PARROT: Interactive Privacy-Aware Internet of Things Application Design Tool

NADA ALHIRABI, Cardiff University, UK
STEPHANIE BEAUMONT, My Data Fix Ltd, UK
JOSE TOMAS LLANOS, University College London, UK
DULANI MEEDENIYA, University of Moratuwa, Sri Lanka
OMER RANA, Cardiff University, UK
CHARITH PERERA, Cardiff University, UK

Internet of Things (IoT) applications typically collect and analyse personal data that is categorised as *sensitive* or *special category* of personal data. These data are subject to a higher degree of protection under data privacy laws. Regardless of legal requirements to support privacy practices, such as in Privacy by Design (PbD) schemes, these practices are not yet commonly followed by software developers. The difficulty of developing privacy-preserving applications emphasises the importance of exploring the problems developers face to embed privacy techniques, suggesting the need for a supporting tool. An interactive IoT application design tool – PARROT (PrivAcy by design tool foR inteRnet Of Things) – is presented. This tool helps developers to design privacy-aware IoT applications, taking account of privacy compliance during the design process and providing real-time feedback on potential privacy violations. A user study with 18 developers was conducted, comprising a semi-structured interview and a design exercise to understand how developers typically handle privacy within the design process. Collaboration with a privacy lawyer was used to review designs produced by developers to uncover privacy limitations that could be addressed by developing a software tool. Based on the findings, a proof-of-concept prototype of PARROT was implemented and evaluated in two controlled lab studies. The outcome of the study indicates that IoT applications designed with PARROT addressed privacy concerns better and managed to reduce several of the limitations identified. From a privacy compliance perspective, PARROT helps developers to address compliance requirements throughout the design and testing process. This is achieved by incorporating privacy specific design features into the IoT application from the beginning rather than retrospectively.

CCS Concepts: • **Security and privacy** → *Privacy protections*; *Domain-specific security and privacy architectures*; • **Software and its engineering** → **System modeling languages**; **Visual languages**; **Domain specific languages**; • **Human-centered computing** → Ubiquitous and mobile computing design and evaluation methods; **Visualization toolkits**.

Additional Key Words and Phrases: Internet of Things,Privacy Laws, Software Design, Software Developers, Data Protection

Authors' addresses: Nada Alhirabi, Cardiff University, School of Computer Science and Informatics, Cardiff, CF24 3AA, UK, alhirabin@cardiff.ac.uk; Stephanie Beaumont, My Data Fix Ltd, Wenlock Road, London, N1 7GU, UK, stephaniebeaumont@mydatafix.com; Jose Tomas Llanos, University College London, London, WC1E 6JA, UK, j.llanos@ucl.ac.uk; Dulani Meedeniya, University of Moratuwa, Moratuwa, Sri Lanka, dulanim@cse.mrt.ac.lk; Omer Rana, Cardiff University, School of Computer Science and Informatics, Cardiff, CF24 3AA, UK, ranaof@cardiff.ac.uk; Charith Perera, Cardiff University, School of Computer Science and Informatics, Cardiff, CF24 3AA, UK, pererac@cardiff.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Association for Computing Machinery.
2474-9567/9999/9-ART99 $15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Nada Alhirabi, Stephanie Beaumont, Jose Tomas Llanos, Dulani Meedeniya, Omer Rana, and Charith Perera. 9999. PARROT: Interactive Privacy-Aware Internet of Things Application Design Tool. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 9, Article 99 (September 9999), 38 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The Internet of Things (IoT) is a broad term that includes any device with properties such as connectivity, intelligence, sensing, energy and safety [90]. It is defined by the Internet of Things European Research Cluster (IERC) as "a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual things have identities" [93, 94]. GPS, wearable and monitoring devices can also be part of IoT systems where sensors are integrated with analytic algorithms for the purpose of tracking, analysing and guiding users [63, 72, 90]. IoT applications generate and process large amounts of data which requires an efficient architecture to manage [56] as well as to address any potential privacy and data protection concerns.

Researchers have been advocating techno-regulatory approaches that help to minimise and avoid privacy risks in data processing systems. These approaches are commonly discussed within the context of privacy-by-design (PbD) principles [1], which seek to ensure that privacy-related requirements are accounted for in data processing system design and subsequent development [14]. However, PbD cannot be realised without the active role of software developers because only they can ultimately engineer privacy into their technological designs. In addition, data protection authorities have imposed fines and sanctions at an increasing rate since June 2018 [1], as illustrated by the €225 million fine by the Irish Privacy Commissioner imposed on WhatsApp [32]. Therefore, IoT applications that cannot be deemed privacy-compliant present a compliance risk for data controllers and, by extension, may be less appealing than applications developed with PbD in mind. There is therefore a need for a tool that enhances privacy-awareness and is capable of abridging the operational and implementation gap between software developers and privacy requirements.

Using End-User Development (EUD) techniques makes IoT applications more manageable [60]. EUD seeks to empower end-users to develop and adjust systems at a level of complexity that suits their background and skills [7]. Moreover, supporting interactivity can add more value to the tool by ensuring that mechanisms to address privacy and data protection issues do not have to be retrospectively incorporated into the design. Interactivity may also make tools more intuitive for target users compared to static ones, because EUD imitates real-time collaboration instead of a more stagnated user experience [30]. Interaction could be supported in different ways such as by using alerts, notifications, real-time feedback, or highlights [30, 49, 62].

We propose a tool that is intended for the conceptual design phase of the software development life cycle (SDLC). It is intended to support dialogue between lawyers and developers in the context of IoT app design. The key value is following a user(developer)-centric approach to designing user solutions. Engaging developers in software development helps to achieve their needs in addition to fulfilling security and privacy requirements. The current study makes the following research contributions:

- It systematically examines a developer's understanding of privacy through a series of semi-structured interviews. It also conducts an IoT application design exercise to understand a developer's approach to integrating privacy within the software design process. It supports collaboration between developers and privacy lawyers to reduce privacy breakdowns, such as interpreting legal terms or using third party software libraries without knowing their implications on data privacy.
- It presents the design and implementation of PARROT, an interactive IoT application design tool that helps developers design privacy-aware IoT applications, supporting them to consider privacy during the design

[1] PbD principles are captured in seven foundation principles by Anne Cavoukian [15].

process and providing real-time feedback about potential privacy issues. PARROT interactively *nudges* developers into incorporating privacy measures into the app design.
- It presents the results of studies carried out with PARROT, which show that developers can better understand how their IoT applications handle personal data..

The target users of PARROT are software developers, privacy lawyers and those who communicate with privacy lawyers in the application design process. We build this tool to enable software developers make privacy choices, thereby reducing the assessment load on privacy professionals at a subsequent stage. The paper is structured as follows: in Section 2 we describe related work about privacy, privacy engineering methodologies and provide an overview of interactive tools. Section 3 includes a description of the proposed privacy-aware interaction methodology. Sections 4, 5 and 6 present different phases of the methodology which consist of 3 case studies to implement the PARROT prototype tool. Section 7 presents the research findings and Section 8 discusses the research challenges and future opportunities, with concluding comments in Section 9.

2 BACKGROUND AND MOTIVATION

2.1 Software Design Tools

In a software system, a conceptual model is defined as "a high-level description of how a system is organised and operates" [50]. Parush [68] adds extra details and divides the conceptual model for interactive systems into a five-layer framework which is, from the bottom up: function, configuration, navigation and policy, form and detail. There are many different conceptual models but the one that is most relevant in design is the mental model [51] which shows how people understand how things work. A good conceptual model should be clear and understandable with the support of affordances, signifiers and constraints. Adopting a good conceptual model in the IoT application design tool could make it easier for the developer to efficiently predict any likely privacy threats, thereby increasing the tool's useability.

Some surveys [e.g., [4]] have reviewed several design notations and languages commonly used to develop applications. The next step is to make these notations available for software engineers to operate within SDLC. Tools such as Visual Paradigm (visual-paradigm.com) have been developed for software engineers to make design notations and languages (e.g., UML 2, SysML and Business Process Modelling) available. These tools are intended to enhance the software design capabilities of software engineers. They also aim to make the software development process efficient and effective by reducing human errors/mistakes and the design time.

Security and privacy visualisation are essential in IoT apps, especially large-scale systems like smart cities where poor data visualisation and analysis can produce invaluable insight into the system [34, 38, 80, 92]. Having interactivity can make the tool more understandable and useable for the target users. For example, Coconut [58] and FixDroid [66] are IDE plugins that target developers handling security/privacy. While Coconut presents real-time feedback on possible privacy issues, FixDroid highlights code security and privacy issues. On the other hand, My IoT Puzzle [24] and ViSiT [3] both follow the jigsaw puzzle metaphor as a technique to simplify IoT transformation and attract end-users. Describing transformations in this way can resemble mathematical equations, which proves that complicated concepts (e.g., privacy) can be presented in a more friendly way.

2.2 Threats and Privacy

In any system, assets have value and must be protected from threats. According to Geer [36], the primary cause of software vulnerabilities can be identified and eliminated early in the SDLC. Threat modelling methodologies such as STRIDE and LINDDUN are proposed to reveal system attacks and reduce the number of risks. Since Cavoukian [15] introduced the concept of PbD, into the design of information technologies and systems, a significant literature has emerged in this area. For example, Perera et al. [70] proposed a five-phase PbD data life cycle method for IoT systems. The framework was created for engineers to enhance their designs by increasing

privacy awareness. Chaudhuri et al. [17] also presented PbD principles intended to solve privacy problems in IoT devices and smart services. These principles are based on stakeholder demands regarding IoT services.

PbD is underpinned by the knowledge that embedding privacy features from the outset of the design process is preferable to attempts to adapt a product or service at a later stage [25, 37] which would likely increase costs and extend the time to launch when it is generally too late and expensive. The perceptions and understanding of privacy and data protection requirements of software developers play a pivotal role in efforts to devise and implement privacy-compliant systems such as IoT devices [6, 40]. However, for lawyers, PbD may be a coherent and intuitive policy tool. As an ENISA report observes, '[m]any system developers are not familiar with privacy principles or technologies that implement them [37]'. This unfamiliarity is problematic. The GDPR contemplates fines of up to 2% of the infringer's total annual turnover[2] [35].

Despite the importance of integrating PbD into SDLC, developers face numerous complications when handling privacy requirements. Recent research studies have investigated how software developers handle privacy practices and the possible privacy challenges. Awanthika and Nalin [81] stated that developers' experiences and personal opinions are common issues when applying privacy. Negative privacy culture is another obstacle to integrating privacy into software design in software teams [86]. Tianshi et al. [59] stated that Android developers rarely mention privacy when discussing app design or implementation challenges. However, they do so if there are new privacy restrictions from Android OS, app store policies, or privacy laws. Moreover, many developers treat privacy as data security, restricting their understanding of particular privacy threats. Hadar et al. [40] focused on software architects who make high-level design decisions from different domains and reported that developers use data security to approach privacy challenges which limit their perception of privacy. At any rate, IoT applications require an advanced software development life cycle to adapt to their requirements and to help developers integrate privacy engineering into system development [4].

2.3 Privacy Engineering Methodologies:

With rising concerns for software system privacy, Privacy Engineering Methodologies (PEMs) have been developed as a form of privacy engineering to guide software developers in incorporating privacy into their systems design [61, 82]. Several standards and methodologies are available to facilitate privacy. For example, Privacy Impact Assessment (PIA) and Data Protection Impact Assessment (DPIA) are processes that systematically examine an organization's activities to identify and mitigate privacy risks at an early stage of the project. Both are recommended as key tools for PbD and privacy by default, where DPIA is commonly associated with the GDPR and personal data processing [21, 48, 67]. The assessment is typically developed manually with multiple steps. Developers perceive performing a PIA as a complicated task, mainly due to the lack of practical guidance on how to carry out such an assessment, especially with technologies such as IoT that process large-scale data [67, 83]. There are efforts to incorporate PIAs into the IoT architecture, such as [73]. However, they have not been adopted extensively into the IoT context [97]. Tahari [86] explained that despite having standards on how privacy engineering techniques should operate in critical environments, such as NIST [45], privacy advocates affirmed that privacy is difficult to measure, which could lead developers to be less motivated to address privacy in their designs.

Our proposed method differs in multiple way. First, existing methods are proposed at a legal level and do not target software developers of technologies such as IoT [97]. Consequently, deploying privacy features into IoT designs is technically complex where the suggested guidelines are difficult to translate [86]. IoT developers need simple guidelines on how to embed privacy into their applications. Second, developers tend to employ less complex (easy-to-follow) techniques [82] while these methods typically involve multiple and time consuming steps, which limited identification of immediate benefit. If developers are not able to see the result of their actions

[2]GDPR, Article 83(4).

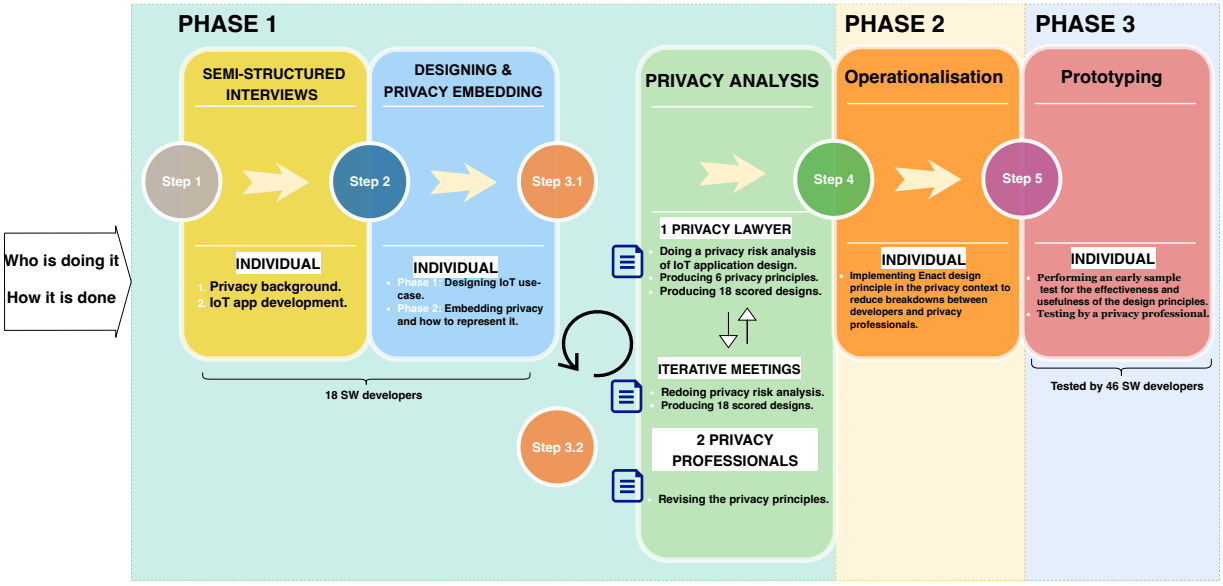

Fig. 1. The proposed methodology to build the PARROT prototype tool. The methodology consists of three phases discussed in Sections 4, 5, and 6.

while using a method, they may be reluctant to adopt it [82]. Third, developers struggle when executing soft decisions, which suggests that privacy guidelines should be presented in a simple, explicit and straightforward way [81], which PARROT proposes. Poor tool support, unclear evaluation criteria and complexity impede software privacy. Moreover, having an automated tool for data flow modelling is proven to be useful in security, yet it is rarely supported [86]. PARROT allows interactive privacy configuration where the developers can see the result of their privacy decisions immediately in a simplified way.

2.4 Motivation

Applying privacy law (such as GDPR) in an IoT application context can be challenging [44, 95] because legal rules tend to be open-ended, unlike programming rules. The terminology that privacy lawyers use and understand is very different to the language developers use. Programming requirements are about meeting stakeholders' goals and needs. Laws are about rights, privileges and obligations. Moreover, legal language can have multiple levels of interpretation and translating these terms and criteria directly into technical terms for developers is challenging. We believe that the tools used to develop privacy-aware IoT application design, such as visual and functional prototypes, are not yet able to support the interactive practices followed by developers.

3 METHODOLOGY

Our methodology is based on an in-depth empirical study comprising a series of phases with different participants with different level of expertise. We define privacy-aware interaction; i.e., when, where and how the system reacts to particular design inputs to produce privacy-aware IoT applications. The work is conducted in three phases (see Sections 4, 5, and 6), as seen in Figure 1.

In the first phase, Section 4, multiple semi-structured qualitative interviews and discussions were conducted with developers. We conducted a design exercise task for an IoT health use case which a privacy lawyer scored.

The lawyer has an in-depth understanding of the PbD principle and is thus well-equipped to perform privacy risk analysis. However, such an assessment could be subjective because the scores are bound to be dependent on the lawyer's perspective. Therefore, a questionnaire-based discussion was held with two privacy professionals to corroborate the soundness of the abovementioned privacy-related criteria. In this way, we sought to mitigate any bias that might have affected the first privacy lawyer's analysis and scoring. The discussion sought to determine whether these criteria are valid and of general applicability in the field. Based on this second opinion, we were able to mitigate any concerns about how objective and sound the first assessment was. In the second phase, (Section 5), we discussed and observed how to implement Enact's [57] four design principles in the PbD context. We intended to implement operationalisation and interaction techniques for the app designs that were produced in Section 4. We worked iteratively with the lawyer, privacy professionals and software developers to verify that we were correctly translating the information from both sides. Thereafter, based on our previous findings, we produced an early proof-of-concept prototype of PARROT, validated by privacy professionals in the third phase. In addition, we evaluated this prototype in controlled lab studies with multiple developers (Section 6).

4 PHASE ONE: UNDERSTANDING PRIVACY BREAKDOWNS

The goal of this study is to understand developers' privacy breakdowns that are not fully addressed when designing IoT applications, and how these breakdowns can be overcome. We thus focused on understanding any interpretation incongruencies/gaps in order to develop a clear and understandable notation design to visualise privacy-aware IoT applications in the subsequent phases. In particular, we were interested in:

- How software (SW) developers manage privacy in practice when developing applications;
- Which apps the SW developers had recently developed and how they treated personal data in these apps;
- How and what the SW developers could add to the design to embed privacy principles using notations or symbols to visualise IoT applications;
- How a privacy professional performs privacy risk analysis and interprets the IoT application design illustrated by others.

4.1 Method:

4.1.1 Participants. Self-nomination [41, 86] and peer nomination [43, 86] are commonly used in prior research methods to identify targeted participants. Our targeted participants were IoT app developers or mobile app developers if they used sensors on their mobile applications (e.g., location, temperature). Restricting the participation of IoT app developers would limit the research due to the difficulty of recruiting IoT specialists [12, 52, 85]. Initially, we contacted those who self-identified as software developers from researcher contacts, Twitter and LinkedIn. We also posted the recruitment invitations on mailing lists. Privacy was explicitly stated as a design goal in recruiting material because we were concentrating on developers' conscious behaviour towards privacy. Participants were also explicitly requested to incorporate privacy during the design task [58, 81].

Throughout the data collection process, we used theoretical sampling to perform our recruitment [40, 41]. We recruited three to four software developers at a time. Then we used snowball and peers' recommendations to contact more developers [41, 86]. We asked participants to nominate someone who is: (1) defined as a software developer or full-stack developer; (2) familiar with developing IoT or mobile applications. Then we selected the following potential participants based on who would be able to contribute additional perspectives on the areas of interest. We recruited 18 developers which is within the recommended range required to find themes in qualitative interviews [39] and close to other related tools listed in the survey [4]. Because we were interested in how non-privacy experts might approach incorporating privacy into their tasks, we did not limit participation to individuals with privacy experience.

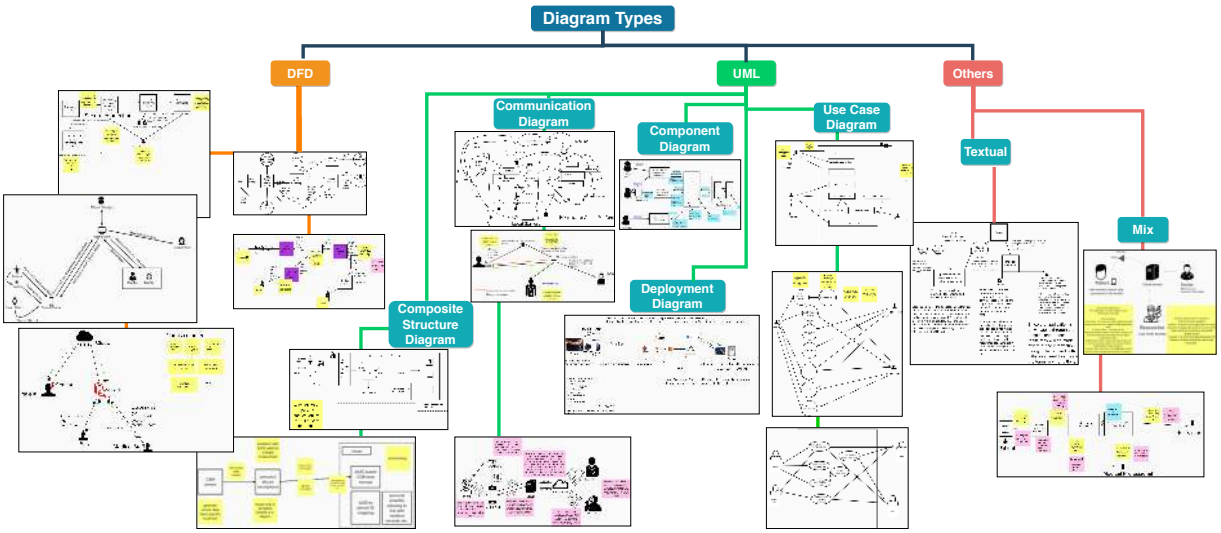

Fig. 2. Eighteen IoT application designs snapshot showing the different approaches the participants used to express their high-level conceptual designs for the diabetes treatment and monitoring use case. Participants used mixed diagrams such as DFD, UML (e.g., use case diagram) and combinations of diagram types.

Participants P1–18 are working in multiple domains such as cyber attack vectors, integration, data ingestion, data streaming, IoT, web and mobile app development, full-stack development. Four of them are full-stack developers with an average of three years of experience. Ten developers stated they had not developed a single IoT app because they are mobile developers who have used multiple sensors in their mobile applications. Fifteen developers actively work on software development projects as software developers, and five of them started to work on SW/IoT development at an undergraduate level. Among the participants, two have more than 20 years of SW development experience, six have 2-10 years in IoT development, and two have 4-10 years' experience of making privacy policies. Furthermore, five participants have received short training about security and privacy as a job requirement. The demographic information of the participants is listed in Appendix B, Table A1.

4.1.2 Procedure. We conducted a systematic study to examine developers' understanding of privacy. All of the study materials and questions are listed in Appendices (A, B, and C). This study was conducted in three steps:

- Firstly, we had general questionnaires about the recent applications they had developed (up to 3 applications).
- Secondly, each software developer was asked to read and complete the design task for the "diabetes treatment and monitoring" use case (A) using Mural. Once they had finished, we briefly introduced GDPR privacy principles and discussed with them how they typically capture privacy requirements. Based on our discussion, the developers were asked to revise their designs to address privacy concerns.
- Thirdly, we had iterative meetings with a privacy lawyer to conduct a privacy risk analysis of developers' IoT application designs. In particular, the privacy lawyer assessed the use case and the output IoT application designs of each participant. To confirm the soundness and general applicability of the lawyer's assessment and privacy-related criteria, we had a questionnaire-based discussion with two privacy professionals who were impartial towards the study.

4.1.3 Data collection. We ended up with 18 different IoT application designs for the proposed use case, as seen in Figure 2. Also, we had 18 scored/interpretations of these designs. During the interviews, we video and audio

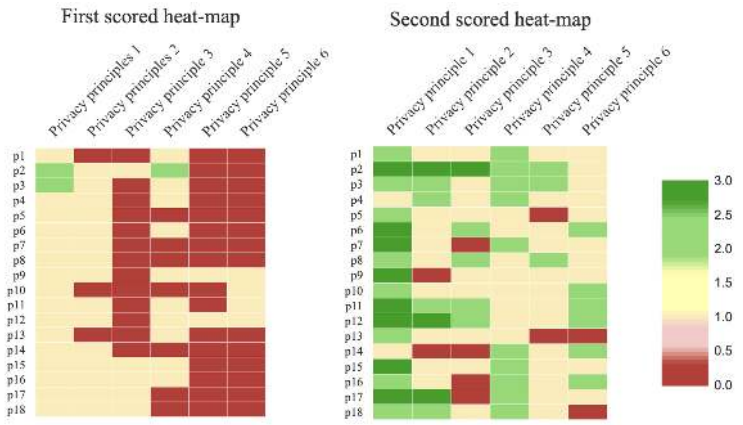

Fig. 3. Privacy assessment heat-map. The x-axis denotes the privacy principle score identified by the participants (red represents no identified principle; pink, yellow and light green represent partially identified principle; dark green represents fully identified principle). The y-axis denotes the participants' ID. Note: the score for each principle is: 0 – privacy is not considered; 1 – privacy is considered; 2 – Privacy is considered, and the issue is identified; 3 – privacy is considered, the issue is identified and the solution is correct.

recorded the participants' designs and discussions for qualitative analysis purposes. We produced a list of the privacy principles that can be applied to the use case in the IoT context (listed and explained in Appendix F).

4.2 Results and Discussion

We assessed the app designs based on six principles inspired by Cavoukian PbD principles [15], which are: *(1) Privacy requirements intrinsic in design and analysis, (2) Privacy embedded in the design, (3) Full functionality, (4) End-to-end security, (5) Visibility and transparency,* and *(6) Respect for user privacy.* We expected each participant to recognise six privacy principles (each scored with a maximum of 3 points, the total score is 18), as shown in Appendix F (Table A2). We assigned a score for each principle as 0: if no privacy requirement is considered; 1: if privacy is considered; 2: if privacy is considered, and the issue is identified; and 3: if privacy is considered, the issue is identified and the solution is correct. The participants were able to demonstrate knowledge of these principles by either using their development experience, privacy training background, searching the web or using the brief privacy discussion during the interview. We present an overview of the scores using two heat-maps in Figure 3 where the results of the first and second scoring rounds are presented separately. The lawyer produced the first-round scoring solely based on an assessment of the conceptual designs of each individual. The second-round scoring includes the interview transcript for each participant in addition to the design. If privacy principles are fully recognised, the scorecard should collect 324 points in total (18 scores × 18 participants). The first and second heat-maps clearly show that software developers were able to identify a higher number of privacy principles after the lawyer interpreted the interview transcripts. In fact, the total score increased significantly (from 58 total points in the first-round to 157 points in the second-round). The participants noted that applying concepts such as PbD is challenging due to the difficulties of relating privacy rules to their design and into techniques they can implement, which was also noted in several research studies [40, 59, 81]. This reflects that most of the developers were able to explain privacy verbally and textually but were unable to clearly illustrate them in their design. Our findings provide insight into how to design prototype tools that encourage thinking about privacy practices and also lay the groundwork for the design of PARROT in Sections 5 and 6.

4.3 Interview Results and Effects on the Design of PbD Prototype Tool for IoT Applications

To conduct the qualitative analysis, we used Miles' methods [64]. Furthermore, we used Richards' coding techniques: descriptive coding, topic coding and analytic coding [76]. We applied descriptive observation, descriptive coding and topic coding to acquire as much information as possible and to create an initial interpretation of the cases, before assigning them to topics [100]. Then we use analytical coding to gain a better understanding. To analyse the 18 app designs, we applied thematic analysis [11, 74]. First, we transcribed the interview and organised the data. Then we assigned codes to the notes drawn from the interviews and the designs' observations. We filtered the materials to identify themes, relationships between variables, similar phrases, patterns and variances.

4.3.1 Developers from large vs. small companies. Is privacy self-learned or guided by others? In general, large companies tend to give general privacy guidance. Five participants said they were given general privacy guidance as to their responsibilities to make everything secure. P7, who was working for a large company and is now working for a start-up said "we certainly had lawyers". However, the other developers had to learn and apply privacy by themselves. P16 initially worked at a small company as a full stack developer. He said: "when the website was hacked by hackers, at that time I personally experienced privacy issues. . . and I needed to take some steps back before re-joining the industry." Later, once he had joined a medium-sized company, he was introduced to security and privacy by his manager. Other developers deal with third parties to help them ensure privacy standards are met. P18 said: "we got help from third parties to figure out the problems and they showed us the way to correct and fix them... they have a privacy-related lawyer." In conclusion, from our limited participant pool, it is clear that there is no single, unified approach to introducing privacy into software development.

4.3.2 Embedding privacy requirements during SW development. Is it a straightforward or iterative process? Typically, ensuring privacy involves several stakeholders with conflicting interests, such as developers, managers and legal stakeholders which makes it a complex and not a straightforward process. Tahaei [86] noted that communication complexity is due to differences in conceptualisations, vocabulary and distinct backgrounds. From multiple interviews we found that there are multiple ways to embed privacy into software. The majority of the developers said that there is a set of privacy standards that need to be met and included before submitting any applications to clients, such as applying GDPR rules. "GDPR is the standard process and known by everyone on the team" P16 stated. Additional privacy and security standards can be incorporated based on client requests at any time."Based on the client requests; the price can be increased," P16 said.

In small-to-medium sized companies, developers can have direct contact with the client to appreciate the level of privacy they require from the onset. Conversely, in larger companies, it is usually the case that a separate team is tasked with communicating with the client and the legal team iteratively. "It is a kind of management team task…" P16 said. Additionally, P18 said that adding privacy is not a single process when dealing with a legal third-party company. "We have a couple of meetings back and forth… we have to redesign most of our systems… until security and privacy are confirmed and there is no data leaking or other vulnerabilities," P18 added.

4.3.3 Privacy attitudes: Privacy does get much attention from SW developers. It became mandatory to apply security and privacy on most applications as a result of client requirements predetermined by mandatory privacy law. Five developers stated that they have to apply security measures, otherwise companies such as VISA and Mastercard will stop dealing with them. P18, who works on payment gateway applications, stated: "security is forced on us." The majority of developers that we interviewed agreed that it is important to know and apply privacy policies either by using a checklist or as a group discussion. P7 explained that privacy had not been considered as important as security in the past but "with GDPR we have to be aware of what is legal to store and we cannot store anything not necessary, and if someone wants to have the data removed, we have to make that accessible to them easily." Moreover, most of the developers have a partial understanding of privacy as a result of self-training or based on guidance provided in their workplace. "The employees talk about GDPR and privacy

issues day-by-day," P16 said. "In terms of training, [those] who work in the organisation know privacy is an important issue, especially [in] the last couple of years," P7 noted. Despite the developers' general understanding of privacy and its importance, they lack an appreciation of how to apply it. Tahaei [87] has reported that when analysing stack overflow privacy-related answers, respondents generally stated why privacy technology is needed or why a behaviour occurs but not how to accomplish a task or an activity.

4.3.4 The need for a supportive tool. Lack of knowledge of viable alternatives. Privacy policies and practices are an overwhelming topic for SW developers. When we asked them to incorporate privacy into their designs, they said it was challenging because they usually talk about privacy. Awanthika and Nalin [81] reported that developers have difficulty embedding privacy and validating their work and their perspectives affect how they incorporate privacy into the design. Moreover, converting privacy goals into technical needs is complex. The technical complexity could derive from a lack of knowledge about privacy-preserving techniques to mitigate the privacy risk, as Tahaei [86] reported. P7 said, "No one [is] telling us what to do at the moment and what [are] the best practices on privacy." On the other hand, privacy professionals may struggle to understand what software developers are doing in terms of privacy if there is no such explanation.

When we asked developers to write a privacy notice for the app user to explore how extensive their privacy policies knowledge is, P6 wrote ". . . personal data is not accessible to researchers. You have full control of your data; you can delete it at any time." In turn, P2 wrote: ". . . Only authorised users will be able to access your data for the purposes outlined in this policy." P2, P3, P7 and P17 stated that all personal data would be anonymised when used within the application. All participants except P3, P4, P9, P10 and P14 stated that personal data would be deleted from the app upon request by the user. Almost two thirds of them said that, even though the app handled personal data, this data included medical data as well, so the hospital is the entity that decides when to delete it and not the user. P11 talked about the data storage limitations with security measures: "Your data is only kept as long as is necessary. It is stored in accordance with certain standards for data encryption." In turn, P7, P11, P12, P14, P15 and P17 expressed to the user that they would not share any personal data with third parties.

Developers generally could not demonstrate their knowledge of privacy properly on their designs. Certainly, words such as anonymisation, pseudonymisation, encryption, authorisation, consent and GDPR compliance were repeated by most of the developers. The majority of them also represented privacy properties as sticky notes or texts alongside some of the nodes or within the design space. P1 used key and lock icons to represent the encryption and decryption process. P7 said that as there was no tool to help him in terms of privacy; he had to rely on his experience. When asked about who reviews his designs from a privacy perspective, P7 said "we have to be aware of some of the issues, especially with developing software for healthcare." Therefore, we conclude that having a tool that assists developers in the application of privacy measures is highly desirable, not least given that most of them stated that they needed such a tool.

4.4 Why Is PARROT Useful?

Based on the previous discussion and findings, we find that having a tool that applies the PbD principle visually will be beneficial to developers working in companies of different sizes. Developers, especially in small-sized companies, struggle to apply privacy and may not afford the cost of iterative communication with a legal firm. The availability of this tool would also be beneficial for privacy professionals when reviewing developers' designs, since privacy requirements would be supported within the design.

With PARROT, there is no need to use code or technical visual notations, e.g. UML, to embed privacy principles into IoT application design. PARROT allows developers to visually integrate and define privacy components and properties by using simple visual notations while keeping complexity hidden. Sometimes, complicated visual tools can impede the learning ability of developers with basic or no privacy law knowledge. Providing developers

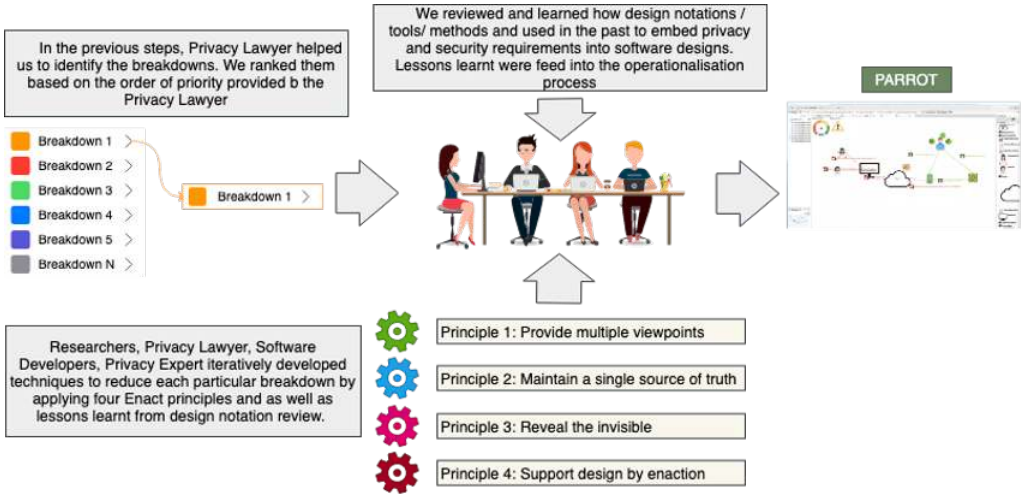

Fig. 4. Operationalisation process (phase 2 of overall methodology illustrated in Figure 1 Section 3) building on the findings from the survey in [4], iterative meeting with privacy lawyer, and Enact's principles.

with a simple tool-supported approach for embedding privacy could empower them to adequately implement privacy requirements during IoT app design. We believe this tool enables early communication between privacy professionals and software developers, ultimately resulting in privacy-aware designs.

5 PHASE TWO: OPERATIONALIZATION

In this phase, as seen in Figure 4, we aim to apply the notation that was employed in phase one using Enact's four design principles: provide multiple viewpoints, maintain a single source of truth, reveal the invisible, and support design by enaction. Because the Enact [57] principles are expected to reduce gaps between designer-developers, we wanted to test whether the same principles could help to reduce gaps between developers and privacy professionals. We thus focused on understanding the design principles produced by Enact and how to make them fit in the privacy context. Then we came up with a visual notation that helps to build a basic structure for an interactive tool in the later phase. In particular, this study was concerned with:

- How to implement Enact's four design principles for PbD purposes.
- How developers can correct the mistakes/gaps that were identified by the privacy lawyers in phase one.
- How to assist developers in the implementation of privacy during the design process by providing real-time feedback about potential privacy issues.

5.1 Method:

5.1.1 Participants. For the operationalisation element, we collaborated with two SW engineers, a privacy lawyer and two privacy professionals who have an in-depth understanding of the PbD principle. Collaboration with the privacy professionals took place iteratively in multiple meetings before and after prototyping the tool to ensure the prototype satisfied the majority of the privacy lawyers' requirements for the case study.

5.1.2 Procedure and discussion. At the beginning of the meetings, we sought to map common terms between a software developer and privacy lawyer in the context of a PbD framework from Perera's framework [71]. We based the mapping on 30 privacy guidelines by Perera [69, 71] which work as a framework to help software

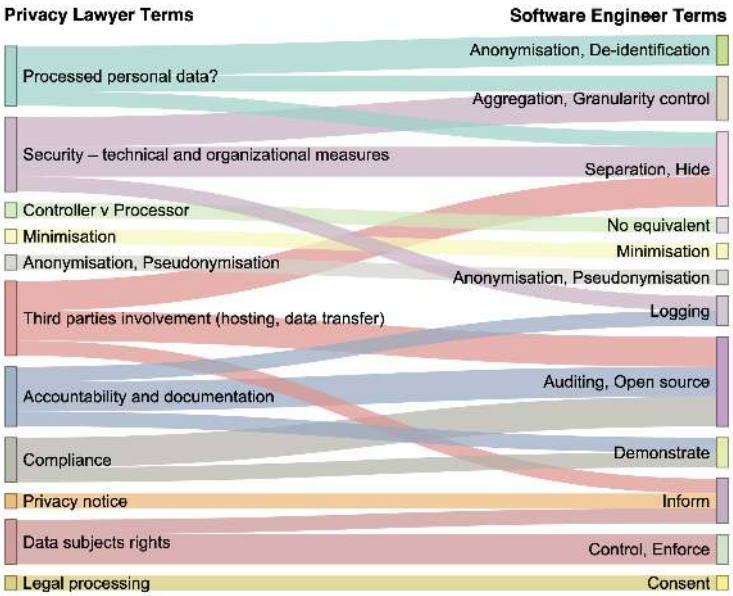

Fig. 5. Translation of common terms used in PbD between the engineering and legal fields.

engineers with the IoT application design process. These guidelines are categorised into eight categories: *Minimise, Hide, Separate, Aggregate, Inform, Control, Enforce,* and *Demonstrate*. First, we mapped each of the eight guidelines to privacy lawyers' terms. For example, the lawyer interprets the Minimise guideline as a concept of minimisation applied across the entire data life cycle: collection (type), source, frequency, replacement, accuracy, storage and use (retention). Inform was interpreted into three parts: pre-inform (privacy notice) aspects, post-inform (data subject rights), data sharing and transfer. Then we did the reverse process from the privacy lawyer's perspective and mapped those requirements to the developer's guidelines. For example, data subject right is related to Inform and Control guidelines. Having finished the mapping processing, we consulted two experienced software engineers (with 12 and 10 years of experience in SW engineering development) to verify the mapping results and if something was missing. They confirmed the mapping and suggested including encryption and multi-layering terms for security purposes which are already covered under the *Separate* and *Hide* guidelines. The result of this process is illustrated in Figure 5.

After that, we focused on the other breakdowns that developers tend to overlook. At the beginning, we thought that the data minimisation[3] principle was the first requirement we needed to implement. In our case study, most of the designs collected geolocation data with different frequencies (i.e., this data is not necessary). Data minimisation applies to: (i) the type of data; (ii) the frequency with which it is collected; and (iii) whether the new set of data replaces the previous set if possible. However, "launching straight into this concept means we have already bypassed some points such as the decision to process personal data and the need for a privacy notice to be displayed," the lawyer said. Software engineers must factor these issues into their design, irrespective of whether or not they are data controllers or processors under the GDPR. If the application owner is a processor, the controller will still need to display a privacy notice, so this functionality will always be needed.

[3]Note, data minimisation is listed third in Art 5(1)(c) of the GDPR with lawful, fair and transparent processing listed first (Art. 5(1)(a)).

In addition, the type of personal data that is being processed is important for developers to know. The UK GDPR defines any information relating to an identified or identifiable data subject as *personal data*, (Art. 4). On the other hand, it defines *the special category of personal data* as any data revealing "racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation"(Article 9) [91]. From our previous findings, the majority of SW engineers do not understand the difference between personal data and special or sensitive personal data which can usually only be processed with consent. This means that a functionality to obtain consent needs to be part of the design.

Moreover, understanding the use of personal data by third parties, whether this is a cloud hosting provider or a process support provider (e.g., Kafka; kafka.apache.org) is essential. When personal data is transferred to non-EU countries which have no adequacy decision from the European Commission (such as the US after the Schrems II landmark ruling of the European Court of Justice[4][88]), the GDPR provisions on international transfers must be taken into account [89]. This is a delicate issue because many IoT applications involve data passing through tools provided by third parties [19] which may result in the application owner being classified as a processor and subject to privacy law compliance. Indeed, most of the designs produced in phase one used third-party servers for cloud processing without thinking about the server location. Thus, if the above mentioned considerations are not duly accounted for, we may end up with good SW decisions about what is collected, containerisation/separation of data, frequency of collection and accuracy, yet the privacy lawyer is nevertheless left to address such considerations at a later stage.

Even though security is not the primary focus of the study, it is one of the aspects that was considered in the discussion and prototyping. "Security is one of the most important aspect of data collection and use. What security do you have around your entire system from start to finish?" the lawyer said. GDPR compliance is irrelevant if there is no personally identifiable information (PII). However, with IoT apps that hold and transfer PII, leakage could happen. Consequently, the security of any data is important. For example, containerisation is a key tool for preventing leakage, as are pseudonymisation, anonymisation and deletion. In addition to containerisation, penetration testing, vulnerability scanning, patching and logging are all key to the security of data, which we tried to cover on the tool.

As noted above, the data minimisation principle is essential in privacy-aware designs. Whilst Hoepman considered data minimisation to be the most important strategy out of the eight privacy design strategies proposed to support the software development lifecycle in [46], data minimisation was not a top priority of the privacy lawyer to avoid missing fundamental considerations, such as asking for consent. As a result, we produced a high-level flowchart diagram (see Figure 6) that can serve as the basis for devising the prototype in phase three, as well as for applying Enact's principles to the identified breakdowns. According to the lawyer, the diagram aims to apply the privacy techniques to the proposed use-case and to help "replicate the thought process of a collector and user of personal data to trigger different compliance points at different stages." In general terms, the method is only based on GDPR requirements, which are not prescriptive. The method permits the user to adopt a risk-based approach to the GDPR compliance requirements, enabling the developer to rely on the tool's output towards achieving better GDPR compliance. To sum up, we need the developer to think about who is "looking after" the data (e.g., third-party hosting provider). What if an attacker gathers PII from sensors and tracks the holder of the sensor? What access controls are required to be designed (both the granting and removal of such access)? Also, who has access to the data, why and for how long?

[4]C-311/18 Data Protection Commissioner v Facebook Ireland and Maximillian Schrems [88].

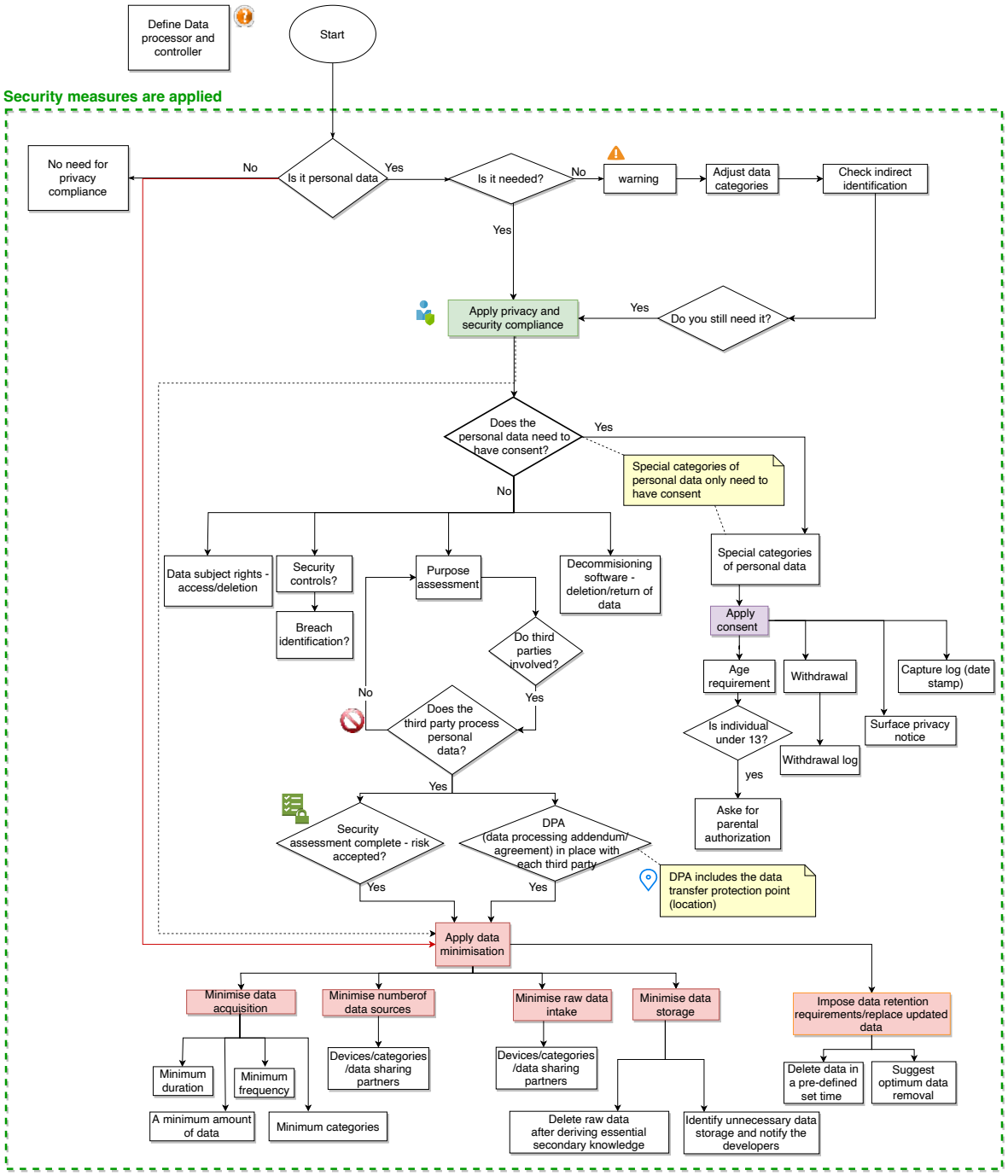

Fig. 6. A high-level flowchart aiming to apply privacy techniques to the proposed use-case. The diagram is part of the operationalisation process (phase 2 of the overall methodology illustrated in Section 3). It also serves as a basis for devising the prototype in a later process (phase 3). Note: we assumed the developers defined the data processor and controller.

6 PHASE THREE: PROTOTYPING AND EVALUATION

The goal of this study is to implement a tool that supports interactive techniques to an extent that it complements the designs produced in phase one. We focused on acting as a privacy assistant, based on the fundamental aspects derived from the previous phases. Thus, we engaged two privacy professionals to discover the prototype tool, to see if it overcomes the first phase's breakdowns and if the privacy representation needs improvement. The prototype was tested in two controlled lab studies with 46 developers by a privacy lawyer. The results will be used to comprehend whether or not these designs developed using the prototype tool are more privacy-aware than the previously produced designs. In particular, this study is concerned with:

- Creating a privacy-aware interactive prototype tool;
- Determining how the tool prompts developers to think about privacy throughout the software development process.

6.1 The Design of PARROT

We designed and implemented the tool using Eclipse Sirius (eclipse.org/sirius), that offers the ability to build domain-specific modelling tools. First, we created the domain model based on the Eclipse Modelling Framework (EMF) and then the graphical editor at run time. Furthermore, to encourage early lawyer involvement in the development process, we supported a lawyer-developer friendly representations "sandbox" for handling the system under construction. Ultimately, developers can use the tool as a dynamic tool to develop privacy aware IoT application designs, as seen in Figure 7.

Nudging developers to select better privacy practices when designing SW can be challenging. For example, applications must provide a privacy notice to their users if personal data is collected, ideally. "This should be a standard part of any software development for any developer," the lawyer said. However, presenting this in a user-friendly way may pose an additional challenge. Well known tools such as Node-RED display a status message and icon below the node to indicate the state of the node [9]. In PARROT, we tried to identify a trade-off between overwhelming developers with possible suggestions and relying on the limited knowledge a developer may have to embed privacy techniques in their system. Visualisation with many notifications and pop-up warning messages could result in developers ignoring them and, thus, failing to address privacy concerns [58]. One means of visualisation is the use of one or more of the three modes of representation. According to Bruner:

> "Any domain of knowledge [. . .] can be represented in three ways: by a set of actions appropriate for achieving a certain result (enactive representation); by a set of summary images or graphics that stand for a concept without defining it fully (iconic representation); and by a set of symbolic or logical propositions drawn from a symbolic system that is governed by rules or laws for forming and transforming propositions (symbolic representation)." [13]

Privacy-preserving systems such as Aquilis [55] rely on a simple three-colour coding system, that is standard and commonly used in the health sector, to remind users about potential privacy breaches. In PARROT, we relied on different criteria while selecting the notations, such as whether they were semiotically clear and visually expressive. Therefore, we sought to identify simplified visual notations using shapes, size and colour [65]. We adopted a colouring scheme using risk code colouring (green, yellow, orange and red), in addition to using icons to visually reflect which privacy issue is at stake, as explained in Figure 8. The iconic representation and colours are supported by a window featuring properties which has a combination of Yes/No questions. The questions could be a direct, such as "did you consider authentication for the doctor?" or indirect, such as"Are you planning to collect data other than that necessary for the purpose at hand such as heart-beat rate?" The purpose of using this combination is to help the developer think without rushing to answer Yes/No without reflecting on the potential privacy consequences. Changing these properties changes the colours in the design sketch. For example, if the privacy issue is related to a cloud location, there will be a circular node which has an icon indicating

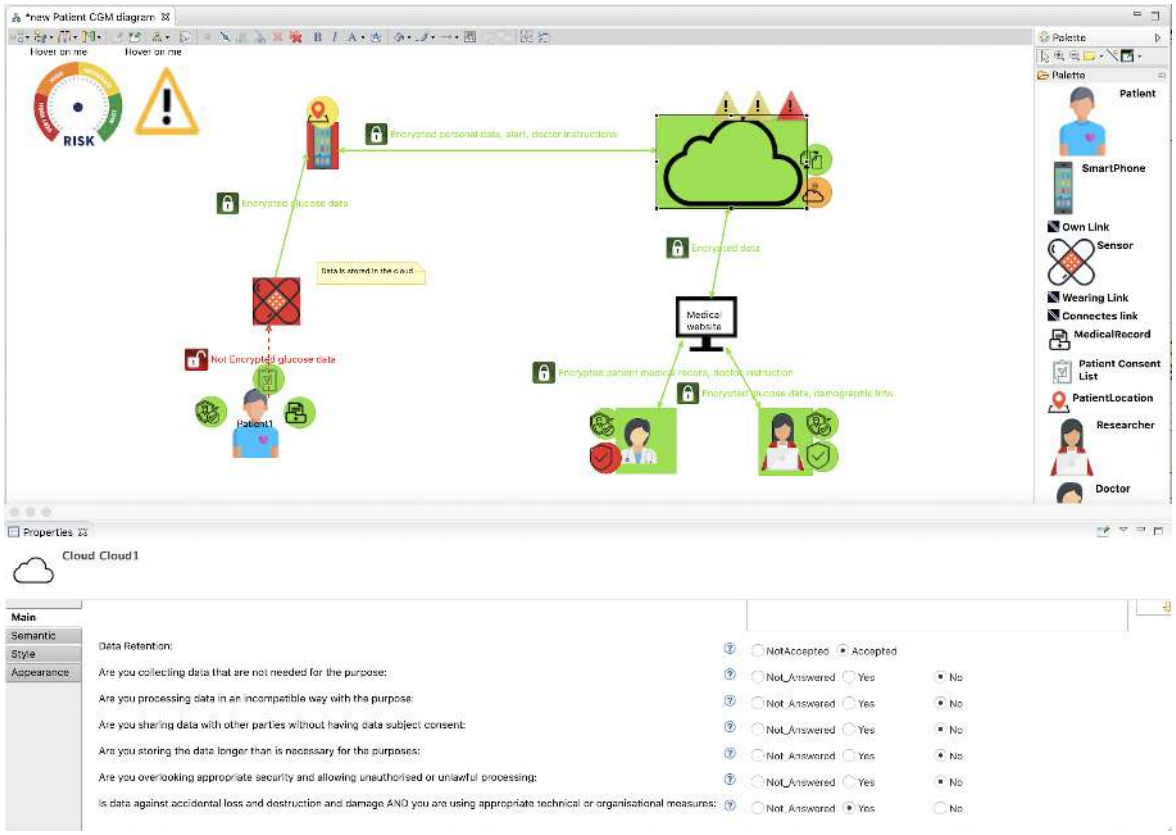

Fig. 7. PARROT prototype tool interface. In the middle, there is the design area. At the right, there is the palette where the developer drags form and drops in the design area. At the bottom, there is the properties section where the developer configures the privacy properties by answering multiple questions related to the selected node or sub-node.

the location, and the background colour of the node reflects the privacy risk, if any. We have listed the privacy breakdowns and explained how and why the nodes are visually represented in Appendix G, (Table A3). Also, we explained how to nudge the developer into embedding privacy whilst designing. In short, our intention is to enhance developers' PbD capabilities as if a lawyer were guiding them. Figure 9 shows that using PARROT to represent privacy makes the visual design simpler than expressing the privacy threats using text.

6.2 Evaluation

Our evaluation methodology was inspired by comparable techniques, particularly the one used for Coconut [58] and LINDDUN [26] which included the adoption of a use case-based evaluation technique [70]. To minimise the effect of any bias, we followed some techniques such as randomisation and partial blinding. We used randomised assignment, where participants were assigned randomly to control and treatment groups [53]. In the evaluation, we adopted a strict procedure so all the participants receive the same amount of attention throughout the experiment to reduce the risk of differential behaviours [33]. In medical experiments, double-blinded trials are used to prevent bias, which is an effective solution [33]. This experimental protocol is not feasible in software engineering experiments (SE) since they rely on a subject performing a human-intensive task [53]. However, we

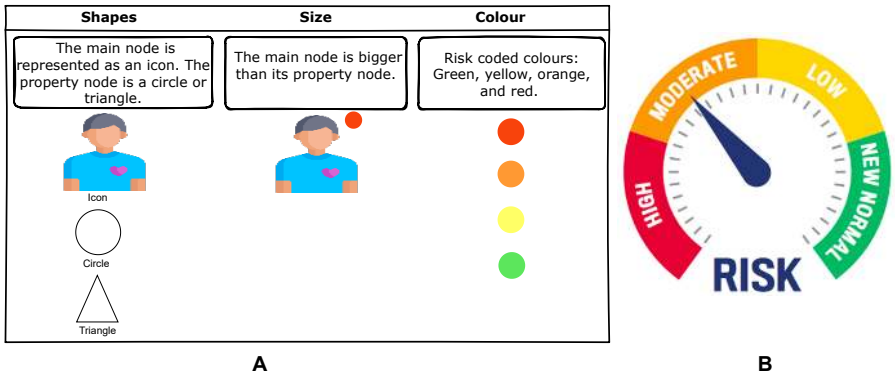

Fig. 8. A: Simplified visual symbol set for representing the IoT use-case. Shapes: icon inside a circle represents a property node, where the circle means it is a part of the PbD phase that the developer needs to consider; the triangle represents a good practice but not necessary part of the design phase (good to pay attention); the icon not surrounded by a circle represents the main node. Size: the main node is larger than its own property node. Colour: green, yellow, orange and red are used to represent the level of privacy/security risks. B: Risk coded colours: green, yellow, orange and red. Each colour reflects a degree of privacy/security risk. Red means the privacy/security risk is very high. Orange means the risk is high. Yellow means the risk is moderate. Green means the risk is low.

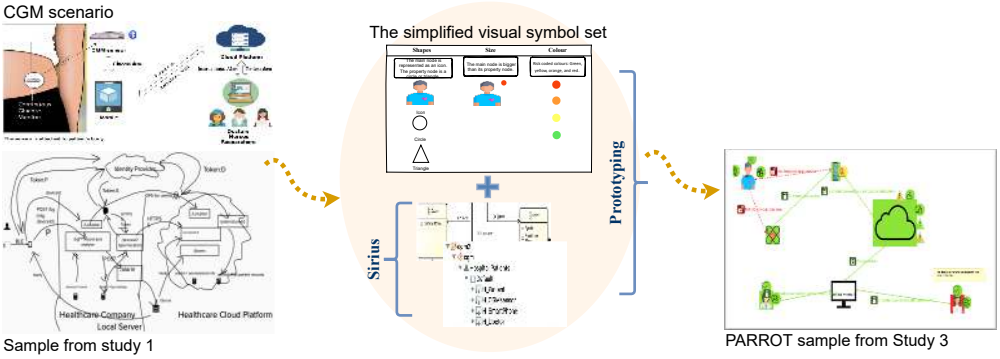

Fig. 9. Integrating privacy without using PARROT (left-side) and with using PARROT (right-side) for the same use case.

adopted blind data analysis and duplicating the data analysis which are commonly used in SE studies [31, 77]. The evaluation is based on two studies as follows:

(1) **Study 1 (Primary):** This is our primary study in which we tested our main hypothesis: 'Can the proposed tool help software developers design more privacy-aware IoT applications than they would otherwise?' Besides, we explored developers' perceptions of the tool features and their usefulness. The study focused on both qualitative discussions of participants' thoughts and ideas and quantitative data (to test the hypothesis).
(2) **Study 2 (Secondary):** We sought to strengthen and generalise our findings from the primary study. Because the requirements were captured using a CGM use case, we assessed if other use cases would present further ones. We added two further use cases (Smart home and Bus routing) to cover any missing requirements.

6.2.1 *Study 1 (Primary):*

Objectives. The aim of this study is to explore how the tool helps developers to design privacy-aware IoT applications. We conducted two lab studies to answer the following questions: (1) Can the proposed tool reduce the previously identified privacy issues? (2) Can the proposed tool help developers to create more privacy-preserving IoT applications (3) Can the proposed tool offer help to developers?

Participants. We sent invitations to multiple SW developers (who designs IoT apps or mobile apps with sensors) via email, LinkedIn and Twitter. Only half of them accepted to participate in the study after multiple reminders. We recruited 34 software developers where 12 have 6+ years of experience; the rest have 1-5 years. Among the 34, we managed to re-recruit 14 participants from phase 1 to test if the tool reduced the issue identified previously.

Procedure. During the design, we conducted a pilot study to obtain feedback from two developers and two privacy professionals to improve usability and validity. We followed an iterative design methodology by prototype and then assessed the prototype again until the initial tool was produced. For evaluation, we conducted two lab studies to examine the tool's usability and effectiveness. First, we conducted a within-subject study for 14 of the participants who previously participated in phase 1 to compare their designs from Mural and PARROT. Second, we undertook a between-subject study with 10 participants as a control group (using MURAL) and 10 as an experimental group (using PARROT). We divided this process into three steps, as follows:

- Firstly, 15-minute study procedure and tutorial. We gave the experimental group remote access to PARROT. Then we asked them to design the CGM use case (see Appendix A) while configuring the privacy properties that the tool offers. For the control group, we gave each participant a Mural link.
- After the design, we asked the developers who experimented with PARROT to answer three questions to gauge the usability and usefulness of the tool based on a 1-7 Likert scale and Microsoft reaction cards. Moreover, we asked all of the participants open-ended questions for qualitative analysis.
- Lastly, we held iterative meetings with a privacy lawyer to perform privacy risk analysis. This step produced 34 scored designs based on the same six design principles relied upon in phase 1.

Data collection. We had 34 different IoT application designs of the proposed CGM use case. During the interviews, we video and audio recorded the participants' designs and discussions for qualitative analysis purposes.

6.2.2 Study 2 (Secondary):

Objectives. The purpose of this study was to examine the ability of the tool to offer help in different use cases (see Appendix A). We sought to generalise our findings from the primary study. We conducted a lab study to answer the following questions: (1) Is the proposed tool able to scale to different domains? (2) Can the proposed tool help developers create more privacy-preserving IoT applications in different domains?

Participants. We hired 12 students in computer science who have worked for at least one year in IoT apps or mobile apps with sensor. Students are often used in software engineering studies instead of professional software developers [28, 47, 79]. We sent invitations via email and mailing lists. Among the participants were 2 PhD students, 4 masters students and 6 bachelor's students. The bachelor's students developed IoT application projects as part of the Network Communication module. None of the participants had received official privacy training.

Procedure. We conducted a between-subjects evaluation for the 12 participants. Therefore, each participant was allocated to one of the two conditions (using or not using PARROT). We divided the participants into experimental (E) and control (C) groups. Each group comprised 6 participants and the participants in both groups worked in pairs [20, 27, 29]. We divided this stage into three steps, as follows:

- Firstly, each group was given a 5 minute warm-up tutorial (PARROT for Group E and Mural for Group C). Then both groups were asked to undertake design tasks for smart home and bus route use cases. The pairs

Fig. 10. A comparison between the privacy scores in three rounds for the same participants from phase 1 (Section 4). (Round 1: using the Mural design only; Round 2: using the Mural design with an audio transcript; Round 3: using PARROT). The x-axis denotes the participants' ID.

in the experimental group were given remote access to the PARROT tool, whereas the pairs in the control group were given a link to Mural.
- We asked all of the participants open questions about how they typically address privacy requirements.
- Finally, the privacy lawyer assessed the design from a legal perspective.

Data collection. At the end of this study, we collected 12 IoT application designs (6 designs for the smart home and 6 for the bus route use cases). We also had 12 scored IoT application designs.

7 FINDINGS AND RESULTS

7.1 Quantitative Analysis (exploring effectiveness)

7.1.1 Privacy assessment. In **the primary study**, a Wilcoxon test was performed for the within-subject study to determine if the median score using Mural was less than the median score using PARROT for the same participants [5]. The Wilcoxon test revealed that there was a significant difference (p-value = 0.0005301 <0.05), with Mural producing lower privacy scores than PARROT. The Mann-Whitney U test revealed a significant difference in the privacy scores of Mural and PARROT (p-value = 0.0001717<0.05), whereby the scores of the participants who used PARROT were better than those who used Mural. The results for both studies are shown in Figures 10 and 11 (a and b) . In **the secondary study**, the Mann-Whitney U test revealed a significant difference in privacy scores between Mural and PARROT for the smart-home (p-value = 0.0463<0.05) and bus use case (p-value = 0.0463<0.05), whereby the scores of the participants who used PARROT were better than those who used Mural. The score results for both studies are shown in Figure 12 (a and b).

The privacy lawyer affirmed that the tool appears to be working adequately to trigger privacy thinking overall. Mixing direct and indirect questions ensures that users read and comprehend their options for each node. Even though some nodes turned to amber or red, which means there are some privacy issues, the developers attempted to justify their choices on a node level which is easier for privacy experts to understand, as the lawyer expressed.

7.1.2 How developers perceive PARROT?. Because adding privacy properties to the design can impose an additional cost for developers, we evaluated if they perceived the prototype features as disruptive, difficult to use and time-consuming using a Likert scale [8, 42, 58]. In Figure 13, we illustrate the complete results of the questions as a coordinated set of diverging stacked bar charts. The results indicate that developers perceived a cost for adding sub-nodes and configuring them, which is reasonable. Three of them said that after a couple of minutes of using the tool, they felt more comfortable using it. The hover box, on the other hand, was found to be difficult to understand and time-consuming. Overall, most of the developers found PARROT's features to be useful, as seen

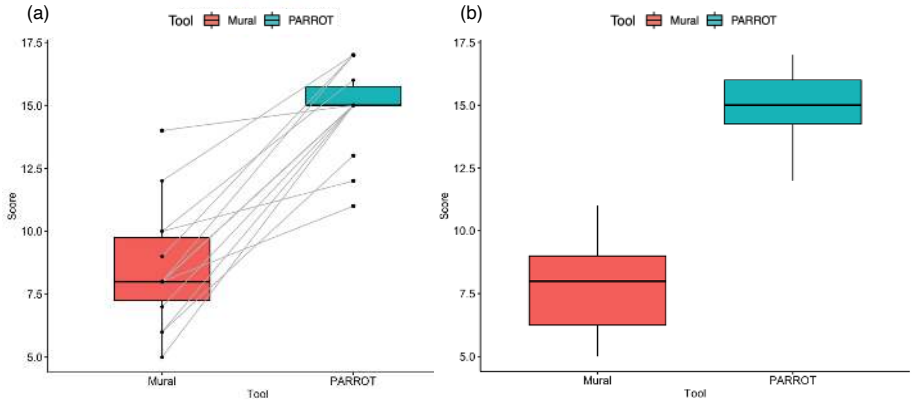

Fig. 11. (a)Mean rates of privacy principles scores in rounds 2 and 3 of the within subjects study; (b) Mean rates of privacy principles scores in Mural and PARROT. Note: both are for CGM use case.

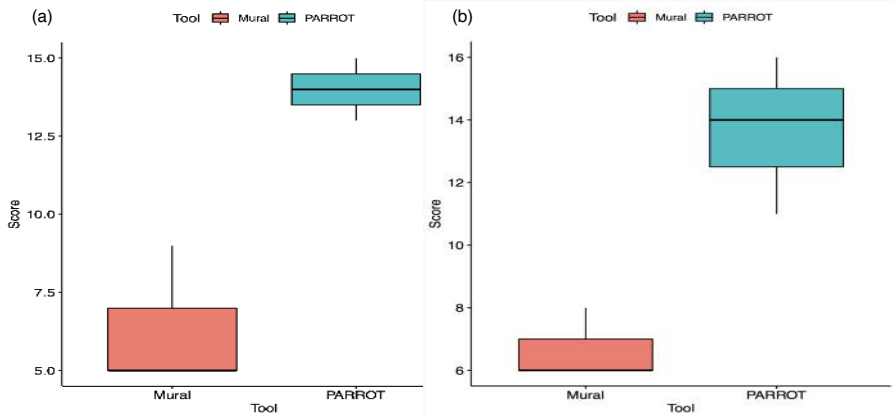

Fig. 12. (a) Mean rates of privacy principles scores for smarthome use-case; (b)Mean rates of privacy principles scores in bus use-case.

in the stacked column in Figure 14. Features such as the use of solid or dashed lines and different node sizes were perceived to be less helpful as compared to others. All of the questions regarding usability and usefulness are listed in Appendices D and E.

Using standard methods, such as satisfaction scores for some usability attributes such as efficiency and effectiveness can have limitations. Because general usability statements are commonly written positively, participants are more likely to agree than disagree with them, thus biasing the results. Therefore, we used Microsoft's reaction words to evaluate the usability of the tool generally [3, 99], as seen in the bar-chart in Figure 15. Generally, PARROT was described positively by most of the participants. Each participant selected five words that describe using PARROT. It can be seen from the chart that 71% of the participants considered PARROT to be useful and helpful (positive description). In contrast, no one described it as 'not valuable,' 'stressful' or 'annoying,' all of which are negative words. When the participants were asked why they found the tool useful, one of the developers

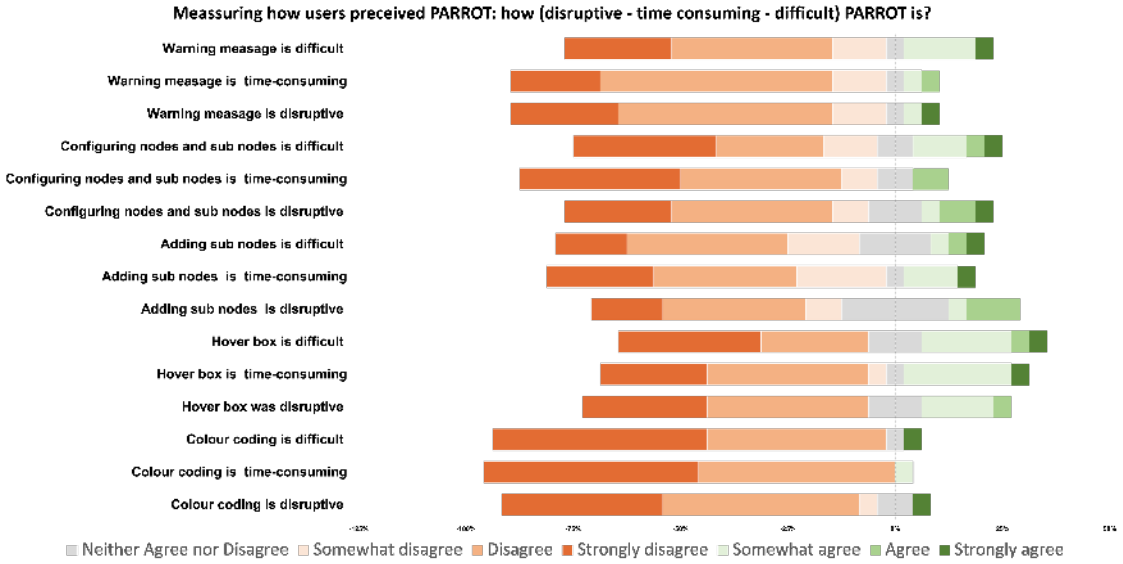

Fig. 13. Measuring how PARROT is perceived in terms of disruptiveness, time-consumption and difficulty. A subjective rating on a 1-7 Likert scale applies to all of the questions, where 7 indicates 'strongly agree' with the negative description and 1 indicates 'strongly disagree.'

explained that he would use this kind of tool to apply privacy during design immediately. Other developers said this tool is useful for showing the team manager the design at the weekly meeting and discussing privacy at a glance. On the other hand, they described it as helpful because it makes privacy more visually approachable. P11 stated that this tool will help him identify the privacy issues at the node level which is very focused, rather than the textual description on a checklist. P18 explained that this tool would help the design architect to follow up with the team. As a member of a large development team, "this tool would help us see and discuss privacy issues on each part of the design," P19 noted.

'Easy to use' and 'efficient' were the second most selected words by the participants (46% for each). In turn, 38% of the developers found the tool to be clear, and 21% and 13% found it to be complex and time-consuming, respectively. When asked why they perceived the tool to be complex and time-consuming, they said because it is a 'new' concept to them and it takes time to become familiar with the tool. Three of them said that after a couple of minutes, they felt comfortable using the tool because it became obvious instead of complex. This situation could happen with similar SW development tools when introducing a new concept, such as Coconut [58]. Further analysis was conducted regarding the negative descriptions. One of the participants comes from the underlying framework (Sirius) and not from the prototype tool itself (PARROT). In general, most of the participants reacted positively towards PARROT. However, because they had only been exposed to the tool for a short period of time, their design ability might have been adversely affected by their unfamiliarity with the new notation.

7.2 Qualitative Analysis and Discussion

For quantitative analyses, we used Miles' methods [64] and Richards' three coding techniques (descriptive, topic and analytic coding) [76]. Then we extracted the common themes similar to what we did in phase 1 Section 4.2. Below, we go over the qualitative analysis results alongside the lessons learned.

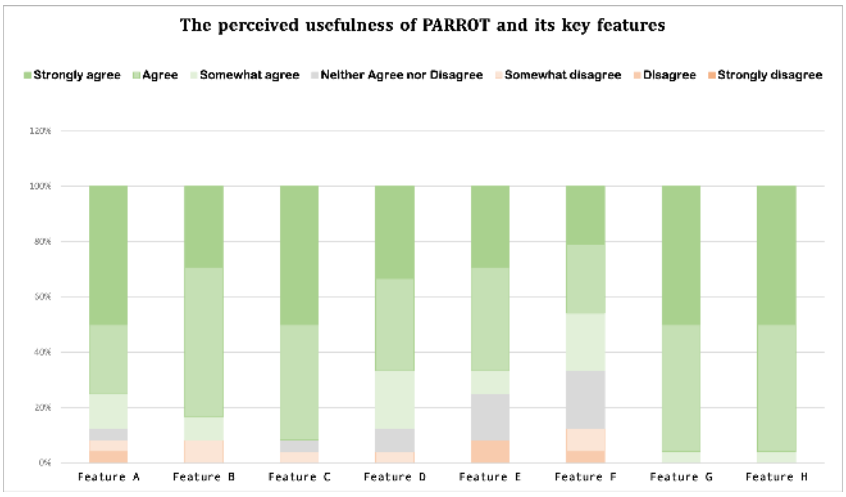

Fig. 14. Interpreted usefulness of PARROT and key features (7 for very useful, 1 for not useful at all). The features are A: sub nodes configuration; B: mouseover features; C: icons; D: shapes; E: sizes; F: dashed or solid line; G: colour coding for the line; H: colour coding for the nodes.

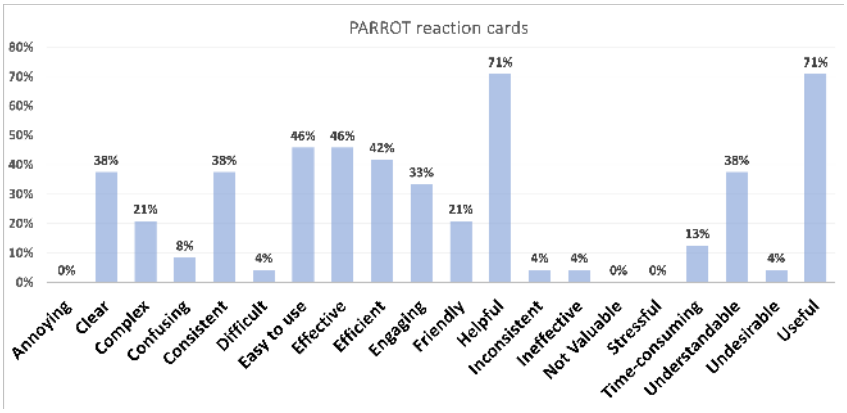

Fig. 15. Using Microsoft reaction cards to test the usability of the tool. There is a list of 20 words that could describe PARROT (10 negative words, 10 positive words). Generally, PARROT was described positively by most of the participants.

7.2.1 PbD principles are important Vs. challenges in implementing them. In this section, we discuss the privacy principles, how developers embed them during software development and which tools help them, if any. When we asked who looks at privacy initially, P24 said, "in our company, it is basically the architect's job, then the security team." Meanwhile, P12 said, "it is a team effort as developers but on top of us we have some security specialists working for our company, so they also check on those issues." P5, P6, P12 and P20 said that their companies integrate security and privacy as early as possible such as in the design phase when we asked them at which SW development stage privacy is applied. P5 stated that "at an early stage, we use data protection impact assessment (DPIA), we look at which personal information is going to be used or stored." In case there is a need for personal data, they apply "a combination of tokenisation and minimisation," P5 continued. P20 said if we

have privacy issues at the design we investigate and then, "if we have further questions from that, I suppose, talk to more senior developers and see if they can give us any insight."

Most developers send their designs to another team or legal company to ensure appropriate privacy. P24 states that even though "we had multiple training sessions and workshops about privacy, once the design is done another security team review it and we refine it again and again and again." P18 said when it comes to "privacy concerns, we don't handle it, we just transfer it to the team." P21 explains that applying privacy for small companies is challenging. They tend to deal with other security companies providing a framework to help them comply with GDPR. "Because smaller companies just have not got the time and the people to do it," P21 stated.

Many developers believed that the PbD concept is important but it is complicated and there is not much support in the form of tools. P12, P15, P21 and P24 said they did not use a tool to help embed privacy and instead relied on their experience but they noted that having a tool to support privacy would be helpful. P21 said, "we are a very small company, so I take that responsibility" (from his own experience). P5, P6, P14 P20 said that they try to avoid using any personal data with the software, so they do not have to deal with privacy compliance regulation at all. When we asked if using a checklist could help apply privacy, P12, P14, P15 and P21 said it could help in general but not on each node of the system such as PARROT does.

7.2.2 Does PARROT help incorporate privacy? We discuss how the tool helps integrate privacy principles with the privacy lawyer and two privacy professionals (PP1 and PP2) in addition to the study participants. From a privacy perspective, PARROT embedded privacy design properties in the IoT application "from the beginning rather than retrospectively," the privacy lawyer said. Applying privacy principles to protect data in the early development stage can save resources when achieving privacy compliance throughout a commercial environment's design and testing process. When we asked how PARROT could help the developers and what the expected interpretation for each node in the tool is, PP2 stated, "I would primarily be focused on the sub-nodes as opposed to the nodes. This is where compliance with data protection obligations starts to be considered." For example, the Surface Privacy Notice (Yes/No) sub-node is one instance where "privacy risks would be flagged." PP2 also continued, "the colour coding means they are clearly informed about what combinations lead to privacy compliant solutions."

As stated, applying privacy principles is mandatory due to data privacy laws in many countries such as the California Consumer Privacy Act (CCPA) in the US, Brazil's General Data Protection Law (LGPD), and The Personal Information Protection and Electronic Documents Act (PIPEDA) etc [96]. However, "transferring these concepts to software architecture is challenging," PP1 said. For example, in the GDPR, the first principle is lawfulness, fairness and transparency. It essentially means that data needs to be processed for lawful purposes, "which is a very broad term and subject to interpretation," PP1 said. By offering privacy configurations on the node level and using colour coding, the tool helps to overcome the complications. These data might be derived and processed for different purposes with different parties. In order to ensure compliance with the purpose limitation principle, "data has to be monitored on an ongoing basis to ensure it is not used for other purposes," PP1 stated. It is an important principle but the process of embedding it in the system architecture is complicated. Having the consent sub-nodes on PARROT will trigger the developers to think about how important it is.

P24 said PARROT could also help the project manager shorten the time spent reviewing and refining designs "since if the tool is there, we can have the design faster by identifying threats early." While P18 thought that the solution architect is the one who will use this tool, P20 and P21 assumed that the senior developer is the one who will most benefit from PARROT. "Quite early in the project, the senior developers and an architect will use the PARROT tool" P20 stated. P20 described how PARROT is useful, "it gives you an idea of how the sort of systems need to work together to gain the best privacy measures. It also gives you an indication of what factors contribute to that privacy." P21 said visual notation in PARROT helps "not only testing the privacy stuff but you want them (developers) to be engaged as well, so it's an opportunity for them to learn." P21 also stated "the traffic light system is useful to know where you have to look and the questions sort of lead you to consider things you

haven't considered." P12, P14, P15, P20 and P21 preferred PARROT to a privacy checklist. P12 said PARROT is faster than the list "having a pictorial representation rather than texts is easier to grasp at a glance other than going through each and every item in the list." Meanwhile, P14's reason for favouring PARROT was "because you can see it visually and that's how the brain works rather than a checklist; it'd be very difficult." Both P15 and P20 preferred the tool due to the colour coding feature. P20 said "it is interactive and because you have the colour-coding it is easier to see whether you are making the right decision or not." P15 said "the checklist does not have a colour code. The checklist will be complex if we have to look at them on the aspect of a particular node spectrum."

7.2.3 The scalability of using PARROT to ensure compliance with changes in privacy laws. Privacy regulations typically evolve with time. For example, e-privacy is likely to succeed the GDPR, which came into force on May 25, 2018. As with all legislative updates, there is usually a compliance period factored into its publication and an understanding from regulators that updates can not be put in place immediately. Our focus is currently considering the existing legislation that is already known (i.e. GDPR). We also expect PARROT to be applicable to other legislation, such as e-privacy. The tool is designed to be adaptive enough that as legislation changes, these new additional requirements can be captured. For example, we plan to add a tool box that enables the developer to select a specific privacy legislation, and based on that choice, the privacy configuration for the app will change.

7.2.4 Recommendation to ensure privacy by design over time. PbD principles apply across the lifecycle of an application. In this paper, we explore and illustrate the potential of using PARROT as a PbD development tool to help developers to satisfy better compliance. We also expect this idea to be applicable to other parts of the development life cycle such as maintenance and support. For example, the tool could be used as a PbD audit tool by checking the status of compliance on an existing IoT app. The tool could also be used for the annual audit of all apps, and it could also be used in respect of any app updates. We plan to add some features such as a signed document. This document is generated once the developers finish the design and then signed by other parties such as a lawyer and the business analyst. This document can be carried out for the whole development cycle where other developers, support managers and technicians can add more details without changing the design.

7.2.5 Design recommendations for addressing privacy issues. Most of the participants explicitly mentioned how they benefited from the tool and they stated helpful suggestions. Participants P1, P2, P5, P20 and P23 suggested having help icon explaining the law/regulation behind each privacy issue. "It would be good to have a deeper explanation of the implications of the decisions the developer has to make to give a better understanding" P20 said. In addition, many participants suggested having more features (P7, P8, P9, P11, P12, P19 and P21). For example, two of them suggested having a simple indicator of how good the model is. "The tool should say to the developer that you have x number of issues to fix" P19 said. P11 also stated that prompts when there are any missing privacy properties in a node could be helpful. "When the use case gets too big, it will become hard to follow and having guiding symbols will be helpful" P11 explained. Other suggestions included, but were not limited to, having an import/export function and more design interactive feedback.

In addition, having more privacy questions and advice were suggested. However, at the stage tool prototyping, we were trying to strike a balance between adding more advice and improving the tool's usability. Providing a lot of security and privacy questions and recommendations is not a feasible approach because it is likely to affect the tool's usability for developers [2]. PARROT is not meant to replace privacy professionals, as previously stated. Instead, it seeks to establish and encourage communication between privacy experts and software developers.

8 LIMITATION AND FUTURE WORK

8.1 Limitation of the Evaluation of the Methodology

The tool is initially implemented based on a health use case where privacy may be straightforward. To assess the tool's capabilities in other domains, we added two non-health-related use cases. The privacy lawyer develops high-level privacy principles to ensure they are legally sound in different domains. Thus, translating these to other complex domains such as criminal law could be challenging. It is necessary to point out that the existing tool might be more beneficial for developers that have limited privacy knowledge. Moreover, recruiting a large number of developers for lab studies, as well as assessing the design processes has long been acknowledged as a challenge. First, the length of such studies' designing task can take several hours, which is difficult to scale-up. The risk is that their designs may not reflect what they would do in real-life development [2]. In addition, recruiting software professionals for research is difficult and time-consuming. Second, software developers usually work in teams but it was challenging to maintain this during the first phase, which we attempted to do in the secondary study (6.2.2) by making participants work collaboratively in pairs [20].

In the current study, we managed to recruit 18 developers in the first lab study (see Section 4) and 46 in the second (see Sections 6.2.1 and 6.2.2), for 1-2 hours. The number of participants and the duration of the current study is comparable to many prior studies that have contemplated design methodologies [10, 18, 54, 57, 58, 98]. We admit that the duration may be arbitrary. In practice, the time is very much dependent on the task and the expertise of the individual who agrees to participate in the interviews. To give participants a long time to think about the use-case, all the study materials including the use-case and personalised link to Mural (for the first phase) were emailed days before the interview. We use this study duration as a benchmark but it is necessary to emphasise that duration is only part of the bigger problem. Lastly, because privacy was explicitly stated as a design goal in the participants' recruiting invitation, it could affect their focus on this aspect of the design task. Consequently, it might influence the participants' results. Because we were concentrating on developers' PbD behaviour, not considering privacy as a study goal could make the study's results incomparable because developers might ignore privacy completely.

8.2 Future Work

We would like to explore the use of PARROT to improve privacy in three directions. The first is to enhance the current tool. For example, we observed the clear importance of integrating privacy laws into PARROT. Such integration would allow developers to see how their actions lead to better compliance with privacy laws. Profiling, cookies, and advertisements are also privacy threats that we plan to address. The second is to develop PARROT into an education tool. During this study, we realised that PARROT could be used to educate developers despite their level of expertise. Currently, no such tool support exists that could be used within university-level teaching. Even though considerable research has been conducted [22, 23, 84] relating to PbD techniques, there is no clear path for developers to learn and understand how such privacy-preserving measures could be applied into an application design. Finally, we believe that there is an opportunity to improve PARROT by developing a dedicated viewpoint for privacy lawyers. Currently, PARROT focuses on encouraging and guiding developers to integrate the most critical privacy-preserving measures into their IoT applications which reduces the breakdown between privacy lawyers and developers. However, a dedicated 'viewpoint' that shares a 'single source of truth' [57] will enable privacy lawyers to engage with software developers much more easily and naturally.

9 CONCLUSION

In this paper we present the design, implementation and evaluation results for PARROT. We made three contributions in this research paper: (i) investigate privacy awareness of software developers and with the help of a privacy lawyer and privacy professionals, we discovered privacy threats in early software design; (ii) introduced

PARROT, an interactive IoT application design tool which encourages developers to consider privacy during the design process by showing real-time feedback regarding potential privacy issues. To have such a prototype tool, we used operationalisation processes; (iii) we show the findings of our PARROT lab studies which indicated that PARROT developers create more privacy-preserving IoT applications. Our results show that PARROT-designed IoT apps better address privacy issues, enabling between discussion of privacy requirements between developers and privacy lawyers. The tool could be used and adapted for a wide range of applications. However, we specifically focus on IoT applications and GDPR compliance in the first instance. We focus on IoT applications that gather, store and analyse private data across multiple hardware and software systems, (potentially) revealing private information. It is important to bear in mind that PARROT offers no guarantees that IoT systems built using it will be free from all privacy issues. We believe, however, that software developers will be able to better understand where privacy principles are applicable.

ACKNOWLEDGMENTS

This work is partially supported by EPSRC PETRAS (EP/S035362/1) and PACE (EP/R033439/1). We would like to acknowledge the scholarship and support provided by King Saud University.

REFERENCES

[1] 2021. GDPR Enforcement Tracker - List of GDPR Fines' (2021). (2021). https://www.enforcementtracker.com/

[2] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. 2016. You are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users. In *2016 IEEE Cybersecurity Development (SecDev)*. 3–8. https://doi.org/10.1109/SecDev.2016.013

[3] Pierre A Akiki, Arosha K Bandara, and Yijun Yu. 2017. Visual Simple Transformations: Empowering End-Users to Wire Internet of Things Objects. *ACM Trans. Comput.-Hum. Interact* 24, 10 (2017). https://doi.org/10.1145/3057857

[4] Nada Alhirabi, Omer Rana, and Charith Perera. 2021. Security and Privacy Requirements for the Internet of Things: A Survey. *ACM Trans. Internet Things* 2, 1 (feb 2021). https://doi.org/10.1145/3437537

[5] Majedah Alrehiely, Parisa Eslambolchilar, and Rita Borgo. 2018. Evaluating Different Visualization Designs for Personal Health Data. April (2018). https://doi.org/10.14236/ewic/hci2018.205

[6] Oshrat Ayalon, Eran Toch, Irit Hadar, and Michael Birnhack. 2017. How developers make design decisions about Users' Privacy: The place of professional communities and organizational climate. *CSCW 2017 - Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (2017), 135–138. https://doi.org/10.1145/3022198.3026326

[7] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *The Journal of Systems and Software* 149 (2019), 101–137. https://doi.org/10.1016/j.jss.2018.11.041

[8] Alex Barth, Emmanuel Caillaud, Bertrand Rose, and Others. 2011. How to validate research in engineering design?. In *DS 68-2: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011.* 41–50.

[9] Michael Blackstock and Rodger Lea. 2014. Toward a Distributed Data Flow Platform for the Web of Things (Distributed Node-RED). In *Proceedings of the 5th International Workshop on Web of Things (WoT '14)*. Association for Computing Machinery, New York, NY, USA, 34–39. https://doi.org/10.1145/2684432.2684439

[10] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R Klemmer. 2010. *Example-Centric Programming: Integrating Web Search into the Development Environment.* Association for Computing Machinery, New York, NY, USA, 513–522. https://doi.org/10.1145/1753326.1753402

[11] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[12] Justine Brown. 2017. Why is IoT talent so hard to find? https://www.ciodive.com/news/why-is-iot-talent-so-hard-to-find/449576/

[13] Jerome Seymour Bruner and Others. 1966. *Toward a theory of instruction.* Vol. 59. Harvard University Press.

[14] Lee A Bygrave. 2017. Data protection by design and by default: Deciphering the EU's legislative requirements. *Oslo Law Review* 4, 02 (2017), 105–120.

[15] Ann Cavoukian. 2009. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada* 5 (2009), 12.

[16] A Cavoukian and S Kingsmill. 2016. Privacy by Design Setting a new standard for privacy certification. https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/risk/ca-en-ers-privacy-by-design-brochure.PDF

[17] Abhik Chaudhuri and Ann Cavoukian. 2018. The Proactive and Preventive Privacy (3P) Framework for IoT Privacy by Design. *EDPACS* 57, 1 (2018), 1–16.

[18] Yulia Cherdantseva. 2014. *Secure * BPMN - a graphical extension for BPMN 2 . 0 based on a Reference Model of Information Assurance & Security*. Ph.D. Dissertation. Cardiff University. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.655937

[19] Saksham Chitkara, Nishad Gothoskar, Suhas Harish, Jason I Hong, and Yuvraj Agarwal. 2017. Does this app really need my location? Context-aware privacy management for smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–22.

[20] Eric S. Chung, Jason I. Hong, Lin James, Madhu K. Prabaker, James A. Landay, and Alan L. Liu. 2004. Development and evaluation of emerging design patterns for ubiquitous computing. *DIS2004 - Designing Interactive Systems: Across the Spectrum* (2004), 233–242. https://doi.org/10.1145/1013115.1013148

[21] CNIL. 2021. The open source PIA software helps to carry out data protection impact assessment. https://www.cnil.fr/en/home

[22] Collaboration. 2015. Privacy patterns org. https://privacypatterns.org/

[23] Collaboration. 2016. Privacy patterns-collecting patterns for better privacy. https://privacypatterns.eu/#/?limit=6&offset=0

[24] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2019. My IoT Puzzle: Debugging IF-THEN Rules Through the Jigsaw Metaphor. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11553 LNCS. Springer Verlag, 18–33. https://doi.org/10.1007/978-3-030-24781-2_2

[25] George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Le Metayer, Rodica Tirtea, and Stefan Schiffner. 2015. Privacy and data protection by design-from policy to engineering. *arXiv preprint arXiv:1501.03726* (2015).

[26] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. 2011. A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering* 16, 1 (2011), 3–32. https://doi.org/10.1007/s00766-010-0115-7

[27] Paloma Díaz, Ignacio Aedo, Mary Beth Rosson, and John M Carroll. 2010. A visual tool for using design patterns as pattern languages. In *Proceedings of the International Conference on Advanced Visual Interfaces*. ACM, 67–74.

[28] Paloma Díaz, Ignacio Aedo, Daniel Sanz, and Alessio Malizia. 2008. A model-driven approach for the visual specification of Role-Based Access Control policies in web systems. *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008* (2008), 203–210. https://doi.org/10.1109/VLHCC.2008.4639087

[29] Paloma Díaz, Ignacio Aedo, Daniel Sanz, and Alessio Malizia. 2008. A model-driven approach for the visual specification of Role-Based Access Control policies in web systems. *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008* (2008), 203–210. https://doi.org/10.1109/VLHCC.2008.4639087

[30] Alan Dix and Geoffrey Ellis. 1998. Starting simple: adding value to static visualisation through simple interaction. In *Proceedings of the working conference on Advanced visual interfaces*. 124–134.

[31] Tore Dybå and Torgeir Dingsøyr. 2008. Strength of Evidence in Systematic Reviews in Software Engineering. *ESEM'08: Proceedings of the 2008 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* 7465 (2008), 178–187. https://doi.org/10.1145/1414004.1414034

[32] EDPB. 2021. Binding decision 1/2021 on the dispute arisen on the draft decision of the Irish Supervisory Authority regarding WhatsApp Ireland under Article 65(1)(a) GDPR. https://edpb.europa.eu/our-work-tools/our-documents/binding-decision-board-art-65/binding-decision-12021-dispute-arisen_en

[33] Dorothy Forbes. 2013. Blinding: An essential component in decreasing risk of bias in experimental designs. *Evidence-Based Nursing* 16, 3 (2013), 70–71. https://doi.org/10.1136/eb-2013-101382

[34] Abdur Rahim Mohammad Forkan, Geoff Kimm, Ahsan Morshed, Prem Prakash Jayaraman, Abhik Banerjee, and Weidong Huang. 2019. AqVision: A tool for air quality data visualisation and pollution-free route tracking for smart city. *Proceedings - 2019 23rd International Conference in Information Visualization - Part II, IV-2 2019* (2019), 47–51. https://doi.org/10.1109/IV-2.2019.00018

[35] GDPR. 2018. *Art. 83 GDPR. General conditions for imposing administrative fines*. Technical Report. https://gdpr-text.com/read/article-83/

[36] David Geer. 2010. Are companies actually using secure development life cycles? *Computer* 43, 6 (2010), 12–16.

[37] Daniel Le Métayer George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman and Stefan Schiffner Rodica Tirtea. 2014. *Data Protection by Design – From Policy to Engineering (European Union Agency for Network and Information Security 2014)*. Technical Report. https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design

[38] Giacomo Ghidini, Vipul Gupta, and Sajal K Das. 2010. SNViz: Analysis-oriented Visualization for the Internet of Things. In *IoT 2010 Workshop: The Urban Internet of Things*. Citeseer.

[39] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods* 18, 1 (2006), 59–82. https://doi.org/10.1177/1525822X05279903

[40] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. 2018. Privacy by designers: software developers' privacy mindset. *Empirical Software Engineering* 23, 1 (2018), 259–289. https://doi.org/10.1007/s10664-017-9517-1

[41] Julie M. Haney and Wayne G. Lutters. 2019. Motivating cybersecurity advocates: Implications for recruitment and retention. *SIGMIS-CPR 2019 - Proceedings of the 2019 Computers and People Research Conference* (2019), 109–117. https://doi.org/10.1145/3322385.3322388

[42] Richard M. Heiberger and Naomi B. Robbins. 2014. Design of diverging stacked bar charts for Likert scales and other applications. *Journal of Statistical Software* 57, 5 (2014), 1–32. https://doi.org/10.18637/jss.v057.i05
[43] Michael S.H. Heng, Eileen M. Trauth, and Sven J. Fischer. 1999. Organisational champions of IT innovation. *Accounting, Management and Information Technologies* 9, 3 (1999), 193–222. https://doi.org/10.1016/S0959-8022(99)00008-9
[44] Mireille Hildebrandt and Bert-Jaap Koops. 2010. The challenges of ambient law and legal protection in the profiling era. *The Modern Law Review* 73, 3 (2010), 428–460.
[45] Janine S Hiller and Roberta S Russell. 2017. Privacy in Crises: The NIST Privacy Framework. *Journal of Contingencies and Crisis Management* 25, 1 (2017), 31–38. https://doi.org/10.1111/1468-5973.12143
[46] Jaap-Henk Hoepman. 2014. Privacy Design Strategies. In *ICT Systems Security and Privacy Protection*, Nora Cuppens-Boulahia, Frédéric Cuppens, Sushil Jajodia, Anas Abou El Kalam, and Thierry Sans (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 446–459.
[47] Martin Höst, Björn Regnell, and Claes Wohlin. 2000. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering* 5, 3 (2000), 201–214.
[48] Information Commissioner's Office (ICO). 2014. Conducting privacy impact assessments code of practice. (2014), 1–55.
[49] Lukasz Jedrzejczyk, Blaine A Price, Arosha K Bandara, and Bashar Nuseibeh. 2010. On the impact of real-time feedback on users' behaviour in mobile location-sharing applications. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*. 1–12.
[50] Jeff Johnson and Austin Henderson. 2002. Conceptual models: begin by designing what to design. *interactions* 9, 1 (2002), 25–32.
[51] Esther Jun, Huafei Liao, April Savoy, Liang Zeng, and Gavriel Salvendy. 2008. *The design of future things, by D. A. Norman, basic books, New York, NY, USA*. Vol. 18. 480–481 pages. https://doi.org/10.1002/hfm.20127
[52] Himmet Karadal and A Mohammed Abubakar. 2021. Internet of things skills and needs satisfaction: do generational cohorts' variations matter? *Online Information Review* (2021).
[53] Barbara A Kitchenham and Tore Dybå. 2004. Evidence-based Software Engineering. (2004).
[54] Amy J Ko and Brad A Myers. 2004. Designing the Whyline: A Debugging Interface for Asking Questions about Program Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. Association for Computing Machinery, New York, NY, USA, 151–158. https://doi.org/Ko2004
[55] Abhishek Kumar, Tristan Braud, Young D Kwon, and Pan Hui. 2020. Aquilis: Using contextual integrity for privacy protection on mobile devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–28.
[56] Sachin Kumar, Prayag Tiwari, and Mikhail Zymbler. 2019. Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data* 6, 1 (2019). https://doi.org/10.1186/s40537-019-0268-2
[57] Germán Leiva, Nolwenn Maudet, Wendy Mackay, and Michel Beaudouin-Lafon. 2019. Enact: Reducing designer–developer breakdowns when prototyping custom interactions. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 3 (2019), 1–48.
[58] Tianshi Li, Yuvraj Agarwal, and Jason I. Hong. 2018. Coconut: An IDE Plugin for Developing Privacy-Friendly Apps. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 178 (Dec 2018), 35 pages. https://doi.org/10.1145/3287056
[59] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 2021. How Developers Talk About Personal Data and What It Means for User Privacy. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–28. https://doi.org/10.1145/3432919
[60] Diego Martin, Ramon Alcarria, Tomas Robles, and Augusto Morales. 2013. A systematic approach for service prosumerization in IoT scenarios. *Proceedings - 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013* (2013), 494–499. https://doi.org/10.1109/IMIS.2013.89
[61] Yod Samuel Martín García and José María del Álamo Ramiro. 2017. A metamodel for privacy engineering methods. CEUR Workshop Proceedings.
[62] Roberto Martinez-Maldonado, Andrew Clayphan, Kalina Yacef, and Judy Kay. 2014. MTFeedback: providing notifications to enhance teacher awareness of small group work in the classroom. *IEEE Transactions on Learning Technologies* 8, 2 (2014), 187–200.
[63] Ben Mathews and Delphine Collin-Vézina. 2016. Data for life: Wearable technology and the design of self-care. *Journal of Public Health Policy* 37, 3 (2016), 304–314. https://doi.org/10.1057/jphp.2016.21
[64] Matthew B Miles, A Michael Huberman, and Johnny Saldaña. 2018. *Qualitative data analysis: A methods sourcebook*. Sage publications.
[65] Daniel L. Moody, Patrick Heymans, and Raimundas Matulevičius. 2010. Visual syntax does matter: Improving the cognitive effectiveness of the i* visual notation. *Requirements Engineering* 15, 2 (2010), 141–175. https://doi.org/10.1007/s00766-010-0100-1
[66] Duc Cuong Nguyen, Dominik Wermke, Yasemin Acar, Michael Backes, Security Lancaster, and Sascha Fahl. 2017. A Stitch in Time: Supporting Android Developers in Writing Secure Code Charles Weir. (2017). https://doi.org/10.1145/3133956.3133977
[67] Marie Caroline Oetzel and Sarah Spiekermann. 2014. A systematic methodology for privacy impact assessments: a design science approach. *European Journal of Information Systems* 23, 2 (2014), 126–150.
[68] Avi Parush. 2015. Conceptual Design for Interactive Systems Designing for Performance and User Experience. Morgan Kaufmann, Boston, 164. https://doi.org/10.1016/B978-0-12-419969-9.09992-7
[69] Charith Perera. 2017. Privacy Guidelines for Internet of Things: A Cheat Sheet. (2017), 1–9. arXiv:1708.05261 http://arxiv.org/abs/1708.05261

[70] Charith Perera, Mahmoud Barhamgi, Arosha K Bandara, Muhammad Ajmal, Blaine Price, and Bashar Nuseibeh. 2020. Designing privacy-aware internet of things applications. *Information Sciences* 512 (2020), 238–257. https://doi.org/10.1016/j.ins.2019.09.061

[71] Charith Perera, Ciaran McCormick, Arosha K. Bandara, Blaine A. Price, and Bashar Nuseibeh. 2016. Privacy-by-design framework for assessing internet of things applications and platforms. *ACM International Conference Proceeding Series* 07-09-Nove (2016), 83–92. https://doi.org/10.1145/2991561.2991566

[72] Ferry Pramudianto, Carlos Alberto Kamienski, Eduardo Souto, Fabrizio Borelli, Lucas L. Gomes, Djamel Sadok, and Matthias Jarke. 2014. IoT Link: An Internet of Things Prototyping Toolkit. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*. 1–9. https://doi.org/10.1109/UIC-ATC-ScalCom.2014.95

[73] Insan Laksana Pribadi and Muhammad Suryanegara. 2017. Regulatory recommendations for IoT smart-health care services by using privacy impact assessment (PIA). In *2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering*. IEEE, 491–496.

[74] Keith F Punch. 2013. *Introduction to social research: Quantitative and qualitative approaches*. Sage publications.

[75] Shirley Radack. 2009. *The system development life cycle (sdlc)*. Technical Report. National Institute of Standards and Technology.

[76] Lyn Richards. 2020. *Handling qualitative data: A practical guide*. Sage publications.

[77] Simone Romano, Davide Fucci, Giuseppe Scanniello, Maria Teresa Baldassarre, Burak Turhan, and Natalia Juristo. 2021. On researcher bias in Software Engineering experiments. *Journal of Systems and Software* 182 (2021), 111068. https://doi.org/10.1016/j.jss.2021.111068

[78] Nayan B Ruparelia. 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes* 35, 3 (2010), 8–13.

[79] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. 2015. Are students representatives of professionals in software engineering experiments? *Proceedings - International Conference on Software Engineering* 1 (2015), 666–676. https://doi.org/10.1109/ICSE.2015.82

[80] Panagiotis Sarigiannidis, Eirini Karapistoli, and Anastasios A. Economides. 2015. VisIoT: A threat visualisation tool for IoT systems security. *2015 IEEE International Conference on Communication Workshop, ICCW 2015* (2015), 2633–2638. https://doi.org/10.1109/ICCW.2015.7247576

[81] Awanthika Senarath and Nalin A.G. Arachchilage. 2018. Why developers cannot embed privacy into software systems? An empirical investigation. *ACM International Conference Proceeding Series* Part F1377 (2018). https://doi.org/10.1145/3210459.3210484 arXiv:1805.09485

[82] Awanthika Senarath, Marthie Grobler, and Nalin Asanka Gamagedara Arachchilage. 2019. Will they use it or not? Investigating software developers' intention to follow privacy engineering methodologies. *ACM Transactions on Privacy and Security* 22, 4 (2019). https://doi.org/10.1145/336422

[83] Laurens Sion, Pierre Dewitte, Dimitri Van Landuyt, Kim Wuyts, Ivo Emanuilov, Peggy Valcke, and Wouter Joosen. 2019. An architectural view for data protection by design. *Proceedings - 2019 IEEE International Conference on Software Architecture, ICSA 2019* i (2019), 11–20. https://doi.org/10.1109/ICSA.2019.00010

[84] Spanish Data Protection Agency. 2019. *A Guide to Privacy by Design*. Number october. https://www.aepd.es/sites/default/files/2019-12/guia-privacidad-desde-diseno_en.pdf

[85] S Steffee. 2017. IOT HELP WANTED: A lack of Internet of Things knowledge–and skills–leaves businesses struggling to recruit talent. *Internal Auditor* 74, 5 (2017), 11–13.

[86] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Privacy champions in sofware teams: Understanding their motivations, strategies, and challenges. *Conference on Human Factors in Computing Systems - Proceedings* (2021). https://doi.org/10.1145/3411764.3445768

[87] Mohammad Tahaei, Li Tianshi, and Vaniea. Kami. 2022. Understanding Privacy-Related Advice on Stack Overflow. *Proceedings on Privacy Enhancing Technologies* 2022, 2 (2022), 1–5. https://doi.org/10.2478/popets-2022-0032

[88] The Court of Justice and of the European Union. 2020. C-311/18 - Facebook Ireland and Schrems. https://curia.europa.eu/juris/liste.jsf?num=C-311/18

[89] The Members and staff of the European Parliament. [n.d.]. The CJEU judgment in the Schrems II case. https://www.europarl.europa.eu/RegData/etudes/ATAG/2020/652073/EPRS_ATA(2020)652073_EN.pdf

[90] Soe Ye Yint Tun, Samaneh Madanian, and Farhaan Mirza. 2021. Internet of things (IoT) applications for elderly care: a reflective review. *Aging Clinical and Experimental Research* 33, 4 (2021), 855–867. https://doi.org/10.1007/s40520-020-01545-9

[91] Council of the European Union and European Parliament. 2016. 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) (Text with EEA relevance). https://op.europa.eu/en/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1

[92] Onoriode Uviase and Gerald Kotonya. 2018. IoT architectural framework: Connection and integration framework for IoT systems. *Electronic Proceedings in Theoretical Computer Science, EPTCS* 264 (2018), 1–17. https://doi.org/10.4204/EPTCS.264.1

[93] Ovidiu Vermesan and Peter Friess. 2013. *Internet of things: converging technologies for smart environments and integrated ecosystems*. River publishers.

[94] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, et al. 2011. Internet of things strategic research roadmap. *Internet of things-global technological and societal trends* 1, 2011 (2011), 9–52.

[95] Sandra Wachter. 2018. Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR. *Computer law & security review& security review* 34, 3 (2018), 436–449.

[96] Wirewheel. 2021. Data Privacy Laws in 2021: What You Need to Know. https://wirewheel.io/data-privacy-laws-guide/

[97] Steven A. Wright. 2019. Privacy in IoT Blockchains: With Big Data comes Big Responsibility. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019* (2019), 5282–5291. https://doi.org/10.1109/BigData47090.2019.9006341

[98] Kim Wuyts. 2015. *Privacy Threats in Software Architectures*. Ph.D. Dissertation. https://lirias.kuleuven.be/retrieve/295669

[99] Shea Tinn Yeh and Cathalina Fontenelle. 2012. Usability study of a mobile website: The Health Sciences Library, University of Colorado Anschutz Medical Campus, experience. *Journal of the Medical Library Association* 100, 1 (2012), 64–68. https://doi.org/10.3163/1536-5050.100.1.012

[100] Gail A. Zieman. 2012. Participant observation. *Action Research Methods: Plain and Simple* (2012), 49–67. https://doi.org/10.1057/9781137046635

A APPENDIX: STUDIES USE CASES

Use case 1: Diabetes treatment and monitoring.

This use case scenario is presented from a problem owner's perspective. We want to develop an IoT application that can solve this problem. This application should analyse patient health data and produce an alarm to notify the patient and the medical staffs. This scenario has many privacy challenges that need to be considered and highlighted.

Sara is a researcher in a healthcare company that studies diabetes and how to find a treatment that can cure this chronic disease. To do that, the healthcare company wants to develop an IoT application that can assist diabetic patients in controlling the symptoms to some degree where patients with diabetes require treatment and continual monitoring. Sara is concerned about gathering and analysing data from Continuous Glucose Monitor (CGM) devices worn by patients where the sensor is placed into the patient's body, not into his bloodstream as seen in the figure below. The sensor measures the glucose in the patient interstitial fluid by taking readings at regular intervals for several days. Sara has a monitoring application that can recognise any triggers or patterns for abnormal glucose levels. Sara could study some of the factors that affect diabetes by analysing the collected glucose readings from many patients with other values such as age, food habits, demographic group etc. Sara has limited access to the data, and she should not have access to the personal information of the patient such as name, medical record and exact location. Also, Sara should not have any direct communication with the patient at all.

Medical staffs are other users of this application, nurses and doctors in particular, where they have a level of access to patient data such as the medical record for following-up and provide essential instructions when required. These instructions may include suitable insulin dosage, an exercise plan, daily meals\snacks, and types\dosage of medications. Medical staffs have different access than the researcher where the level of access is set by the hospital itself.

Q: The data is transferring from one place to another as seen in the figure, could you please draw the previous IoT use case while highlighting some of the privacy issues based on your experience?

Use case 2: Smart home scenario

Consider a service that aims to build a smart home app. In this use case, the smart home has four sensors: thermostat, light, camera, and door lock. The app's primary purpose is to control all these sensors, such as opening the car door lock or truing the light on/off when the user wants. The data transfers back and forth between the sensors to the cloud to the phone app. In this app, we aim to control the sensor on the stated purpose for each one of the sensors while keeping the client's privacy protected. Note: We have assumed that the app provider is not profiling the user in relation to the data collected by the sensors.

Q: Based on the previous scenario, could you please draw an IoT use case while considering GDPR privacy compliance?

Use case 3: Video processing on bus scenario

Consider a service that aims to optimise bus route planning in the urban commuter network. In this use case, a bus has front and rear cameras. The primary purpose of the cameras is to do people counting and know which stop people get on and off. The bus has a GPS coordinate that enables recording which bus stop people get off. It aims to know which stop each person is using without revealing the person's identity by recognising other features such as coat colour or type of clothing they are wearing. Therefore, by identifying the most popular stops, a fixable bus schedule can be created.

Q: Based on the previous scenario, could you please draw an IoT use case while considering GDPR privacy compliance?

B APPENDIX: STUDY PARTICIPANTS BACKGROUND AND DEMOGRAPHICS

Table A1. Study participants background information regarding SW/IoT development (Qualification? Education level; Years of exp? How many years of experience do they have in SW development; Field of expertise?; # of IoT apps*? How many IoT apps they had developed; Tools? What tool have you worked on to develop/design apps; Privacy training? Whether they had the training in building privacy). - Means they prefer not to say. *Zero app means the developer did not implement an IoT app but had implemented a mobile app with sensors.

ID	Age	Qualification	Years of exp	Field of expertise	# of IoT apps	Tools	Privacy training
P1	[40-49]	Doctoral	8-10	Development techniques into cyber-attack vectors	0	NA	✗
P2	[20-29]	Bachelor	10+	Cyber-security	0	Balsamiq, Visual Studio, Android Studio, Xcode	✓, four years
P3	[40-49]	Bachelor	10+	Internet of Things, telecommunications, utilities	2	Davra IoT, Amazon	✗
P4	[30-39]	Master	8-10	Mobile apps (iOS/Android)	2	Arduino, Blink	✗
P5	[30-39]	Diploma	8-10	Software development	0	UML diagrams	✗
P6	[40-49]	Doctoral	10+	Internet of Things	7	Many tools, Many Ides	✓, several times in 15 years
P7	[50-59]	Bachelor	10+	Telecoms and Internet of Things	3	Many tools	✗
P8	[30-39]	Master	1-3	Integration, cyber-security, product development	3	NA	✗
P9	[20-29]	Master	1-3	Software engineering and cyber-security.	0	NA	✗
P10	[20-29]	Master	1-3	Data ingestion, data streaming, API Development	5	NA	✗
P11	[20-29]	Bachelor	4-5	Security app development	0	NA	✗
P12	[20-29]	Master	4-5	Security, Internet of Things, full-stack development	7	✗	
P13	[20-29]	Bachelor	1-3	Full-stack developer, compiler engineering	2	IntelliJ, React JS, HTML, PHP, JAVA	✗
P14	[20-29]	Bachelor	1-3	Web and mobile development	0	PhpStorm,	✗
P15	[20-29]	Bachelor	1-3	Identity and access management	0	NA	✗
P16	[20-29]	Master	1-3	Web, app development, full-stack development	0	NA	✗
P17	[20-29]	Bachelor	4-5	Full-stack development	0	Jira, Draw.io	✗
P18	[20-29]	Bachelor	1-3	Payment gateway	0	NA	✗

C APPENDIX: SEMI-STRUCTURED INTERVIEWS QUESTIONS

C.1 General Background Questions

- Email Address:
- Participant ID (your first name . e.g. Nada) [Study use only]
- Age group:[20-29][30-39][]40-49][50-59][60+]
- Qualification:- Diploma - Bachelor's degree (or equivalent) - Master's degree (or equivalent) - Doctoral Degree (or equivalent)
- How many years of experience do you have in SW development?
- What is your area of expertise?

C.2 App Development and Privacy Training Background

Note: The questions may differ from one interview to another based on participants answers. Some of the questions are inspired by Coconut paper [58].

- Are you actively working on any development project as a software developer?
- Are you actively working on any IoT development project as a software developer? if yes give an example.
- Did you work as a professional IoT developer?
- How many IoT apps you had developed?
- Do you have experience in making or applying privacy policies? if "yes" for how long?
- When did you start to learn and apply privacy in general SW / IoT development? if not applicable type NA
- What general SW / IoT apps have you developed before? if not applicable type NA
- What tool have you used to develop/design apps? if not applicable type NA
- Did you participate in building any SW / apps (individually or as part of a team)?
- What were these applications built for?
- Was it developed by a team or individually?

- If you work in teams: How did you divide your work and collaborate? What's your responsibilities?
- Is there anyone working on determining what feature in your app and personal data is needed to implement or not?

When you answer these questions, put in mind the previously presented use case (CGM) as an example for IoT apps.

- Did you decide what the feature the app should have and what personal data you might need to collect prior to the development process?
  Did any of these apps use personally identifiable information (PII) /Personal Data (PD) ?
  Did you store identifiable information (PII) /Personal Data (PD)? Where?
  Did you send any identifiable information (PII) /Personal Data (PD) out of the phone?
  Do you think your app users are clearly know about what Personal Data (PD) are used and how they are used?
  If you use PD in your app: what's the purpose for that?
- Do your apps use unique identifier(UID)?
  If yes, Are you familiar with the best practices for unique identifier?
  Do you know how to reset the unique identifier?
  Do you know if UID will be shared by what apps?
  Do you think your app users are clearly know about what UID are used and how they are used?
  Did you send UID data out of the phone?
  Did you know where it is stored?
- If any of these apps collect and use the some of listed above information (Personal data):
  What's the purpose for that?
  Did you store them?
  Did you send them out of the phone?
- If any of these apps use the some of listed above information : How frequent did you access this information? In foreground or background?
  Did you follow the data collection practices with your user?
  Did you use an analytical third party library?
- Did you use any advertising third party library?
  If yes: What are these libraries?
  Why did you use them?
  Are you familiar with the data collection practices of the libraries that you used?
  If yes: How did you know?
- Do you have any comments or suggestions (any comments or additions are valuable):

D APPENDIX: PROTOTYPING FEEDBACK QUESTIONS

D.1 Measuring how (Disruptive – Time-consuming – Difficult)PARROT is? (A subjective rating on a 1–7 likert scale.)

Note: (1:strongly disagree for not disruptive/time-consuming/difficult at all, 7: strongly agree for very disruptive, time-consuming and difficult).

1. I felt that having to use colour coding was disruptive.
2. I felt that understanding colour coding over the design was time-consuming.
3. I felt that understanding colour coding over the design was difficult.
4. I felt that having to use mouseover or hover box over the design was disruptive.
5. I felt that having to use mouseover or hover box over the design was time-consuming.
6. I felt that having to use mouseover or hover box over the design was difficult.
7. I felt that the way of adding sub nodes was disruptive.
8. I felt that the way of adding sub nodes was time-consuming.
9. I felt that the way of adding sub nodes was difficult.
10. I felt configuring nodes and sub nodes properties was time-consuming.
11. I felt configuring nodes and sub nodes properties was difficult.
12. I felt that understanding the privacy properties in properties section was difficult.
13. I felt warning was time-consuming.
14. I felt warning was difficult.
15. I felt that understanding the warning was difficult.

D.2 Interpreted usefulness of PARROT and its key features (A subjective rating on a 1–7 likert scale.)

Feature code	Feature explanation
A	Sub nodes configuration
B	Mouseover features
C	Icons
D	Shapes
E	Sizes
F	Line
G	Colour coding for the line
H	Colour coding for nodes

A. I found the sub nodes configuration such as "Consent checked list" sub node of patient node is useful to learn privacy principals/policies.
B. I found having mouseover features over each point in "Consent checked list" is useful to learn further about privacy principles.
C. I found the using icons such as "lock icon" over the data transfer link is useful to understand it is encryption/decryption issue.
D. I found the using different shapes for the sub-nodes (circles and triangles) is useful to reflect that some properties are part of the design phase AND some are good practices but not necessary part design phase.
E. I found the using different sizes for the nodes and their sub-nodes is useful to understand that sub-nodes are the once that have privacy compliance with data protection obligations.
F. I found using "dashed or solid line" for the data transfer link is useful to understand (there is or there is no) security/privacy issue.
G. I found using "colour coding" for the data transfer is useful to understand there (there is or there is no) security/privacy issue.
H. I found using "colour coding" for the nodes and sub nodes is useful to understand there (there is or there is no) security/privacy issue.

D.3 Open-ended questions : to measure their understanding of the tool general features (does the tool help the participants to grasp new concepts quickly and efficiently)

- In short sentence, could you describe what you think the (red, yellow, amber or green) colour means and what you would do if you get it during the design.
- In short sentence, could you describe what you think the small circles above the nodes means.
- In short sentence, could you describe what you think the small triangles above the nodes means.
- Do you have any suggestions for other features for this tool to assist developers with privacy when designing IoT apps?

E APPENDIX: INTERVIEW SCRIPT

SW development procedure (in the real world)

- How do you develop a system in your work?
- Specify your work size (small/medium/big)?
- What are the most common privacy issues you have faced/ your work focus on?
- How do you learn about privacy?
  - Do you struggle while learning and applying privacy? How?
- How do you integrate privacy with the system as a developer? If it is not you who does it?
  - If it is text based explain from where? And how you do it? From where you get it?
    - Is it easy?
  - If it is chick list explain from where? And how you do it? From where you get it?
    - Is it easy?

The available tools that help to apply PbD in the SW development process (in the real world) and what PARROT offers:

- Do you use tools to help you integrate privacy with your design?
  - If yes list them?
  - If no? how about PARROT
- Is PARROT useful for you? explain?
  - If yes? In which way do you think it is useful?
  - If not? Why do you think it is not useful?

F APPENDIX: PRIVACY BY DESIGN SCORECARD AND ASSESSMENT

F.1 Privacy by Design scorecard

Table A2. Privacy by Design scorecard produced by the privacy lawyer during the privacy assessment step (step 4 of phase 1). Note: PD means Personal Data.

Principle	Explanation	Applicability to Case Study	Score
1: Privacy requirements intrinsic in design and analysis.	Understand and commit to privacy as BAU practice rather than as compliance add-on.	Identify key privacy issues: •Is PD needed for this process? •PD or anonymous data – de-identification methods?? •Type of PD (regular or special category); •Purpose limitation; •Minimisation and proportionality; •Data flows and different parties; •Retention/deletion.	3
2: Privacy embedded in the design.	Ensure privacy is integral to the architecture without impairing functionality being.	Consider issues throughout the process: •Different privacy concerns with different process steps (Patient, Sara, Medical staff) •Different processing purposes in respect of the same PD. •PD guardianship and responsibilities. •Documenting the above (accountability).	3
3: Full functionality.	Privacy is valued alongside the other aspects of the project: design, objectives, security, third parties etc.	Clear identification of privacy issues from the beginning, which should include: •Testing on dummy data; •Implementing and monitoring access controls.	3
4: End-to-end security.	Lifecycle protection of the data (including PD) - collection, use, disclosure, retention, and deletion.	All project participants must comply with a minimum standard: •Define and implement the minimum standard. •Ensure third parties (cloud hosting provider/local server healthcare company) comply with the minimum standard – how? •Security incident identification.	3
5: Visibility and transparency.	End-user trust: accountability, openness, and compliance.	Does the patient "know" what is happening to their personal data, and do we "do" what we say we are doing with their personal data? •Privacy notice – where? •Obtaining consent – where and how? •Withdrawing consent – where and how?	3
6: Respect for User Privacy.	All PD belongs to the end user, not to us. Respect for and understanding of this principle support the implementation of functionality that enables end-user to understand PD processing and access their PD.	•What controls does the diabetic patient have over their PD? •Can they cancel their use of the app at any time? •Do they "know" how to do this? •What happens to their PD once they cancel? •Purging/archiving?	3
			18

F.2 Privacy Assessment

In the beginning, the privacy lawyer was given the "diabetes treatment and monitoring" use case. The lawyer then elaborated on this basis six Privacy by Design principles. She then gave a score for each principle, thus assessing each conceptual design from a privacy standpoint. Each principle is given a score ranging between one and three (maximum 18 scores in total for the six principles, as seen in Table A2). The principles in Table A2 are suggested by the lawyer, which adheres to Ann Cavoukian's foundational principles of privacy by design [15][16], where a privacy certification is based on these principles too. The table is supplemented with different examples covering different phases of the Software Development Life Cycle (SDLC). Therefore, it is worth noting that not all of them are included in the scoring process since they are not in the design phase. For instance, upgrade functionality and decommissioning system at the 6th principle is related to the operations, maintenance and decommissioning (termination) phases [78][75], which not all developers are expected to think about in the study setting.

G APPENDIX: REDUCING BREAKDOWNS PROCESS

Table A3. List of identified privacy gaps or incongruencies between developers and lawyers. To address each gap, each was reviewed and reassessed using Enact's design principles. The background colour of the breakdown is blue for privacy-related issues and orange for more security-related issues. The note section explains more about the gap or incongruency, as well as an example of how it is visually represented in the second column.

#	Break Down	Enact design principles*				Visual representation example	Notes
		P1	P2	P3	P4		
1	Is it personal data (special categories)?	X	X	X	X	**Sample of the consent list of personal data special category:**  **Consent list icon:** 	The challenge here is what the special category distinction needs to trigger from the developer. Processing of special category data usually requires consent from the data subject. There are exceptions, but developers should always factor in obtaining consent using a check box as part of their design when the special category personal data prompt is triggered. If any part of the consent list is not considered, the icon background of the consent list will be red until all the list is checked to be yes. Also, any action done on any Sirius screen will be reflected on the other screens immediately, thus supporting automatic transformations among multiple representations.
2	Data subjects' rights	X	X	X	X	(Note: representation is similar to # 1.)	Awareness of these rights should make a developer "think" about the use, storage, and sharing (3rd party providers) of the personal data, and hopefully guide those decisions. In terms of delivering on data subjects' rights, the neatest way to deal with these is via a self-service model through the app/web portal. E.g. sending a request, viewing personal data that has been collected, stored and used.
3	Minimisation	X	X	X			Collecting data that is not needed breaches the data minimisation principle, such as where the app is collecting location data because it can, not because it needs it. Developers should always think about the data, if it is necessary or not, using a check box as part of their design. For example, the location of the patient can be determined by the developer using the checkbox. If the developer chooses to capture the exact geographical location of the patient, the location icon will be red, which means the presence of other privacy issues here, unless the developer changes this to the 'not needed' choice.
4	Processing activity	X	X	X			If personal data is being processed by the software, a privacy notice will need to be served to the user before the collection of any personal data. Developers should always think about the data, if it is necessary or not, using a check box as part of their design. For example, the location of the patient is not necessary for the glucose monitoring use-case. Based on the choice of the developer, the colour of the background of the location icon will be follow a risk colour coding: green, yellow, orange, and red. Red means the privacy/security risk is very high. Orange means the risk is high. Yellow means the risk is moderate. Green means the risk is low.

5	Data transfers /transmissions	X	X	X	**Cloud provider and server location options:** US_Provider_and_EU_Server_for_US_Patient US_Provider_and_US_Server_for_German_Patient US_Provider_and_EU_Server_for_UK_or_Ne_Patient EU_Provider_and_EU_Server_for_German_Patient US_Provider_and_US_Server_for_US_Patient **Cloud location icon:** 	Here we need to think about third parties and cross border transfers, so checkboxes are added to make sure the developer is aware of where the third-party provider and cloud server are located. Awareness of data transfer issues could result in the developer choosing a tool that is hosted in a 'more favourable' location, e.g. personal data stored in an EU server. Moreover, when there is data transmission, the developer should pay attention to whether security measures are applied. The colour of the link showing data transfer will change from red to green based on the action and the choices of the developer that solve one or more privacy concerns.
6	Sharing personal data that results in a data transfer	X	X	X	**Cloud location icon:** 	Developers lack knowledge about the server location, while lawyers consider that as a privacy issue, and thinking about the location of the servers within the hosting solution is mandatory. Developers need to be aware of this, and ideally use checkboxes and tooltip expression messages to explain why the colour is currently is red, green or amber. The background colour for the third-party cloud location changes based on the cloud provider location and the server location choice.
7	Data retention can have budgetary implications with storage costs	X		X	**Data retention options:** **Sensor Background colour:** 	The data retention period depends on the organization, which must be commitment to the laws and regulations that apply to it. Therefore, we use two options for the data retention period as: Accepted and Not accepted. These choices will depend on the organization and the application itself. Therefore, the developer needs to look at the system requirement document of the SW. Also, the CGM sensor background will be red when the data retention is not accepted. When it is accepted, the colour will be green.
8	Development of a self-service solution for data rights.	X	X	X	**Data sharing options:** **Cloud location icon:** 	The tool should help the developer to think about data sharing and who has access to all data from the app. The developer needs to ensure that when any patient data is shared, it is shared in compliance with security and privacy standards. For example, the researcher should have anonymised data only.

9	Containerisation and aggregation of data through API calls	X	X		Containerisation is used in case i one container is compromised; not all data is compromised. Aggregation: If the data is no longer needed, instead of deleting it, it can be aggregated, thus becoming anonymized. Since this is a security practice and thus not part of privacy by design, it is represented as a triangle rather than a circle to make it easier to differentiate
10 11 12 13 14	Risk of sensor data manipulation Risk of credential disclosure Risk of log data access Risk of unauthorised access Risk of data eavesdropping	X	X	**Data encryption option at the sensor node:**  **The reaction based on the choice:**  **If the node is authenticated or not:** 	One of the solutions to solve these issues is supporting data encryption and authentication in the tool. Based on the developer's choice, if the data is encrypted while being transferred, the colour of the data link and the icon will change. In addition, based on the developer's choice, the background colour of the authentication and authorisation icons will be either green or red.