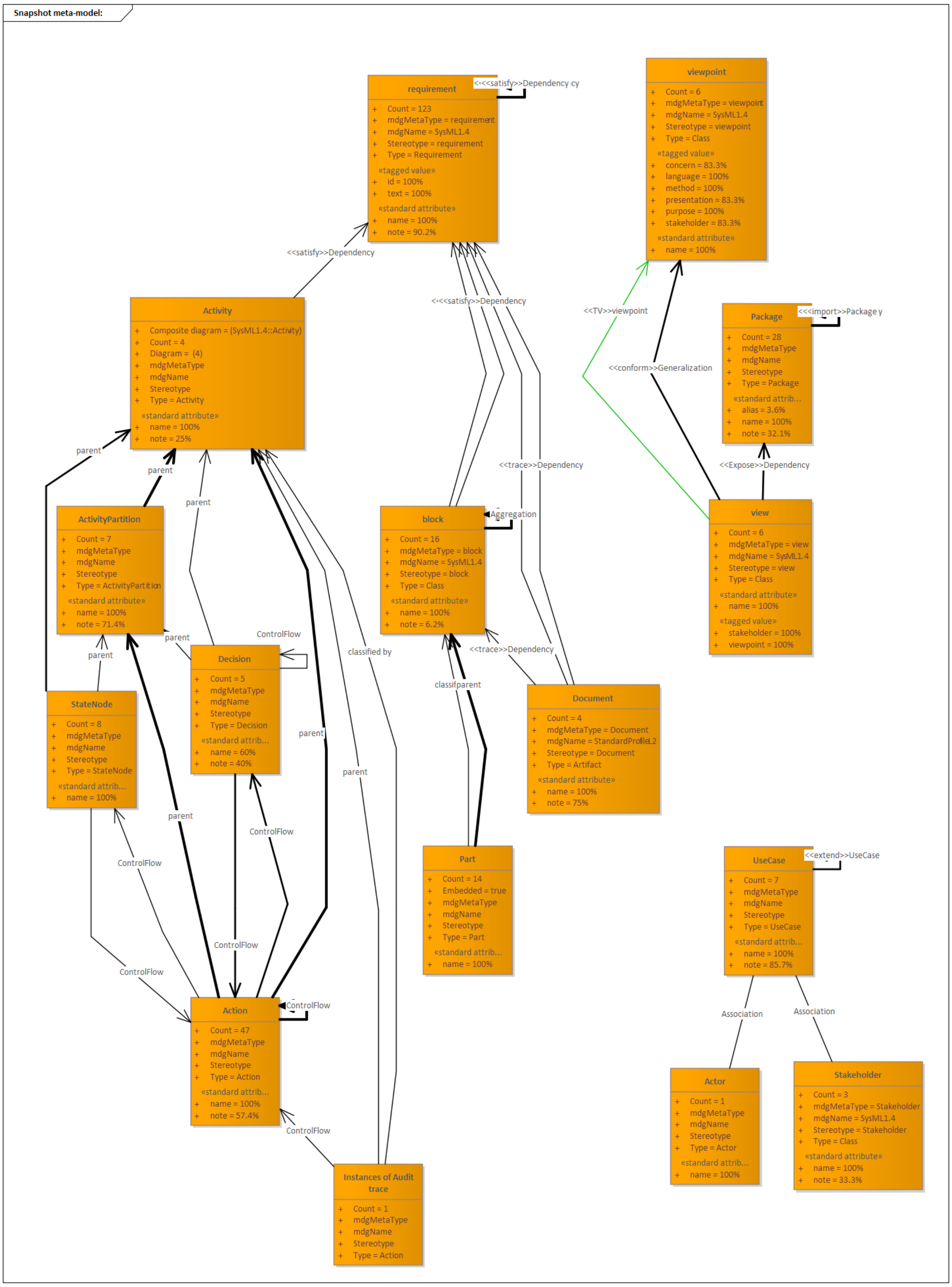


Figure 9.9: DISCREET implementation Snapshot meta-model

statements in this instance[6].

[6]This assertion is also supported by the common use of case studies to test domain extensions, as discussed in Subsection 9.5.

### 9.3.1 Governing method viewpoint

Since DISCREET is intended for use by systems engineers it is not surprising to find that they are stakeholders in each of the four instances presented here. However, the associated viewpoints are slightly different depending on the context and therefore also spawn different views in each case. Subsection 7.2.1 introduced DISCREET's governing method viewpoint and associated modelling view, with Figure 7.5 showing the viewpoint package. Included in the model elements exposed to this viewpoint via the modelling view, is the DISCREET meta-model which is a unique feature of this view. The meta-model is needed here since the governing method must be used by the systems engineer to structure all following processes, including options regarding full or partial implementation of DISCREET. Any case study examining the use of DISCREET therefore has to include the governing method, not only as a necessary condition for any further work but also as a starting point for such work.

For the governing method, systems engineers are the only stakeholders and the associated view therefore includes the full DISCREET model library. The stated purpose for the viewpoint then, is "import and implement DISCREET". In the case study explored in the previous chapter we showed that DISCREET could indeed be imported and used as intended. This use included applying the governing method to determine which elements of DISCREET to use, which in this instance was a full implementation. Furthermore, although the verification checks had a different aim, their outcome also supports the contention that the domain extension was successfully imported and integrated into the case study model.

### 9.3.2 Auditing method viewpoints

Within the auditing context both systems engineers and auditors are stakeholders and although they have different viewpoints, the views they spawn include the same model elements. These two auditing viewpoints were first discussed in Subsection 7.5.1, following on from the introduction of the audit trace method as a whole. Although the audit trace method includes three potential forks for modellers to follow, these options do not impact on the overall viewpoint aims, which is why both the audit elements (systems engineer) and audit compliance (compliance auditor) views can be included in a single package as shown in Figure 7.8. For systems engineers in this context their primary aim is to prepare for auditing, so as to demonstrate compliance during auditing. For auditors on the other hand, the task is compliance auditing to assess if compliance targets have been met. Consequently, there is a full overlap of the model elements exposed by the two views included in this viewpoints package.

The application of DISCREET to the case study then, showed that audit requirements can be linked to model elements or used to identify missing but required functionality. It also showed that this can be done during the design phase and that associated reporting can be produced at this phase too. Furthermore, the successful assessment of one model component shows that DISCREET does indeed deliver on these needs in the most direct sense. On a deeper and possibly more valuable level, the location of a non-compliance instance and the systematic development of remedial actions show that problems are accurately located and dealt with during the design phase. As such we can consider the concerns of both stakeholders addressed.

### 9.3.3 Compliance method viewpoints

As with the previous context, compliance again presents two stakeholders and their related viewpoints as introduced in Subsection 7.6.1. In this case it is systems engineers and regulators and again the two viewpoints spawn fully overlapped views. These views are also very similar to those in the previous context with just the audit requirements package being swapped out for the compliance requirements package and resulting in a very similar package diagram which is shown in Figure 7.9. Given the connection between the two contexts this is not surprising. Here too we find the case study based demonstration shows that the concerns of both stakeholders were met. For systems engineers that was the ability to demonstrate compliance, while the regulators wanted to assess compliance.

Of specific note here is that the connection between consumer IoT and the GDPR is not limited to IoT technology falling under the purview of the GDPR like many other areas. Instead, the capacity for consumer IoT systems to collect, store and process personal data far outstrips that of other fields [15], especially since the IoT has a tendency to subsume other fields such as health and entertainment. Lastly, since the GDPR explicitly requires data controllers to implement Privacy by Design [15], importing DISCREET into a system model is a clear and concrete step towards complying with this requirement. Something which would be far more difficult to attest to if the systems engineer attempted to address GDPR compliance in a less structured manner.

### 9.3.4 Privacy by Design method viewpoint

For this final context, initially discussed in Subsection 7.8.1, the systems engineer is again the only stakeholder. In this case the primary concern is the ability to address privacy and related threat modelling, leading to the eventual generation of remedial requirements or the introduction of PETs[7]. The ability of DISCREET to deliver on these needs was demonstrated in the case study with a non-compliance event mitigated via requirement generation. This followed the standard audit trace method process which has LINDDUN threat modelling at its heart. LINDDUN is also expressly included in the associated viewpoint package shown in Figure 7.17.

Subsection 8.4.5 discusses case study application of the privacy trace method and shows the ease with which the systems engineer can follow the method, apply LINDDUN threat modelling, and generate mitigation requirements. For the sake of completeness, it was also shown that the systems engineer in the case study would have been able to locate an appropriate PET to use should requirement formulation not have been possible.

[7] Please see Section 3.9 for the discussion on PETs.

## 9.4 DISCREET deployment and integration

The verification and validation work presented in this chapter examined if DISCREET meets its stated aims with regards to internal functioning and integration into a system model. Although these checks were all completed successfully, they do not yet speak to why presenting DISCREET as a model library is such a powerful option or how this is superior to other avenues. On this count in Section 3.18, when introducing SysML4IoT, the two options for developing a domain extension were briefly discussed. These are either a model library as opted for in DISCREET or profile based extension as opted for by SysML4IoT. For its primary functioning DISCREET only relies on standards SysML or imports such as the MGF which are fully compliant with standard SysML. Where DISCREET does touch on non-standard elements, such as the CNIL audit documentation, these elements are all dealt with via a SysML wrapper, which can take the form of a model artefact, requirement or action. As such, even these non-standard elements can be addressed in standard SysML tooling, which would of course not be the case for tool dependent systems such as language profiles or automated modelling tools [111]. Though there have been efforts to provide domain specific languages with more suited tooling [110], these still do not provide the cross tool implementation, ease of use, and simple utility that comes with a language compliant model library.

Furthermore, the provision of reusable model libraries within domain specific work does not just include the tools a system engineer might need for a portion of their modelling, as is the case with DISCREET, but could even extend to a template for the entire model within that specific domain. An example of the latter is the availability of a CubeSat meta-model and a full example model[8] implementing that meta-model [139]. The CubeSat example is also interesting in the manner in which a significant portion of this good work is undone by opting for a modelling and analytics suite that is linked to a specific tool. This tool is the same one that SysML4IoT opted for. Doing so might have seemed an obvious choice given the market dominance of that tool, but as mentioned previously, that is no longer the case and for non-enterprise clients the feasibility of this option is questionable[9]. DISCREET therefore not only offers a broad range of functionality to the systems engineer, but does so in a language compliant manner making DISCREET significantly more robust than other domain extensions. This is also underscored by the positive results gained from the case study and verification and validation tests, which are discussed in more detail in the following section.

[8]This model includes satellite hardware, behaviours, constraints and mission specifications.

[9]From the No Magic website it also appears as if the version in which SysML4IoT was developed has now reached end of life and is being retired.

## 9.5 Verification and validation summation

### 9.5.1 The path taken

The objective of the case study presented in the previous chapter and analysed in this one, is to enable verification and validation of DISCREET against a range of standard requirements, show that DISCREET can be imported in to a system model, and finally that DISCREET can bring such a system model into GDPR and privacy alignment. Section 3.16 introduced the notion of IoT modelling in SysML. This was done with specific reference to the SysML4IoT domain extension. Not only does that work clearly illustrate the need for a SysML-based IoT domain extension, but it also diverges from DISCREET on a number of counts. The most significant of these are SysML4IoT's formulation as domain extension of the type profile, while DISCREET is of the type *«modelLibrary»*, and that DISCREET targets GDPR compliance and privacy by design across the system lifecycle while SysML4IoT does not. Though these differences are significant, SysML4IoT as a project still presents valuable lessons[10] for DISCREET, not least of which is an avenue for testing functionality and implementation.

On the topic of testing SysML4IoT, Costa et al. make use of an evaluation case study which specifically aims to demonstrate that their work can be used to generate a system model. This then is also the path taken in evaluating DISCREET with a minor deviation. Both SysML4IoT and DISCREET limited the testing environment to a manageable size but where DISCREET does show the use of MFG in the case study, SysML4IoT omits the use of OOSEM[11]. Thereafter a per view analysis is performed of the implementation, which is also the option taken for our analysis of DISCREET and presented in the preceding section. Finally, a qualitative analysis is performed to ensure that the use of SysML4IoT provides dividends and an overall result which exceeds the results gained by not using it. This again is done here for DISCREET not only through the verification and validation checks performed but also by way of reference to the benefits of it being a model library as opposed to a profile.

However, SysML4IoT is not the only comparable work with other authors also tackling the question of testing SysML domain extensions and ensuring they address their intended purposes. For instance, Maschotta et al. worked on an automotive domain specific language as SysML profile, which the authors tested by way of application to a case study. Although this case study was constrained to a single subsystem and the system being tested is smaller in scope than DISCREET, the authors also reported issues with the bulk of test documentation generated. Finally an applicable viewpoint is brought in and since all use requirements are met and demonstrated, the authors conclude that the practical applicability of their approach is established.

[10] Figure 3.17 presents the IDeA methodology meta-model, which also partially inspired the DISCREET meta-model shown in Figure 7.3.

[11] The Object-oriented Systems Engineering Method introduced in Section 5.4

City GML is an open 3d modelling and data handling standard for city topography. In their influential[12] article on modelling a City GML Application Domain Extension (ADE) in UML, Van den Brink et al. compares and contrasts a number of alternatives, selects their preferred option, and then tests it by way of application and model generation. This approach was subsequently described as “essential reading for developing an ADE” by Biljecki et al..

The rapid pre-validation of competing architectures in thermal modelling is targeted by the TheReSE[13] SysML domain extension [13]. Modelling thermal effects on components including convection and radiation, TheReSE also provides traceability between simulation results and requirements. Here too the authors chose a case study to test the functioning and validity of their domain extension. Again, the authors limit the case study to a functional but manageable subset. While a final and interesting take on the use of case studies is that taken by Koltun and Pundel, who use two SysML domain extension based case studies to test a modelling framework.

Since domain extensions are by their nature bespoke endeavours, it follows that development and ultimately testing and evaluation will also present with bespoke elements. That not withstanding, the preceding discussion showed that the testing approach used here is well within the bounds of accepted practice. This relates specifically to the use of a case study to show the functioning of the new domain extension, as well as the extent to which it is suited to its stated aims. Also often used, are clear rules to limit the case study to a manageable size or subset containing all the elements which need to be tested, without generating repetitive and/or extensive additional documentation. In this testing of DISCREET we also added the verification and validation assessments which go beyond[14] the testing performed in the work referenced above and is further discussed in the following subsection.

### 9.5.2 Verification and validation results

It is tempting to take a near naive approach to GDPR compliance treating headline issues and procedural points as the main events. This type of discourse also seems to point to the line of thought that GDPR compliance is the end all and be all of privacy. That assertion is of course false, which is why LINDDUN is used by DISCREET to assess privacy beyond legal compliance. That being said though, there is some merit in briefly considering how DISCREET might measure up against such a naive line of questioning. For this we turn to the work of Bastos et al. who consider the impact of the GDPR on consumer IoT devices, but generally keeps their discussion to systems at the highest level. They propose the following questions:

- Can a data subject find out if their devices are collecting data beyond the stated purposes?
- Could such data be shared with third-parties and are data subjects aware thereof?

[12]Their work was officially adopted by the standard’s working group.
[13]**The**rmics **Re**lated **S**ysML **E**xtension
[14]Given the differences in tooling, the further testing performed here would not have been available to most of the sources sited above.

- Are there privacy dangers inherent in the online availability of personal data, including diet, exercise and general health data?
- Are data subjects aware of associated dangers?

In answer to the first of these questions we saw that current home IoT devices do indeed routinely collect such data unbeknownst to data subjects. The implementation of DISCREET then would promptly deal with both failures by limiting data collection to stated purposes, ensuring that data subjects have clarity on said purposes, and declaring all data collection and storage to data subjects. For the second question we revisit one of the major findings in Chapter 4, which was that consumer IoT devices routinely collect and share data with third-parties without the related data subject being aware thereof. To the extent that the data subjects are often not even aware that the data exists in the first instance. Here too, DISCREET would nullify the problem by stopping unwarranted data collection and declaring all data flows and data processors to data subjects. The third question relates to the threats of linkability, identifiability, information disclosure, unawareness and non-compliance. These form a significant part of the IoT threat taxonomy presented in Table 6.3 and the bulk of LINDDUN. Accordingly, DISCREET would not only identify all of these issues as significant privacy threats, but would act to mitigate them. This then also provides a two-part answer to the fourth and final question since DISCREET would act to directly eliminate these threats, thereby negating any need to inform the data subject of a threat that no longer exists. If however the threat is unavoidable due to system functionality requirements, DISCREET would ensure that the data subject is made aware thereof.

Although DISCREET easily addresses these high level concerns, this in and of itself is proof of very little since these hurdles are so easily passed. For this reason the testing presented throughout this and the previous chapter was conducted. As stated in the previous subsection, most of this testing is in line with that conducted by other comparable projects though the inclusion of verification and validation checks goes beyond those projects. Starting with checking against DISCREET on its own, we saw in Figure 9.4 that model verification checks were passed. Repeating this test after importing DISCREET into the case study again yielded the same result. This showed that DISCREET not only passed the applicable tests but that importing it into a model did not have any distorting effects. This then leads to traceability checking.

Checking for traceability by way of meta-model snapshots as we have done here, provides clear evidence of the manner in which model elements, by type, relate to each other and across the model. This latter point specifically allowed us to compare the expected model graph to the one actually realised. Here we found the exact graph shape we expected both in relation to DISCREET on its own and when imported into the case study. However, proving that graph shape is what we expected is not yet proof that the models are truly correct. For this we need to confirm that all model elements are of the correct type, appropriately connected, and do not violate other language rules. For this we used the Sparx Enterprise Architect built in tool to check against 13 metrics. After exclusions for warnings relating to unused requirements and a system error, no further warnings and no errors were reported, thereby verifying the model.

Finally we attended to model validation by revisiting each included viewpoint and determining if the concerns of each stakeholder per context were met. Here too we found that all such concerns were met and as such we conclude that DISCREET functions as intended.

## 9.6 DISCREET assessment overview

As discussed in Subsection 9.5.1, the method of assessment used herein in has been deployed by many other projects, provides the information needed to make that assessment, and also demonstrates DISCREET's functionality beyond that assessment[15]. A purely model based approach to testing is also found in other projects or with a slightly different area of focus, such as the CubeSat project introduced in Section 9.4, which had as its first milestone the use of its newly generated model to produce common CubeSat specifications [139]. No assessment of any system would be complete without discussing the systems limitations though.

In the case of DISCREET, the most obvious are those that relate to it being a SysML domain extension. As such, DISCREET can only be used by practitioners who at the very least, are familiar with SysML and have access to a modern and capable modelling tool. Though DISCREET's position as model library, as opposed to a profile, goes some way to advance ease of use and broaden the range of tools available, there will always be some skills requirement placed on the modeller. Further measures to aid in this regard are the inclusion of detailed methods[16] which contain associated view points, activity diagrams for their implementation and also fully worked examples for GDPR compliance and auditing.

Although DISCREET was developed with the aim of addressing GDPR compliance and auditing, as well as general privacy by design, within the consumer IoT sphere, the way in which each method is structured allows for the systems engineer to take DISCREET outside of this targeted domain and apply it to most any field with regulatory compliance and/or additional privacy requirements. This adds additional functionality, but due to the sheer volume of work involved, and in depth knowledge of the compliance regime needed, it was not possible to examine that functionality.

The third limitation is that of SysML v2. SysML v2 has not yet been finalised or adopted, while mainstream commercial tools have also not been released. Consequently, it would not have been a good candidate for use in the development of DISCREET given the need for direct application and reusability for system engineers. This may, of course, change in future and is therefore a clear candidate for future work, as discussed in Chapter 10.

With DISCREET now developed, tested and packaged as a stand alone domain extension, the bulk of the work for this project is completed. What remains though, is to cast an eye to the

[15] DISCREET is tested in a case study, thereby demonstrating that it can be imported into and used to change the compliance and privacy outcomes of that case study, and in so doing also provides the opportunity for verification and validation to be performed which goes beyond the testing typically done in the referenced literature.

[16] Governing, audit trace, compliance trace, and privacy.

future not just for the further development of DISCREET itself, but also the exploration of other related work. In this, proposed work on new SysML model libraries and the linking of SysML, threat modelling and formal methods already shows great promise. These and related topics are further discussed in the following chapter.

# Chapter 10

# Conclusion and Future Work

## 10.1 Project overview

The work presented here is both relevant and timely with regards to consumer IoT, privacy, and compliance. This is due to three factors of which the first two are strongly linked. The first is the massive and continued rise in IoT adoption across the board [140]. The second is that, as we showed herein, the privacy characteristics of consumer IoT devices can worsen over time. The third factor is the continued increase in systems complexity, which requires a well grounded and capable approach to systems engineering, preferably with a strong focus on reusable elements.

In this work we set out to investigate if consumer IoT devices are designed to meet both the general privacy requirements of consumers and legislative privacy requirements, and if the current answer is "no", to then develop a remedy for this state of affairs. We established the answer to this question is indeed "no" and from a combination of anecdotes on IoT failures and research conducted by others, we found that the primary area of failure is in the design of these devices. That is to say, there are no technological or other reasons which preclude the manufacture of sufficiently compliant and private consumer IoT devices. To affirm this point and develop a solution and guide the research needed to do so, we presented the following research question:

- Can a single SysML domain extension address both compliance and privacy by design in consumer IoT, and if so, what are its components?

From this primary research question we developed a number of supplementary questions to direct the work to follow. In this the first port of call was to set out the exact nature of the research to do, with reference to standard research practice. This includes both a review of existing literature and conducting our own primary research. From the existing literature we found a large number of other initiatives which also identify these problems and in some cases offer solutions. However, these solutions are either narrow in their application, entirely theoretical, or do not address the possibility of industry application during systems design. To structure the discussion in Chapter

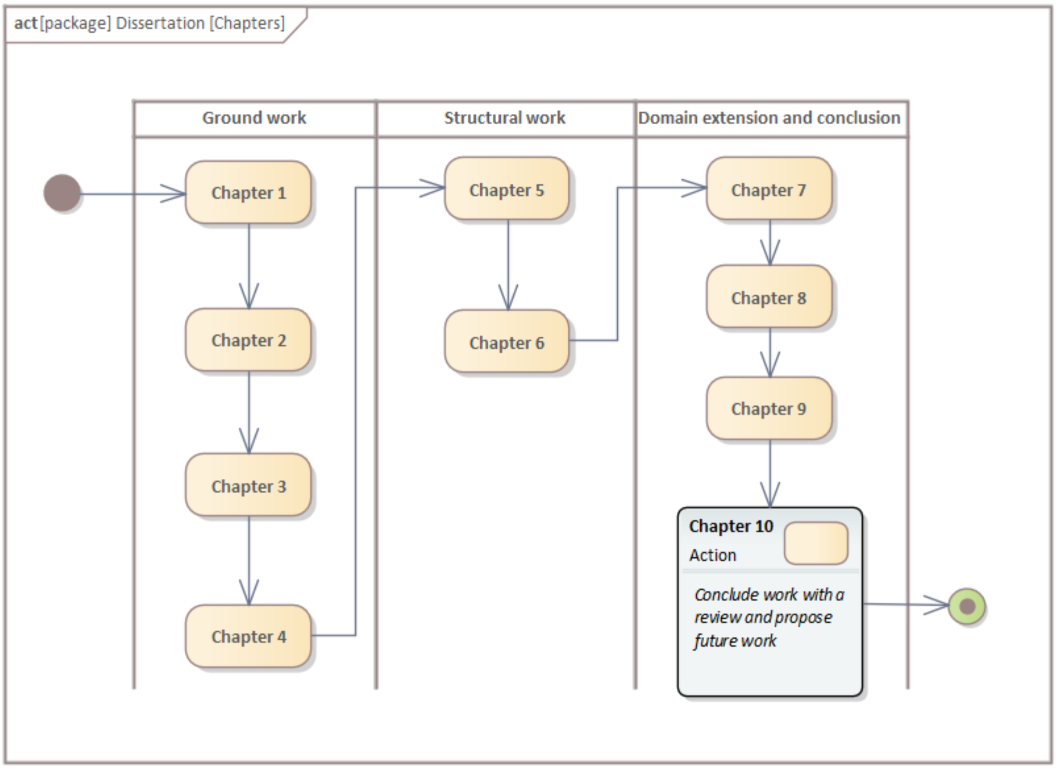



Figure 10.1: Chapter 10 overview

3, eight metrics that speak to the primary elements of the problem space were introduced and the presented literature reviewed accordingly. This showed areas of overlap and areas of need. To gain a deeper understanding though, especially with a view to developing a solution for the areas of need, primary research was conducted using an IoT testbed. Based on the results thereof we found a series of failings which were in many ways far worse than expected but also affirmed our position that the primary failure is one of design and that it was indeed possible that failings could worsen over time.

To analyse the testbed results in more detail CNIL's PIA was used and significant failings around the processing and collection of user data were found, with this extending to communicating these activities to users. As a first step to developing a solution, we considered the work previously reviewed using the testbed results as filter and found that both the CNIL's PIA and the LINDDUN threat modelling methodology were excellent fits. However, simply pointing others to these two resources would be borderline pointless as neither cover all the elements required of a full and industry usable answer to the challenges facing compliance and privacy by design in consumer IoT. Consequently, the DISCREET domain extension was developed to meet these needs and is the primary contribution of this work.

Finally the newly formed domain extension had to be tested. For this a case study based approach was used, incorporating a model built by a tool vendor which was wholly unrelated to DISCREET. From this third-party case study a subset of the model elements were selected and a narrative constructed to guide the use of the domain extension. This approach allowed for DISCREET to be tested at a manageable level.

## 10.2 Research questions revisited

To guide the process of moving from the main research question to the final answer outlined above, six additional sub research questions were formulated and are recapped below:

- RQ1: Which current compliance assessment, privacy enhancement or threat mitigation methodologies, strategies, and frameworks relate to consumer IoT implementations, or can be brought to bear on them?
- RQ2: Considering the defined problem space, what are the coverage gaps in prominent current frameworks and methodologies, including those not directly aimed at IoT?
- RQ3: Which of these methodologies would best serve to assess the testbed results presented herein?

The first three sub research questions were answered by way of a literature review which presented a series of divergent work covering the IoT, privacy by design, and compliance. From this it was evident that no current offerings fit the bill and a novel approach was needed. The next three questions then drove the process of scoping and developing this novel approach.

- RQ4: Do real-world observations of a consumer IoT implementation match up to the expected shortcomings, both as a snapshot and over time?
- RQ5: Can the lessons learned from the preceding investigations be translated into an actionable mitigation of the privacy and compliance related design challenges in consumer IoT?
- RQ6: What form could an approach to engineering compliance and privacy by design into IoT devices and systems take, if it were expressed in standardised systems engineering terms?

From this real-world observation it was found that consumer IoT not only massively under performed against compliance and PbD metrics, but that this under performance could worsen over time. Furthermore, these issues did not relate to technological challenges but rather to design choices. This insight was the key motivator for the choice of a systems engineering approach which eventually took shape as the DISCREET domain extension. Though the domain extension itself is not just a response to the testbed findings but also to the literature review in Chapter 3. Linking the DISCREET domain extension to the literature review presented in Chapter 3 is a series of eight metrics. These were used as a means of assessment for the work analysed in the literature review and are:

- A comprehensive approach covering both technical and legal considerations.
- Focused on implementation as opposed to only assessment or suggested courses of action.
- Focused on auditing outcomes.
- An IoT specific focus.
- The use of data flow diagrams or other measures to track data flow.
- Privacy by design is the primary focus.
- Systems modelling using a language currently in use in industry[1].
- Can the specifics be represented in a standard systems modelling tool.

These focus areas are all directly reflected in DISCREET, but possibly its standout feature is that fact that it is a SysML Domain extension as a model library. This specifically services the need to develop a novel contribution which also has the potential for industry application. In our estimation the alternative approach, developing a domain extension via profile, would undermine any potential real-world application.

Lastly, we were able to present the constituent parts of our domain extension and usage examples in answer to the primary research question. That is to say, yes a single SysML domain extension can address both compliance and privacy by design in consumer IoT and its components are those of DISCREET.

## 10.3 Future work

This section presents proposed future work starting with the most concrete proposal and progressing to the least. The prospect of SysML v2 could play a major role in determining the sequence of this work, while a SysML v2 version of DISCREET (with potential additions proposed below) is a high priority.

We will not delve into the exact changes coming with SysML v2[2] but will mention that they present a much larger change than between the preceding releases. Most notably, SysML v2 will explicitly move further away from UML than previous releases. Having a new version of DISCREET ready after the release of SysML v2 would be advantageous but also brings some difficulties, most notably around tooling. As a result of this latter concern we have decided to wait for full vendor support before proceeding with this update. The state of SysML tool support is problematic not only due to some variance in implementation, but also due to the decidedly outdated look, and functionality of many tools.

[1] One of the primary reasons we will opt for the use of SysML is its industry application and standardised tooling. Any system which adheres to these requirements will have an automatic advantage.

[2] Full details are available here `https://github.com/Systems-Modeling/SysML-v2-Release`

### 10.3.1 Threat modelling and model checking

The first piece of proposed future work is already well under way. As part of Arm's Morello programme[3] there is a need to understand the functioning of highly secure, but bespoke, hardware in environments where other system components are off-the-shelf devices, typically with minimal to no built in security traits. This work takes a case study based approach focused on a smart ballot box. The smart ballot box has a Morello board at its core but all other sensors and actuators are low cost off-the-shelf components. Interestingly, the manner in which these components link to the Morello board, and the board to a wider network, is similar to the manner in which consumer IoT devices link to a smartphone and that again links to a wider network.

For this study we are using STRIDE to conduct threat modelling and generate two classes of threat related requirements. The first are those which relate directly to system software and dictate some software-based mitigation or secure operation. Fully addressing these requirements will form part of the model checking activity to be carried out in Event-B. The second set of STRIDE threats will be those relating to either physical threats or threats localised outside of the Morello board. These will be formed into mitigation strategies and included in a SysML model as requirements. Thereafter the work conducted in the Event-B model will also be subsumed into the SysML model and finally, the general provisions of the case study will be drawn in. The resulting SysML model will then present the "single version of the truth" for the smart ballot box and allow for prototyping to start.

This project is ongoing and envisioned to yield a number of papers and articles. The initial work here deals with the interaction between STRIDE, SysML and Event-B. Specifically focusing on how SysML can be used in conjunction with model checking in Event-B to capture and address a full range of STRIDE threats. The linchpin here, is a set of requirements formed into a new and reusable specification for governing the SysML to Event-B interaction[4].

### 10.3.2 Closer threat modelling integration

The work of Johnstone, as discussed in Chapter 3, presents an interesting avenue for advancing DISCREET. In that work, the author contends that threat modelling, specifically via STRIDE, can be greatly improved by substituting traditional DFD-based models for UML activity diagrams. A complimentary point, though not exclusively focused on UML, is also made by Van Landuyt and Joosen. There an argument is made against the singular use of DFDs due to their rigidity and limited level of detail.

The limitations of a DFD-based approach, as used in STRIDE, is best demonstrated by there only being four element types, namely data flow, external entity, process, and data store. A UML activity diagram on the other hand has eleven distinguishable elements. As such, a much finer grained analysis is possible as can be seen from Table 10.1.

[3] More information is available here: `https://www.arm.com/why-arm/architecture/cpu/morello`

[4] An extended abstract and poster on this topic as work in progress, have been accepted for MODELS 2022.

| Element | Description |
|---|---|
| Initial node | A filled in circle is the starting point of the diagram |
| Activity final node | A filled circle with a border is the end point |
| Activity | Represents physical or electronic activities that occur |
| Activity edge | Presented as arrows on the diagram |
| Fork | A black bar denoting the beginning of parallel processing |
| Join | A black bar denoting the simultaneous end of parallel processing |
| Condition | A guard which must evaluate to true for node traversal to occur |
| Decision | A diamond with one flow entering and several leaving |
| Merge | Several flows entering and one leaving, with one entering flow required |
| Swimlanes | Also referred to as partitions, these indicate who performs the contained activities |
| Flow final | Indicates that one split of a parallel process stops at this point |

Table 10.1: Activity diagram components [86]

From this finer grained analysis Johnstone was able to show all the threat modelling options available in STRIDE could be covered and that significant further insight could be gained. We therefore want to investigate if this switch to activity diagrams could be used for a more advanced integration of LINDDUN into DISCREET. Such a study would require significant testing. However, the fact that DISCREET uses SysML instead of UML is not seen as a hindrance here. It is rather the relative complexity of LINDDUN and the functioning of DISCREET that could be problematic. As such, we propose to do a future study involving first an analysis of LINDDUN using activity diagrams and if successful, the inclusion thereof in DISCREET.

### 10.3.3 SysML4IoT integration

Due to the significant differences between the two domain extensions, SysML4IoT can not be used in or incorporated into DISCREET. However, this is not to say that they are mutually exclusive in application. SysML4IoT targets the IoT at a much deeper level than DISCREET does, while the latter addresses critical issues in compliance and privacy in a way that the former does not. As such, modellers could certainly deploy both.

However, where DISCREET opts for a lightweight and modular approach to its components, SysML4IoT presents much greater formalism and opts for larger feature sets, for instance in its choice for OOSEM. These differences can be dealt with by selectively importing DISCREET elements into a model using SysML4IoT, specifically with the exclusion of the MBSE Grid

Framework. Although this is a workable stopgap measure, the potential for greater synergy between DISCREET and SysML4IoT is such that we would advocate for a proper focus on this issue. Targeting as output an alternate version of DISCREET which is aligned with SysML4IoT.

There are however also a couple of significant hurdles in this undertaking. Firstly, the tool used in IoT4SysML is a plugin for a major SysML modelling tool. However the vendor of this SysML tool has had some difficulties recently after a takeover and seems to have continuing issues in dealing with individual and academic users. Secondly, the IoT4SysML plugin is written for SysML 1.4. This is not yet a "deal breaker" but could become an insurmountable challenge after the full release of SysML v2. This depends on how tool vendors go about adopting the new standard and also what work is done on compatibility for the various components of SysML4IoT. Given these concerns, an alternate route is to wait for the adoption of SysML v2, conduct the update of DISCREET and select from IoT4SysML both the IoT-A domain model and IoT model library, to advance DISCREET's value proposition.

### 10.3.4 Security modelling

The need for specific privacy protections can be determined by way of privacy engineering methodologies, but the implementation of such protections is the domain of cyber security [24]. The need for more research into security by design is clearly identified by Weir et al. who also highlight the significant contribution being made by the Cyber Security Body of Knowledge (CYBoK) hosted out of the University of Bristol in the UK. These two points indicate a promising further line of inquiry, which is the expansion of DISCREET to include cyber security for consumer IoT devices. This involves not only the development of devices by way of modelling the device and ecosystem lifecycles, but also constructing models of existing devices to use in verification and testing against predetermined metrics[5]. A link to cyber security in SysML is not a new idea though and there have been a number of efforts in this regard. One of the best known is CyberML[6], which styles itself as "*a UML/SysML profile and model library for specifying the architectures and designs of cybersecurity applications*".

If a cyber security related expansion of DISCREET was undertaken it would again proceed as a model library based extension only and would best be implemented after an upgrade to SysML v2. It could however also be a stand alone module, using the procedures and principles of DISCREET, but forming its own domain extension.

[5]Such metrics may relate to security issues which are evident at the model level, including via the use of state machines.

[6]More information is available at `https://cyberml.com/`

## 10.4 Conclusion

The project presented in this work directly tackles a series of significant threats to the privacy and personal lives of millions of consumers the world over. While the focus is very much on the GDPR and related challenges, the proposed solution explicitly includes the steps of its own creation allowing for reuse not only by others dealing with these issues but also in radically different legislative environments. Although this work was conducted during the time of a global pandemic, and that necessarily closed some doors, the principles of SysML models as single versions of the truth, model and element reusability, and refraining from dictating solutions, were constantly used to guide decisions.

Although DISCREET provides tools for use in the black box, white box and finally the solutions phases, these remain tools only as it is up to the systems engineer to develop the correct implementation for their context. This also informed the composition of the various methods included in DISCREET, where these methods described the process of drawing in the needed auditing, compliance and PbD requirements, even though the resulting requirements are already included in DISCREET. Thereby providing end users with a more robust tool set to use and enhancing the contribution made.

It is hoped that this approach will allow others to find value in the work presented here, continuing both the research and the advancement of privacy and compliance in consumer IoT.

# Appendix A

# DISCREET methods

In Chapter 7 we introduced and developed the DISCREET methodology. The name is derived from its core functionality, which is **D**oma**I**n exten**S**ion for **C**ompliance and p**R**ivacy by d**E**sign in consum**E**r io**T**. In this appendix we present the methods making up the central process for executing DISCREET[1].

## A.1 Method 1: The governing method

This method guides the use of the domain extension in three distinct ways:

1. As a collection of system design resources which practitioners can include in their models in an *ad hoc* manner.
2. A structured methodology for conducting compliance and privacy by design for consumer smart electronics, using SysML.
3. A ready made *«modelLibrary»* for inclusion in other system models.

The method sequence is as follows:

- Derive system model, including stakeholders, viewpoints, and requirements not related to DISCREET[2]
- Determine which elements of DISCREET are needed
- Clarify if MGF or another methodology is used to structure overall execution

[1]The DISCREET model library and testbed dataset are available for download here https://doi.org/10.5258/SOTON/D2434.

[2]DISCREET only directs design and corrections relating to compliance and PbD for the system concerned. All other model elements, including business process issues relating to compliance and PbD still need to be drawn in as per the user's needs.

- Import the related DISCREET[3] meta requirements to the system model
- Follow the DISCREET methods for each meta requirement
- Maintain and produce DISCREET outcomes
- Maintain continuous reference to DISCREET in the system model to allow for changes over the system's lifecycle

## A.2 Method 2: The audit trace method

**Fork 1: An official audit methodology**

- Determine context
- Determine if there is an official audit methodology
- Include a requirement that all audit requirements should be drawn into the model as appropriately formulated "shall statements" and in accordance with the compliance trace method (black box)
- Determine that all audit requirements are included and up to date, if not, then return to the previous step[4]
- Record all requirements to the requirements matrix for later use in the compliance trace method

**Fork 2: An adjacent audit methodology**

- Determine context
- Determine if there is an official audit methodology
- Determine if there is an applicable adjacent audit methodology
- Include a requirement that all audit requirements should be drawn into the model as appropriately formulated "shall statements" and in accordance with the compliance trace method (black box)
- Determine that all audit requirements are included and up to date, if not, then return to the previous step[5]

[3] Each element of DISCREET, as discussed below, is hooked into a system model by the introduction of a single meta requirement.
[4] This is both a check for the correctness of the work done and a lifecycle change check.
[5] This is both a check for the correctness of the work done and a lifecycle change check.

- Determine that no new official audit methodology has been released, if so, return to step two[6]
- Record all requirements to the requirements matrix for later use in the compliance trace method

**Fork 3: Stand in audit via provenance graphs**

- Determine context
- Determine if there is an official audit methodology
- Determine if there is an applicable adjacent audit methodology
- Develop a system wide DFD based provenance graph
- Split the graph into inter and intra-context segments, per IoT context[7]
- For each compliance requirement, formulate an audit requirement requiring proof of compliance
- To each newly formulated audit requirement, attach a provenance graph describing the associated data actions
- Develop standard documentation to capture compliance outcome and associated graph
- Determine that no new official, or adjacent, audit methodology has been released, if so, return to step two[8]
- Record all requirements to the requirements matrix for later use in the compliance trace method

## A.3 Method 3: The compliance trace method

The compliance trace method consists of the following discrete steps, with the audit trace method subsumed therein:

1. Determine applicable legislation
2. Cast the compliance requirements as "shall statements" and include in requirements database
3. Complete the audit trace method

[6]This is a lifecycle change check.
[7]Processing context can change the regulations at play and therefore the compliance requirements.
[8]This is a lifecycle change check.

4. Include meta requirements for reporting artefacts generation as per Subsection 7.6.3.
5. Build out the system model in SysML
6. Attach the applicable requirements to model elements using the *«satisfy»* relationship
7. Generate reporting artefacts where requirements and model elements intersect or should intersect
8. Link meta requirements to reporting artefacts using the *«realization»* relationship
9. Link compliance requirements, audit requirements, and system components to the newly generated reporting artefact using the *«trace»* relationship

## A.4 Method 4: The privacy by design trace method

- Derive a DFD diagram for the system at issue
- Type all items on the DFD as P, DF, DS, or E[9]
- Use the LINDDUN mapping template to map threat areas on to the DFD components as L, I, Nr, D, Di, U, or Nc[10]
- Determine if any portions of the LINDDUN threat tree are not applicable and create a system artefact to document these assumptions
- Refer mapped threats to the threat tree to locate detailed privacy threats as misuse cases
- Create model artefacts to document misuse cases
- Prioritise risks using preferred method[11]
- Determine if a concealing association or guarding association is at play[12]
- Consult LINDDUN mitigation strategy taxonomy based on the previous step
- Determine if the system is under design or the threat is realised
- For realised threats determine one or more PETs and run trade studies if a choice is to be made
- Derive a requirement to implement the selected PET
- Implement the PET and link it to the requirement above with a *«satisfy»*

[9] Process, data flow, data store. or entity.

[10] Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance.

[11] Risk is seen as the function of likelihood and severity, but the exact method of calculation is left to the systems engineer.

[12] Is the exposure of data being prevented before the fact or managed after the fact?

- For systems under design, work the results back to the privacy threats found and develop privacy requirements
- Link both the requirement above and the PET implementation requirement via *«trace»* relationships to the meta requirement used to initiate the PbD trace method

## A.5 Note on generating reporting artefacts

1. Class 1: Full intersection (Compliance requirement, audit requirement, and model element.). Generates class one output: Compliance documentation, as per the linked audit requirement.
2. Class 2: No audit intersection (Compliance requirement and model element). Generates class two output: provisional documentation and proposed remedy.
3. Class 3: No compliance intersection (Audit requirement and model element). Generates class three output: Non-compliance documentation relationship with a compliance requirement, while an audit requirement seeks proof of compliance.
4. Class 4: No intersection. A model element should, on the face of it, be under the requirements of the applicable regulation but has no compliance or audit intersections. Generates class four output: Non-compliance documentation and proposed action to determine cause of the issue.

# Appendix B

# DISCREET Requirements

In this appendix we present all the requirements included in the DISCREET domain extension. These requirements are grouped in the same way as in the model library and are presented using SysML values, that is “NAME” for the title of the requirement, “ID” for the requirements ID, “NOTES” for use notes on the requirement, and “TEXT” for the text of the requirement. As indicated in Chapter 7, we include the ID as a prefix to the requirement name. Although this is not standard practice it is a “quality of life” change to enhance the utility of matrix views as the IDs directly relate to the legal, audit, or other documents they are based on. Since it is also often the case that compliance or audit requirements in their original form will be too long or include multiple provisions, these are split to make requirements containing singular shall statements. We construct the new requirement’s ID by using a letter[1] to indicate the source and then the numbering system of the original requirements and appending Roman numerals in brackets.

## B.1 DISCREET mission needs statements

The first subsection below contains a broad mission needs statement and is used to govern the overall undertaking but will never be directly satisfied by any single other model component. The four subsections listed thereafter each correspond to a single requirement which directly forms part of the mission needs statement. These four requirements are accordingly linked to the mission needs statement via a containment relationship and will also not be directly satisfied by model elements. There will however be a larger number of other requirements derived from these four, where those derived requirements will have a full scope of interactions with the rest of the system model. This is demonstrated in sections B.2, B.3, B.4, and B.5. However, due to the large number of requirements involved, it is not possible to include a requirements diagram for each section to follow.

[1] G for GDPR and C for CNIL. The one variation is the four meta requirements which use the word Class and the numbers 1 to 4.

### B.1.1 The mission needs statement

**NAME:**
DMNS: DISCREET Mission Needs Statement

**ID:**
DMNS

**TEXT:**
Compliance with privacy-focused legislation and general privacy by design are both severely lacking in consumer IoT deployments. These failings could be addressed through a single, design-led solution expressed in a systems engineering context whilst also addressing additional vital factors such as auditing against compliance requirements.

This design-led solution must include meta requirements to structure the inclusion of, use, and reporting on, compliance and auditing requirements. Second, the related compliance requirements must be drawn in and cast as appropriately phrased shall statements. Third, the audit requirements for assessing performance against the compliance requirements must be included as shall statements. Fourth, a suitable privacy by design methodology must be included and enacted by way of requirements.

**NOTES:**
This Mission Needs Statement governs the requirement and other elements needed to enact DISCREET in full. After importing the DISCREEET *«modelLibrary»*, the DMNS serves as starting point for enabling DISCREET.

### B.1.2 Audit and compliance meta requirements

**NAME:**
CANS: Compliance assessment meta requirement needs statement

**ID:**
CANS

**TEXT:**
Meta requirements must be present in the system model to structure the inclusion of, use, and reporting on, compliance and auditing requirements.

**NOTES:**
Meta requirements are needed to guide the linking of compliance and audit requirements as well as the production of related artefacts.

### B.1.3 Compliance requirement needs

**NAME:**
CRNS: Compliance requirement needs statement

**ID:**
CRNS

**TEXT:**
Compliance requirements must be drawn in from applicable legislation and cast as appropriately phrased shall statements.

**NOTES:**
Once the appropriate legislation is identified, the dictates thereof must be drawn into the system model as requirements formulated as shall statements.

### B.1.4 Audit requirements

**NAME:**
ARNS: Audit requirement needs statement

**ID:**
ARNS

**TEXT:**
Audit requirements for assessing performance against the compliance requirements must be included as shall statements.

**NOTES:**
The best suited audit requirements must be determined by way of the audit trace method and drawn in as shall statements.

### B.1.5 Privacy by design requirement needs

**NAME:**
PRNS: PbD requirement needs statement

**ID:**
PRNS

**TEXT:**
A suitable privacy by design methodology must be included and enacted by way of requirements.

**NOTES:**

Requirements are used to initiate and guide the use of an appropriate privacy by design methodology.

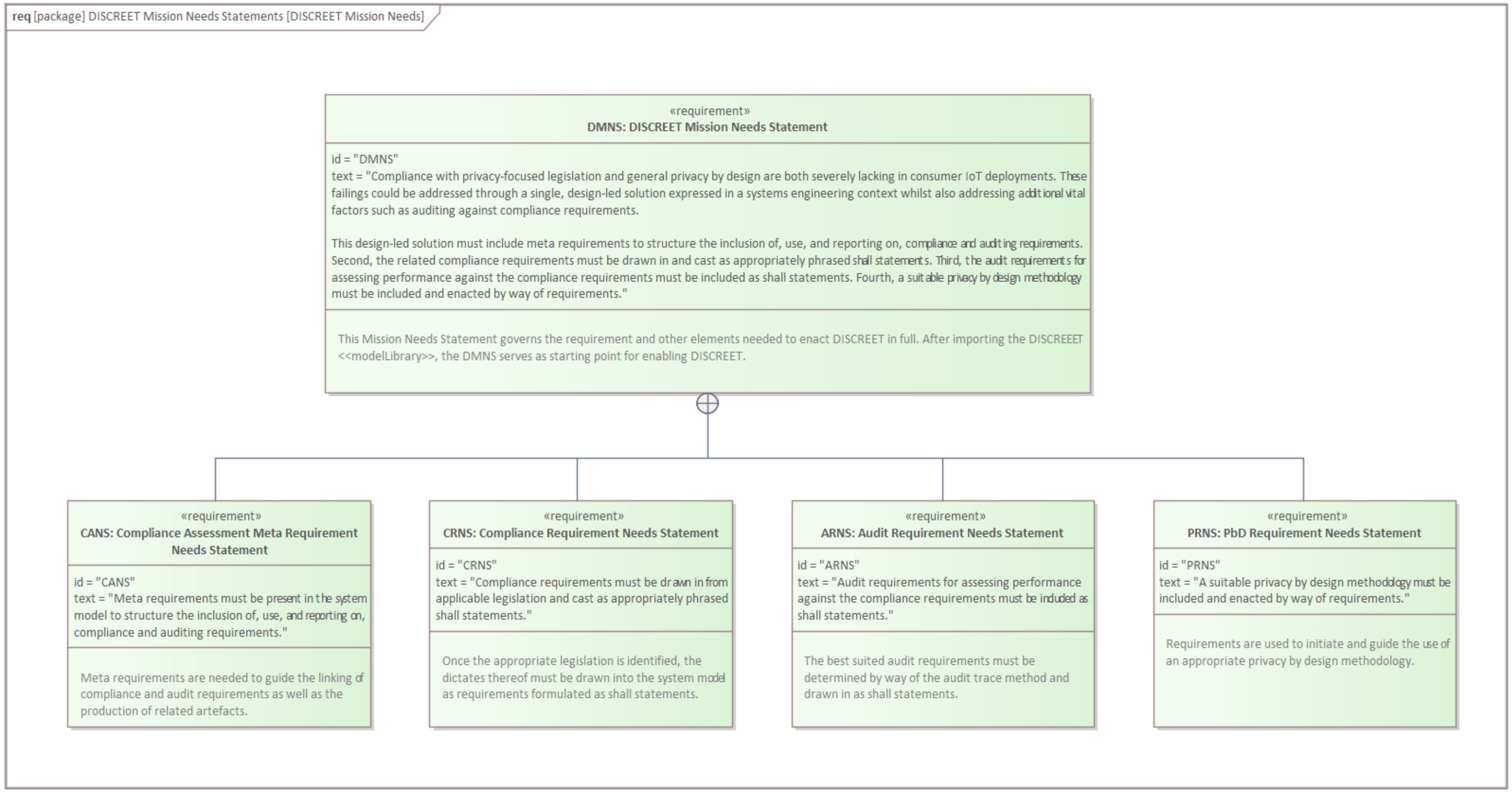



Figure B.1: DISCREET mission needs statement and requirements

## B.2 DISCREET compliance assessment meta requirements

All the requirements in this section flow from ID:CANS by way of a *«deriveReqt»* relationship.

**NAME:**
Class1: Full intersection meta requirement

**ID:**
Class1

**TEXT:**
Where compliance requirement, audit requirement and model element intersect to indicate full compliance, compliance documentation as per the audit requirement shall be generated.

**NOTES:**
Full intersection meta requirement (Compliance requirement, audit requirement, and model element.). Generates class one output: Compliance documentation, as per the linked audit requirement.

**NAME:**
Class2: No audit intersection meta requirement

**ID:**
Class2

**TEXT:**
Where a model element and compliance requirement intersect without a satisfied audit requirement, a report including proposed remedies shall be generated.

**NOTES:**
No audit intersection (Compliance requirement and model element). Generates class two output: provisional documentation and proposed remedy. The contents of the audit requirement that has not been satisfied, that's its requirements and documentation, can be used to formulate the report and remedies.

**NAME:**
Class 3: No compliance intersection meta requirement

**ID:**
Class3

**TEXT:**
Where an audit requirement and model element intersect but no compliance requirement, a non-compliance report including proposed remedies shall be generated.

**NOTES:**
If a model element was not designed in accordance with a compliance requirement and therefore

fails to meet a related audit requirement, the link between audit requirement and model element must change from *«satisfy»* to *«trace»*. This allows for traceability to be maintained whilst reflecting the failure to satisfy the audit requirement.

**NAME:**
Class4: No intersection meta requirement

**ID:**
Class4

**TEXT:**
Where a model element does not intersect with compliance or audit requirements but is expected to do so, a non-compliance report including proposed remedies shall be generated.

**NOTES:**
In this case the model element concerned will directly *«satisfy»* the class four meta requirement.

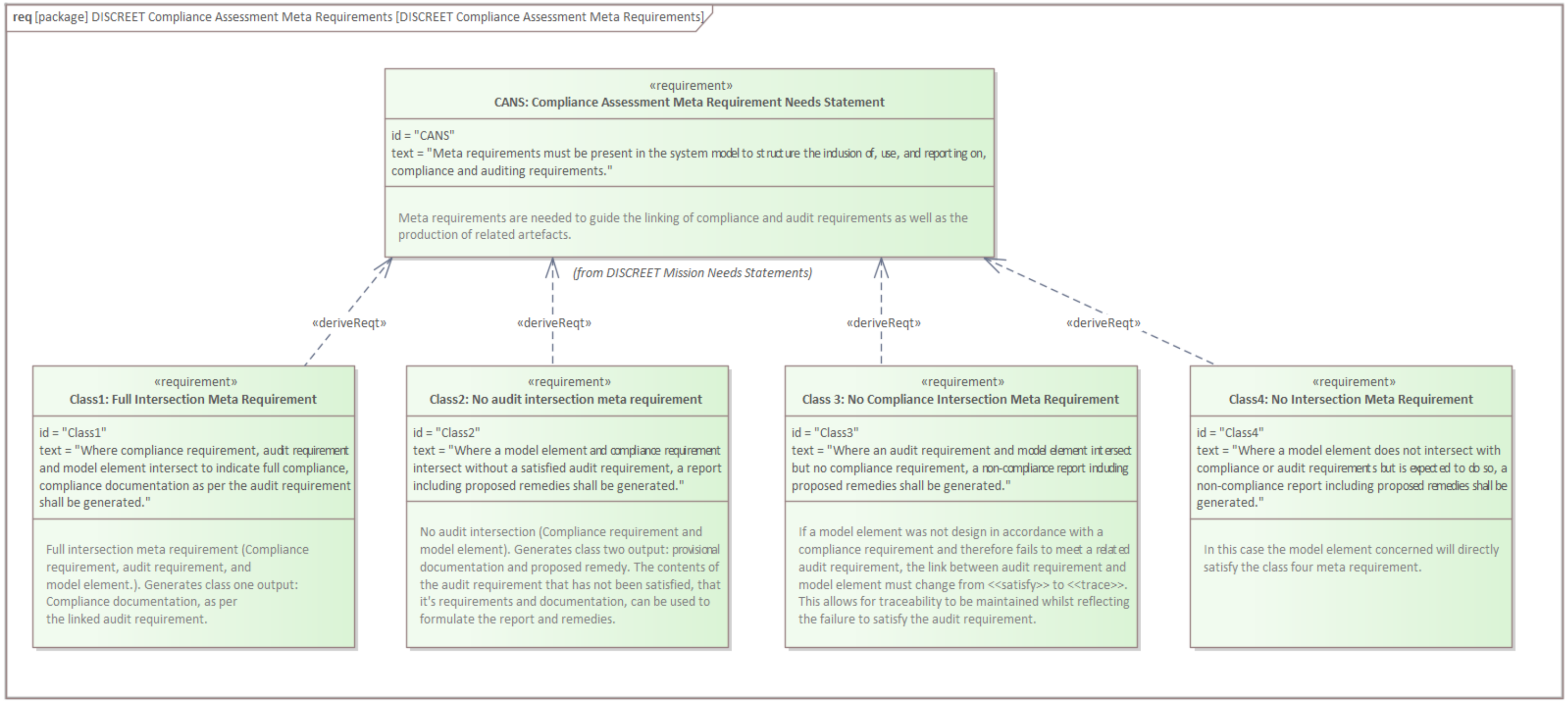



Figure B.2: DISCREET meta requirements and needs statement

## B.3 DISCREET compliance requirement needs

The following compliance requirements are drawn from the GDPR and include all articles addressing the potential functioning of a system. Full inclusion of all potential requirements in the *«modelLibrary»* is a core tenet of DISCREET and accordingly the only exclusions are those articles relating to issues of context, scope, or other issues outside of the system being modelled. Article one[2] sets the stage for the GDPR and is a prime example of such an exclusion. All the requirements included in this section flow from ID: CRNS by way of a *«deriveReqt»* relationship. Where possible, the exact wording of the English GDPR text will be used, though larger Articles that are split into more than one requirement will necessitate some changes purely for linguistic reasons. Lastly, DISCREET is intended for use in building IoT devices and systems, which may involve the collecting and processing of personal details, but not to deal with the trade in, or acquisition of, personal data. According to Article 14[3] of the GDPR, "Information to be provided where personal data have not been obtained from the data subject" is excluded.

### B.3.1 Article 4: Definitions

**NAME:**
G4: Definitions

**ID:**
G4

**TEXT:**
This model shall adopt the GDPR Article 4 definitions of personal data, processing, restriction of processing, profiling, pseudonymisation, filing system, controller, processor, recipient, third party, consent, personal data breach, genetic data, biometric data, data concerning health, main establishment, representative, enterprise, group of undertakings, binding corporate rules, supervisory authority, supervisory authority concerned, cross-border processing, relevant and reasoned objection, information society service, and international organisation.

**NOTES:**
The systems engineer may chose any additional route to include the Article 4 definitions or may view them as read. The text is located at: `https://gdpr.eu/article-4-definitions/`

### B.3.2 Article 5: Principles relating to processing of personal data

**NAME:**
G5.1.a: Lawful Processing

[2] Article one is as follows: "*This Regulation lays down rules relating to the protection of natural persons with regard to the processing of personal data and rules relating to the free movement of personal data.*"[125]

[3] Available here: `https://gdpr.eu/article-14-personal-data-not-obtained-from-data-subject/`

**ID:**
G5.1.a

**TEXT:**
Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility.

**NAME:**
G5.1.b: Purpose Limitation

**ID:**
G5.1.b

**TEXT:**
Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility. Please note that Article 89 allows for the purpose limitation to be relaxed in cases of archiving purposes in the public interest, scientific or historical research purposes or statistical purposes.

**NAME:**
G5.1.c: Data Minimisation

**ID:**
G5.1.c

**TEXT:**
Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility.

**NAME:**
G5.1.d: Accuracy

**ID:**
G5.1.d

**TEXT:**
Personal data shall be accurate and, where necessary, kept up to date; every reasonable step must

be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility.

**NAME:**
G5.1.e: Storage Limitation

**ID:**
G5.1.e

**TEXT:**
Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility. Personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89.

**NAME:**
G5.1.f: Integrity and Confidentiality

**ID:**
G5.1.f

**TEXT:**
Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures.

**NOTES:**
Paragraph 2 of Article 5 establishes this as the data controller's responsibility.

#### B.3.2.1 Article 6: Lawfulness of processing

**NAME:**
G6: Lawful Processing Check

**ID:**
G6

**TEXT:**
The processing of data shall only proceed if lawful.

**NOTES:**
The check for lawfulness specified in Article 6 is that at least one of the following must be the case: the data subject has given consent, processing is necessary for the performance of a contract to which the data subject is party, processing is necessary for compliance with a legal obligation to which the controller is subject, processing is necessary in order to protect the vital interests of the data subject or of another natural person, processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller, or processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child.

### B.3.3 Article 7: Conditions for consent

**NAME:**
G7.1: Consent

**ID:**
G7.1

**TEXT:**
Where processing is based on consent, the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data.

**NOTES:**
This requirement must be read in conjunction with Article 6: Lawfulness of processing, Article 8 Conditions applicable to child's consent in relation to information society services, Article 9: Processing of special categories of personal data, Article 10: Processing of personal data relating to criminal convictions and offences. Please pay special attention to the text of Article 9[4].

**NAME:**
G7.2: Consent Withdrawal

**ID:**
G7.2

**TEXT:**
The data subject shall have the right to withdraw his or her consent at any time.

**NOTES:**
The withdrawal of consent shall not affect the lawfulness of processing based on consent before its withdrawal. Prior to giving consent, the data subject shall be informed thereof. It shall be as easy to withdraw as to give consent. This requirement must be read in conjunction with

[4] https://gdpr.eu/article-9-processing-special-categories-of-personal-data-prohibited/

Article 6: Lawfulness of processing, Article 8 Conditions applicable to child's consent in relation to information society services, Article 9: Processing of special categories of personal data, Article 10: Processing of personal data relating to criminal convictions and offences. Please pay special attention to the text of Article 9, which can be found here: `https://gdpr.eu/article-9-processing-special-categories-of-personal-data-prohibited/`

### B.3.4 Article 8: Conditions applicable to child's consent in relation to information society services

**NAME:**
G8: Consent for Children

**ID:**
G8

**TEXT:**
The controller shall make reasonable efforts to verify that consent is given or authorised by the holder of parental responsibility over the child.

**NOTES:**
For children under the age of 16, parental consent must be given for the use of information society services.

### B.3.5 Article 11: Processing which does not require identification

**NAME:**
G11.1: Processing Anonymous Data

**ID:**
G11.1

**TEXT:**
The data controller shall not be required to identify data subjects if identification is not needed for the specified processing.

**NOTES:**
Dealing with fully anonymous data only, means that the GDPR no longer applies. However, please note Article 11 paragraph 2 as presented herein in G11.2(I) and G11.2(II).

**NAME:**
G11.2(I): Notice of Non-identification

**ID:**
G11.2(I)

**TEXT:**
The data controller shall inform the data subject, if possible, that it is not in a position to identify the data subject due to the data in question being anonymous.

**NOTES:**
Dealing with fully anonymous data only, means that the GDPR no longer applies. However, please note Article 11 paragraph 2 as presented herein and G11.2(II).

**NAME:**
G11.2(II): Identification Through Additional Information

**ID:**
G11.2(II)

**TEXT:**
Data subject's rights under the GDPR shall not apply to anonymous data except where the data subject, for the purpose of exercising their rights under those articles, provides additional information enabling their identification.

**NOTES:**
This provision impacts on G11.1 and G11.2(I), while data subject rights relate to Articles 15 to 20.

### B.3.6 Article 13: Information to be provided where personal data are collected from the data subject

**NAME:**
G13.1.a: Controller Details

**ID:**
G13.1.a

**TEXT:**
During the collection of personal information the controller shall provide the identity and the contact details of the controller and, where applicable, of the controller's representative.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/`

**NAME:**
G13.1.b: Data Protection Officer Details

**ID:**
G13.1.b

**TEXT:**
During the collection of personal information the controller shall provide the contact details of the data protection officer, where applicable.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/` Also see articles 37, 38, and 39 for further details on data protection officers.

**NAME:**
G13.1.c: Processing Details

**ID:**
G13.1.c

**TEXT:**
During the collection of personal information the controller shall provide the purposes of the processing for which the personal data are intended as well as the legal basis for the processing.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/`

**NAME:**
G13.1.d: Legitimate Interests Details

**ID:**
G13.1.d

**TEXT:**
During the collection of personal information the controller shall provide the details of any legitimate interest used as the basis of processing.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/` Legitimate interest is covered in Article 6(1)(f): `https://gdpr.eu/article-6-how-to-process-personal-data-le`

**NAME:**
G13.1.e: Data Recipient Details

**ID:**
G13.1.e

**TEXT:**
During the collection of personal information the controller shall provide the recipients or categories of recipients of the personal data, if any.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/`

**NAME:**
G13.1.f: Data Transfer Details

**ID:**
G13.1.f

**TEXT:**
During the collection of personal information the controller shall provide, where applicable, the fact that the controller intends to transfer personal data to a third country or international organisation and the existence or absence of an adequacy decision by the Commission, or reference to the appropriate or suitable safeguards and the means by which to obtain a copy of them or where they have been made available.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/`

**NAME:**
G13.2.a: Data Storage Period Details

**ID:**
G13.2.a

**TEXT:**
The controller shall, at the time when personal data are obtained, provide the data subject with the period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: `https://gdpr.eu/article-13-personal-data-collected/`

**NAME:**
G13.2.b: Data Access Details

**ID:**
G13.2.b

**TEXT:**
The controller shall, at the time when personal data are obtained, inform the data subject of the existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

**NAME:**
G13.2.c: Withdrawal of Consent Details

**ID:**
G13.2.c

**TEXT:**
The controller shall, at the time when personal data are obtained, inform the data subject of the existence of the right to withdraw consent at any time, without affecting the lawfulness of processing based on consent before its withdrawal.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

**NAME:**
G13.2.d: Supervisory Authority Details

**ID:**
G13.2.d

**TEXT:**
The controller shall, at the time when personal data are obtained, inform the data subject of the right to lodge a complaint with a supervisory authority.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

**NAME:**
G13.2.e: Statutory and Contractual Requirement Details

**ID:**
G13.2.e

**TEXT:**
The controller shall, at the time when personal data are obtained, inform the data subject whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

**NAME:**
G13.2.f: Automated Decision-making Details

**ID:**
G13.2.f

**TEXT:**
The controller shall, at the time when personal data are obtained, inform the data subject of the the existence of automated decision-making, including profiling and meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

**NAME:**
G13.3: Further Processing Details

**ID:**
G13.3

**TEXT:**
The data controller shall provide the data subject with details on any further processing of their personal data beyond the stated purpose, prior to such processing.

**NOTES:**
This provision does not apply if the data subject already has the information in question. Please see Article 13: https://gdpr.eu/article-13-personal-data-collected/

### B.3.7 Article 15: Right of access by the data subject

**NAME:**
G15.1.a: Processing Purposes

**ID:**
G15.1.a

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation as to whether or not their personal data are being processed.

**NOTES:**

**NAME:**
G15.1.b: Data Categories

**ID:**
G15.1.b

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation as to the categories of personal data concerned.

**NOTES:**

**NAME:**
G15.1.c: Data Recipients

**ID:**
G15.1.c

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation of the recipients or categories of recipient to whom the personal data have been or will be disclosed.

**NOTES:**

**NAME:**
G15.1.d: Data Retention Period

**ID:**
G15.1.d

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation of, where possible, the envisaged period for which the personal data will be stored, or, if not possible, the criteria used to determine that period.

**NOTES:**

**NAME:**
G15.1.e: Data Rectification and Erasure

**ID:**
G15.1.e

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation of the existence of the right to request from the controller rectification or erasure of personal data or restriction of processing of personal data.

**NOTES:**

**NAME:**
G15.1.f: Complaint Lodging

**ID:**
G15.1.f

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation of the right to lodge a complaint with a supervisory authority.

**NOTES:**

**NAME:**
G15.1.g: Other Data Sources

**ID:**
G15.1.g

**TEXT:**
The data subject shall have the right to obtain from the controller, where the personal data are not collected from the data subject, any available information as to their source.

**NOTES:**

**NAME:**
G15.1.h: Automated Decision-making Confirmation

**ID:**
G15.1.h

**TEXT:**
The data subject shall have the right to obtain from the controller confirmation of the existence of automated decision-making, including profiling, and meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject.

**NOTES:**

**NAME:**
G15.2: Data Transfer Safeguards

**ID:**
G15.2

**TEXT:**
The data subject shall have the right to be informed of the appropriate safeguards relating to data transferred to a third country or to an international organisation.

**NOTES:**

**NAME:**
G15.3: Data Copy Provision

**ID:**
G15.3

**TEXT:**
The controller shall provide a copy of the personal data undergoing processing.

**NOTES:**
Please note the additional stipulation: For any further copies requested by the data subject, the controller may charge a reasonable fee based on administrative costs. Where the data subject makes the request by electronic means, and unless otherwise requested by the data subject, the information shall be provided in a commonly used electronic form. Also, paragraph 4 of this Article holds that: The right to obtain a copy referred to in paragraph 3 shall not adversely affect the rights and freedoms of others.

### B.3.8 Article 16: Right to rectification

**NAME:**
G16: Data Rectification

**ID:**
G16

**TEXT:**
The data subject shall have the right to obtain from the controller without undue delay the rectification of inaccurate personal data.

**NOTES:**
This includes the updating of incomplete data, with the onus on the data controller to facilitate such amendments.

### B.3.9 Article 17: Right to erasure

**NAME:**
G17.1.a: Erasure of Unnecessary Data

**ID:**
G17.1.a

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal data if the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): `https://gdpr.eu/article-17-right-to-be-forgotten/`

**NAME:**
G17.1.b: Erasure After Withdrawal of Consent

**ID:**
G17.1.b

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal data if the data subject withdraws consent on which the processing is based.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): `https://gdpr.eu/article-17-right-to-be-forgotten/`

**NAME:**
G17.1.c: Erasure due to Objection

**ID:**
G17.1.c

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal

data if the data subject objects to the processing and there are no overriding legitimate grounds for the processing.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): https://gdpr.eu/article-17-right-to-be-forgotten/

**NAME:**
G17.1.d: Erasure of Unlawfully Processed Data

**ID:**
G17.1.d

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal data if the personal data have been unlawfully processed.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): https://gdpr.eu/article-17-right-to-be-forgotten/

**NAME:**
G17.1.e: Erasure due to Compliance Obligations

**ID:**
G17.1.e

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal data if the personal data have to be erased for compliance with a legal obligation.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): https://gdpr.eu/article-17-right-to-be-forgotten/

**NAME:**
G17.1.f: Erasure of Data Belonging to Children Using ISS

**ID:**
G17.1.f

**TEXT:**
The data subject shall have the right to obtain from the controller the erasure of their personal data if the personal data have been collected in relation to the offer of information society services to children.

**NOTES:**
Please note, information society services are "Any service normally provided for remuneration, at a distance, by electronic means and at the individual request of a recipient of services." The practitioner can find more information in the ICO discussion [5]. However, this provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): https://gdpr.eu/article-17-right-to-be-forgotten/

**NAME:**
G17.2: Erasure of Data Made Public

**ID:**
G17.2

**TEXT:**
A request for the erasure of data which has been made public, shall be met with erasure and informing any known third parties of such erasure.

**NOTES:**
This provision is undone if processing is needed for the right of freedom of expression and information, for compliance with a legal obligation, for reasons of public interest in the area of public health, or archiving purposes in the public interest, scientific or historical research purposes or statistical purposes, or for the establishment, exercise or defence of legal claims. Please see Article 17(3): https://gdpr.eu/article-17-right-to-be-forgotten/

### B.3.10 Article 18: Right to restriction of processing

**NAME:**
G18.1.a: Contested Data Processing Restriction

**ID:**
G18.1.a

[5] https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/children-and-the-uk-gdpr/what-are-the-rules-about-an-iss-and-consent/

**TEXT:**
The data subject shall have the right to obtain from the controller restriction of processing if the accuracy of the personal data is contested by the data subject, for a period enabling the controller to verify the accuracy of the personal data.

**NOTES:**
After such restriction and excluding data storage, personal data can only be processed with the data subject's consent or for legal and public interest purposes.

**NAME:**
G18.1.b: Lawfulness Processing Restriction

**ID:**
G18.1.b

**TEXT:**
The data subject shall have the right to obtain from the controller restriction of processing if the processing is unlawful and the data subject opposes the erasure of the personal data and requests the restriction of their use instead.

**NOTES:**
After such restriction and excluding data storage, personal data can only be processed with the data subject's consent or for legal and public interest purposes.

**NAME:**
G18.1.c: Processing Restriction on Unnecessary Data

**ID:**
G18.1.c

**TEXT:**
The data subject shall have the right to obtain from the controller restriction of processing if the controller no longer needs the personal data for processing, but they are required by the data subject for the establishment, exercise or defence of legal claims.

**NOTES:**
After such restriction and excluding data storage, personal data can only be processed with the data subject's consent or for legal and public interest purposes.

**NAME:**
G18.1.d: Processing Restriction due to Objection

**ID:**
G18.1.d

**TEXT:**
The data subject shall have the right to obtain from the controller restriction of processing if the data subject has objected to processing and verification of legitimacy for that claim is pending.

**NOTES:**
This requirement halts processing, while an objection to processing is processed. After such restriction and excluding data storage, personal data can only be processed with the data subject's consent or for legal and public interest purposes.

**NAME:**
G18.3: Lifting of Processing Restrictions

**ID:**
G18.3

**TEXT:**
The controller shall provide advance notification to data subjects if any prior restriction on processing is to be lifted.

**NOTES:**

### B.3.11 Article 19: Notification obligation regarding rectification or erasure of personal data or restriction of processing

**NAME:**
G19: Notification Obligation to Third Parties

**ID:**
G19

**TEXT:**
The controller shall communicate any rectification or erasure of personal data or restriction of processing carried out to each recipient to whom the personal data have been disclosed and inform the data subject about said recipients, if requested.

**NOTES:**
Compliance is not necessary if: "this proves impossible or involves disproportionate effort".

### B.3.12 Article 20: Right to data portability

**NAME:**
G20.1.a: Consent Based Data Portability

**ID:**
G20.1.a

**TEXT:**
The data subject shall have the right to receive their personal data from the controller in a structured, commonly used and machine-readable format, where the data subject consented to processing.

**NOTES:**
Data subjects may also request that one controller provides these data directly to another controller, but only if this is technically feasible.

**NAME:**
G20.1.b: Data Portability for Automated Processing

**ID:**
G20.1.b

**TEXT:**
The data subject shall have the right to receive their personal data from the controller in a structured, commonly used and machine-readable format, where processing was conducted by automated means.

**NOTES:**
Data subjects may also request that one controller provides these data directly to another controller, but only if this is technically feasible.

### B.3.13 Article 21: Right to object

**NAME:**
G21: Right to Object

**ID:**
G21

**TEXT:**
The data subject shall have the right to object, at any time to processing of personal data concerning them and shall be informed of this right by the data controller.

**NOTES:**
Any such requests must be responded to within one calendar month. Were the personal data are processed for direct marketing purposes, the right to object is absolute. Beyond such purposes

the request must be judged against Article 6(1)(e) and (f), as well as Article 89(1). The ICO provides a guide[6]

### B.3.14 Article 22: Automated individual decision-making, including profiling

**NAME:**
G22: Recourse Against Automated Decision-making

**ID:**
G22

**TEXT:**
The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling.

**NOTES:**
This right is curtailed if such processing is needed to meet contractual or legal obligations, or is based on the data subject's explicit consent. If the processing involved includes "special categories of personal data" or there is other uncertainty, then it is strongly advised that the modeller draws in the ICO Article 22 checklist to guide compliance. The checklist is available on the ICO website[7]

### B.3.15 Article 28: Processor

**NAME:**
G28: Processor Requirements

**ID:**
G28

**TEXT:**
The controller shall only contract in processors who can provide sufficient guarantees to implement appropriate technical and organisational measures in such a manner that processing will meet GDPR requirements.

**NOTES:**
This also means that the processor can only act on the instruction of the controller and not beyond or without it, and only once a contract is in place. Both controller and processor must take note of the extended Article 28 requirements which can be found here: `https://gdpr.eu/article-28-processor/`

---

[6] `https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/individual-rights/right-to-object/`

[7] `https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/individual-rights/rights-related-to-automated-decision-making-including-profiling/`

### B.3.16 Article 44: General principle for transfers

**NAME:**
G44: Data Transfer Principles

**ID:**
G44:

**TEXT:**
The controller shall only transfer data internationally or to third parties if all provisions for data transfer are met.

**NOTES:**
This is an extensive provision which requires that all provisions of articles 45, 46, 47, 48, 49, and 50 are met. More information is provided on the gdpr.eu site[8]. It is however, recommended that the modeller also consult Section 2.2.7 in the CNIL methodology's IoT guidance[9] and the main methodology at the same section number[10].

## B.4 DISCREET audit requirements

All the requirements in this section flow from ID:ARNS by way of a *«deriveReqt»* relationship. Each of the following requirements include one or more application tips to aid the modeller. These specifically include pointers to which GDPR requirements are being tested by including the relevant DISCREET IDs. Also, following the same logic as with compliance requirements, not all stipulations are included as requirements as a significant portion do not deal with devices or systems but rather with organisational or structural issues. Finally, the CNIL PIA groups requirements into larger numbered sections, divided into unnumbered check marks which are in turn divided into numbered sub-points, with only these sub-points being actionable requirements and the rest being guidance.

Where a further subdivision is needed to form these requirements into single shall statements, Roman numerals are appended in brackets. Lastly, under DISCREET, each of the following requirements will, in conjunction with the previously introduced meta requirements, spawn a compliance artefact containing the required data. As basis for these the applicable CNIL templates[11] should be used. A further aid for the model builder is the CNIL knowledge base[12] which provides a catalogue of potential compliance controls. The final point to note is that the CNIL's PIA

[8] `https://gdpr.eu/article-44-transfer-of-personal-data/`

[9] `https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-piaf-connectedobjects-en.pdf`

[10] `https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-1-en-methodology.pdf`

[11] Available from the CNIL PIA downloads page, or directly here: `https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-2-en-templates.pdf`

[12] Available form the CNIL PIA downloads page, or directly here: `https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-3-en-knowledgebases.pdf`

application for IoT devices comes after the fact, in a manner of speaking. The PIA methodology is used in all cases, but for IoT implementations, the "Application to IoT Devices" guidance[13] is used to fill in the templates provided. At a minimum then, and although the CNIL material is referenced, this section of DISCREET should be used to generate compliance reporting artefacts that adhere to the format of the CNIL's IoT device application.

### B.4.1 CNIL: Assessment of the controls governing processing

**NAME:**
C2.1.1: Purpose Limitation Assessment

**ID:**
C2.1.1

**TEXT:**
The controller shall explain and justify their choices with regards to the purpose limitation.

**NOTES:**
This requirement tests compliance to DISCREET requirement G5.1.b.

**NAME:**
C2.1.2(I): Lawfulness Assessment for Processing

**ID:**
C2.1.2(I)

**TEXT:**
The controller shall explain and justify their choices with regards to the lawfulness of the processing conducted.

**NOTES:**
This requirement tests compliance to DISCREET requirement G5.1.a.

**NAME:**
C2.1.2(II): Lawfulness Assessment for Integrity and Confidentiality

**ID:**
C2.1.2(II)

**TEXT:**
The controller shall explain and justify their choices with regards to the lawfulness of the processing conducted with reference to integrity and confidentiality.

[13] Available form the CNIL PIA downloads page, or directly here: `https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-piaf-connectedobjects-en.pdf`

**NOTES:**
This requirement tests compliance to DISCREET requirement G5.1.f.

**NAME:**
C2.1.2(III): Lawfulness Assessment

**ID:**
C2.1.2(III)

**TEXT:**
The controller shall explain and justify their choices with regards to the lawfulness of the processing conducted with reference to the Article 6 lawfulness check.

**NOTES:**
This requirement tests compliance to DISCREET requirement G6.

**NAME:**
C2.1.3: Data Minimisation Assessment

**ID:**
C2.1.3

**TEXT:**
The controller shall explain and justify their choices with regards to data minimisation.

**NOTES:**
The targets here are to keep data adequate, relevant, and limited. This requirement tests compliance to DISCREET requirement G5.1.c.

**NAME:**
C2.1.4: Data Accuracy Assessment

**ID:**
C2.1.4

**TEXT:**
The controller shall explain and justify their choices with regards to data quality.

**NOTES:**
Data quality in this sense includes bot accuracy and being up to date. This requirement tests compliance to DISCREET requirement G5.1.d.

**NAME:**
C2.1.5: Data Storage Assessment

**ID:**
C2.1.5

**TEXT:**
The controller shall explain and justify their choices with regards to data storage.

**NOTES:**
This requirement tests compliance to DISCREET requirement G5.1.e.

### B.4.2 CNIL: Assessment of the controls protecting data subjects' rights

**NAME:**
C2.2.1(I): Information Provision Assessment for Processing Anonymous Data

**ID:**
C2.2.1(I)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G11.1.

**NAME:**
C2.2.1(II): Information Provision Assessment for Non-identification

**ID:**
C2.2.1(II)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G11.2(I).

**NAME:**
C2.2.1(III): Information Provision Assessment for Identification Through Additional Information

**ID:**
C2.2.1(III)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G11.2(II).

**NAME:**
C2.2.1(IV): Information Provision Assessment for Controller Details

**ID:**
C2.2.1(IV)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.a.

**NAME:**
C2.2.1(V): Information Provision Assessment for Data Protection Officer Details

**ID:**
C2.2.1(V)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.b.

**NAME:**
C2.2.1(VI): Information Provision Assessment for Processing Details

**ID:**
C2.2.1(VI)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.c.

**NAME:**
C2.2.1(VII): Information Provision Assessment for Legitimate Interests Details

**ID:**
C2.2.1(VII)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.d.

**NAME:**
C2.2.1(VIII): Information Provision Assessment for Data Recipient Details

**ID:**
C2.2.1(VIII)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.e.

**NAME:**
C2.2.1(IX): Information Provision Assessment for Data Transfer Details

**ID:**
C2.2.1(IX)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.1.f.

**NAME:**
C2.2.1(X): Information Provision Assessment for Data Storage Period Details

**ID:**
C2.2.1(X)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.a.

**NAME:**
C2.2.1(XI): Information Provision Assessment for Data Access Details

**ID:**
C2.2.1(XI)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.b.

**NAME:**
C2.2.1(XII): Information Provision Assessment for Withdrawal of Consent Details

**ID:**
C2.2.1(XII)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.c.

**NAME:**
C2.2.1(XIII): Information Provision Assessment for Supervisory Authority Details

**ID:**
C2.2.1(XIII)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.d.

**NAME:**
C2.2.1(XIV): Information Provision Assessment for Statutory and Contractual Requirement Details

**ID:**
C2.2.1(XIV)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.e.

**NAME:**
C2.2.1(XV): Information Provision Assessment for Automated Decision-making Details

**ID:**
C2.2.1(XV)

**TEXT:**
The controller shall identify and describe the controls for compliance with the provision of information to data subjects.

**NOTES:**
This requirement tests compliance to DISCREET requirement G13.2.f.

**NAME:**
C2.2.2(I): Consent Assessment

**ID:**
C2.2.2(I)

**TEXT:**
The controller shall identify and describe the controls used in obtaining and maintaining consent.

**NOTES:**
This requirement tests that consent was obtained but also that the withdrawal thereof was correctly managed. This is compliance to DISCREET requirement G7.1.

**NAME:**
C2.2.2(II): Consent Withdrawal Assessment

**ID:**
C2.2.2(II)

**TEXT:**
The controller shall identify and describe the controls used in obtaining and maintaining consent.

**NOTES:**
This requirement tests that consent was obtained but also that the withdrawal thereof was correctly managed. This is compliance to DISCREET requirement G7.2.

**NAME:**
C2.2.2(III): Consent Assessment for Children

**ID:**
C2.2.2(III)

**TEXT:**
The controller shall identify and describe the controls used in obtaining and maintaining consent.

**NOTES:**
This requirement tests that consent was obtained but also that the withdrawal thereof was correctly managed. This is compliance to DISCREET requirement G8.

**NAME:**
C2.2.3(I): Data Access Assessment for Processing Purposes

**ID:**
C2.2.3(I)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.a.

**NAME:**
C2.2.3(II): Data Access Assessment for Data Categories

**ID:**
C2.2.3(II)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.b.

**NAME:**

C2.2.3(III): Data Access Assessment for Data Recipients

**ID:**
C2.2.3(III)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.c.

**NAME:**
C2.2.3(IV): Data Access Assessment for Data Retention Period

**ID:**
C2.2.3(IV)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.d.

**NAME:**
C2.2.3(V): Data Access Assessment for Data Rectification and Erasure

**ID:**
C2.2.3(V)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.e.

**NAME:**
C2.2.3(VI): Data Access Assessment for Complaint Lodging

**ID:**
C2.2.3(VI)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirements G15.1.f.

**NAME:**
C2.2.3(VII): Data Access Assessment for Other Data Sources

**ID:**
C2.2.3(VII)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.g.

**NAME:**
C2.2.3(VIII): Data Access Assessment for Automated Decision-making Confirmation

**ID:**
C2.2.3(VIII)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.1.h.

**NAME:**
C2.2.3(IX): Data Access Assessment for Data Transfer Safeguards

**ID:**
C2.2.3(IX)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G15.2.

**NAME:**
C2.2.3(X): Data Access Assessment for Data Copy Provision

**ID:**
C2.2.3(X)

**TEXT:**
The controller shall identify and describe the controls for compliance with data access requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirements G15.3.

**NAME:**
C2.2.3(XI): Consent Based Data Portability Assessment

**ID:**
C2.2.3(XI)

**TEXT:**
The controller shall identify and describe the controls for compliance with data portability requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G20.1.a.

**NAME:**
C2.2.3(XII): Data Portability Assessment for Automated Processing

**ID:**
C2.2.3(XII)

**TEXT:**
The controller shall identify and describe the controls for compliance with data portability requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G20.1.b.

**NAME:**
C2.2.4(I): Rectification of Data Assessment

**ID:**
C2.2.4(I)

**TEXT:**
The controller shall identify and describe the controls for compliance with data rectification requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirements G16 and G19.

**NAME:**
C2.2.4(I): Rectification of Data Assessment

**ID:**
C2.2.4(I)

**TEXT:**
The controller shall identify and describe the controls for compliance with data rectification requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G16.

**NAME:**
C2.2.4(II): Assessment of Rectification Notification Obligation to Third Parties

**ID:**
C2.2.4(II)

**TEXT:**
The controller shall identify and describe the controls for compliance with data rectification requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G19.

**NAME:**
C2.2.4(III): Assessing Erasure of Unnecessary Data

**ID:**
C2.2.4(III)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.a.

**NAME:**
C2.2.4(IV): Data Erasure Assessment for Consent Withdrawal

**ID:**
C2.2.4(IV)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.b.

**NAME:**
C2.2.4(V): Data Erasure Assessment for Objections

**ID:**
C2.2.4(V)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.c.

**NAME:**
C2.2.4(VI): Data Erasure Assessment for Unlawfully Processed Data

**ID:**
C2.2.4(VI)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.d.

**NAME:**
C2.2.4(VII): Assessment of Erasure due to Compliance Obligations

**ID:**
C2.2.4(VII)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.e.

**NAME:**
C2.2.4(VIII): Data Erasure Assessment for Data Belonging to Children Using ISS

**ID:**
C2.2.4(VIII)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.1.f.

**NAME:**
C2.2.4(IX): Erasure Assessment of Data Made Public

**ID:**
C2.2.4(IX)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G17.2.

**NAME:**
C2.2.4(X): Data Erasure Assessment Relating to Third Party Notification Obligation

**ID:**
C2.2.4(X)

**TEXT:**
The controller shall identify and describe the controls for compliance with data erasure requirements.

**NOTES:**
This requirement tests compliance to DISCREET requirement G19.

**NAME:**
C2.2.5(I): Processing Restriction Assessment for Contested Data

**ID:**
C2.2.5(I)

**TEXT:**
The controller shall identify and describe the controls for compliance with requirements for the restriction of data processing.

**NOTES:**
This requirement tests compliance to DISCREET requirement G18.1.a.

**NAME:**
C2.2.5(II): Processing Restriction Assessment for Lawful Processing

**ID:**
C2.2.5(II)

**TEXT:**
The controller shall identify and describe the controls for compliance with requirements for the restriction of data processing.

**NOTES:**
This requirement tests compliance to DISCREET requirement G18.1.b.

**NAME:**
C2.2.5(III): Processing Restriction Assessment for Unnecessary Data

**ID:**
C2.2.5(III)

**TEXT:**
The controller shall identify and describe the controls for compliance with requirements for the restriction of data processing.

**NOTES:**
This requirement tests compliance to DISCREET requirement G18.1.c.

**NAME:**
C2.2.5(IV): Processing Restriction Assessment for Objection

**ID:**
C2.2.5(IV)

**TEXT:**
The controller shall identify and describe the controls for compliance with requirements for the restriction of data processing.

**NOTES:**
This requirement tests compliance to DISCREET requirement G18.1.d.

**NAME:**
C2.2.5(V): Processing Restriction Assessment for Lifting of Processing Restrictions

**ID:**
C2.2.5(V)

**TEXT:**
The controller shall identify and describe the controls for compliance with requirements for the restriction of data processing.

**NOTES:**
This requirement tests compliance to DISCREET requirement G18.3.

**NAME:**
C2.2.5(VI): Right to Objection Assessment

**ID:**
C2.2.5(VI)

**TEXT:**
The controller shall identify and describe the control for enabling the data subject's right to objection.

**NOTES:**
This requirement tests compliance to DISCREET requirement G21.

**NAME:**
C2.2.6: Processor Details Assessment

**ID:**
C2.2.6

**TEXT:**
The controller shall identify and describe the controls for compliance with the need to identify and formalise the interaction with all processors.

**NOTES:**
This requirement tests compliance to DISCREET requirement G28.

**NAME:**
C2.2.7: Data Transfer Assessment

**ID:**
C2.2.7

**TEXT:**
The controller shall identify and describe the controls for compliance with all requirements relating to data transfers outside the European Union.

**NOTES:**
This requirement tests compliance to DISCREET requirement G44.

## B.5 DISCREET PbD requirements

All the requirements in this section flow from ID:PRNS. ID:L1 by way of a *«deriveReqt»* relationship to ID:PRNS, and ID:L2 by way of a *«deriveReqt»* relationship to ID:L1. LINDDUN is intended for iterative application throughout the device or system lifecycle and as such, can be applied and reapplied as and when needed.

**NAME:**
L1: LINDDUN Use

**ID:**
L1

**TEXT:**
The LINDDUN threat modelling methodology shall be utilised in accordance with the DISCREET Privacy by Design trace method.

**NOTES:**

**NAME:**
L2: LINDDUN Results Implementation

**ID:**
L2

**TEXT:**

Results from the LINDDUN threat modelling methodology, including requirement formulation and PET recommendations, shall be imported into the system model as appropriate.

**NOTES:**

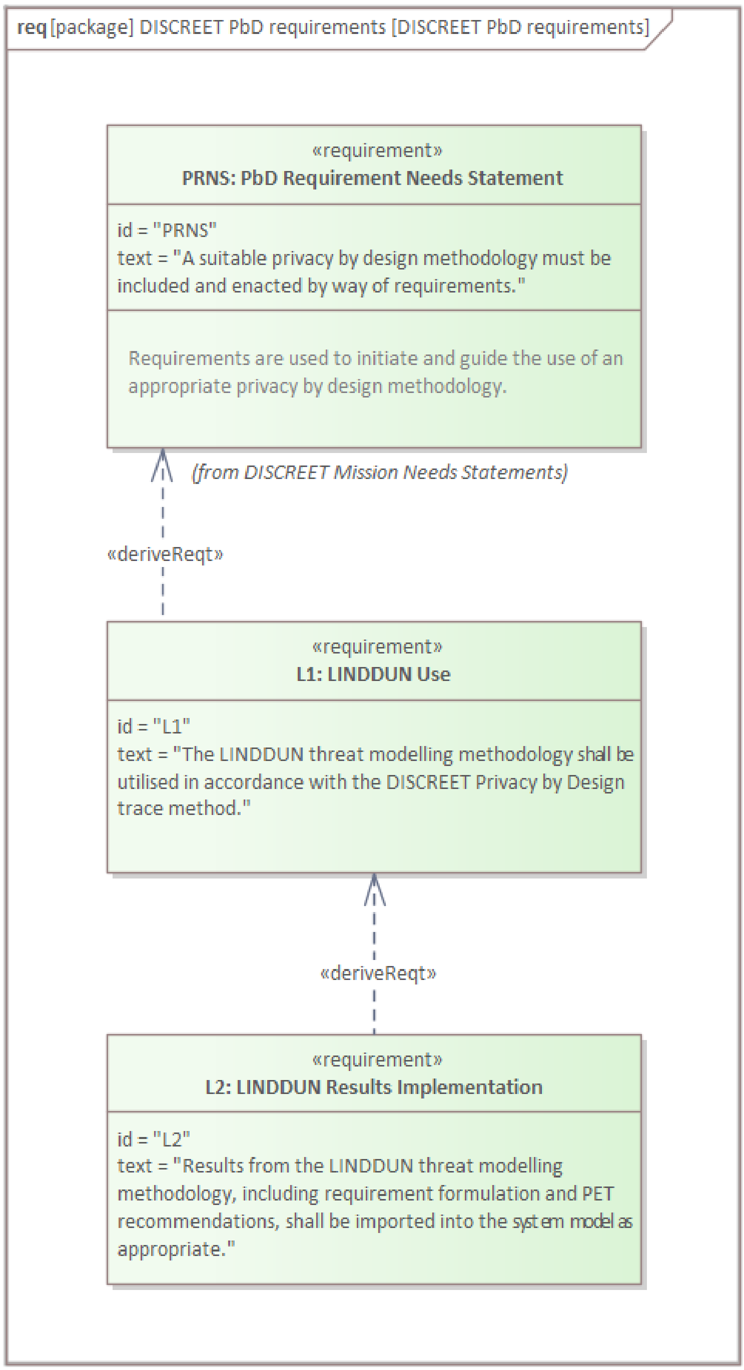



Figure B.3: DISCREET Privacy by Design requirements

# Appendix C

# LINDDUN Privacy Threat Tree Catalogue

## C.1 Threat tree use

The threat tree is included herein for ease of reference but does not contain the full description of all its elements. These, plus the needed discussion thereof, can be found in the LINDDUN Privacy Threat Tree Catalogue V2 which is available for download[1]. Since the threat trees are in the first instance only developed in so far as a specific threat is mapped to a given DFD element, the mapping is also included below in Figure C.1.

| | L | I | N | D | D | U | N |
|---|---|---|---|---|---|---|---|
| Data store | X | X | X | X | X | | X |
| Data flow | X | X | X | X | X | | X |
| Process | X | X | X | X | X | | X |
| Entity | X | X | | | | X | |

Figure C.1: LINDDUN mapping table

## C.2 Linkability

All four DFD element types have linkability concerns.

[1]https://7e71aeba-b883-4889-aee9-a3064f8be401.filesusr.com/ugd/cc602e_d7cf949767b7486d8bff0ecc05b91db6.pdf

## Linkability of an entity

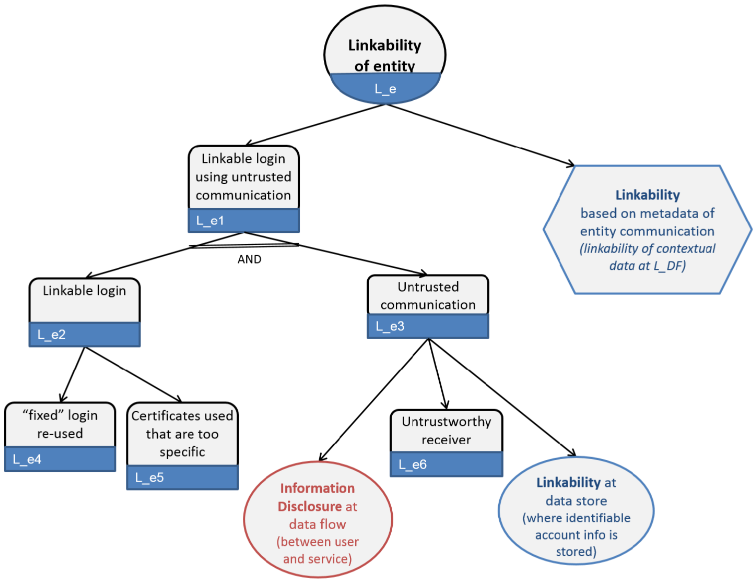



Figure C.2: LINDDUN entity linkability threat tree

## Linkability of a dataflow

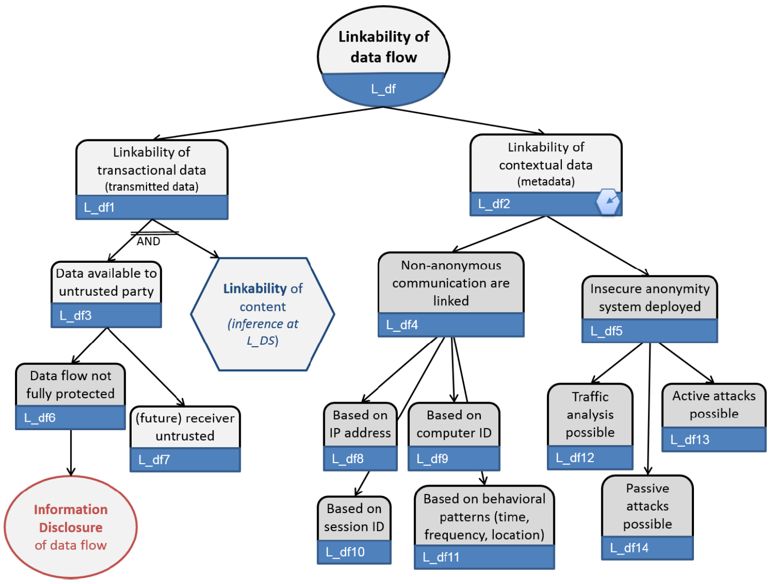



Figure C.3: LINDDUN data flow linkability threat tree

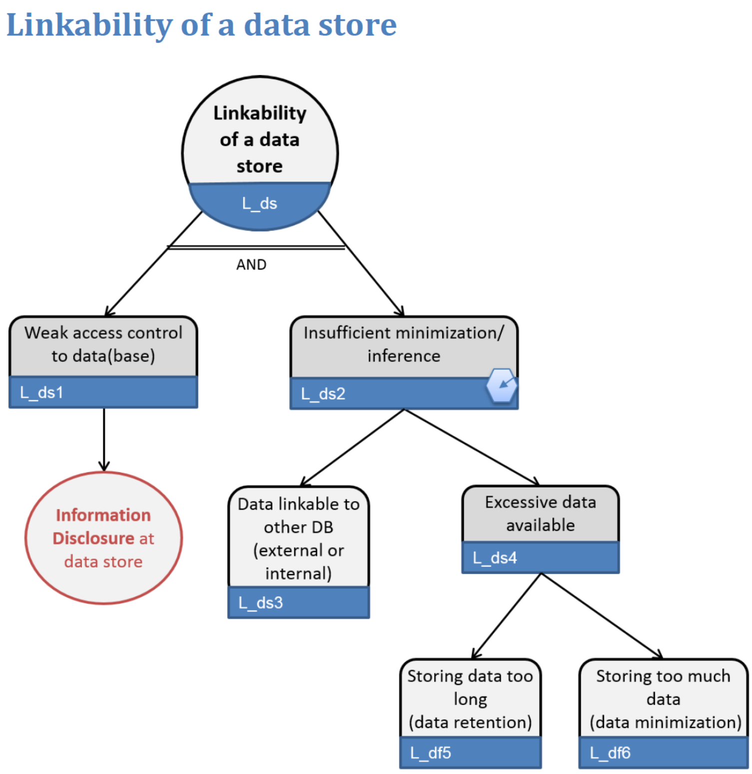



Figure C.4: LINDDUN data store linkability threat tree

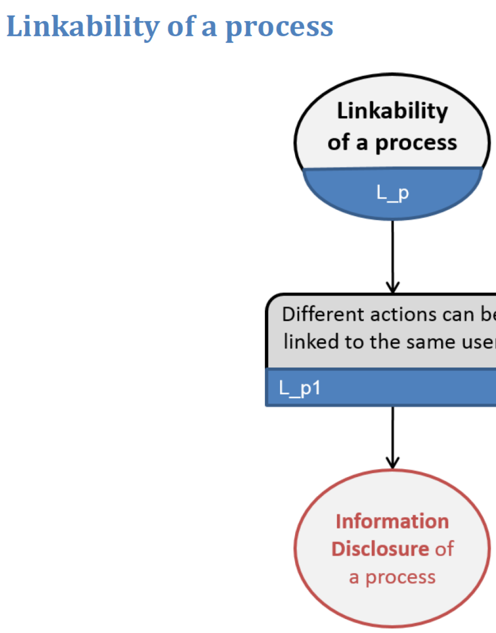



Figure C.5: LINDDUN process linkability threat tree

## C.3 Identifiability

All four DFD element types have identifiability concerns.

### Identifiability of entity

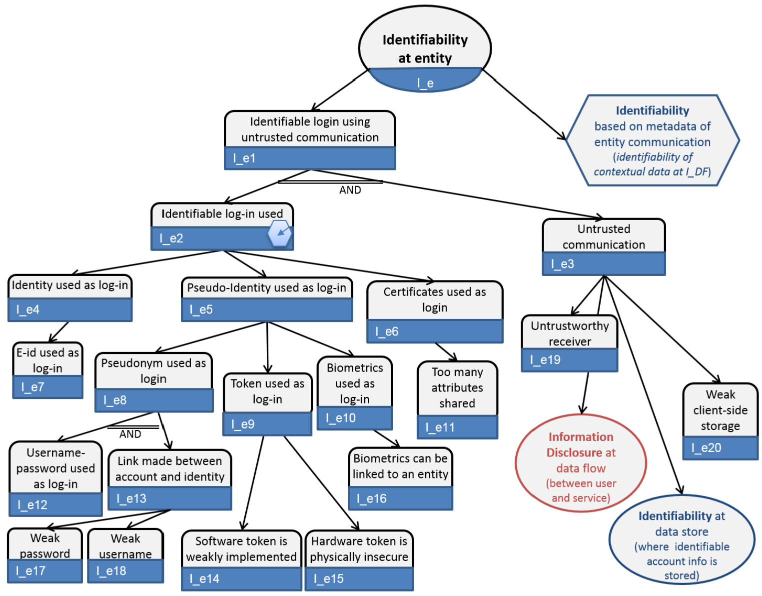



Figure C.6: LINDDUN entity identifiability threat tree

### Identifiability of data flow

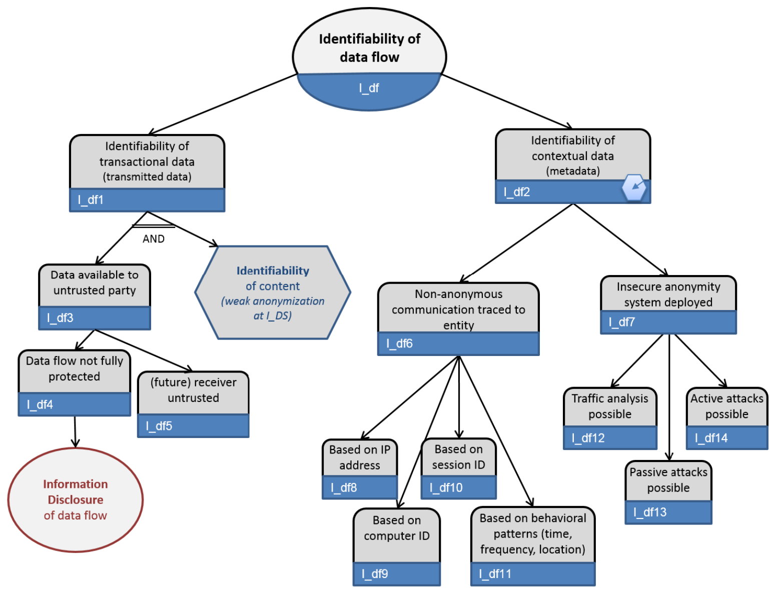



Figure C.7: LINDDUN data flow identifiability threat tree

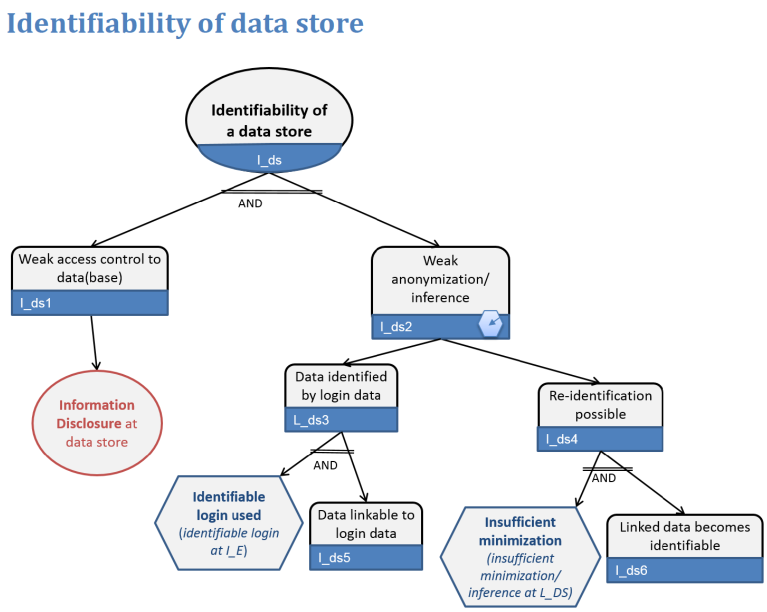



Figure C.8: LINDDUN data store identifiability threat tree

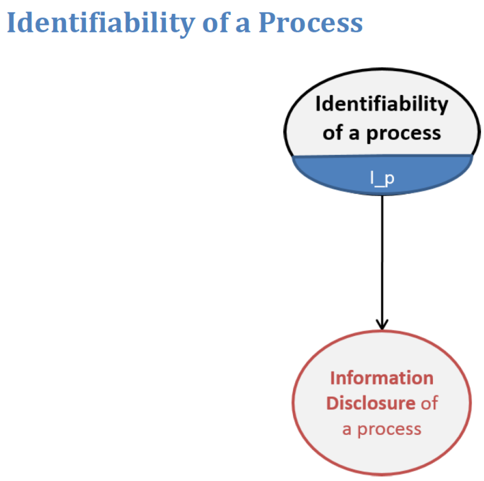



Figure C.9: LINDDUN process identifiability threat tree

## C.4 Non-repudiation

Three of the DFD element types have non-repudiation concerns, with “entity” being the odd one out.

## Non-repudiation of data flow

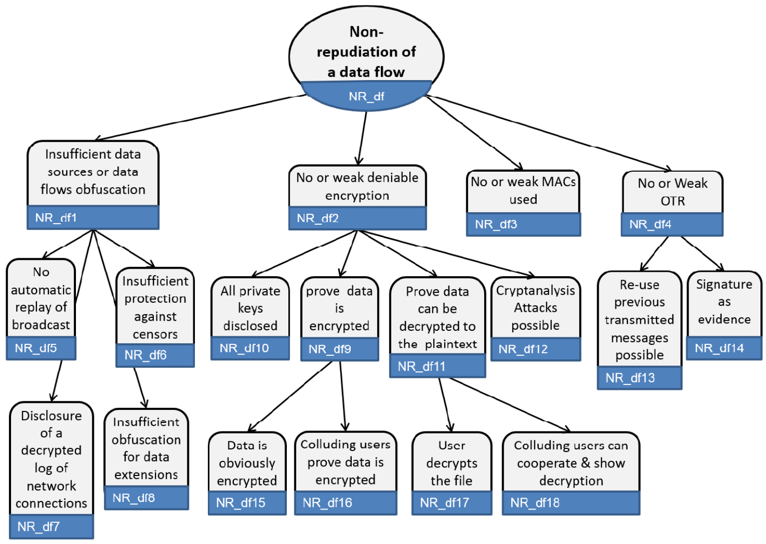



Figure C.10: LINDDUN data flow non-repudiation threat tree

## Non-repudiation of data store

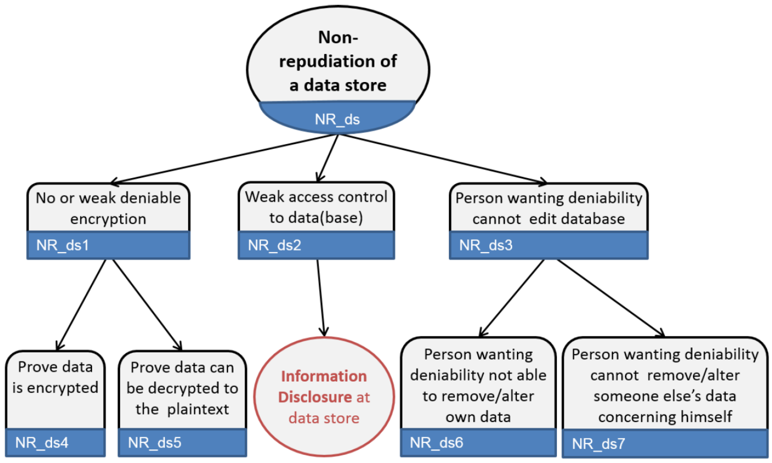



Figure C.11: LINDDUN data store non-repudiation threat tree

### Non-repudiation of a process

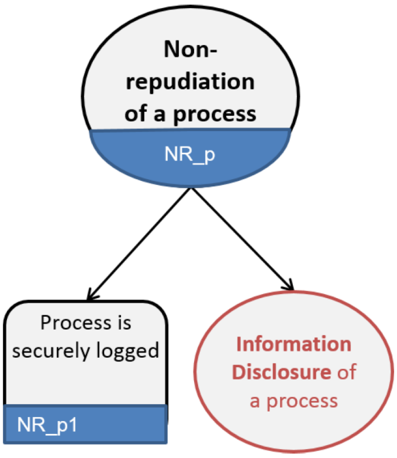



Figure C.12: LINDDUN process non-repudiation threat tree

## C.5 Detectability

Three of the DFD element types have detectability, with “entity” being the odd one out.

### Detectability of a dataflow

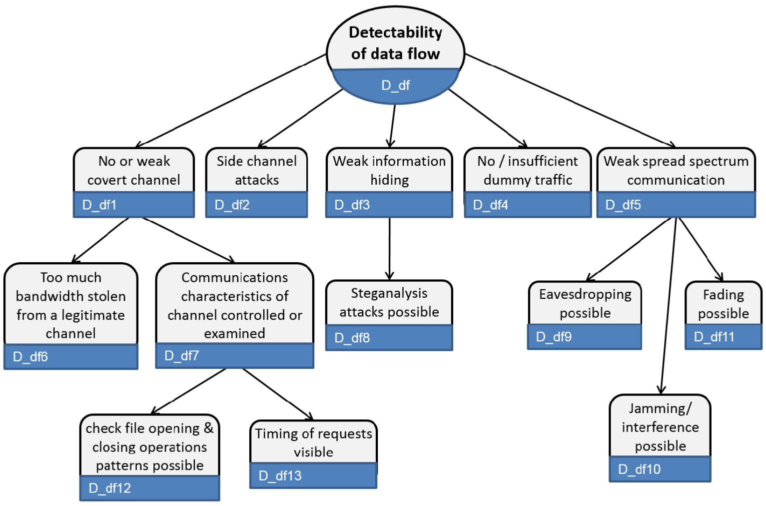



Figure C.13: LINDDUN data flow detectability threat tree

### Detectability of a Data store

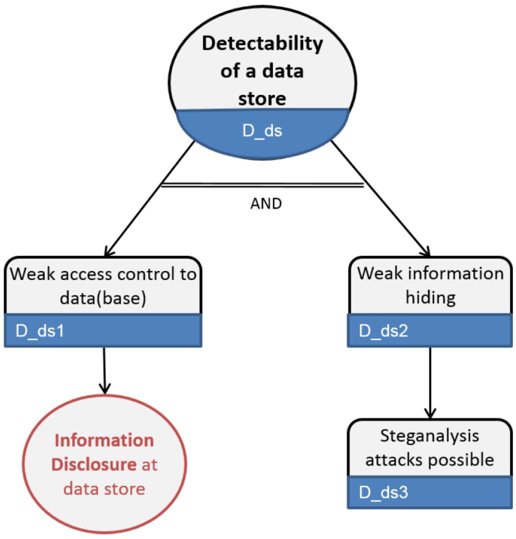



Figure C.14: LINDDUN data store detectability threat tree

### Detectability of a Process

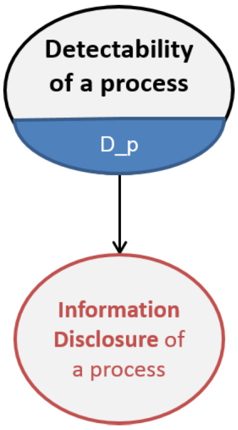



Figure C.15: LINDDUN process detectability threat tree

## C.6 Disclosure of Information

Three of the DFD element types have disclosure of information concerns, with "entity" being the odd one out. However, only one tree is presented as this is drawn in from STRIDE and is the same for all applicable DFD elements.

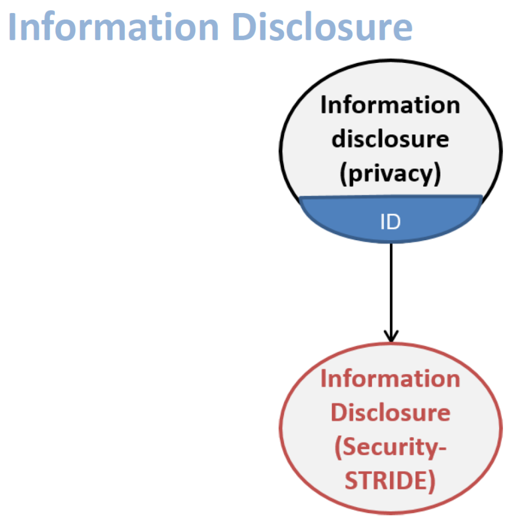



Figure C.16: LINDDUN disclosure of information threat tree

## C.7 Unawareness

Only the “entity” DFD element presents with unawareness concerns.

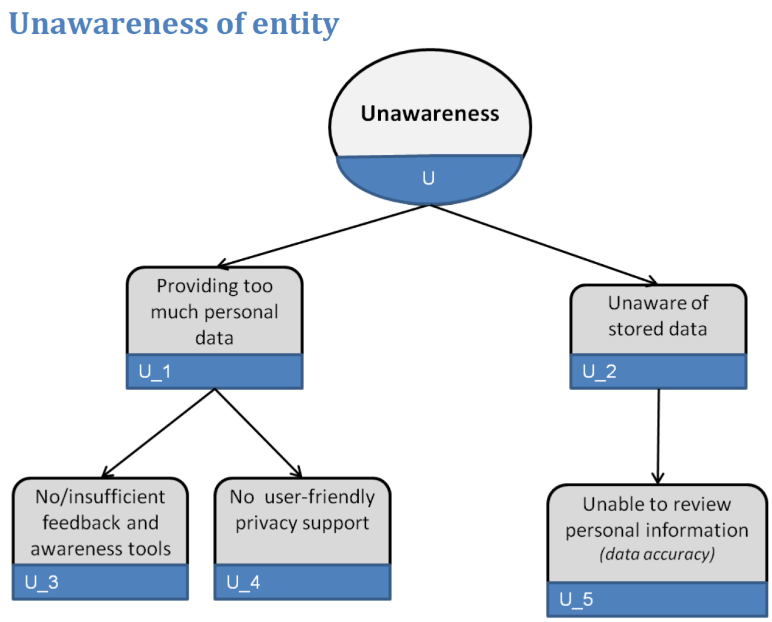



Figure C.17: LINDDUN entity unawareness threat tree

## C.8 Non-compliance

Three of the DFD element types have non-compliance concerns, with “entity” being the odd one out. However, for non-compliance LINDDUN only presents a single tree to be used for all affected elements.

## Non-Compliance

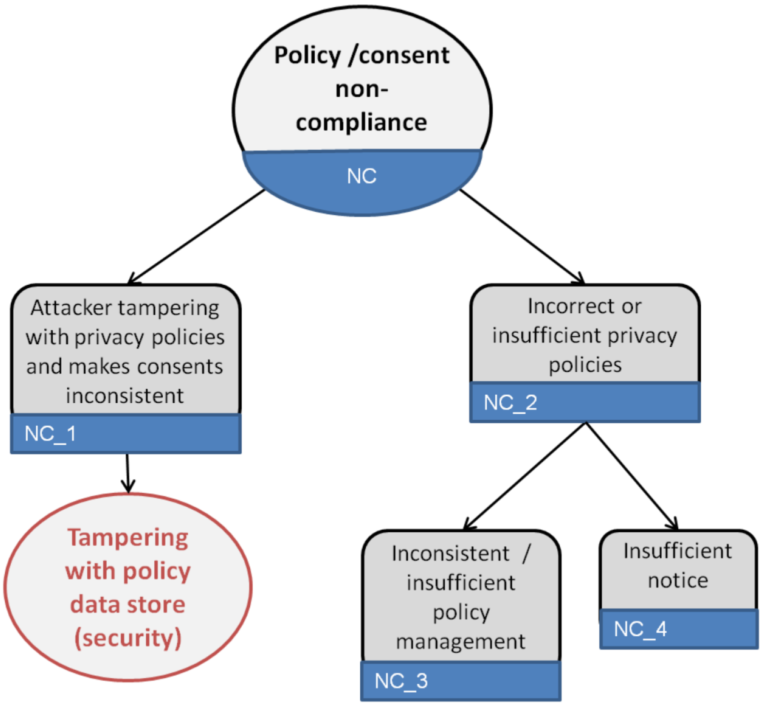



Figure C.18: LINDDUN entity non-compliance threat tree

# References


[1] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. Privacy and Human Behavior in the Age of Information. *Science*, 347(6221):509–514, 2015.

[2] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Mousa Ayyash. The internet of things: a Survey on Enabling Technologies, Protocols, and Applications. *IEEE COMMUNICATION SURVEYS & TUTORIALS*, 17(April 2014):243–259, 2015. .

[3] Shorouq Alansari. *PERSONA : A Decentralized Framework for Personalized Data Protection*. PhD thesis, University of Southampton, 2016.

[4] Katherine Albrecht and Liz McIntyre. Privacy Nightmare: When Baby Monitors Go Bad [Opinion]. *IEEE Technology and Society Magazine*, 34(3):14–19, 2015. ISSN 02780097. .

[5] Arwa Y Aleryani. Comparative Study Between Data Flow Diagram and Use Case Diagram. *International Journal of Scientific and Research Publications*, 6(3):124–126, 2016.

[6] Israa Alqassem and Davor Svetinovic. A taxonomy of security and privacy requirements for the Internet of Things (IoT). In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1244–1248. IEEE, 2014.

[7] Badr Alsamani and Husam Lahza. A taxonomy of IoT: Security and privacy threats. In *2018 International Conference on Information and Computer Technologies (ICICT)*, pages 72–77. IEEE, 2018.

[8] Nishadh Aluthge. *IoT device fingerprinting with sequence-based features Department of Computer Science*. PhD thesis, UNIVERSITY OF HELSINKI, 2017.

[9] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[10] Kevin Ashton et al. That 'internet of things' thing. *RFID journal*, 22(7):97–114, 2009.

[11] Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, and Ramjee Prasad. Proposed security model and threat taxonomy for the Internet of Things (IoT). In *International Conference on Network Security and Applications*, pages 420–429. Springer, 2010.

[12] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT press, 2008.

[13] Romain Barbedienne, Olivia Penas, Jean-Yves Choley, and Laurent Gasser. TheReSE: SysML Extension for Thermal Modeling. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, pages 301–308. IEEE, 2015.

[14] David Basin, Søren Debois, and Thomas Hildebrandt. On purpose and by necessity: compliance under the GDPR. *FC. Springer, Berlin Heidelberg*, 2018.

[15] Daniel Bastos, Fabio Giubilo, Mark Shackleton, and Fadi El-Moussa. GDPR Privacy Implications for the Internet of Things. In *4th Annual IoT Security Foundation Conference*, volume 4, pages 1–8, 2018.

[16] Mark S Beasley, Joseph V Carcello, and Dana R Hermanson. Top 10 audit deficiencies. *Journal of Accountancy*, 19(1):63, 2001.

[17] Kristian Beckers, Isabelle Côté, Ludger Goeke, Selim Güler, and Maritta Heisel. A Structured Method for Security Requirements Elicitation concerning the Cloud Computing Domain. *International Journal of Secure Software Engineering*, 5(2):20–43, 2014.

[18] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[19] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. Iotsense: Behavioral fingerprinting of IoT devices. *arXiv preprint arXiv:1804.03852*, 2018.

[20] Filip Biljecki, Kavisha Kumar, and Claus Nagel. CityGML Application Domain Extension (ADE): Overview of Developments. *Open Geospatial Data, Software and Standards*, 3(1):1–17, 2018.

[21] Kaitlin Boeckl, Michael Fagan, William Fisher, Naomi Lefkovitz, Katerina Megas, Ellen Nadeau, Ben Piccarreta, D Gabel O'Rourke, and Karen Scarfone. Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks. *National Institute of Standards and Technology, NISTIR 8228 (Draft)*, 2018.

[22] Barry W. Boehm. Verifying and Validating Software Requirements and Design Specifications. *IEEE software*, 1(1):75, 1984.

[23] John J. Borking and Ronald Hes. Privacy-Enhancing Technologies: The Path to Anonymity. I(Volume I):1–60, 1995. .

[24] Courtney Bowman, Ari Gesher, John K Grant, Daniel Slate, and Elissa Lerner. *The Architecture of Privacy: On Engineering Technologies that can Deliver Trustworthy Safeguards*. " O'Reilly Media, Inc.", 2015.

[25] Mitchel Broussard. Withings Co-Founder Buys Back Digital Health Company From Nokia, Relaunch Planned This Year, 2018. URL https://www.macrumors.com/2018/05/31/withings-co-founder-nokia/.

[26] Ian Brown. Britain's smart meter programme: A case study in privacy by design. *International Review of Law, Computers & Technology*, 28(2):172–184, 2014.

[27] G. Buttarelli. The Commission Proposal for a Regulation on ePrivacy: Why Do We Need a Regulation Dedicated to ePrivacy in the European Union? *European Data Protection Law Review*, 3(2):155–159, 2017. ISSN 23642831. URL http://edpl.lexxion.eu/article/EDPL/2017/2/5.

[28] Ann Cavoukian. Privacy by design: The 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada*, 5, 2009.

[29] Shi Cho Cha, Tzu Yang Hsu, Yang Xiang, and Kuo Hui Yeh. Privacy Enhancing Technologies in the Internet of Things: Perspectives and Challenges. *IEEE Internet of Things Journal*, 6(2):2159–2187, 2019. ISSN 23274662. .

[30] Alan Chamberlain, Andy Crabtree, Hamed Haddadi, and Richard Mortier. Special theme on privacy and the Internet of things. *Personal and Ubiquitous Computing*, 22(2):289–292, 2018. ISSN 16174909. .

[31] Poornima M Chanal and Mahabaleshwar S Kakkasageri. Security and privacy in IOT: a survey. *Wireless Personal Communications*, 115(2):1667–1693, 2020.

[32] Heting Chu and Qing Ke. Research methods: What's in the name? *Library & Information Science Research*, 39(4):284–294, 2017.

[33] CNIL. Privacy Impact Assessment Application to IoT Devices. Technical report, Commission Nationale de l'Informatique et des Libertés, Paris, 2018.

[34] CNIL. Privacy Impact Assessment Knowledge Bases. Technical report, Commission Nationale de l'Informatique et des Libertés, Paris, 2018. URL https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-3-en-knowledgebases.pdf.

[35] CNIL. Privacy Impact Assessment Methodology. Technical report, Commission Nationale de l'Informatique et des Libertés, Paris, 2018.

[36] CNIL. Privacy Impact Assessment Templates. Technical report, Commission Nationale de l'Informatique et des Libertés, Paris, 2018. URL https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-2-en-templates.pdf.

[37] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016.

[38] Bruno Costa, Paulo F Pires, and Flávia C Delicato. Modeling IoT Applications with SysML4IoT. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 157–164. IEEE, 2016.

[39] Xiaoyi Cui. The internet of things. In *Ethical ripples of creativity and innovation*, pages 61–68. Springer, 2016.

[40] MH Davis, U Lang, and S Shetye. A Cybermodel for Privacy by Design: Building Privacy Protection into Consumer Electronics. *IEEE Consumer Electronics ...*, (December 2014): 41–49, 2015. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6985930.

[41] Lenny Delligatti. *SysML Distiled: A Brief Guide to the Systems Modeling Language*. Pearson Education, Crawfordsville, 2014.

[42] Department of Education. Privacy Impact Assessment For Debt Management and Collection System, 2014. URL https://www2.ed.gov/notices/pia/dmcs.pdf.

[43] Concetta Di lorio and Fabrizio Carinci. The BIRO project: Privacy Impact Assessment, 2009. URL http://www.biro-project.eu/documents/downloads/D14_4_BIRO_Monograph.pdf.

[44] Digital Catapult. Internet of Things Taxonomy. Technical Report December, 2016. URL http://www.idc.com/downloads/iot_taxonomy_map_nov2014.pdf.

[45] DistriNet. LINNDUN framework, 2020. URL https://www.linddun.org/linddun.

[46] DistriNet. LINDDUN GO, 2020. URL https://www.linddun.org/go.

[47] Bruno Dorsemaine, Jean-Philippe Gaulier, Jean-Philippe Wary, Nizar Kheir, and Pascal Urien. Internet of things: a definition & taxonomy. In *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 72–77. IEEE, 2015.

[48] Jean Jacques Du Plessis, Anil Hargovan, and Jason Harris. *Principles of contemporary corporate governance*. Cambridge University Press, 2018.

[49] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[50] M Eigner, T Dickopf, C Huwig, et al. An Interdisciplinary Model-based Design Approach for Developing Cybertronic Systems. In *DS 84: Proceedings of the DESIGN 2016 14th International Design Conference*, pages 1647–1656, 2016.

[51] Ahmed M Elmisery, Seungmin Rho, and Dmitri Botvich. A fog based middleware for automated compliance with OECD privacy principles in internet of healthcare things. *IEEE Access*, 4:8418–8441, 2016.

[52] E Allen Emerson. The Beginning of Model Checking: A Personal Perspective. In *25 Years of Model Checking*, pages 27–45. Springer, 2008.

[53] Epic. EU Privacy and Electronic Communications (e-Privacy Directive), 2017. URL `https://bit.ly/3lT8WaP`.

[54] European Union. Directive 2002/58/EC on privacy and electronic communications, 2002. URL `https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML`.

[55] FDIC External Service. PRIVACY IMPACT ASSESSMENT: Financial Advisory Service for Mortgage Servicing Rights, 2013. URL `https://www.fdic.gov/about/privacy/documents/msr-public-pia.pdf`.

[56] Wen Feng, Edward F. Crawley, Olivier De Weck, Rene Keller, and Bob Robinson. Dependency structure matrix modelling for stakeholder value networks. In *Managing Complexity by Modelling Dependencies - Proceedings of the 12th International DSM Conference*, number July, pages 3–16, 2010. ISBN 9783446424739.

[57] Nicolas Ferry, Phu Nguyen, Hui Song, Pierre-Emmanuel Novac, Stéphane Lavirotte, Jean-Yves Tigli, and Arnor Solberg. Genesis: Continuous Orchestration and Deployment of Smart IoT Systems. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 870–875. IEEE, 2019.

[58] Noria Foukia, David Billard, and Eduardo Solana. Privacy verification chains for IoT. In *International Conference on Network and System Security*, pages 737–752. Springer, 2017.

[59] Jeremiah Fowler. Report: Fitness Tracker Data Breach Exposed 61 Million Records and User Data Online, 2021. URL `https://www.websiteplanet.com/blog/gethealth-leak-report/`.

[60] Robert Edward Freeman. *Strategic Management: A Stakeholder Approach.* Pitman, Boston, 1984.

[61] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A Practical Guide to SysML: the Systems Modeling Language*. Morgan Kaufmann, Waltham, 3rd edition, 2014. ISBN 978-0-12-800202-5.

[62] Unabhängiges Landeszentrum für Datenschutz. The Standard Data Protection Model: A concept for inspection and consultation on the basis of unified protection goals, 2017.

[63] Thomas R Gruber. Toward Principles for the Design of Ontologies used for Knowledge Sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1993.

[64] Seda Gürses, Bettina Berendt, and Thomas Santen. Multilateral Security Requirements Analysis for Preserving Privacy in Ubiquitous Environments. In *Proceedings of the UKDU Workshop*, pages 51–64, 2006.

[65] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering Privacy By Design. In *Computers, Privacy & Data Protection*. Springer, London, aug 2011.

[66] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering Privacy by Design Reloaded. In *Amsterdam Privacy Conference 2015 (APC15)*, number 610613, pages 1–21, 2015. . URL https://software.imdea.org/{~}carmela.troncoso/papers/Gurses-APC15.pdf{%}0Ahttp://carmelatroncoso.com/papers/Gurses-APC15.pdf.

[67] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering privacy by design reloaded. In *Amsterdam Privacy Conference 2015 (APC15)*, pages 1–21, 2015.

[68] Saida Haidrar, Adil Anwar, and Ounsa Roudies. Towards a Generic Framework for Requirements Traceability Management for SysML Language. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 210–215. IEEE, 2016.

[69] Marit Hansen, Meiko Jensen, and Martin Rost. Protection goals for privacy engineering. In *2015 IEEE Security and Privacy Workshops*, pages 159–166. IEEE, 2015.

[70] Stephen Hart, Anna Lisa Ferrara, and Federica Paci. Fuzzy-based approach to assess and prioritize privacy risks. *Soft Computing*, pages 1–11, 2019.

[71] Waël Hassan and Luigi Logrippo. A governance requirements extraction model for legal compliance validation. *2009 2nd International Workshop on Requirements Engineering and Law, RELAW 2009*, pages 7–12, 2009. .

[72] Matthew Hause et al. The SysML Modelling Language. In *Fifteenth European Systems Engineering Conference*, volume 9, pages 1–12, 2006.

[73] Jeffrey Haynes, Maribette Ramirez, Thaier Hayajneh, and Md. Zakirul Alam Bhuiyan. A Framework for Preventing the Exploitation of IoT Smart Toys for Reconnaissance and Exfiltration. In Guojun Wang, Mohammed Atiquzzaman, Zheng Yan, and Kim-Kwang Raymond Choo, editors, *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, pages 581–592, Cham, 2017. Springer International Publishing. ISBN 978-3-319-72395-2.

[74] David Hetherington. *Simple SysML for Beginners: Using Sparx Enterprise Architect*. Asatte Press, Austin, 2019.

[75] Rich Hilliard. Frequently Asked Questions: ISO/IEC/IEEE 42010, 2013. URL http://www.iso-architecture.org/ieee-1471/faq.html#wharch.

[76] Jon Holt. *Systems Engineering Demystified: A practitioner's handbook for developing complex systems using a model-based approach*. Packt Publishing Ltd, 2021.

[77] Peter Hustinx. Privacy by design: delivering the promises. *Identity in the Information Society*, 3(2):253–255, 2010.

[78] Inesc Id, Instítuto Superior Técnico, and Universidade De Lisboa. A Catalogue of Reusable Security Concerns : Focus on Privacy Threats. In *IEEE 20th Conference on Business Informatics (CBI)*, pages 1–10, 2018.

[79] IEEE. IEEE Standards Association (IEEE-SA) Internet of Things (IoT) Ecosystem Study. Technical report, 2015. URL https://manualzz.com/doc/47535908/ieee-standards-association--ieee-sa--internet-of-things--iot.

[80] INCOSE. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons Ltd, Hoboken, 4th edition, 2015.

[81] Information Commissioner's Office. Introduction to anonymisation: Draft anonymisation, pseudonymisation and privacy enhancing technologies guidance. Technical Report May, 2021. URL https://ico.org.uk/media/about-the-ico/consultations/2619862/anonymisation-intro-and-first-chapter.pdf.

[82] Intersoft Consulting. Art. 5 GDPR Principles relating to processing of personal data, 2018. URL https://gdpr-info.eu/art-5-gdpr/.

[83] IEC ISO. Iso/iec 15228: Systems and software engineering — system life cycle processes. *ISO, IEC*, 2015.

[84] Shafagh Jafer, Bharvi Chhaya, and Umut Durak. OWL Ontology to Ecore Metamodel Transformation for Designing a Domain Specific Language to Develop Aviation Scenarios. In *Proceedings of the symposium on model-driven approaches for simulation engineering*, pages 1–11, 2017.

[85] Sofia Johansson, Ann Werner, Patrik Åker, and Greg Goldenzwaig. *Streaming music: Practices, media, cultures*. Routledge, 2017.

[86] Michael N Johnstone. Threat Modelling with STRIDE and UML. 2010.

[87] Kangsoo Jung and Seog Park. Grayscale access control: Applying differential privacy to access control for Internet of Thing environment. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 849–854. IEEE, 2017.

[88] Atreyi Kankanhalli, Yannis Charalabidis, and Sehl Mellouli. IoT and AI for smart government: A research agenda, 2019.

[89] Stephen J Kapurch. NASA Systems Engineering Handbook. *NASA Special Publication*, page 360, 2007. URL http://adsabs.harvard.edu/full/1995NASSP6105.....S.

[90] Mahsa Keshavarz and Mohd Anwar. The Automatic Detection of Sensitive Data in Smart Homes. In *International Conference on Human-Computer Interaction*, pages 404–416. Springer, 2019.

[91] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. STRIDE-based Threat Modeling for Cyber-physical Systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6. IEEE, 2017.

[92] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D Breaux, Annie I Antón, James R Cordy, Luisa Mich, and John Mylopoulos. Extracting rights and obligations from regulations: toward a tool-supported process. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pages 429–432, 2007.

[93] Gennadiy Koltun and Mathis Pundel. Using Two Case Studies to Explore the Applicability of VIATRA for the Model-driven Engineering of Mechatronic Production Systems. *Software and Systems Modeling*, pages 1–22, 2022.

[94] Bert Jaap Koops and Ronald Leenes. Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design' provision in data-protection law. *International Review of Law, Computers and Technology*, 28(2):159–171, 2014. ISSN 13646885. .

[95] Antonio Kung, Frank Kargl, Santiago Suppan, Jorge Cuellar, Henrich C Pöhls, Adam Kapovits, Nicolás Notario Mcdonnell, and Yod Samuel Martin. A Privacy Engineering Framework for the Internet of Things. In *Data Protection and Privacy:(In) visibilities and Infrastructures*, volume 36, pages 163–202. 2017. ISBN 978-3-319-56177-6. . URL http://link.springer.com/10.1007/978-3-319-50796-5.

[96] Antonio Kung, Frank Kargl, Santiago Suppan, Jorge Cuellar, Henrich C Pöhls, Adam Kapovits, Nicolás Notario McDonnell, and Yod Samuel Martin. A privacy engineering framework for the internet of things. In *Data Protection and Privacy:(In) visibilities and Infrastructures*, pages 163–202. Springer, 2017.

[97] In Lee and Kyoochun Lee. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015. ISSN 00076813. . URL http://dx.doi.org/10.1016/j.bushor.2015.03.008.

[98] Franco Loi, Arunan Sivanathan, Hassan Habibi Gharakheili, Adam Radford, and Vijay Sivaraman. Systematically Evaluating Security and Privacy for Consumer IoT Devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pages 1–6, 2017.

[99] Yod-Samuel Martin and Antonio Kung. Methods and tools for gdpr compliance through privacy and data protection engineering. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 108–111. IEEE, 2018.

[100] Ralph Maschotta, Alexander Wichmann, Armin Zimmermann, and Kristina Gruber. Integrated Automotive Requirements Engineering with a SysML-based Domain-specific Language. In *2019 IEEE International Conference on Mechatronics (ICM)*, volume 1, pages 402–409. IEEE, 2019.

[101] Surya Mattu. Your Smart Home Is Spying on You. Here's How to Spy Back., 2018. URL https://bit.ly/3LTzOSN.

[102] Jonathan McGruer. Emerging Privacy Legislation in the International Landscape: Strategy and Analysis for Compliance. *Wash. JL Tech. & Arts*, 15:120, 2019.

[103] Emily Mcreynolds, Sarah Hubbard, Timothy Lau, Aditya Saraf, Maya Cakmak, and Franziska Roesner. Toys that Listen: A Study of Parents, Children, and Internet-Connected Toys. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.*, pages 5197–5207, 2017. ISBN 978-1-4503-4655-9. . URL https://blogs.law.nyu.edu/privacyresearchgroup/wp-content/uploads/ToysThatListen-CHI2017.pdf.

[104] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. IoT sentinel: Automated device-type identification for security enforcement in IoT. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[105] Sophia Moganedi and Jabu Mtsweni. Beyond the convenience of the internet of things: Security and privacy concerns. In *2017 IST-Africa Week Conference (IST-Africa)*, pages 1–10. IEEE, 2017.

[106] Aurelijus Morkevicius. Architectural Frameworks for SysML, 2016. URL https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/CATIA/NoMagic/pdf/system-engineering-architectural-framework.pdf.

[107] Aurelijus Morkevicius, Aiste Aleksandraviciene, Donatas Mazeika, Lina Bisikirskiene, and Zilvinas Strolia. MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems. In *INCOSE International Symposium*, volume 27, pages 136–150. Wiley Online Library, 2017.

[108] Miranda Mourby, Elaine Mackey, Mark Elliot, Heather Gowans, Susan E Wallace, Jessica Bell, Hannah Smith, Stergios Aidinlis, and Jane Kaye. Are 'Pseudonymised'Data Always Personal Data? Implications of the GDPR for Administrative Data Research in the UK. *Computer Law & Security Review*, 34(2):222–233, 2018.

[109] Victor Muntés-Mulero, Jacek Dominiak, Elena González, and David Sanchez-Charles. Model-driven Evidence-based Privacy Risk Control in Trustworthy Smart IoT Systems. 2019.

[110] Blazo Nastov, Vincent Chapurlat, Christophe Dony, and François Pfister. Towards V&V Suitable Domain Specific Modeling Languages for MBSE: A Tooled Approach. In *INCOSE International Symposium*, volume 26, pages 556–570. Wiley Online Library, 2016.

[111] Christian Nigischer, Sébastien Bougain, Rainer Riegler, Heinz Peter Stanek, and Manfred Grafinger. Multi-domain Simulation Utilizing SysML: State of the Art and Future Perspectives. *Procedia CIRP*, 100:319–324, 2021.

[112] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *Ucla L. Rev.*, 57:1701, 2009.

[113] Xabiel García Pañeda, David Melendi, Manuel Vilas, Roberto García, Víctor García, and Isabel Rodríguez. FESORIA: An integrated system for analysis, management and smart presentation of audio/video streaming services. *Multimedia Tools and Applications*, 39 (3):379–412, 2008.

[114] Thomas Pasquier, Jatinder Singh, Julia Powles, David Eyers, Margo Seltzer, and Jean Bacon. Data Provenance to Audit Compliance with Privacy Policy in the Internet of Things. *Personal and Ubiquitous Computing*, 22(2):333–344, 2018. ISSN 16174909. .

[115] Niklas Paul, Welderufael B Tesfay, Dennis-Kenji Kipker, Mattea Stelter, and Sebastian Pape. Assessing Privacy Policies of Internet of Things Services. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 156–169. Springer, 2018.

[116] Siani Pearson. Taking account of privacy when designing cloud computing services. In *Software Engineering Challenges of Cloud Computing, 2009*, pages 44–52, 2009. ISBN 9781424437139. . URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5071532.

[117] Mauricio Peña and Ricardo Valerdi. Characterizing the Impact of Requirements Volatility on Systems Engineering Effort. *Systems Engineering*, 18(1):59–70, 2015. ISSN 15206858. .

[118] Charith Perera, Mahmoud Barhamgi, Arosha K. Bandara, Muhammad Ajmal, Blaine Price, and Bashar Nuseibeh. Designing Privacy-aware Internet of Things Applications. *arXiv preprint arXiv:1703.03892*, pages 1–35, 2017. URL http://arxiv.org/abs/1703.03892.

[119] Luca Piras, Mohammed Ghazi Al-Obeidallah, Andrea Praitano, Aggeliki Tsohou, Haralambos Mouratidis, Beatriz Gallego-Nicasio Crespo, Jean Baptiste Bernard, Marco Fiorani, Emmanouil Magkos, Andres Castillo Sanz, et al. DEFeND Architecture: a Privacy by Design Platform for GDPR Compliance. In *16th International Conference on Trust, Privacy and Security in Digital Business-TrustBus 2019*, 2019.

[120] Klaus Pohl. *Requirements Engineering Fundamentals: a Atudy Guide for the Certified Professional for Requirements Engineering Exam*. Rocky Nook, Inc., 2016.

[121] Mary Popeck and Nataliya Shevchenko. Benefits and Challenges of Model-Based Systems Engineering, 2021. URL https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=736242.

[122] Pawani Porambage, Mika Ylianttila, Corinna Schmitt, Pardeep Kumar, Andrei Gurtov, and Athanasios V Vasilakos. The quest for privacy in the internet of things. *IEEE Cloud Computing*, 3(2):36–45, 2016.

[123] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, and Linlin Chen. De-anonymizing social networks and inferring private attributes using knowledge graphs. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[124] André Queirós, Daniel Faria, and Fernando Almeida. Strengths and limitations of qualitative and quantitative research methods. *European Journal of Education Studies*, 2017.

[125] General Data Protection Regulation. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294, 2016.

[126] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.*, page 14, 2016. ISBN 9781450342698. . URL http://arxiv.org/abs/1507.00255{%}0Ahttp://dx.doi.org/10.1145/2906388.2906392.

[127] Allen F Repko and Rick Szostak. *Interdisciplinary research: Process and theory*. Sage Publications, 2020.

[128] Rick Robinson. Data Privacy vs. Data Protection, 2020. URL https://blog.ipswitch.com/data-privacy-vs-data-protection.

[129] Veronica Root. The compliance process. *Ind. LJ*, 94:203, 2019.

[130] Jessica Ryan, Shahram Sarkani, and Thomas Mazzuchim. Framework for Architecture Trade Study Using MBSE and Performance Simulation. In *Selected Papers Presented at MODSIM World 2011 Conference and Expo*, 2012.

[131] By Mark Saunders and Paul Tosey. The Layers of Research Design. *Rapport: The Magazine for NLP Professionals*, 14(4):58–59, 2012. ISSN 1944-2866. .

[132] Parvaneh Shayegh and Sepideh Ghanavati. Toward an approach to privacy notices in IoT. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 104–110. IEEE, 2017.

[133] Anne Shepherd, Chalermpon Kesa, and James Cooper. Internet of things medical security: Taxonomy and perception. *Issues in Information Systems*, 21(3), 2020.

[134] Nataliya Shevchenko. An Introduction to Model-Based Systems Engineering (MBSE), 2020. URL `https://bit.ly/3M3qtaS`.

[135] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[136] Laurens Sion, Dimitri Van Landuyt, and Wouter Joosen. The Never-Ending Story: On the Need for Continuous Privacy Impact Assessment. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 314–317. IEEE, 2020.

[137] Daniel J. Solove. A Taxonomy of Privacy. *U. Pa. L. Rev.*, 154:477–560, 2006.

[138] Daniel J Solove. Understanding privacy. 2008.

[139] Sara C Spangelo, David Kaslow, Chris Delp, Bjorn Cole, Louise Anderson, Elyse Fosse, Brett Sam Gilbert, Leo Hartman, Theodore Kahn, and James Cutler. Applying Model based Systems Engineering (MBSE) to a Standard CubeSat. In *2012 IEEE aerospace conference*, pages 1–20. IEEE, 2012.

[140] Kahkashan Tabassum, Ahmed Ibrahim, and Sahar A El Rahman. Security Issues and Challenges in IoT. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–5. IEEE, 2019.

[141] Robert Tairas, Marjan Mernik, and Jeff Gray. Using Ontologies in the Domain Analysis of Domain-specific Languages. In *International Conference on Model Driven Engineering Languages and Systems*, pages 332–342. Springer, 2008.

[142] Hun-Tong Tan. Effects of expectations, prior involvement, and review awareness on memory for audit evidence and judgment. *Journal of Accounting Research*, 33(1):113–135, 1995.

[143] Don Tapscott and Alex Tapscott. *Blockchain revolution: how the technology behind bitcoin and other cryptocurrencies is changing the world*. Portfolio, 2018.

[144] Emmeline Taylor and Katina Michael. Smart Toys that are the Stuff of Nightmares [Editorial]. *IEEE Technology and Society Magazine*, 35(1):8–10, 2016. ISSN 02780097.
.

[145] The OMG. *OMG Systems Modeling Language (OMG SysML) v1.0*. The Object Management Group, 2007. URL `http://www.omg.org/spec/SysML/20161101`.

[146] The OMG. *OMG Systems Modeling Language (OMG SysML) v1.6*. The Object Management Group, 2019. URL `http://www.omg.org/spec/SysML/20161101`.

[147] Keerthi Thomas, Arosha K Bandara, Blaine A Price, and Bashar Nuseibeh. Distilling privacy requirements for mobile applications. In *Proceedings of the 36th International Conference on Software Engineering*, pages 871–882, 2014.

[148] Robert Thorburn, Andrea Margheri, and Federica Paci. Towards an Integrated Privacy Protection Framework for IoT: Contextualising Regulatory Requirements with Industry Best Practices. In *IET Conference Proceedings*, pages 1–6. The Institution of Engineering & Technology, 2019.

[149] Robert Thorburn, Federica Paci, Vladimiro Sassone, and Sophie Stalla-Bourdillon. Connecting Regulatory Requirements to Audit Outcomes: A Model-driven Approach to Auditable Compliance. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 641–642. IEEE, 2021.

[150] Unabhängiges Landeszentrum für Datenschutz. The Standard Data Protection Model: A method for Data Protection advising and controlling on the basis of uniform protection goals v2.0b. Technical report, 2020. URL https://www.datenschutzzentrum.de/uploads/sdm/SDM-Methodology_V2.0b.pdf.

[151] European Union. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016. ISSN 1977-0677.

[152] Lachlan Urquhart, Neelima Sailaja, and Derek McAuley. Realising the right to data portability for the domestic internet of things. *Personal and Ubiquitous Computing*, 22(2): 317–332, 2018.

[153] Linda Van den Brink, Jantien Stoter, and Sisi Zlatanova. UML-based Approach to Developing a CityGML Application Domain Extension. *Transactions in GIS*, 17(6):920–942, 2013.

[154] Dimitri Van Landuyt and Wouter Joosen. A Descriptive Study of Assumptions in STRIDE Security Threat Modeling. *Software and Systems Modeling*, pages 1–18, 2021.

[155] Janus Varmarken, Hieu Le, Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. The TV is Smart and Full of Trackers: Measuring Smart TV Advertising and Tracking. *Proceedings on Privacy Enhancing Technologies*, 2020(2), 2020.

[156] W. Gregory Voss. European Union data privacy law reform: General Data Protection Regulation, privacy shield and the right to delisting. *Business Lawyer*, 72:221–234, 2016. ISSN 00076899. URL https://ssrn.com/abstract=2894571.

[157] Qi Wang, Wajih Ul Hassan, Adam Bates, and Carl Gunter. Fear and logging in the internet of things. In *Network and Distributed Systems Symposium*, 2018.

[158] Tim Weilkiens. *SYSMOD-The systems modeling toolbox-pragmatic MBSE with SysML*. Lulu. com, 2016.

[159] Charles Weir, Sammy Migues, Mike Ware, and Laurie Williams. Infiltrating security into development: exploring the world's largest software security stud. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1326–1336, 2021.

[160] Christopher J White and Bryan L Mesmer. Research Needs in Systems Engineering: Report from a University of Alabama in Huntsville Workshop. *Systems Engineering*, 23 (2):154–164, 2020.

[161] Roel J Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[162] Carrie Williams. Research methods. *Journal of Business & Economics Research (JBER)*, 5(3), 2007.

[163] David Wright. The state of the art in privacy impact assessment. *Computer Law and Security Review*, 28(1):54–61, 2012. ISSN 02673649. . URL http://dx.doi.org/10.1016/j.clsr.2011.11.007.

[164] Jacob Wurm, Khoa Hoang, Orlando Arias, Ahmad-Reza Sadeghi, and Yier Jin. Security analysis on consumer and industrial IoT devices. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 519–524, 2016. ISBN 978-1-4673-9569-4. . URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7428064.

[165] Kim Wuyts. LIND(D)UN privacy threat tree catalog. Technical Report September, 2014. URL http://www2.cs.kuleuven.be/publicaties/rapporten/cw/CW675.pdf.

[166] Kim Wuyts and Wouter Joosen. LINDDUN privacy threat modeling: a tutorial. *CW Reports*, 2015.

[167] Kim Wuyts, Dimitri Van Landuyt, Aram Hovsepyan, Wouter Joosen imec DistriNet, and Ku Leuven Belgium. Effective and Efficient Privacy Threat Modeling through Domain Refinements. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing.*, pages 1175–1178, 2018. ISBN 9781450351911. . URL https://doi.org/10.1145/3167132.3167414.

[168] Yan Zexian and Yan Xuhui. A Revolution in the Field of Systems Thinking, a Review of Checkland's System Thinking. *Systems Research and Behavioral Science: The Official Journal of the International Federation for Systems Research*, 27(2):140–155, 2010.

[169] Jan Henrik Ziegeldorf, Oscar Garcia Morchon, and Klaus Wehrle. Privacy in the Internet of Things: threats and challenges. *Security and Communication Networks*, 7(12):2728–2742, 2014.